ORACLE

Cloud Infrastructure

Private Cloud Appliance

March 27, 2026

Contents

Private Cloud Appliance 6
Architecture and Design 6
Compatibility with OCI 7
Enclaves and Interfaces 8
Layered Architecture 10
Key Platform Properties 11
High Availability 13
System Hardware 14
Base Rack Configuration 14
GPU Expansion Rack Configuration 21
Compute Expansion Rack Configuration 23
Server Nodes 25
Storage Hardware 27
Network Infrastructure 28
Platform and Service Limits 48
Compute Node Physical Resources 48
Tenancy Resource Configuration Limits 48
System Load and Concurrency Limits 51
Guest OS Matrix 53
View as PDF 54
Installation and Configuration 54
Prepare Installation Site 55
Prepare Data Center Network 62
Site Checklists 68
Initial System Installation Checklist 74
Install Appliance Rack 82
Configure Appliance 95
Cabling Reference 125
Infrastructure Administration 145
Using the Service Web UI 146
Using the Service CLI 150
Change Component Passwords 155
Compute Node Operations 156
Integrate GPU Expansion Nodes 166
Integrate Compute Expansion Rack 167
Compute Instance Availability 168
Administrator Account Management 174
Tenancy Management 191
Configure Resource Limits 195
Configure Flex Networks 207
Reconfigure Network Environment 235
Configure Active Directory for File Storage Service 246
Use Custom CA Trust Chain 247
Status and Health Monitoring 250
System Backup and Restore 292
Disaster Recovery 297
Controller Software Upgrade 335
Upgrade Plan and History 335
Pre-Checks 336

Full Rack Upgrade....336
Component Upgrades....337
Important Upgrade Notes....338
Prepare for Upgrade....339
Check Upgrade Readiness and Status....350
Perform Full Rack Upgrade....362
Individual Component Upgrades....371
Accessing User Interfaces....396
Signing in to the Compute Web UI....396
Installing and Using the OCI CLI....397
Using the API....413
Identity and Access Management (IAM)....414
Creating and Managing Compartments....414
Managing User Accounts....419
Creating and Managing User Groups....426
User Account Federation with Active Directory....429
Configuring Instances for Calling Services....430
Creating and Managing Dynamic Groups....433
Managing Policies....435
Tagging Resources (IAM on the Appliance)....446
Creating and Managing Tag Namespaces (IAM on the Appliance)....447
Creating and Managing Tag Key Definitions (IAM on the Appliance)....451
Configuring Tag Defaults (IAM on the Appliance)....455
Using Resource Tags (IAM on the Appliance)....458
Using Resource Tags to Extend Functionality (IAM on the Appliance)....460
Virtual Cloud Networks....465
Name Resolution....466
Networking Scenarios....467
Managing VCNs and Subnets....478
Configuring VCN Rules and Options....483
Configuring VCN Gateways....505
Configuring VNICs....518
Managing Private IP Addresses....523
Managing Public IP Addresses....528
Enabling IPv6....534
Configuring SR-IOV....535
Managing Public DNS Zones....537
Managing Traffic with Steering Policies....542
Load Balancers....548
Load Balancer as a Service....550
Network Load Balancers....601
Images....620
Listing Images and Viewing Details....621
Platform Images....621
Managing Custom Images....622
Bring Your Own Image (BYOI)....629
Compute Instances....634
Compute Shapes....636
Physical Resource Allocation....638
Compute Instance Lifecycle....639
Extending Compute Resources....639
Tutorial: Launching Your First Instance....640
Working with Instances....649
Configuring Instances for Calling Services....663
Working with Instance Configurations....666
Connecting to a Compute Instance....674

Backing Up and Restoring an Instance....677
Instance Pools....682
Creating an Instance Pool....683
Using Schedule-Based Autoscaling....686
Updating an Instance Pool....695
Attaching an Instance to an Instance Pool....696
Detaching an Instance from an Instance Pool....697
Managing Instance Pool Load Balancer Attachments....698
Stopping and Starting Instances in an Instance Pool....699
Deleting an Instance Pool....700
Container Instances....701
Container Instance Shapes....701
Working with Container Instances....702
Working with Containers in Container Instances....711
Block Volume Storage....712
Creating and Attaching Block Volumes....716
Managing Block Volumes....725
Managing Boot Volumes....731
Resizing Volumes....735
Managing Volume Groups....740
Backing Up Block Volumes....744
Managing Backup Policies....747
File Storage....754
VCN Security Rules for File Storage....757
File Storage Network Ports....758
Deleting File System Resources Overview....758
Export Options for File Storage....758
Creating a File System, Mount Target, and Export....763
Controlling Access to File Storage....771
Mounting File Systems on UNIX-based Instances....774
Mounting File Systems on Microsoft Windows Instances....778
Managing Mount Targets and Exports....782
Managing File Systems....786
Managing File System Snapshots....789
Managing File System Clones....792
Object Storage....793
Obtaining the Object Storage Namespace....794
Managing Object Storage Buckets....794
Managing Storage Objects....797
Managing Object Versioning....807
Using Preauthenticated Requests....811
Defining Retention Rules....817
Kubernetes Engine (OKE)....821
OKE Workflow....822
OKE Best Practices....830
Creating OKE Network Resources....833
Creating and Managing OKE Clusters....904
Managing OKE Cluster Add-ons....916
Creating and Managing OKE Worker Node Pools....932
Exposing Containerized Applications....947
Adding Storage for Containerized Applications....949
Troubleshooting....961
Known Issues....961
Upgrade might impact Load Balancer as a Service functionality....961
GPU Drivers Not Included in Oracle Linux Platform Images....962
After an API key is created or changed, the initial CLI command might fail....964

The `oci ccc get infrastructure` and `oci ccc infrastructure update` CLI commands return `null` for the `provisioning_pin` value..... 964
When using the `ccc infrastructure list` CLI command with the `--compartment-id-in-subtree true` option, no results are returned.....965
Output from `oci iam user get` doesn't list user capabilities.....966

Index.....967

Private Cloud Appliance

Use Private Cloud Appliance to deploy Oracle Cloud Infrastructure (OCI) services on-premises to comply with data residency requirements. Operate disconnected from OCI, maintain full control over your infrastructure, and expand its capacity as your resource consumption grows.

What's new

Get Started

- Installation task map
- The Service CLI
- Access user interfaces
- Tutorial: launch your first instance
- Legacy documentation
- Product page

Expand Infrastructure

- Add a GPU rack
- Add a compute expansion rack
- Connect other systems

Maintenance

- Software upgrade
- Reconfigure the network
- Change component passwords
- Use your own CA chain

Appliance Administration

- Manage tenancies
- Compute node operations
- Monitor system health
- Configure disaster recovery
- Control instance HA

Key Concepts

- Architecture
- Hardware components
- Data center uplinks
- Compute instances

Support

- Register for ASR
- Sign in to My Oracle Support

Community

- OCI blog
- OCI community forum

Set Up Resources

- Manage VCNs and subnets
- Create and attach a block volume
- Create and mount a file system
- Deploy a load balancer
- Create an OKE cluster

Manage Resources

- Bring your own image
- Use instance backup
- Create an instance pool
- Manage block volumes
- Manage file storage
- Use object storage

Troubleshooting

- Known issues

Developer Tools

- SDKs and the CLI
- Cloud Shell

Architecture and Design of Private Cloud Appliance

Private Cloud Appliance brings a comprehensive suite of Oracle Cloud Infrastructure services into the secure environment of your on-premises network. It is an engineered system composed of hardware and software components that have been tested, configured, and tuned for optimal performance.

Private Cloud Appliance is a general purpose IaaS (Infrastructure as a Service) solution in the sense that it supports the widest variety of workloads. You access the core IaaS services using the same methods, tools and interfaces as with OCI. A Private Cloud Appliance installation represents a customer region. Workloads are portable between your private and public cloud environments, but the private cloud is disconnected from OCI and thus runs its own control plane components in order to host its set of compatible services.

As an engineered system, Private Cloud Appliance complies with the highest business continuity and serviceability requirements. It has the capabilities to monitor all components, detect potential problems, send out alerts, and automatically log a service request. Subsequent troubleshooting and repair can be performed without affecting the uptime of the environment.

Because a Private Cloud Appliance is customer owned and operated, the administration environment known as the Service Enclave is available to the administrator accounts you configure.

More detailed information is provided in these topics:

- Compatibility with Oracle Cloud Infrastructure on page 7
- Enclaves and Interfaces on page 8
- Layered Architecture on page 10
- Key Properties of the Platform Layer on page 11
- High Availability on page 13

Compatibility with Oracle Cloud Infrastructure

To allow you to consume core Oracle Cloud Infrastructure services within your on-premises network, and migrate workloads between your public and private cloud infrastructures with minimal or no modification required, Private Cloud Appliance offers API compatibility with OCI.

Private Cloud Appliance can be considered a rack-scale deployment unit of OCI, aligned with the physical hierarchy of the public cloud design.

Hierarchy Concept	Oracle Cloud Infrastructure Design	Private Cloud Appliance Mapping
Realm	A Realm is a superset of Regions, and the highest physical subdivision of the Oracle cloud. There are no cross-realm features. OCI currently consists of a Realm for Commercial Regions and a Realm for Government Cloud Regions.	The concept of a Realm exists in Private Cloud Appliance, but it has no practical function. It allows the appliance to participate in any Realm.
Region	A Region is a geographic area. An OCI Region is composed of at least three Availability Domains. It is possible to migrate or replicate data and resources between Regions.	Private Cloud Appliance is designed as a single Region. Because this private region is disconnected from any other systems, it has no practical function. Domain and system identifiers are used in system configuration instead, and mapped to the region and realm values.
Availability Domain	An Availability Domain consists of one or more data centers. Availability Domains are isolated from each other; they have independent power and cooling infrastructure and separate internal networking. A failure in one Availability Domain is highly unlikely to impact others. Availability Domains within the same region are interconnected through an encrypted network with high bandwidth and low latency. This is a critical factor in providing high availability and disaster recovery.	Each Private Cloud Appliance is configured as an Availability Domain. Multiple installations are distinct from each other: they do not function as Availability Domains within the same region.

Hierarchy Concept	Oracle Cloud Infrastructure Design	Private Cloud Appliance Mapping
Fault Domain	A Fault Domain is a grouping of infrastructure components within an Availability Domain. The goal is to isolate downtime events due to failures or maintenance, and ensure that resources in other Fault Domains are not affected. Each Availability Domain contains three Fault Domains. Fault Domains provide anti-affinity: the ability to distribute instances so that they do not run on the same physical hardware.	Private Cloud Appliance adheres to the public cloud design: each Availability Domain contains three Fault Domains. A Fault Domain corresponds with one or more physical compute nodes.

Private Cloud Appliance also aligns with the logical partitioning of OCI. It supports multiple tenancies, which are securely isolated from each other by means of tunneling and encapsulation in the appliance underlay network. Tenancies are hosted on the same physical hardware, but users and resources that belong to a given tenancy cannot interact with other tenancies. In addition, the Compute Enclave – which refers to all tenancies collectively, and to the cloud resources created and managed within them – is logically isolated from the Service Enclave, from where the appliance infrastructure is controlled. Refer to Enclaves and Interfaces on page 8 for more information.

The Compute Enclave interfaces provide access in the same way as OCI. Its CLI is identical while the browser UI offers practically the same user experience. API support is also identical, but limited to the subset of cloud services that Private Cloud Appliance offers.

The consistency of the supported APIs is a crucial factor in the compatibility between the public and private cloud platforms. It ensures that the core cloud services support resources and configurations in the same way. More specifically, Private Cloud Appliance supports the same logical constructs for networking and storage, manages user identity and access in the same way, and offers the same compute shapes and images for instance deployment as OCI. As a result, workloads set up in a private cloud environment are easily portable to Private Cloud Appliance and vice versa. However, due to the disconnected operating mode of the private cloud environment, workloads must be migrated offline.

Enclaves and Interfaces

From a cloud user perspective, the Private Cloud Appliance Compute Enclave offers a practically identical experience to Oracle Cloud Infrastructure. However, the appliance also runs its own specific and separate administration area known as the Service Enclave. Each enclave and its interfaces are intended for different groups of users and administrators, with clearly distinct access profiles.

Enclave Boundaries

The **Compute Enclave** was deliberately designed for maximum compatibility with OCI. Users of the Compute Enclave have certain permissions to create and manage cloud resources. These privileges are typically based on group membership. The Compute Enclave is where workloads are created, configured and hosted. The principal building blocks at the users' disposal are compute instances and associated network and storage resources.

Compute instances are created from a *compute image*, which contains a preconfigured operating system and optional additional software. Compute instances have a particular *shape*, which is a template of virtual hardware resources such as CPUs and memory. For minimal operation, a compute instance needs a boot volume and a connection to a virtual cloud network (VCN). As you continue to build the virtual infrastructure for your workload, you will likely add more compute instances, assign private and public network interfaces, set up NFS shares or object storage buckets, and so on. All those resources are fully compatible with OCI and can be ported between your private and public cloud environments.

The **Service Enclave** is the part of the system where the appliance infrastructure is controlled. Access is closely monitored and restricted to privileged administrators. It runs on a cluster of three management nodes. Because Private Cloud Appliance is operationally disconnected from OCI, it needs a control plane of its own, specific to the design

and scale of the appliance. The API is specific to Private Cloud Appliance, and access is very strictly controlled. Functionality provided by the Service Enclave includes hardware and capacity management, service delivery, monitoring, and tools for service and support.

Both enclaves are strictly isolated from each other. Each enclave provides its own set of interfaces: a web UI, a CLI and an API per enclave. An administrator account with full access to the Service Enclave has no permissions whatsoever in the Compute Enclave. The administrator creates the tenancy with a primary user account for initial access, but has no information about the tenancy contents and activity. Users of the Compute Enclave are granted permission to use, manage and create cloud resources, but they have no control over the tenancy they work in, or the hardware on which their virtual resources reside.

Access Profiles

Each enclave has its own interfaces. To access the Compute Enclave, you use either the Compute Web UI or OCI CLI. To access the Service Enclave, you use either the Service Web UI or Service CLI.

> **Note:**
>
> You access the graphical interfaces of both enclaves using a web browser. For support information, please refer to the Oracle software web browser support policy.

The properties of your account determine which operations you are authorized to perform and which resources you can view, manage or create. Whether you use the web UI or the CLI makes no difference in terms of permissions. All operations result in requests to a third, central interface of the enclave: the API. Incoming requests from the Service Enclave API or Compute Enclave API are evaluated and subsequently authorized or rejected by the API service.

Different categories of users interact with the appliance for different purposes. At the enclave level we distinguish between administrators of the appliance infrastructure on the one hand, and users managing cloud resources within tenancies on the other hand. Within each enclave, different access profiles exist that offer different permissions.

In the Service Enclave, only a select team of administrators should be granted full access. There are other administrator roles with restricted access, for example for those responsible specifically for system monitoring, capacity planning, availability, upgrade, and so on. For more information about administrator roles, see Private Cloud Appliance Infrastructure Administration on page 145. Whenever Oracle accesses the Service Enclave to perform service and support operations, an account with full access must be used.

When a tenancy is created, it has only one Compute Enclave user account: the tenancy administrator, who has full access to all resources in the tenancy. Practically speaking, every additional account with access to the tenancy is a regular Compute Enclave user account, with more or less restrictive permissions depending on group membership and policy definitions. It is the task of the tenancy administrator to set up additional user accounts and user groups, define a resource organization and management strategy, and create policies to apply that strategy.

Once a resource management strategy has been defined and a basic configuration of users, groups and compartments exists, the tenancy administrator can delegate responsibilities to other users with elevated privileges. You could decide to use a simple policy allowing a group of administrators to manage resources for the entire organization, or you might prefer a more granular approach. For example, you can organize resources in a compartment per team or project and let a compartment administrator manage them. In addition, you might want to keep network resources and storage resources contained within their own separate compartments, controlled respectively by a network administrator and storage administrator. The policy framework of the Identity and Access Management service offers many different options to organize resources and control access to them. For more information, see Identity and Access Management (IAM) on page 414.

When creating scripts or automation tools to interact directly with the API, make sure that the developers understand the authentication and authorization principles and the strict separation of the enclaves. Basic API reference documentation is available for both enclaves.

To view the API reference, append `/api-reference` to the base URL of the Compute Web UI or Service Web UI. For example:

- Service Web UI base URL: https://adminconsole.myprivatecloud.example.com.

Service Enclave API reference: https://adminconsole.myprivatecloud.example.com/*api-reference*.

- Compute Web UI base URL: https://console.myprivatecloud.example.com.

  Compute Enclave API reference: https://console.myprivatecloud.example.com/*api-reference*.

Layered Architecture

For the architecture of Private Cloud Appliance a layered approach is taken. At the foundation are the hardware components, on which the core platform is built. This, in turn, provides a framework for administrative and operational services exposed to different user groups.

The layers are integrated but not monolithic: they can be further developed at different rates as long as they maintain compatibility. For example, supporting a new type of server hardware or extending storage functionality are enhancements that can be applied separately, and without redeploying the entire controller software stack.

Hardware Layer

The hardware layer contains all physical system components and their firmware and operating systems.

- The three management nodes form a cluster that runs the base environment for the controller software.
- The compute nodes provide the processing capacity to host compute instances.
- The storage appliance provides disk space for storage resources used by compute instances. It also provides the storage space required by the appliance internally for its operation.
- The network switches provide the physical connections between all components and the uplinks to the public (data center) network.

Platform Layer

Private Cloud Appliance uses a service-based deployment model. It is divided into functional areas that run as services within separate containers. The platform provides the base for this model. The management node cluster orchestrates the deployment of the containerized services, for which it also hosts the image registry.

The platform provides a set of foundational services that are required by all other services: message transport, secrets management, database access, logging, monitoring, and so on. These foundational services are standardized so that all services deployed on top of the platform can plug into them in the same way, which makes new service integrations easier and faster.

The platform also plays a central role in hardware administration, managing the data exchange between the hardware layer and the services. Information about the hardware layer, and any changes made to it, must be communicated to the services layer to keep the inventory up-to-date. When operations are performed at the service layer, an interface is required to pass down commands to the hardware. For this purpose, the platform has a tightly secured API that is only exposed internally and requires the highest privileges. This API interacts with management interfaces such as the server ILOMs and the storage controllers, as well as the inventory database and container orchestration tools.

For more information about this layer in the appliance architecture, see Key Properties of the Platform Layer on page 11.

Infrastructure Services Layer

This layer contains all the services deployed on top of the platform. They form two functionally distinct groups: user-level cloud services and administrative services.

Cloud services offer functionality to users of the cloud environment, and are very similar in operation to the corresponding OCI services. They constitute the Compute Enclave, and enable the deployment of customer workloads through compute instances and associated resources. Cloud services include the compute and storage services, identity and access management, and networking.

The administrative services are either internal or restricted to administrators of the appliance. These enable the operation of the cloud services and provide support for them. They constitute the Service Enclave. Administrator operations include system initialization, compute node provisioning, capacity expansion, tenancy management,

upgrade, and so on. These operations have no externalized equivalent in OCI, where Oracle fulfills the role of the infrastructure administrator.

Key Properties of the Platform Layer

In the Private Cloud Appliance architecture, the platform layer is the part that provides a standardized infrastructure for on-premises cloud services. It controls the hardware layer, enables the services layer, and establishes a secure interface that allows those layers to interact in a uniform and centralized manner. Common components and features of the infrastructure services layer are built into the platform, thus simplifying and accelerating the deployment of those microservices.

Foundational Services

To fulfill its core role of providing the base infrastructure to deploy on-premises cloud services, the platform relies on a set of foundational internal services of its own.

Hardware Management

When a system is initialized, low level platform components orchestrate the provisioning of the management node cluster and the compute nodes. During this process, all nodes, including the controllers of the ZFS Storage Appliance, are connected to the required administration and data networks. When additional compute nodes are installed at a later stage, the same provisioning mechanism integrates the new node into the global system configuration. Additional disk trays are also automatically integrated by the storage controllers.

The first step in managing the hardware is to create an inventory of the rack's components. The inventory is a separate database that contains specifications and configuration details for the components installed in the rack. It maintains a history of all components that were ever presented to the system, and is updated continuously with the latest information captured from the active system components.

The services layer and several system components need the hardware inventory details so they can interact with the hardware. For example, a component upgrade or service deployment process needs to send instructions to the hardware layer and receive responses. Similarly, when you create a compute instance, a series of operations needs to be performed at the level of the compute nodes, network components and storage appliance to bring up the instance and its associated network and storage resources.

All the instructions intended for the hardware layer are centralized in a hardware management service, which acts as a gateway to the hardware layer. The hardware management service uses the dedicated and highly secured platform API to execute the required commands on the hardware components: server ILOMs, ZFS storage controllers, and so on. This API runs directly on the management node operating system. It is separated from the container orchestration environment where microservices are deployed.

Service Deployment

Private Cloud Appliance follows a granular, service-based development model. Functionality is logically divided into separate microservices, which exist across the architectural layers and represent a vertical view of the system. Services have internal as well as externalized functions, and they interact with each other in different layers.

These microservices are deployed in Kubernetes containers. The container runtime environment as well as the registry are hosted on the three-node management cluster. Oracle Cloud Native Environment provides the basis for container orchestration, which includes the automated deployment, configuration and startup of the microservices containers. By design, all microservices consist of multiple instances spread across different Kubernetes nodes and pods. Besides high availability, the Kubernetes design also offers load balancing between the instances of each microservice.

Containerization simplifies service upgrades and functional enhancements. The services are tightly integrated but not monolithic, allowing individual upgrades on condition that compatibility requirements are respected. A new version of a microservice is published to the container registry and automatically propagated to the Kubernetes nodes and pods.

Common Service Components

Some components and operational mechanisms are required by many or all services, so it is more efficient to build those into the platform. Microservices can consume these common components and their essential features when they are deployed on top of the platform.

- **Message Transport**

  All components and services are connected to a common transport layer. It is a message broker that allows components to send and receive messages written in a standardized format. This message transport service is deployed as a cluster of three instances for high availability and throughput, and uses TLS for authentication and traffic encryption.
- **Secret Service**

  Secrets used programmatically throughout the system, such as login credentials and certificates, are centrally managed by the secret service. All components and services are clients of the secret service: after successful authentication the client receives a token for use with every operation it attempts to execute. Policies defined within the secret service determine which operations a client is authorized to perform. Secrets are not stored in a static way; they have a limited lifespan and are dynamically created and managed.

  During system initialization, the secret service is unsealed and prepared for use. It is deployed as an active/standby cluster on the management nodes, within a container at the platform layer, but outside of the Kubernetes microservices environment. This allows the secret service to offer its functionality to the platform layer at startup, before the microservices are available. All platform components and microservices must establish their trust relationship with the secret service before they are authorized to execute any operations.
- **Logging**

  The platform provides unified logging across the entire system. For this purpose, all services and components integrate with the Fluentd data collector. Fluentd collects data from a pre-configured set of log files and stores it in a central location. Logs are captured from system components, the platform layer and the microservices environment, and made available through the Loki log aggregation system for traceability and analysis.
- **Monitoring**

  For monitoring purposes, the platform relies on Prometheus to collect metric data. Since Prometheus is deployed inside the Kubernetes environment, it has direct access to the microservices metrics. Components outside Kubernetes, such as hardware components and compute instances, provide their metric data to Prometheus through the internal network and the load balancer. The management nodes and Kubernetes itself can communicate directly with Prometheus.
- **Analytics**

  Logging and monitoring data are intended for infrastructure administrators. They can consult the data through the Service Web UI, where a number of built-in queries for health and performance parameters are visualized on a dashboard. Alerts are sent when a key threshold is exceeded, so that appropriate countermeasures can be taken.
- **Database**

  All services and components store data in a common, central database. It is a MySQL cluster database with instances deployed across the three management nodes and running on bare metal. Availability, load balancing, data synchronization and clustering are all controlled by internal components of the MySQL cluster. For optimum performance, data storage is provided by LUNs on the ZFS Storage Appliance, directly attached to each of the management nodes. Access to the database is strictly controlled by the secret service.
- **Load Balancing**

  The management nodes form a cluster of three active nodes, meaning they are all capable of simultaneously receiving inbound connections. The ingress traffic is controlled by a statically configured load balancer that listens on a floating IP address and distributes traffic across the three management nodes. An instance of the load balancer runs on each of the management nodes.

  In a similar way, all containerized microservices run as multiple pods within the container orchestration environment on the management node cluster. Kubernetes provides the load balancing for the ingress traffic to the microservices.

High Availability

Private Cloud Appliance is engineered to eliminate single points of failure, enabling the system and hosted workloads to remain operational in case of hardware or software faults, and during upgrades and maintenance operations.

Redundancy is built into the architecture at every level: hardware, controller software, master database, services, and so on. Features such as backup, automated service requests and optional disaster recovery further enhance the system's serviceability and continuity of service.

Hardware Redundancy

The minimum base rack configuration contains redundant networking, storage and server components to ensure that failure of any single element doesn't affect overall system availability.

Data connectivity throughout the system is built on redundant pairs of leaf and spine switches. Link aggregation is configured on all interfaces: switch ports, host NICs, and uplinks. The leaf switches interconnect the rack components using cross-cabling to redundant network interfaces in each component. Each leaf switch also has a connection to each of the spine switches, which are also interconnected. The spine switches form the backbone of the network and enable traffic external to the rack. Their uplinks to the data center network consist of two cable pairs, which are cross-connected to two redundant ToR (top-of-rack) switches.

The management cluster, which runs the controller software and system-level services, consists of three fully active management nodes. Inbound requests pass through the virtual IP of the management node cluster, and are distributed across the three nodes by a load balancer. If one of the nodes stops responding and fences from the cluster, the load balancer continues to send traffic to the two remaining nodes until the failing node is healthy again and rejoins the cluster.

Storage for the system and for the cloud resources in the environment is provided by the internal ZFS Storage Appliance. Its two controllers form an active-active cluster, providing high availability and excellent throughput at the same time. The ZFS pools are built on disks in a mirrored configuration for the best data protection. This applies to the standard high-capacity disk tray as well as an optional SSD-based high-performance tray.

System Availability

The software and services layer are deployed on the three-node management cluster, and take advantage of the high availability that's inherent to the cluster design. The Kubernetes container orchestration environment also uses clustering for both its own controller nodes and the service pods it hosts. Several replicas of the microservices are running at any particular time. Nodes and pods are distributed across the management nodes, and Kubernetes ensures that failing pods are replaced with new instances to keep all services running in an active/active setup.

All services and components store data in a common, central MySQL cluster database, of which instances are deployed across the three management nodes. Availability, load balancing, data synchronization, and clustering are all controlled by internal components of the MySQL cluster.

A significant part of the system-level infrastructure networking is software-defined. The configuration of virtual switches, routers and gateways isn't stored and managed by the switches, but is distributed across several components of the network architecture. The network controller is deployed as a highly available containerized service.

The upgrade framework leverages the hardware redundancy and the clustered designs to provide rolling upgrades for all components. During the upgrade of one component instance, the remaining instances ensure that there's no downtime. The upgrade is complete when all component instances have been upgraded and returned to normal operation.

Continuity of Service

Private Cloud Appliance offers several features that further enhance high availability. Health monitoring at all levels of the system is a key factor. Diagnostic and performance data is collected from all components, then centrally stored and processed, and made available to administrators in the form of visualizations on standard dashboards. In addition, alerts are generated when metrics exceed their defined thresholds.

To mitigate data loss and support the recovery of system and services configuration in case of failure, consistent and complete backups are made regularly. A backup can also be run manually, for example to create a restore point before a critical modification. The backups are stored in a dedicated NFS share on the ZFS Storage Appliance, and allow the entire Service Enclave to be restored when necessary.

Optionally, workloads deployed on the appliance can be protected against downtime and data loss through the implementation of disaster recovery. To achieve this, two Private Cloud Appliance systems need to be set up at different sites, and configured to be a replica of each other. Resources under disaster recovery control are stored separately on the ZFS Storage Appliances in each system, and replicated between the two. When an incident occurs at one site, the environment is brought up on the replica system with minimal downtime. We recommend that disaster recovery is implemented for all critical production systems.

Overview of Private Cloud Appliance Hardware Components

The physical hardware configuration of Private Cloud Appliance consists of a base rack with servers, switches, and storage, and optional expansion racks for increased capacity. All components have been engineered for the best possible integration and performance.

This part of the documentation describes the properties and the role of each component. Special attention is given to the appliance network infrastructure, and the way it integrates with the data center environment. The appliance also has reserved ports for optional direct external connections, for example to an Exadata system.

- Base Configuration of Private Cloud Appliance on page 14
- Server Nodes Installed in Private Cloud Appliance on page 25
- Private Cloud Appliance Network Infrastructure on page 28
- Storage Hardware Installed in Private Cloud Appliance on page 27
- Compute Expansion Rack Configuration on page 23
- GPU Expansion Rack Configuration on page 21

Base Configuration of Private Cloud Appliance

Each Private Cloud Appliance consists of an Oracle Rack base, populated with network, compute, and storage components. The appliance storage and compute capacity is customized to meet your capacity requirements.

A rack is populated according to the flex bay concept for rack space management. Flex bays are dedicated 4-rack-unit sections within the rack that can be used for flexible expansion. A flex bay can house either storage or compute resources, but not both in the same bay.

Different base configurations exist:

Private Cloud Appliance X11

The ***X11*** hardware configuration is shipping as of August 2025. It contains 3 to 12 factory-installed Oracle Server X11 compute nodes (2U). Storage is provided by an Oracle ZFS Storage Appliance ZS11-2. Physical network infrastructure consists of Cisco Nexus 9348GC-FXP (management) and Cisco Nexus 9336C-FX2 (data) switches.

Compared to previous configurations, the storage appliance has newer controllers but uses exactly the same disk enclosures. The network infrastructure components have remained the same.

Private Cloud Appliance X10

The ***X10*** hardware configuration is shipping as of December 2023. It contains 3 to 12 factory-installed Oracle Server X10 compute nodes (2U). Storage is provided by an Oracle ZFS Storage Appliance ZS9-2. Physical network infrastructure consists of Cisco Nexus 9348GC-FXP (management) and Cisco Nexus 9336C-FX2 (data) switches.

Compared to the X9 configuration, component order and cabling are slightly different. The storage and network infrastructure components are the same. In terms of compute capacity, a 2U compute node is comparable to two 1U compute nodes.

Private Cloud Appliance X9-2

The ***X9*** hardware configuration was released together with the first appliance software version 3.x, and was replaced by the X10 configuration. The X9 configuration contains 3 to 20 Oracle Server X9-2 compute nodes (1U). Storage is provided by an Oracle ZFS Storage Appliance ZS9-2. Physical network infrastructure consists of Cisco Nexus 9348GC-FXP (management) and Cisco Nexus 9336C-FX2 (data) switches.

X11 Rack Configuration

The hardware platform in ***X11*** configuration, is factory-installed with appliance software release 3.0.2-b1392231 or newer. Its hardware components are identified in the figure below. This figure shows a fully populated rack, however you can customize your rack configuration to include different storage or compute capacity as needed.

The flex bays are dedicated 4 rack unit sections within the rack that can be used for flexible expansion. For each flex bay, except the top flex bay at rack units 38-41, you can choose to add 2 compute nodes, 1 Oracle Storage Drive Enclosure DE3-24C, or 2 Oracle Storage Drive Enclosure DE3-24P. These flex bays can house either storage or compute resources, but not both in the same bay. Only storage is allowed in the top flex bay because of power constraints.

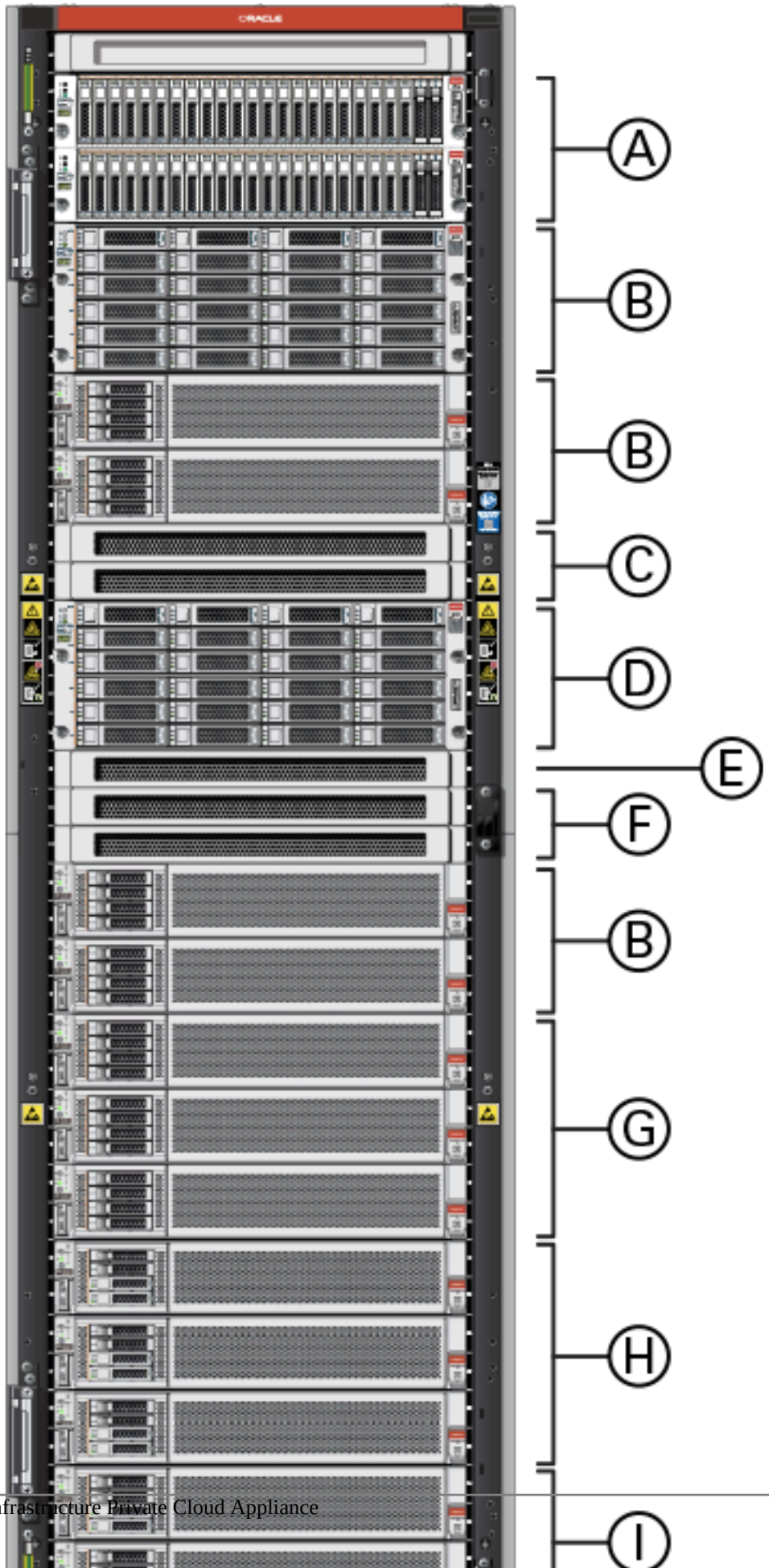

A
B
B
C
D
E
F
B
G
H
I

Figure Legend

Callout	Quantity	Description
A	1	flex bay can accommodate 1-2 storage enclosures
B	1 - 3	flex bay can accommodate 1-2 compute nodes, or 1-2 storage enclosures
C	2	spine switch
D	1	Oracle Storage Drive Enclosure DE3-24C disk shelf
E	1	management switch
F	2	leaf switch
G	3	compute nodes 3 required for minimum configuration
H	3	management nodes
I	2	storage controllers

X10 Rack Configuration

The hardware platform in ***X10*** configuration, is factory-installed with appliance software release 3.0.2-b1001356 or newer. Its hardware components are identified in the figure below. This figure shows a fully populated rack, however you can customize your rack configuration to include different storage or compute capacity as needed.

For each flex bay, you can choose to add one of these options: 1-2 compute nodes, 1 Oracle Storage Drive Enclosure DE3-24C, or 1-2 Oracle Storage Drive Enclosure DE3-24P.

Rack units 39 and 40 are reserved for systems with 6 or more compute nodes: 3 default nodes plus 3 or more expansion nodes. Compute nodes 4 and 5 are installed in the flex bay directly above the management nodes (rack units 14-17), and compute node 6 is installed in the reserved 2U space.

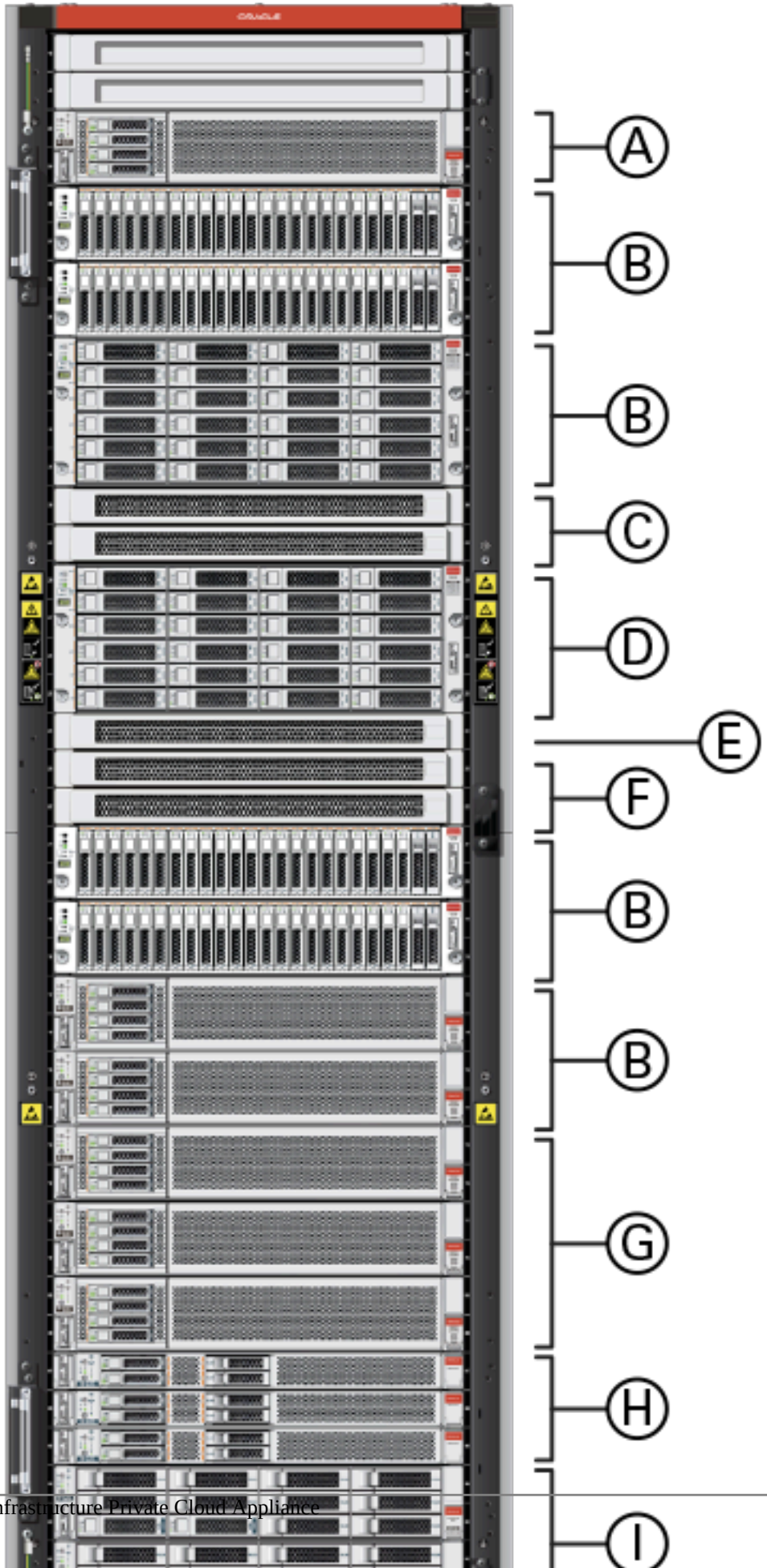

A
B
B
C
D
E
F
B
B
G
H
I

Figure Legend

Callout	Quantity	Description
A	1	reserved space for 6th compute node
B	1 - 4	flex bay can accommodate 1-2 compute nodes, or 1-2 storage enclosures
C	2	spine switch
D	1	Oracle Storage Drive Enclosure DE3-24C disk shelf
E	1	management switch
F	2	leaf switch
G	3	compute nodes 3 required for minimum configuration
H	3	management nodes
I	2	storage controllers

X9 Rack Configuration

The hardware platform in ***X9*** configuration, is factory-installed with appliance software release 3.0.1 or 3.0.2. Its hardware components are identified in the figure below. This figure shows a fully populated rack, however you can customize your rack configuration to include different storage or compute capacity as needed.

For each flex bay, you can choose to add one of these options: 1-4 compute nodes, 1 Oracle Storage Drive Enclosure DE3-24C, or 1-2 Oracle Storage Drive Enclosure DE3-24P.

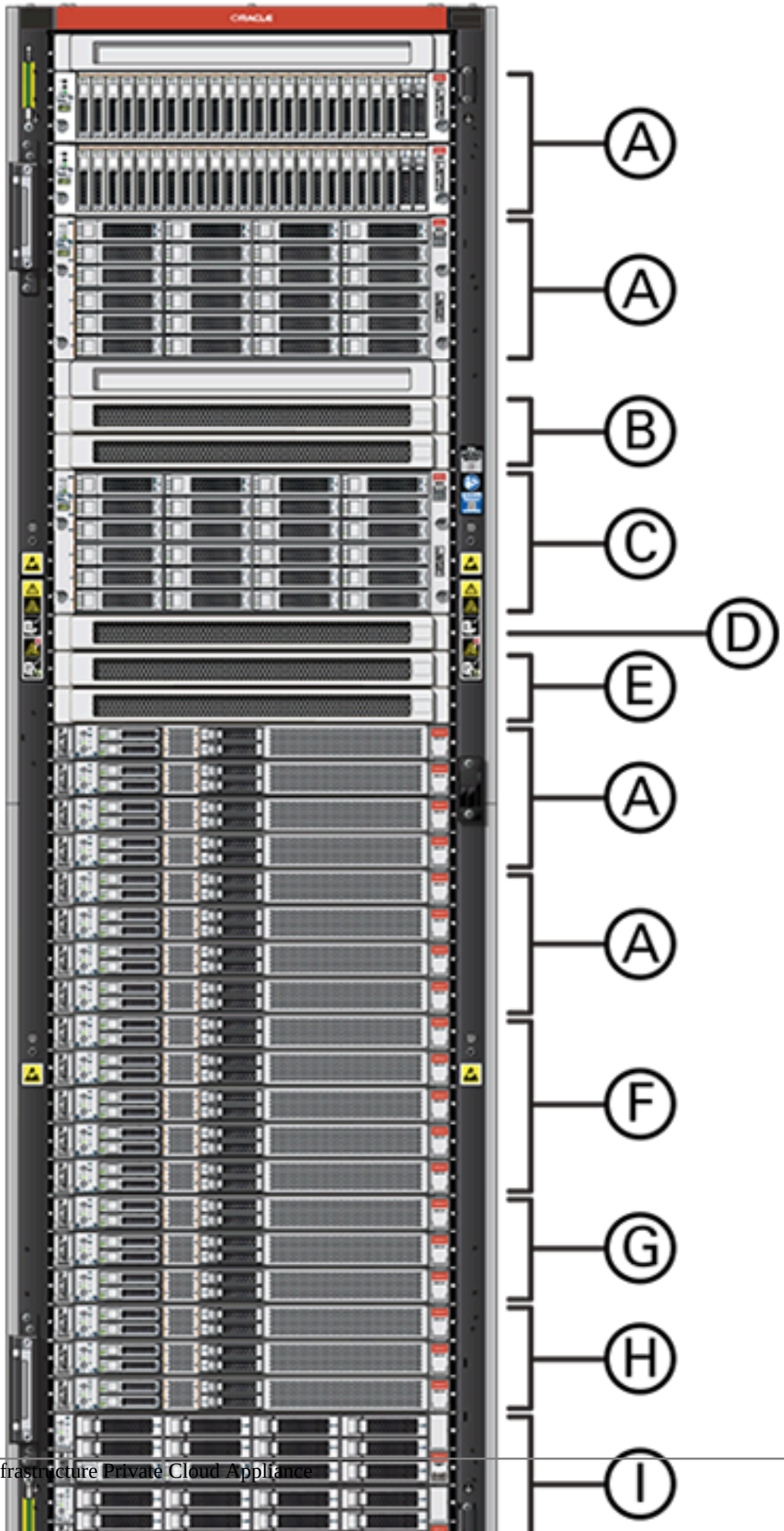

A
A
B
C
D
E
A
A
F
G
H
I

Figure Legend

Callout	Quantity	Description
A	1 - 4	flex bay can accommodate 1-4 compute nodes, or 1-2 storage enclosures
B	2	spine switch
C	1	Oracle Storage Drive Enclosure DE3-24C disk shelf
D	1	management switch
E	2	leaf switch
F	1 - 5	can accommodate 1 - 5 compute nodes
G	3	compute nodes 3 required for minimum configuration
H	3	management nodes
I	2	storage controllers

GPU Expansion Rack Configuration

Private Cloud Appliance with GPU expansion provides a scalable platform to build AI and graphics intensive applications in the private cloud environment.

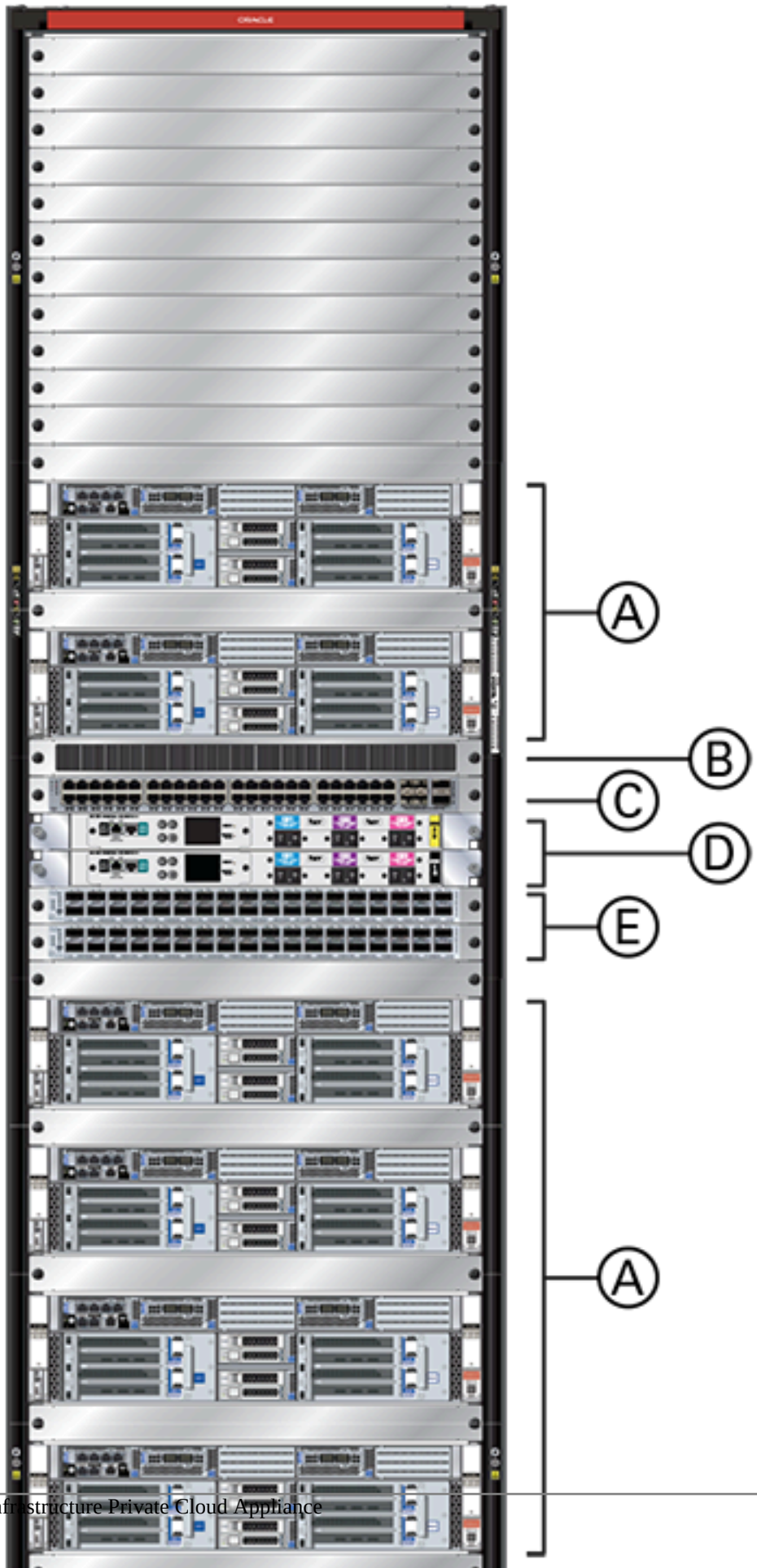
It's built to power the next generation of data center workloads, including:

- Generative AI inference: real time inferencing for multimodel generative AI pipelines (text, image, audio, video)
- LLM training and fine-tuning: accelerated performance for fine-tuning medium LLMs and training small LLMs with NVIDIA's transformer engine and FP8 support
- Graphics-intensive and VDI applications: 3D graphics and rendering workflows with NVIDIA’s RTX and ray tracing capabilities
- Digital twins using NVIDIA Omniverse: develop and operate complex 3D industrial digitization workflows
- Media streaming: increased encode/decode density and AV1 support for 4K video streaming
- HPC: scientific data analysis and simulation workloads with FP32 support

GPU expansions require appliance software version 3.0.2-b1325160 (March 2025) or newer. An X10-2c GPU expansion rack contains 1 to 6 X10-2c GPU L40S Compute Server nodes. To integrate with the base rack physical network infrastructure, two Cisco Nexus 9336C-FX2 leaf switches and a Cisco Nexus 9348GC-FXP management switch are installed. This rack does not include storage hardware.

X10-2c GPU Rack Configuration

The minimum configuration adds 1 factory-installed GPU expansion node. More nodes can be installed at the factory or after deployment. Cabling is preinstalled for a full rack configuration, regardless of the number of factory-installed nodes. A single expansion rack contains up to 6 GPU nodes. Two expansion racks can be connected to the base rack, for a maximum of 12 GPU nodes.


A
B
C
D
E
A

Callout	Quantity	Description
A	6	GPU node minimum configuration: 1, rack maximum: 6
B	1	brush filler - allows cable routing from the back to connectors in the front
C	1	management switch
D	2	universal power distribution unit (UPDU)
E	2	leaf switch
(none)	18	filler panel installed in the top 12 rack units and in empty spaces between components

Compute Expansion Rack Configuration

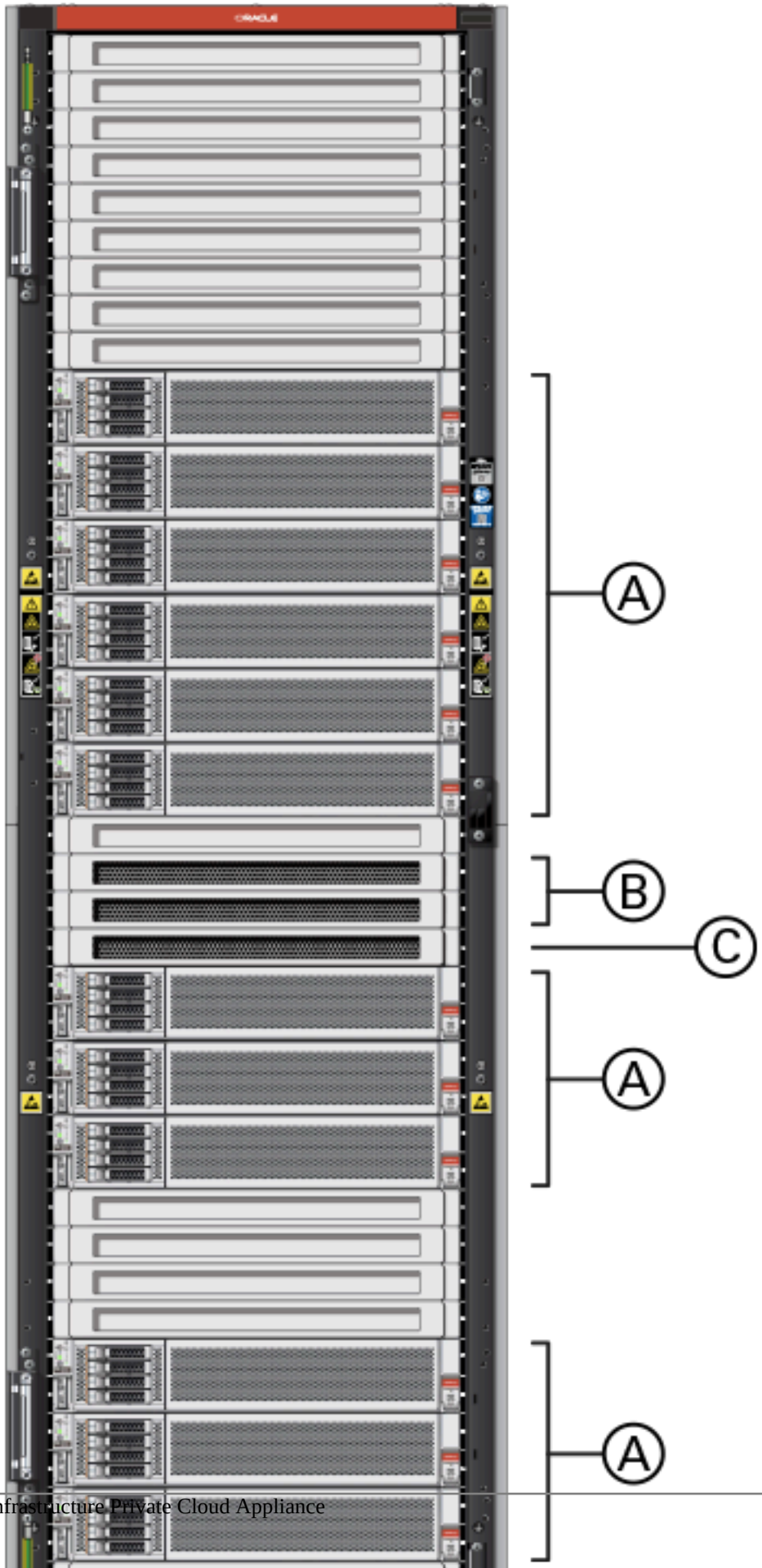
Private Cloud Appliance with compute expansion provides a scalable platform to increase the compute capacity of the system beyond the physical limitations of a base rack.

A compute expansion rack contains a minimum of 3 factory-installed compute nodes. More nodes can be installed after initial deployment, but must be added in sets of 3. Up to two expansion racks can be connected to a base rack. To integrate with the base rack physical network infrastructure, two Cisco Nexus 9336C-FX2 leaf switches and a Cisco Nexus 9348GC-FXP management switch are installed. This rack does not include storage hardware.

For installation information, see Expanding Private Cloud Appliance Capacity with a Compute Expansion Rack on page 123.

X10/X11 Compute Expansion Rack Configuration

The compute nodes are delivered in an expansion rack containing power distribution units (PDUs) and networking components to integrate the additional physical resources with the base rack. The minimum configuration adds 3 factory-installed expansion compute nodes. More nodes can be installed at the factory or after deployment, but must be added in sets of 3. Two expansion racks can be connected to the base rack, for a maximum combined total of 24 expansion compute nodes.


A
B
C
A
A

Callout	Quantity	Description
A	3-12	expansion compute nodes to be installed in sets of 3
B	2	leaf switch
C	1	management switch
(none)	15	filler panel installed in the top 9 rack units and in empty spaces between components

Server Nodes Installed in Private Cloud Appliance

Different models of server nodes might be installed in a Private Cloud Appliance system, depending on factory configuration and optional expansions after initial installation.

Each Private Cloud Appliance system is built with a cluster of management nodes to run the appliance controller software, and an number of compute nodes to provide CPU and memory capacity for hosted workloads.

Information about the system components that provide storage resources and network connectivity, can be found in these topics:

- Storage Hardware Installed in Private Cloud Appliance on page 27
- Private Cloud Appliance Network Infrastructure on page 28

Management Nodes

At the heart of a Private Cloud Appliance installation are three management nodes configured in a fully active cluster for maximum performance and availability. Each node is capable of running the same controller software and system-level services, and has equal access to the system configuration, so the appliance remains operational in case of individual node failures.

The management nodes, running the controller software, provide a foundation for the collection of services responsible for operating and administering Private Cloud Appliance. Responsibilities of the management cluster include monitoring and maintaining the system hardware, ensuring system availability, upgrading software and firmware, backing up and restoring the appliance, and managing disaster recovery.

The part of the system where the appliance infrastructure is controlled is called the Service Enclave, which runs on the management node cluster and can be accessed through the Service CLI or the Service Web UI. Access is closely monitored and restricted to privileged administrators. For more information, see Private Cloud Appliance Infrastructure Administration on page 145. For hardware configuration details of the management nodes, see Main Server Components by Model on page 26.

Compute Nodes

The compute nodes in Private Cloud Appliance are part of the hardware layer and provide the processing power and memory capacity to host compute instances.

When a system is initialized, compute nodes are automatically discovered by the admin service and put in the *Ready-to-Provision* state. Administrators can then provision the compute nodes through the Service Enclave, after which they are ready for use. When additional compute nodes are installed at a later stage, the new nodes are discovered, powered on, and discovered automatically by the same mechanism. Provisioning, monitoring, and other administrative operations are performed from the Service Web UI or Service CLI.

The minimum configuration of the base rack contains three compute nodes, but it can be expanded by three nodes at a time up to 18 compute nodes, if you choose to use all of your flex bay space for compute nodes. The system can

support up to 20 compute nodes in total, with two slots reserved for spares that you can request through an exception process. Contact your Oracle representative about expanding the compute capacity in your system.

For hardware configuration details of the compute nodes, see Main Server Components by Model on page 26. For more information about managing compute resources, see Compute Instances on page 634.

GPU Expansion Nodes

GPU capacity is added to Private Cloud Appliance by incorporating server nodes containing GPU hardware. GPU nodes are always installed in an expansion rack connected to the appliance base rack.

An X10-2c GPU expansion rack adds 1 to 6 GPU nodes to the Private Cloud Appliance installation. Two expansion racks can be connected to the base rack, for a maximum of 12 GPU nodes.

The X10-2c GPU L40S Compute Server is a 3 RU server with Intel Xeon Platinum 8480+ architecture, high-speed Ethernet connectivity, and four NVIDIA L40S GPUs with 48GB GDDR6 memory and 1466 peak FP8 TFLOPS.

For detailed component specifications, see the manufacturer website.

- Intel Xeon Platinum 8480+ Processor specifications
- NVIDIA L40S GPU data sheet

Unlike standard compute nodes, the GPU nodes are not automatically discovered. After physical installation, and assuming an active high-performance storage pool is present in the appliance configuration, the GPU expansion rack is activated by running a script from one of the management nodes. The script tasks are followed by the node installation process, which prepares the nodes for provisioning.

The new nodes are provisioned from the Service Enclave, using the UI or CLI. When the GPU nodes are up and running, appliance administrators can manage and monitor these in the same way as standard compute nodes. Users must deploy compute instances with a dedicated shape to take advantage of the GPUs. For more information, see Integrating GPU Expansion Nodes on page 166. For hardware configuration details of the GPU nodes, see Main Server Components by Model on page 26.

Main Server Components by Model

The following table lists the main components of the server models that might be installed in a Private Cloud Appliance rack, and which are supported by the current software release.

Oracle Server X9-2 Management Node	**Oracle Server X9-2 Compute Node**	**Oracle Server X10 Compute Node**	**X10-2c GPU L40S Compute Server**	**Oracle Server X11 Compute Node**
2x Intel Xeon Gold 5318Y CPU, 24 core, 2.10GHz, 165W	2x Intel Xeon Platinum 8358 CPU, 32 core, 2.60GHz, 250W	2x AMD EPYC 9J14 CPU, 96 core, 2.60GHz, 400W	2x Intel Xeon Platinum 8480+ CPU, 56 core, 2.00GHz, 350W	2x AMD EPYC 9J25 CPU, 96 core, 2.60GHz, 400W
16x 64 GB DDR4-3200 DIMMs (1TB total)	16x 64 GB DDR4-3200 DIMMs (1TB total)	24x 96 GB DDR5-4800 DIMMs (2.25 TB total)	16x 64 GB DDR5-4800 DIMMs (1TB total)	24x 96 GB DDR5-6400 DIMMs (2.3 TB total)
2x 240GB M.2 SATA boot devices configured as RAID1 mirror	2x 240GB M.2 SATA boot devices configured as RAID1 mirror	2x 480GB M.2 NVMe SSD boot devices configured as RAID1 mirror	--	2x 480GB M.2 NVMe SSD boot devices configured as RAID1 mirror
2x 3.84 TB NVMe SSD storage devices configured as RAID1 mirror	--	--	2x 3.84 TB NVMe SSD storage devices configured as RAID1 mirror	--

Oracle Server X9-2 Management Node	Oracle Server X9-2 Compute Node	Oracle Server X10 Compute Node	X10-2c GPU L40S Compute Server	Oracle Server X11 Compute Node
1x Ethernet port for remote management	1x Ethernet port for remote management	1x Ethernet port for remote management	2x Ethernet port for remote management	1x Ethernet port for remote management
1x Dual-port 100Gbit Ethernet NIC module in OCPv3 form	1x Dual-port 100Gbit Ethernet NIC module in OCPv3 form	2x Dual-port 100Gbit Ethernet NIC modules	2x Dual-port 200Gbit Ethernet NIC modules	2x Dual-port 200Gbit Ethernet NIC modules
--	--	--	4x NVIDIA L40S GPU, 48GB GDDR6, 350W	--
2x Redundant power supplies and fans	2x Redundant power supplies and fans	2x Redundant power supplies and fans	4x Redundant power supplies and fans	2x Redundant power supplies and fans

Storage Hardware Installed in Private Cloud Appliance

The physical storage in an Private Cloud Appliance system is provided by an Oracle ZFS Storage Appliance. The default configuration contains two storage controllers and a single DE3-24C capacity disk shelf. Optionally, storage can be expanded with high-capacity (DE3-24C) or high-performance (DE3-24P) disk shelves.

ZFS Storage Appliance

The Oracle ZFS Storage Appliance fulfills the role of 'system disk' for the entire appliance. It is crucial in providing storage space for the Private Cloud Appliance software, configuration, services, backups, and so on. Depending on the base rack configuration, a different storage appliance model is installed.

- X9 and X10 configuration: Oracle ZFS Storage Appliance ZS9-2
- X11 configuration: Oracle ZFS Storage Appliance ZS11-2

The default disk shelf in the appliance has 100+ TB of customer usable storage for public object storage, customer compute images, and customer block storage.

The hardware configuration of the ZFS Storage Appliance is as follows:

- Two clustered storage controller heads
- One fully populated disk chassis with twenty 18TB hard disks
- Two cache disks and two log disks installed in the disk shelf: 2x 200GB SSD and 2x 7.68TB SSD
- Mirrored configuration, for best data protection

The ZFS Storage Appliance is connected to the management subnet and the storage subnet. Both heads form a cluster in active-active configuration to guarantee continuation of service in the event that one storage head fails. The storage heads provide two IP addresses in the storage subnet: one for the default capacity storage pool, the other to access the optional performance (SSD) storage pool. Four management IP addresses are provided: one local to each controller head and one for each storage pool that follows the pool resources between controllers over takeover or failback events for convenient maintenance access. The primary mirrored capacity storage pool contains two projects, named `PCA` and `private_ostore_project`.

Optional Storage Expansion

You can optionally increase the storage in your system by adding disk shelves to a flex bay. If the base rack has no more space available, the expansion disk shelves can be installed in an additional rack cabinet. These storage expansion options are available:

Oracle Storage Drive Enclosure DE3-24C

The supported hardware configuration is as follows:

- Fully populated disk chassis with twenty 18TB hard disks
- Four cache disks installed in the disk shelf: 2x 200GB SSD and 2x 7.68TB SSD

Oracle Storage Drive Enclosure DE3-24P

The supported hardware configuration is as follows:

- Twenty 7.68TB SSD
- Two log disks installed in the disk shelf: 2x 200GB SSD
- Two drive bay fillers

For cabling information when adding disk shelves to the ZFS Storage Appliance, see Oracle ZFS Storage ZS9-2 HE Clustered to Oracle Storage Drive Enclosure DE3-24 Disk Shelves (4 HBAs).

After the disk shelves are installed and cabled to the storage controllers, the new storage resources must be added to the existing storage pools. Storage devices included in additional DE3-24C shelves must be added to the primary capacity storage pool, while devices included in DE3-24P shelves must be added to the optional high performance storage pool.

> **Caution:**
>
> With appliance software versions3.0.2-b1185392 and newer, the storage pools – capacity and/or high-performance – are no longer automatically reconfigured when new storage devices are installed.
>
> Because storage expansions are processed serially, regardless of how many disk shelves are added in a single operation, automated reconfiguration of the storage pools leads to an excessive number of spare drives. To ensure cost-effective and correctly balanced use of storage resources, it was decided to remove this automation. Storage expansions are best configured on a case by case basis, so that the number of spare drives can be adjusted to the specific storage configuration of the rack. Contact Oracle for assistance. Storage expansion scenarios are covered in the note with Doc ID 3020837.1.

Private Cloud Appliance Network Infrastructure

For network connectivity, Private Cloud Appliance relies on a physical layer that provides the necessary high-availability, bandwidth and speed. On top of this, a distributed network fabric composed of software-defined switches, routers, gateways, and tunnels enables secure and segregated data traffic – both internally between cloud resources, and externally to and from resources outside the appliance.

Device Management Network

The device management network provides internal access to the management interfaces of all appliance components. These have Ethernet connections to the 1Gbit management switch, and all receive an IP address from each of these address ranges:

- 100.96.0.0/23 – IP range for the Oracle Integrated Lights Out Manager (ILOM) service processors of all hardware components
- 100.96.2.0/23 – IP range for the management interfaces of all hardware components

To access the device management network, you connect a workstation to port 2 of the 1Gbit management switch and statically assign the IP address 100.96.3.254 to its connected interface. Alternatively, you can set up a permanent connection to the device management network from a data center administration machine, which is also referred to as a *bastion host*. From the bastion host, or from the (temporarily) connected workstation, you can reach the ILOMs and management interfaces of all connected rack components. For information about configuring the bastion host, see Connecting a Bastion Host to Private Cloud Appliance (Optional) on page 120.

> **Note:**
>
> Port 1 of the 1Gbit management switch is reserved for use by support personnel only.

Data Network

The appliance data connectivity is built on redundant 100Gbit switches in two-layer design similar to a leaf-spine topology. The leaf switches interconnect the rack hardware components, while the spine switches form the backbone of the network and provide a path for external traffic. Each leaf switch is connected to all the spine switches, which are also interconnected. The main benefits of this topology are extensibility and path optimization. A Private Cloud Appliance base rack contains two leaf and two spine switches.

The data switches offer a maximum throughput of 100Gbit per port. The spine switches use 5 interlinks (500Gbit); the leaf switches use 2 interlinks (200Gbit) and 2x2 crosslinks to each spine. Each server node is connected to both leaf switches in the rack, through the `bond0` interface that consists of two 100Gbit Ethernet ports in link aggregation mode. The two storage controllers are connected to the spine switches using 4x100Gbit connections.

For external connectivity, 5 ports are reserved on each spine switch. Four ports are available to establish the uplinks between the appliance and the data center network; one port is reserved to optionally segregate the administration network from the data traffic.

Administration Network

In an environment with elevated security requirements, you can optionally segregate administrative appliance access from the data traffic. The administration network physically separates configuration and management traffic from the operational activity on the data network by providing dedicated secured network paths for appliance administration operations. In this configuration, the entire Service Enclave can be accessed only over the administration network. This also includes the monitoring, metrics collection and alerting services, the API service, and all component management interfaces.

Setting up the administration network requires additional Ethernet connections from the next-level data center network devices to port 5 on each of the spine switches in the appliance. Inside the administration network, the spine switches must each have one IP address and a virtual IP shared between the two. A default gateway is required to route traffic, and NTP and DNS services must be enabled. The management nodes must be assigned host names and IP addresses in the administration network – one each individually and one shared between all three.

A separate administration network can be used with both static and dynamic routing. The use of a VLAN is supported, but when combined with static routing the VLAN ID must be different from the one configured for the data network.

Uplinks

The connections between the Private Cloud Appliance and the customer data center are called *uplinks*. They're physical cable connections between the spine switches in the rack and one or, preferably, two next-level network devices in the data center.

Besides the physical aspect, there's also a logical aspect to the uplinks: how traffic is routed between the rack and the external network it's connected to.

Physical Connection

On each spine switch, ports 1 to 4 can be used for uplinks to the data center network. For speeds of 10 or 25 Gbps, the spine switch port must be split using an MPO-to-4xLC breakout cable. For speeds of 40 or 100 Gbps each switch port uses a single MPO-to-MPO cable connection. The correct connection speed must be specified during initial setup so that the switch ports are configured with the appropriate breakout mode and transfer speed.

The uplinks are configured during system initialization, based on information you provide as part of the installation checklist. Unused spine switch uplink ports, including unused breakout ports, are disabled for security reasons. The table shows the supported uplink configurations by port count and speed, and the resulting total bandwidth.

Uplink Speed	Number of Uplinks per Spine Switch	Total Bandwidth
10 Gbps	1, 2, 4, 8, or 16	20, 40, 80, 160, or 320 Gbps

Uplink Speed	Number of Uplinks per Spine Switch	Total Bandwidth
25 Gbps	1, 2, 4, 8, or 16	50, 100, 200, 400, or 800 Gbps
40 Gbps	1, 2, or 4	80, 160, or 320 Gbps
100 Gbps	1, 2, or 4	200, 400, or 800 Gbps

Regardless of the number of ports and port speeds configured, you also select a topology for the uplinks between the spine switches and the data center network. This information is critical for the network administrator to configure link aggregation (port channels) on the data center switches. The table shows the available options.

Topology	Description
Triangle	In a triangle topology, all cables from both spine switches are connected to a single data center switch.
Square	In a square topology, two data center switches are used. All outbound cables from a given spine switch are connected to the same data center switch.
Mesh	In a mesh topology, two data center switches are used as well. The difference with the square topology is that uplinks are created in a cross pattern. Outbound cables from each spine switch are connected in pairs: one cable to each data center switch.

Topology

The physical topology for the uplinks from the rack to the data center network is dependent on bandwidth requirements and available data center switches and ports. Connecting to a single data center switch implies that you select a triangle topology. To increase redundancy you distribute the uplinks across a pair of data center switches, selecting either a square or mesh topology. Each topology allows you to start with a minimum bandwidth, which you can scale up with increasing need. The maximum bandwidth is 800 Gbps, assuming the data center switches, transceivers and cables allow it.

The following diagrams provide a simplified view of supported topologies, and can be used as initial guidance to integrate the rack into the data center network. Use the diagrams and the notes to determine the appropriate cabling and switch configuration for your installation. For more detailed uplink configuration examples, which have been tested by Oracle, see Reference Topologies on page 34.

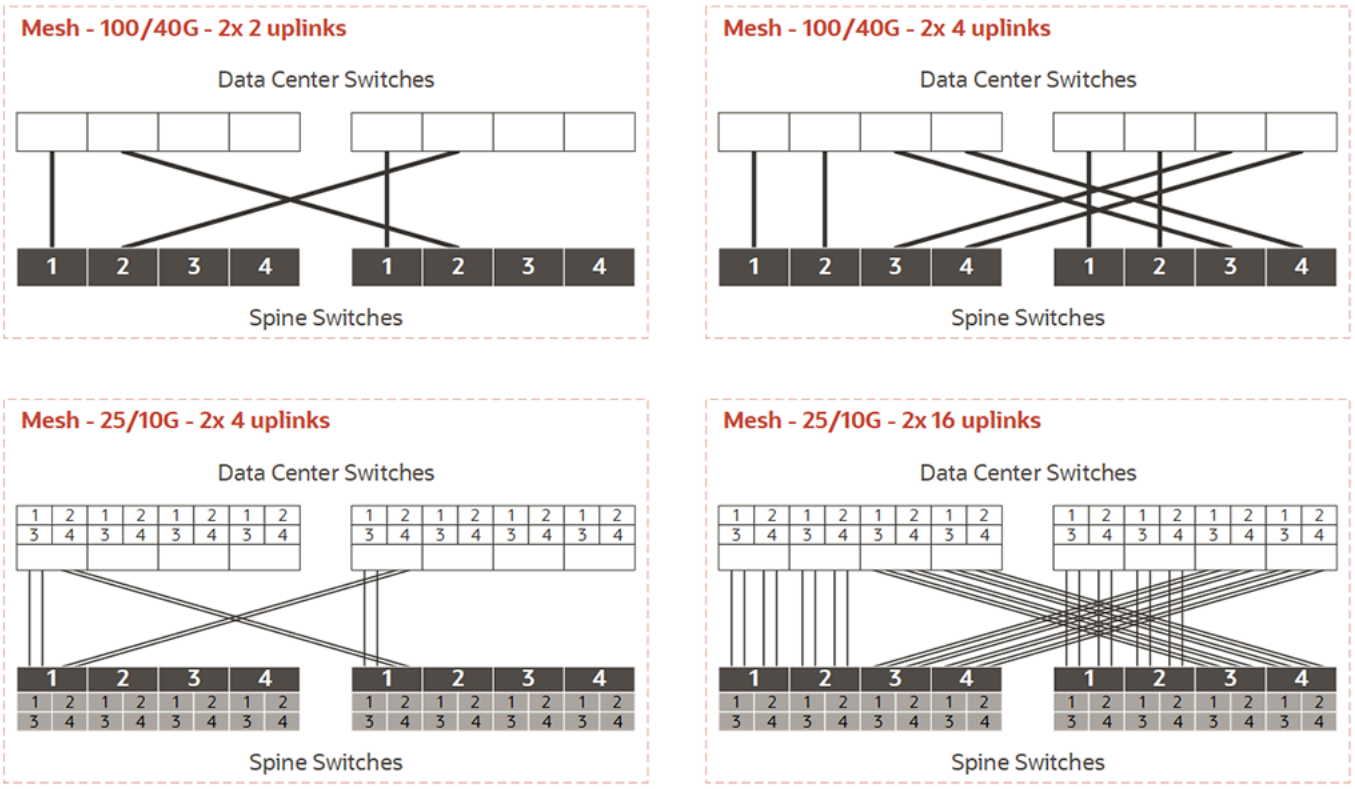


Diagram Notes

On the rack side there are two spine switches that must be connected to the data center network. Both spine switches must have identical port and cable configurations. In each example, the spine switches are shown at the bottom, with all uplink ports identified by their port number. The lines represent outgoing cable connection to the data center switches, which are shown at the top of each example without port numbers.

Cabling Pattern and Port Speed

There are six examples in total, organized in two rows by three columns.

- The top row shows cabling options based on full-port 100-Gbps or 40-Gbps connections. The bottom row shows cabling options using breakout ports at 25-Gbps or 10-Gbps speeds; the smaller boxes numbered 1-4 represent the breakout connections for each of the four main uplink ports per spine switch.
- The third column shows a triangle topology with full-port connections and breakout connections. The difference with column two is that all uplinks are connected to a single data center switch. The total bandwidth is the same, but the triangle topology lacks data center switch redundancy.
- There are no diagrams for the square topology. The square cabling configuration is similar to the mesh examples, but without the crossing patterns. Visually, all connectors in the diagrams would be parallel. In a square topology, all outgoing cables from one spine switch are connected port for port to the same data center switch. Unlike mesh, square implies that each spine switch is peered with only one data center switch.

Link Count

When connecting the uplinks, you are required to follow the spine switch port numbering. Remember that both spine switches are cabled identically, so each uplink or connection corresponds with a pair of cables.

- With one cable per spine port, using 100 or 40 Gbps transceivers, the first uplink pair uses spine switch ports numbered 1, the second uses port 2, and so on. In this configuration, the maximum number of uplinks is four per spine switch.
- When breakout cables are used, with 25 or 10-Gbps port speeds, the first uplink pair uses port 1/1. With two or four uplinks per spine switch there's still only one full port in use. When you increase the uplink count to 8 per

spine switch, ports 1/1-2/4 are in use. At 16 uplinks per spine switch, all breakout connections of all four reserved ports will be in use.

- In a mesh topology, a particular cabling pattern must be followed: connect the first half of all uplinks to one data center switch, and the second half to the other data center switch. For example: if you have four uplinks then the first two go to the same switch; if you have eight uplinks (not shown in the diagrams) then the first four go to the same switch; if you have 16 uplinks then the first eight go to the same switch.

Mesh Topology Implications

- In a mesh topology, the spine switch configuration expects that the first half of all uplinks is connected to one data center switch, and the second half to the other data center switch. When you initially connect the rack to your data center network, it is straightforward to follow this pattern.
- However, if you increase the number of uplinks at a later time, the mesh cabling pattern has a significant implication for the existing uplinks. Compare the diagrams in the first two columns: when you double the uplink count, half of the existing connections must be moved to the other data center switch. For 100/40-Gbit uplinks, recabling is only required when you increase the link count from 2 to 4. Because of the larger number of cables, 25/10-Gbit uplinks require more re-cabling: when increasing uplink count from 2 to 4, from 4 to 8, and from 8 to 16.

Logical Connection

The logical connection between the rack and the data center is implemented entirely in layer 3. In the OSI model (Open Systems Interconnection model), layer 3 is known as the network layer, which uses the source and destination IP address fields in its header to route traffic between connected devices.

Private Cloud Appliance supports two logical connection options: you must choose between static routing and dynamic routing. Both routing options are supported by all three physical topologies.

Connection Type	Description
Static Routing	When static routing is selected, all egress traffic goes through a single default gateway IP address configured on data center network devices. This gateway IP address must be in the same subnet as the rack uplink IP addresses, so it is reachable from the spine switches. The data center network devices can use SVIs (Switch Virtual Interfaces) with VLAN IDs in the range of 2-3899. All gateways configured within a virtual cloud network (VCN) will automatically have a route rule to direct all traffic intended for external destination to the IP address of the default gateway.
Dynamic Routing	When dynamic routing is selected, BGP (Border Gateway Protocol) is used to establish a TCP connection between two Autonomous Systems: the rack network and the data center network. This configuration requires a registered or private ASN (Autonomous System Number) on each side of the connection. Private Cloud Appliance BGP configuration uses ASN 136025 by default, this can be changed during initial configuration. For BGP routing, two routing devices in the data center must be connected to the two spine switches in the rack. Corresponding interfaces (port channels) between the spine switches and the data center network devices must be in the same subnet. It is considered good practice to use a dedicated /30 subnet for each point-to-point circuit, which is also known as a route hand-off network. This setup provides redundancy and multipathing. Dynamic routing is also supported in a triangle topology, where both spine switches are physically connected to the same data center network device. In this configuration, two BGP sessions are still established: one from each spine switch. However, this approach reduces the level of redundancy.

Supported Routing Designs

The following table shows which routing designs are supported depending on the physical topology in your data center and the logical connection you choose to implement.

Note that link aggregation across multiple devices (vPC or MLAG) is only supported with static routing. When dynamic routing is selected, link aggregation is restricted to ports of the same switch.

When the uplinks are cabled in a mesh topology, a minimum of 2 physical connections per spine switch applies. To establish BGP peering, 2 subnets are required. If the uplink count changes, the port channels are reconfigured but the dedicated subnets remain the same.

Logical Connection	Physical Topology	Routing Design		
		Single Subnet	Dual Subnet	vPC/MLAG
Static Routing	Square	Yes	Yes	Yes
	Mesh	Yes	Yes	Yes
	Triangle	Yes	Yes	Yes
Dynamic Routing	Square	Yes	--	--
	Mesh	--	Yes	--
	Triangle	Yes	--	--

Uplink Protocols

On Private Cloud Appliance, the uplinks to the data center run various protocols to provide redundancy and reduce link failure detection and recovery times on these links. These protocols work with the triangle, square, or mesh topologies.

The suite of uplink protocols include:

- Bidirectional Forwarding Detection (BFD)
- Virtual Router Redundancy Protocol (VRRP)
- Hot Spare Router Protocol (HSRP)
- Equal Cost Multipath (ECMP)

Each is briefly described in the following sections of this topic.

BFD

In most router networks, connection failures are detected by loss of the "hello" packets sent by routing protocols. However, detection by this method often takes more than one second, routing a lot of packets on high-speed links to a destination that they can't reach, which burdens link buffers. Increasing the "hello" packet rate burdens the router CPU.

Bidirectional Forwarding Detection (BFD) is a built-in mechanism that alerts routers at the end of a failed link that there's a problem more quickly than any other mechanism, reducing the load on buffers and CPUs. BFD works even in situations where there are switches or hubs between the routers.

BFD requires no configuration and has no user-settable parameters.

VRRPv3

The Virtual Router Redundancy Protocol version 3 (VRRPv3) is a networking protocol that uses the concept of a virtual router to group physical routers together and make them appear as one to participating hosts. This increases the availability and reliability of routing paths through automatic default gateway selections on an IP subnetwork.

With VRRPv3, the primary/active and secondary/standby routers act as one virtual router. This virtual router becomes the default gateway for any host on the subnet participating in VRRPv3. One physical router in the group becomes the primary/active router for packet forwarding. However, if this router fails, another physical router in the group takes over the forwarding role, adding redundancy to the router configuration. The VRRPv3 "network" is limited to the local subnet and doesn't advertise routes beyond the local subnet.

HSRP

Cisco routers often use a redundancy protocol called the Hot Spare Router Protocol (HSRP) to improve router availability. Similar to the methods of VRRP, HSRP groups physical routers into a single virtual router. The failure of a physical default router results in another router using HSRP to take over the default forwarding of packets without stressing the host device.

ECMP

Equal Cost Multipath (ECMP) is a way to make better use of network bandwidth, especially in more complex router networks with many redundant links.

Normally, router networks with multiple router paths to another destination network choose one active route to a gateway router as the "best" path and use the other paths as a standby in case of failure. The decision about which path to a network gateway router to use is usually determined by its "cost" from the routing protocol perspective. In cases where the cost over several links to reach network gateways are equal, the router chooses one based on some criteria. This makes routing decisions easy but wastes network bandwidth as network links on paths not chosen sit idle.

ECMP is a way to send traffic on multiple path links with equal cost, making more efficient use of network bandwidth.

Reference Topologies

These detailed topologies are examples of uplink configurations that have been tested and are known to work as expected. Each example includes a diagram for illustration, and configuration guidelines.

When preparing for the installation of Private Cloud Appliance, you decide how the system will be connected to the data center network: uplink count, cabling pattern, connection speed, routing design, and so on. Decisions about uplinks are based on the network infrastructure in your data center, and the bandwidth and availability requirements of your Private Cloud Appliance environment. These reference topologies help you make informed decisions for your specific environment, and provide guidance to configure your switches for the topology you select.

> **Caution:**
>
> The configuration parameters provided in the reference topologies are intended as guidelines for network administrators. The exact configuration details of your specific environment must be aligned with your data center network design.

Mesh Topology with ECMP Static Routing

ECMP Mesh enables Layer 3 network deployment according to industry-proven best practices. This uplink topology is highly recommended.

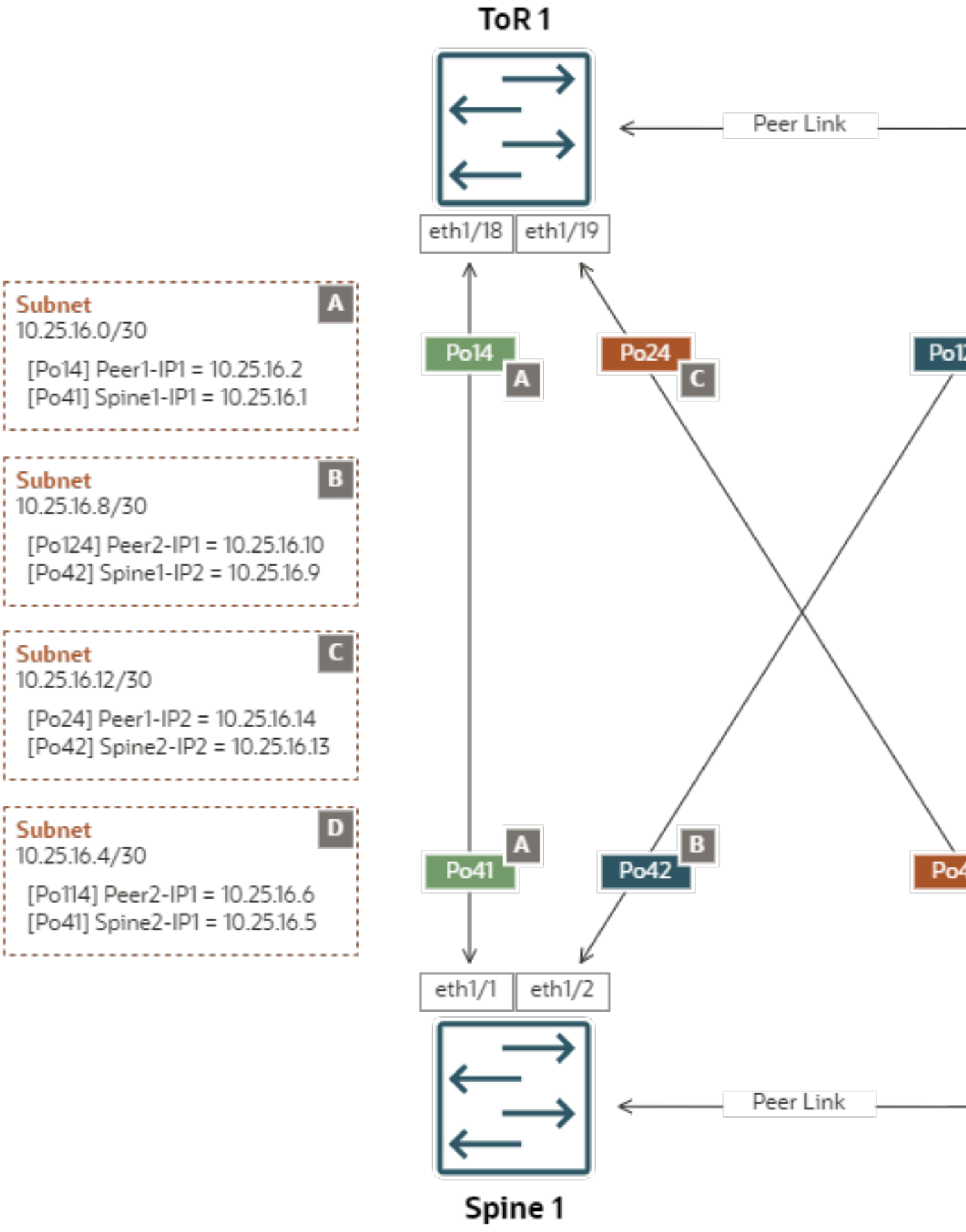
Configuration Properties

- Mesh topology – each spine switch is connected to two independent data center switches
- Static routing – all egress traffic from an uplink goes through a single gateway IP configured on its peer network device in the data center
- ECMP – bandwidth optimization across multiple redundant links or paths
- Separate /30 subnets – each uplink connects one spine switch port channel to one data center switch port channel in a /30 subnet

Topology Highlights

- All uplinks are configured as LACP/active port channels with *rate=fast*

    - Port channel **Po41** represents the first set of links on both spine switches. They **connect straight** to the corresponding ToR switches.
    - Port channel **Po42** represents the second set of links on both spine switches. They **cross-connect** to the corresponding ToR switches.
- ToR switch ports connecting to the spine switches must be set up in access mode. Spanning tree protocol must be disabled.
- Requires 4 unique subnets: a /30 subnet size is recommended, but /31 is possible if the ToR switches support it.
    - Equal cost static routes to both ToR switches are set up automatically.
    - Egress traffic can hash to any of the 4 uplinks.
    - It is NOT possible to isolate specific VCN/VM egress traffic through one particular uplink.


ToR 1
Peer Link
eth1/18
eth1/19
Subnet
10.25.16.0/30
[Po14] Peer1-IP1 = 10.25.16.2
[Po41] Spine1-IP1 = 10.25.16.1
A
Subnet
10.25.16.8/30
[Po124] Peer2-IP1 = 10.25.16.10
[Po42] Spine1-IP2 = 10.25.16.9
B
Subnet
10.25.16.12/30
[Po24] Peer1-IP2 = 10.25.16.14
[Po42] Spine2-IP2 = 10.25.16.13
C
Subnet
10.25.16.4/30
[Po114] Peer2-IP1 = 10.25.16.6
[Po41] Spine2-IP1 = 10.25.16.5
D
Po14
A
Po24
C
Po41
A
Po42
B
eth1/1
eth1/2
Peer Link
Spine 1

Detailed Spine Switch Configuration Example

- Spine Switch 1

```
interface port-channel41
  description "customer uplink"
  no switchport
  mtu 9216
  speed 10000
  no negotiate auto
  ip access-group ingress-ports-acl in
  no ip redirects
  ip address 10.25.16.1/30
  ip nat outside

interface port-channel42
  description "customer uplink 2"
  no switchport
  mtu 9216
  speed 10000
  no negotiate auto
  ip access-group ingress-ports-acl in
  no ip redirects
  ip address 10.25.16.9/30
  ip nat outside

ip route 0.0.0.0/0 po41 10.25.16.2 20
ip route 0.0.0.0/0 po42 10.25.16.10 20
```

  Routes added:

```
0.0.0.0/0, ubest/mbest: 2/0
  *via 10.25.16.2, [20/0], 6d08h, static
  *via 10.25.16.10, [20/0], 6d08h, static
```

- Spine Switch 2

```
interface port-channel41
  description "customer uplink"
  no switchport
  mtu 9216
  speed 10000
  no negotiate auto
  ip access-group ingress-ports-acl in
  no ip redirects
  ip address 10.25.16.5/30
  ip nat outside

interface port-channel42
  description "customer uplink 2"
  no switchport
  mtu 9216
  speed 10000
  no negotiate auto
  ip access-group ingress-ports-acl in
  no ip redirects
  ip address 10.25.16.13/30
  ip nat outside

ip route 0.0.0.0/0 po41 10.25.16.6 20
ip route 0.0.0.0/0 po42 10.25.16.14 20
```

Routes added:

```
0.0.0.0/0, ubest/mbest: 2/0
  *via 10.25.16.6, [20/0], 6d07h, static
  *via 10.25.16.14, [20/0], 6d07h, static
```

Mesh Topology with VRRP Static Routing (Not Supported)
VRRP Mesh would have a negative impact on network performance. This uplink topology is NOT recommended.

When the uplinks are set up in a mesh topology, there are 4 distinct Layer 3 uplinks to the ToR switches. In a Virtual Router Redundancy Protocol switch configuration, egress traffic is typically routed through the uplink with the primary/active role. Thus, a VRRP mesh configuration would reduce the egress bandwidth to only 25 percent of the actual capacity. Therefore, this configuration is considered not supported.

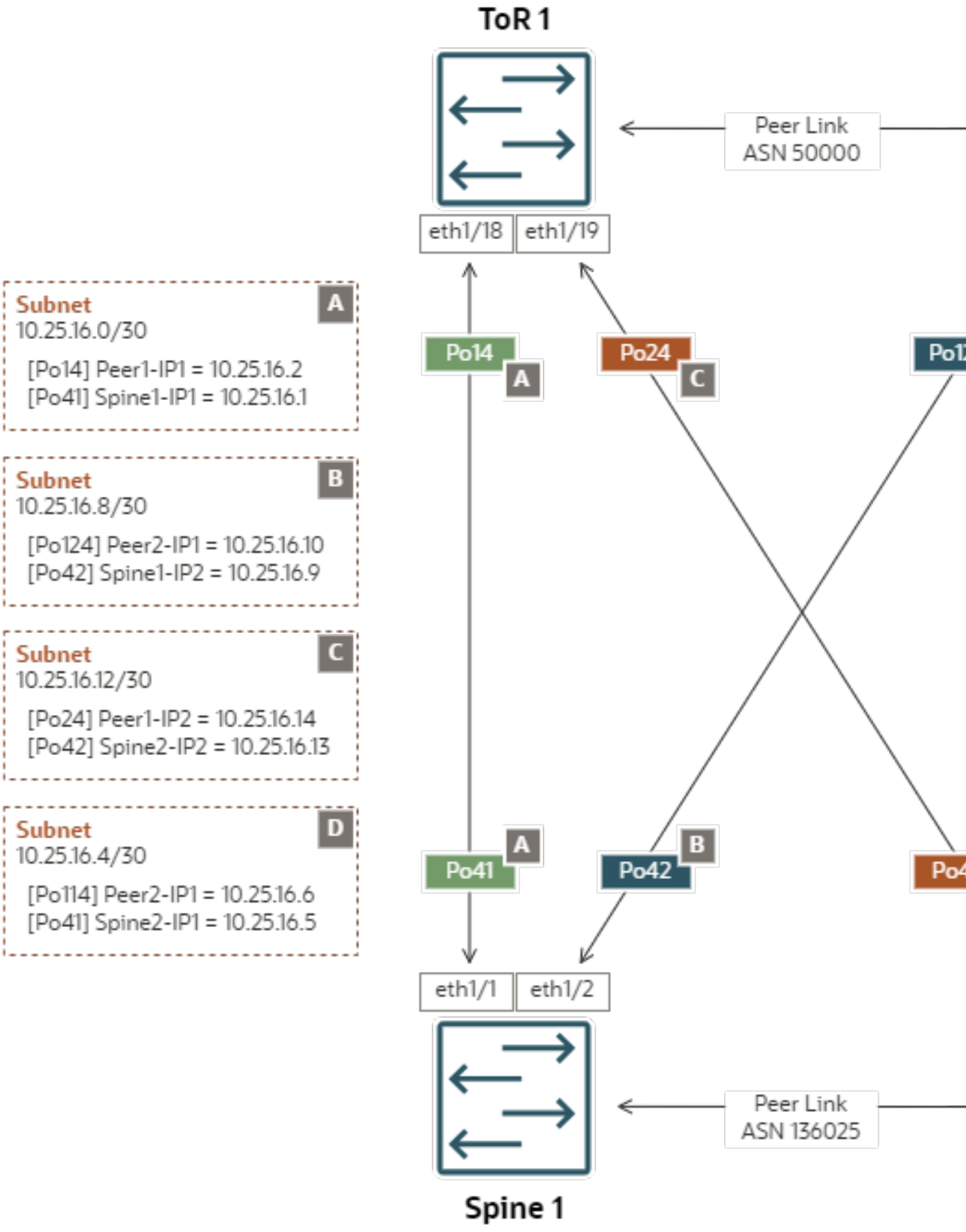
Mesh Topology with Dynamic Routing
Dynamic Mesh enables Layer 3 network deployment according to industry-proven best practices. This uplink topology is highly recommended.

Configuration Properties

- Mesh topology – each spine switch is connected to two independent data center switches
- Dynamic routing – both peered Autonomous Systems, the appliance and the data center, exchange routing information using eBGP (external Border Gateway Protocol). The best routing path is dynamically adjusted based on network availability information advertized by each AS.
- Separate /30 subnets – each uplink connects one spine switch port channel to one data center switch port channel in a /30 subnet

Topology Highlights

- All uplinks are configured as LACP/active port channels with *rate=fast*
  - Port channel **Po41** represents the first set of links on both spine switches. They **connect straight** to the corresponding ToR switches.
  - Port channel **Po42** represents the second set of links on both spine switches. They **cross-connect** to the corresponding ToR switches.
- ToR switch ports connecting to the spine switches must be set up in access mode. Spanning tree protocol must be disabled.
- Requires 4 unique subnets: a /30 subnet size is recommended, but /31 is possible if the ToR switches support it.
  - Two eBGP peering sessions are established between each spine and both ToR switches.
  - Egress traffic can hash to any of the 4 uplinks.
  - It is NOT possible to isolate specific VCN/VM egress traffic through one particular uplink.


ToR 1
Peer Link
ASN 50000
eth1/18
eth1/19
Subnet
10.25.16.0/30
A
[Po14] Peer1-IP1 = 10.25.16.2
[Po41] Spine1-IP1 = 10.25.16.1
Subnet
10.25.16.8/30
B
[Po124] Peer2-IP1 = 10.25.16.10
[Po42] Spine1-IP2 = 10.25.16.9
Subnet
10.25.16.12/30
C
[Po24] Peer1-IP2 = 10.25.16.14
[Po42] Spine2-IP2 = 10.25.16.13
Subnet
10.25.16.4/30
D
[Po114] Peer2-IP1 = 10.25.16.6
[Po41] Spine2-IP1 = 10.25.16.5
Po14
A
Po24
C
Po41
A
Po42
B
eth1/1
eth1/2
Peer Link
ASN 136025
Spine 1

Spine Switch Configuration Details

- Spine Switch 1

```
interface port-channel41
  description "customer uplink"
  no switchport
  mtu 9216
  speed 10000
  no negotiate auto
  ip access-group ingress-ports-acl in
  no ip redirects
  ip address 10.25.16.1/30
  ip nat outside

interface port-channel42
  description "customer uplink 2"
  no switchport
  mtu 9216
  speed 10000
  no negotiate auto
  ip access-group ingress-ports-acl in
  no ip redirects
  ip address 10.25.16.9/30
  ip nat outside

router bgp 136025
  router-id 10.25.16.1
  neighbor 10.25.16.2
    bfd singlehop
    remote-as 50000
    address-family ipv4 unicast
  neighbor 10.25.16.10
    bfd singlehop
    remote-as 50000
    address-family ipv4 unicast

BGP Sessions:
ASN 136025
VRF default, local ASN 136025

Neighbor        ASN    Flaps LastUpDn|LastRead|LastWrit  St  Port(L/R)
 Notif(S/R)
10.25.16.2     50000  0     1w4d    |00:00:50|00:00:20  E   34408/179
   0/0
10.25.16.10    50000  0     1w4d    |00:00:43|00:00:20  E   57322/179
   0/0
```

- Spine Switch 2

```
interface port-channel41
  description "customer uplink"
  no switchport
  mtu 9216
  speed 10000
  no negotiate auto
  ip access-group ingress-ports-acl in
  no ip redirects
  ip address 10.25.16.5/30
  ip nat outside

interface port-channel42
  description "customer uplink 2"
  no switchport
```

```
  mtu 9216
  speed 10000
  no negotiate auto
  ip access-group ingress-ports-acl in
  no ip redirects
  ip address 10.25.16.13/30
  ip nat outside

router bgp 136025
  router-id 10.25.16.5
  neighbor 10.25.16.6
    bfd singlehop
    remote-as 50000
    address-family ipv4 unicast
  neighbor 10.25.16.14
    bfd singlehop
    remote-as 50000
    address-family ipv4 unicast

BGP Sessions:
ASN 136025
VRF default, local ASN 136025

Neighbor        ASN    Flaps LastUpDn|LastRead|LastWrit  St  Port(L/R)
 Notif(S/R)
10.25.16.6      50000  0     1w4d    |00:00:50|00:00:20  E   34408/179
   0/0
10.25.16.14     50000  0     1w4d    |00:00:43|00:00:20  E   57322/179
   0/0
```

Square Topology with ECMP Static Routing

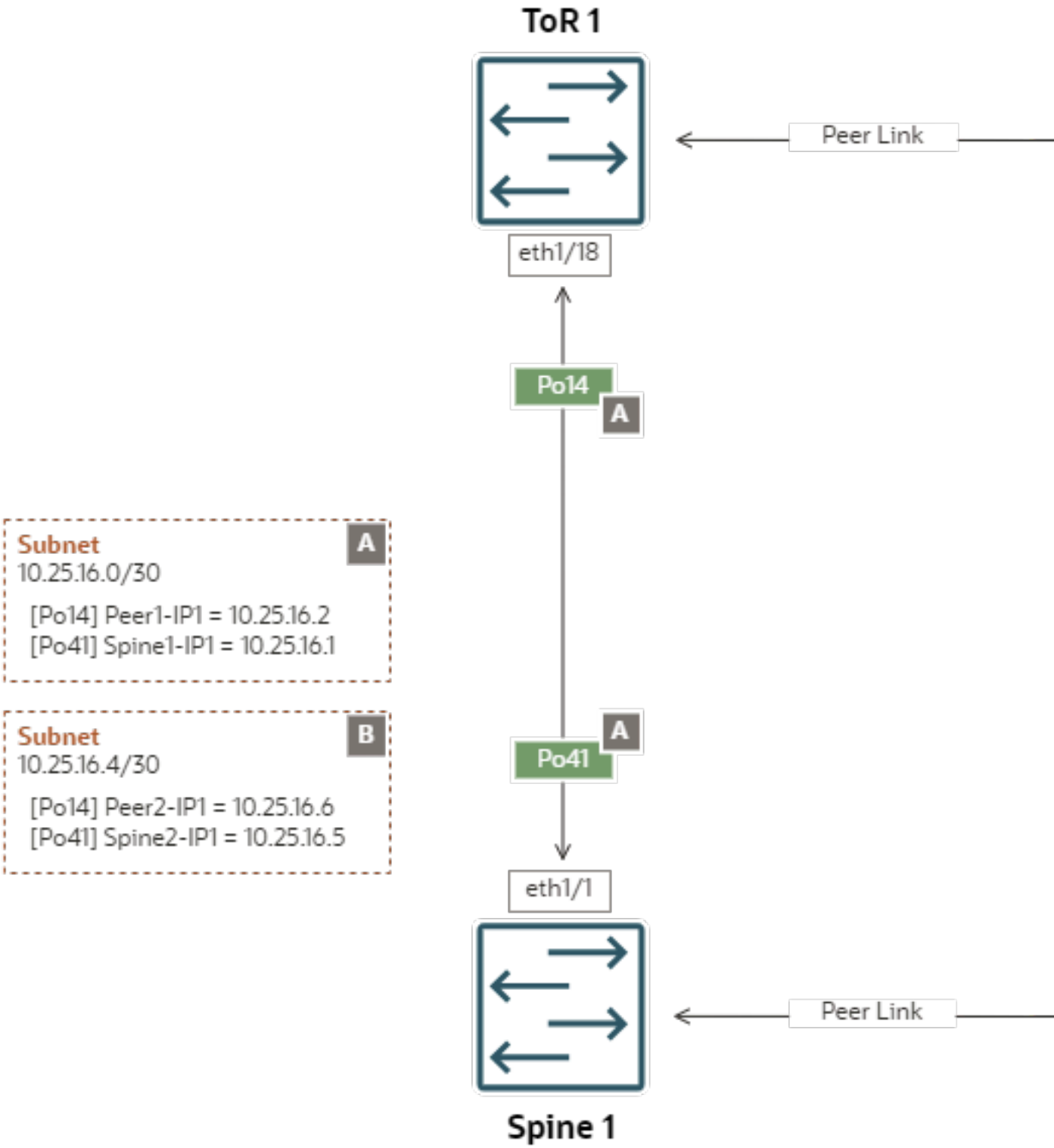
ECMP Square enables Layer 3 network deployment according to industry-proven best practices. This uplink topology is highly recommended.

Configuration Properties

- Square topology – each spine switch is connected to a different independent data center switch
- Static routing – all egress traffic from an uplink goes through a single gateway IP configured on its peer network device in the data center
- ECMP – bandwidth optimization across multiple redundant links or paths
- Separate /30 subnets – each uplink connects one spine switch port channel to one data center switch port channel in a /30 subnet

Topology Highlights

- All uplinks are configured as LACP/active port channels with *rate=fast*
- ToR switch ports connecting to the spine switches must be set up in access mode. Spanning tree protocol must be disabled. The ToR switches must NOT be configured with vPC.
- Requires 2 unique subnets: a /30 subnet size is recommended, but /31 is possible if the ToR switches support it.
  - Equal cost static routes to both ToR switches are set up automatically.
  - Egress traffic can hash to any of the 2 uplinks.
  - It is NOT possible to isolate specific VCN/VM egress traffic through one particular uplink.



Detailed Spine Switch Configuration Example

- Spine Switch 1

```
interface port-channel41
  description "customer uplink"
  no switchport
  mtu 9216
  speed 10000
  no negotiate auto
  ip access-group ingress-ports-acl in
  no ip redirects
  ip address 10.25.16.1/30
  ip nat outside
```

Routes added:

```
0.0.0.0/0, ubest/mbest: 2/0
  *via 10.25.16.2, [20/0], 6d08h, static
  *via 10.25.16.6, [100/0], 6d08h, static
```

- Spine Switch 2

```
interface port-channel41
  description "customer uplink"
  no switchport
  mtu 9216
  speed 10000
  no negotiate auto
  ip access-group ingress-ports-acl in
  no ip redirects
  ip address 10.25.16.5/30
  ip nat outside
```

Routes added:

```
0.0.0.0/0, ubest/mbest: 2/0
  *via 10.25.16.6, [20/0], 6d07h, static
  *via 10.25.16.2, [100/0], 6d07h, static
```

Triangle Topology with ECMP Static Routing

ECMP Triangle enables peering with a single data center router or switch.

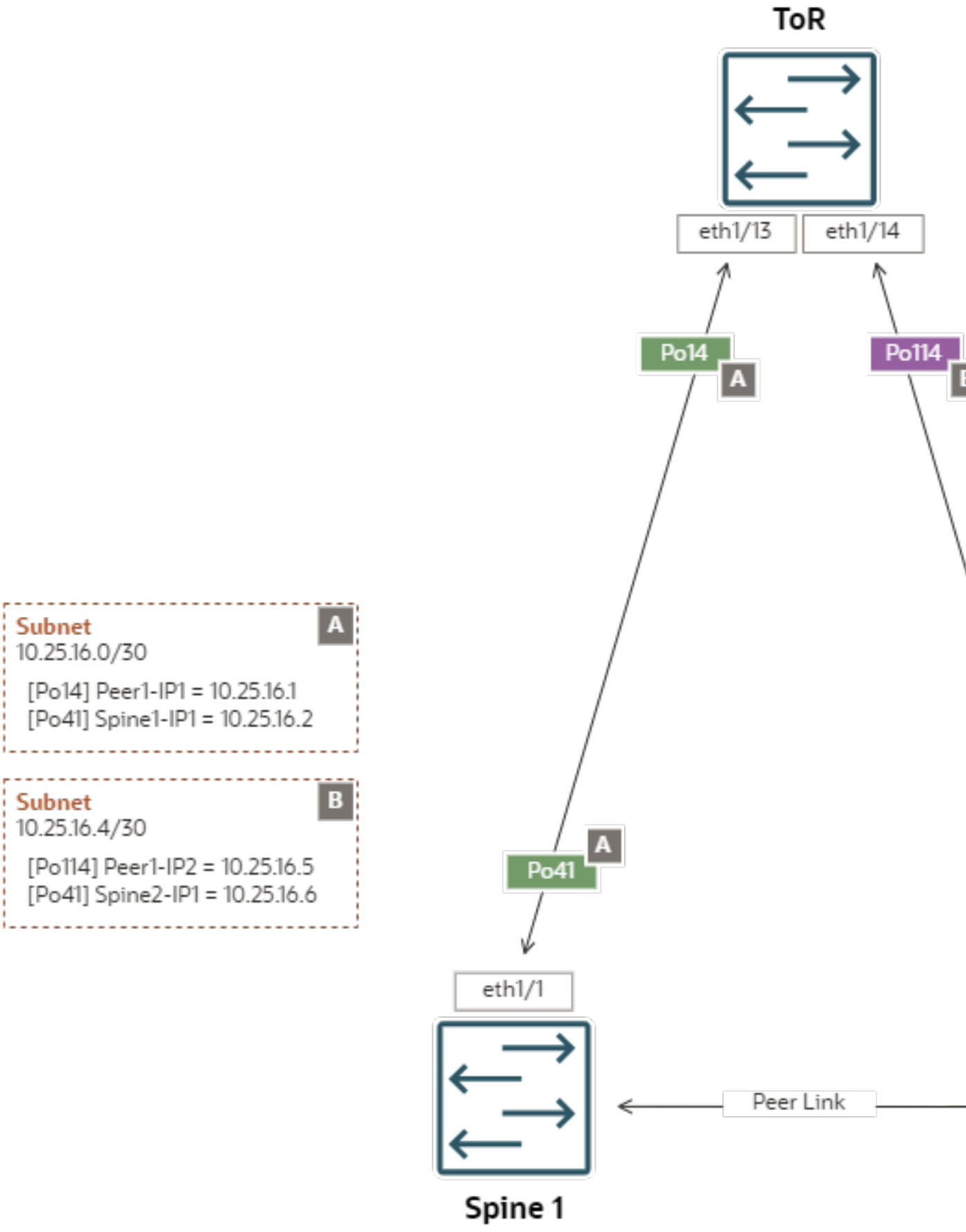
Configuration Properties

- Triangle topology – both spine switches are connected to a single data center switch
- Static routing – all egress traffic from an uplink goes through a single gateway IP configured on its peer network device in the data center
- ECMP – bandwidth optimization across multiple redundant links or paths
- Separate /30 subnets – each uplink connects one spine switch port channel to one data center switch port channel in a /30 subnet

Topology Highlights

- All uplinks are configured as LACP/active port channels with *rate=fast*
- Port channel **Po41** represents the set of links configured on both spine switches. Port channels **Po14** and **Po114** represent the corresponding sets of links configured on the ToR switch.
- ToR switch ports connecting to the spine switches must be set up in access mode. Spanning tree protocol must be disabled.
- Requires 2 unique subnets: a /30 subnet size is recommended, but /31 is possible if the ToR switch supports it.
  - Equal cost static routes to the ToR switch are set up automatically.
  - Egress traffic can hash to any of the 2 uplinks.

- It is NOT possible to isolate specific VCN/VM egress traffic through one particular uplink.


ToR
eth1/13
eth1/14
Po14
A
Po114
Subnet
10.25.16.0/30
A
[Po14] Peer1-IP1 = 10.25.16.1
[Po41] Spine1-IP1 = 10.25.16.2
Subnet
10.25.16.4/30
B
[Po114] Peer1-IP2 = 10.25.16.5
[Po41] Spine2-IP1 = 10.25.16.6
Po41
A
eth1/1
Peer Link
Spine 1

Detailed Spine Switch Configuration Example

- Spine Switch 1

```
interface port-channel41
  no shutdown
  mtu 9216
  ip address 10.25.16.2/30
  no ipv6 redirects
  ip proxy-arp
  ip nat outside

ip route 0.0.0.0/0 po41 10.25.16.1 20
ip route 0.0.0.0/0 po41 10.25.16.5 100
```

- Spine Switch 2

```
interface port-channel41
  no shutdown
  mtu 9216
  ip address 10.25.16.6/30
  no ipv6 redirects
  ip proxy-arp
  ip nat outside

ip route 0.0.0.0/0 po41 10.25.16.5 20
ip route 0.0.0.0/0 po41 10.25.16.1 100
```

Topology for Segregated Administration Traffic

With segregated administration traffic on Private Cloud Appliance, you cable and configure the dedicated connections for the administration network using the same topologies as the data network. All the same principles apply.

The administration network topology does not need to match the data network: it can be based on another reference topology, using a different number of physical ports operating at a different link speed. The uplink configuration does require different subnets, IP addresses and port channels, but otherwise looks exactly like a data network.

On the spine switches, the administration network is hard-coded to use port channel 45 (Po45), whereas the data network uses Po41 and Po42. On the data center side of the uplinks you are allowed to use any valid port channel ID that fits the existing data center configuration.

Reserved Network Resources

On Private Cloud Appliance, the network infrastructure and system components need many IP addresses and several VLANs for internal operation. It's critical to avoid conflicts with the addresses in use in the customer data center and the CIDR ranges configured in the virtual cloud networks (VCNs).

These IP address ranges are reserved for internal use by Private Cloud Appliance:

Reserved IP Addresses	Description
CIDR blocks in Shared Address Space	The Shared Address Space, with IP range 100.64.0.0/10, was implemented to connect customer-premises equipment to the core routers of Internet service providers. To allocate IP addresses to the management interfaces and ILOMs of hardware components, two CIDR blocks are reserved for internal use: 100.96.0.0/23 and 100.96.2.0/23.

Reserved IP Addresses	Description
CIDR blocks in Class E address range	Under the classful network addressing architecture, Class E is the part of the 32-bit IPv4 address space ranging from 240.0.0.0 to 255.255.255.255. At the time, it was reserved for future use, so it can't be used on the public Internet. To accommodate the addressing requirements of all infrastructure networking over the physical 100Gbit connections, the entire 253.255.0.0/16 subnet is reserved. It is further subdivided into multiple CIDR blocks to group IP addresses by network function or type. The various CIDR blocks within the 253.255.0.0/16 range are used to allocate IP addresses for the Kubernetes containers running the microservices, the virtual switches, routers and gateways enabling the VCN data network, the hypervisors, the Private Cloud Appliance chassis components, and so on.
Link Local CIDR block	A link-local address belongs to the 169.254.0.0/16 IP range, and is valid only for connectivity within a host's network segment, because the address isn't guaranteed to be unique outside that network segment. Packets with link-local source or destination addresses aren't forwarded by routers. The link-local CIDR block 169.254.239.0/24, and the IP address 169.254.169.254, are reserved for functions such as DNS requests, compute instance metadata transfer, and cloud service endpoints.

All VCN traffic – from one VCN to another, and between a VCN and external resources – flows across the 100-Gbit connections and is carried by VLAN 3900. Traffic related to server management is carried by VLAN 3901. All VLANs with higher IDs are also reserved for internal use, and VLAN 1 is the default for untagged traffic. The remaining VLAN range of 2 - 3899 is available for customer use.

Integrating External Systems with Flex Networks

Optionally, Private Cloud Appliance can be integrated with external systems in the on-premises network using direct cable connections to reserved ports on the spine switches.

Private Cloud Appliance can be integrated with external systems, such as Oracle Exadata, Oracle ZFS Storage Appliance, Oracle Database Appliance, and so on. In this configuration, external components are directly connected to reserved ports on the spine switches of the Private Cloud Appliance. Four 100Gbit ports per spine switch are reserved and split into 4x25Gbit breakout ports, providing a maximum of 32 total cable connections. Each externally connected component is cabled directly to both spine switches, meaning up to 16 external connections of this type can be made. **This feature was formerly called Exadata Networks**.

> **Note:**
>
> The 25 Gbps breakout configuration applies to connectivity with Oracle Exadata nodes or Oracle Database Appliance nodes and does not indicate that Flex Networks are limited to 25 Gbps.

When the cable connections are in place, the appliance administrator configures a *Flex network*, which enables traffic between the connected system(s) and a set of compute instances. These prerequisites apply:

- The Flex network must not overlap with any subnets in the on-premises network.
- The VCNs containing compute instances that connect to the external nodes, must have a dynamic routing gateway (DRG) configured.
- The relevant subnet route tables must contain rules to allow traffic to and from the Flex network.

The Flex network configuration determines which external components are exposed and which subnets have access to those components. Access can be enabled or disabled per connected system and per compute subnet. In addition, the Flex network can be exposed through the appliance's external network, allowing other resources within the on-

premises network to connect to the external system through the spine switches of the appliance. The Flex network configuration is created and managed through the Service Enclave.

Platform and Service Limits for Private Cloud Appliance

The Private Cloud Appliance limits presented here have been tested and are fully supported by Oracle.

The minimum appliance configuration contains three compute nodes and one high-capacity disk shelf with 100TB of usable disk space. Both compute and storage capacity can be expanded by adding compute nodes and disk shelves.

Review the following sections:

- Capacity Limits Based On Compute Node Physical Resources on page 48
- Tenancy Resource Configuration Limits on page 48
- System Load and Concurrency Limits on page 51

Capacity Limits Based On Compute Node Physical Resources

In Private Cloud Appliance, a part of each compute node's CPU and RAM capacity is reserved for system use. The table shows available physical resources by compute node model.

Compute Node Model	Total Physical Resources	Available Compute Capacity
Oracle Server X9-2	CPU: 64 cores (2x32) RAM: 1024 GB (16x64)	CPU: 60 cores RAM: 960 GB
Oracle Server X10	CPU: 192 cores (2x96) RAM: 2304 GB (24x96)	CPU: 184 cores RAM: 2224 GB
Oracle Server X11	CPU: 192 cores (2x96) RAM: 2304 GB (24x96)	CPU: 184 cores RAM: 2224 GB

Tenancy Resource Configuration Limits

In Private Cloud Appliance, resource limits are dependent on the appliance architecture. The numbers provided here are default limits per tenancy.

Private Cloud Appliance supports up to 8 tenancies. The default resource configuration limits apply per tenancy, unless indicated otherwise. For better allocation of system capacity, certain limits apply across all tenancies combined, as specified in the table.

Some of these limits can be customized. See Viewing and Setting Resource Limits on page 195.

Service	Resource Type	Limit
IAM Service	Users	100
IAM Service	Groups	100
IAM Service	Users per group	100
IAM Service	Groups per user	100
IAM Service	Compartments	50
IAM Service	API keys	3 per user

Service	Resource Type	Limit
IAM Service	Policies	100
IAM Service	Policy statements	50 per policy
IAM Service	Identity providers	3
IAM Service	Group mappings	100 per identity provider
IAM Service	Tag namespaces	100
IAM Service	Tags	100 per tag namespace
IAM Service	Tag defaults	5
Networking Service	VCNs	80 total across all tenancies with up to 16 SR-IOV VCNs
Networking Service	Subnets	40 per VCN 320 across all tenancies
Networking Service	Dynamic routing gateways (DRG)	32 total across all tenancies with up to 16 SR-IOV DRGs
Networking Service	DRG attachments	10 per DRG 80 across all tenancies
Networking Service	Internet gateways	1 per VCN
Networking Service	Local peering gateways	5 per VCN 150 across all tenancies
Networking Service	NAT gateways	1 per VCN
Networking Service	Service gateways	1 per VCN
Networking Service	Storage gateways	2 per VCN 80 across all tenancies, standard and high-performance combined
Networking Service	Ephemeral public IPs	2 per compute instance
Networking Service	DHCP options	30 per VCN 500 across all tenancies
Networking Service	Route tables	20 per VCN 500 across all tenancies

Service	Resource Type	Limit
Networking Service	Route rules	50 per route table 10000 across all tenancies
Networking Service	Network security groups	100 per VCN 5 per VNIC 600 across all tenancies
Networking Service	VNICs in network security group	As many VNICs as are in the VCN. A VNIC can belong to max. 5 network security groups
Networking Service	Security rules	50 per network security group 12000 across all tenancies
Networking Service	Security lists	20 per VCN 5 per subnet 600 across all tenancies
Networking Service	Ingress rules	30 per security list 12000 across all tenancies
Networking Service	Egress rules	30 per security list 12000 across all tenancies
Networking Service	DNS zones	1000 across all tenancies (in addition to any internal zones)
Networking Service	DNS records	25000 per zone 8000000 across all tenancies
Networking Service	Flex networks	128 flex networks 32 DRGs 128 logical ports 32 networks per physical port
Compute Service	Custom images	100
Block Storage Service	Aggregated size of block volumes	100TB (with default storage capacity)
Block Storage Service	Block volume backups	100000 across all tenancies
File Storage Service	File systems	100

Service	Resource Type	Limit
File Storage Service	Mount targets	PCA_POOL 80 total across all tenancies PCA_POOL_HIGH 80 total across all tenancies
File Storage Service	File system size	3.3PB
Object Storage Service	Buckets	10000
(Network) Load Balancing Service	Load balancers (Network LB and LBaaS combined)	Version 3.0.2-b1185392 and earlier: 32 total across all tenancies, 20 in a single VCN Version 3.0.2-b1261765 and newer: 36-144 total, depending on rack capacity and custom configuration
(Network) Load Balancing Service	IP address	1 per load balancer
(Network) Load Balancing Service	Host name	16 per load balancer
(Network) Load Balancing Service	Network security groups	5 per load balancer
(Network) Load Balancing Service	Listeners	16 per load balancer
(Network) Load Balancing Service	Backend sets	16 per load balancer
(Network) Load Balancing Service	Backend servers	512 per load balancer and per backend set
Load Balancing Service	Request routing rules	20 per rule set 50 per load balancer
Kubernetes Engine (OKE)	Clusters	10 per tenancy
Kubernetes Engine (OKE)	Worker nodes	128 per cluster (across all pools)
Kubernetes Engine (OKE)	Pods	110 per node (Kubernetes default)

System Load and Concurrency Limits

This topic shows how many concurrent operations of a given type Private Cloud Appliance can manage at any given time.

The limits presented in the table apply across the entire system and all tenancies. For each of these limits it is assumed that no other operations of any kind are running at the same time. When a limit is exceeded, an error with code 409 or 429 is displayed.

Resource Type	Operation	Concurrency Limit
compute instance	back up or restore an instance	10
compute instance	launch/terminate instance	15

Resource Type	Operation	Concurrency Limit
compute instance	reset/stop/start instance	15
compute instance	update fault domain (live migration)	10
compute image	create image from instance	10
compute image	import image	10
block volume	create/delete volume	10
block volume	attach/detach boot volume	15
block volume	attach/detach data volume	15
block volume	resize volume	15
file system	create/delete file system	10
mount target	create/delete mount target	10
VCN	create/delete VCN	10
VCN gateway	create/delete gateway (all types)	10
subnet	create/delete subnet	10
route table	create/delete route table	10
security list	create/delete security list	10
network security group	create/delete network security group	10
VNIC	attach/detach VNIC	15
public IP	create/delete public IP	10
private IP	create/delete private IP	10
all networking resources	update network resource	10
Kubernetes cluster	create/update/delete cluster	10
Kubernetes node pool	create/update/delete node pool	5
Kubernetes node	create/update/delete node	15

Note:

In addition, there is a system limit on the number of concurrent user sessions:

- Compute Web UI: 10 tenancy users
- Service Web UI: 6 administrators

An authentication error is displayed when the limit is reached. An inactive user session times out after 1 hour.

Guest Operating System Matrix

This topic provides an overview of guest operating systems that can be deployed in Private Cloud Appliance compute instances. It also indicates the OS delivery type and testing status.

Oracle Cloud Infrastructure compute images of Oracle Linux and Oracle Solaris are provided as part of the appliance software, and new image versions are added through upgrades and patches.

Private Cloud Appliance supports other guest operating systems, which you can add to your appliance environment as custom images. Several guest operating systems are part of Oracle testing and are known to work in Private Cloud Appliance compute instances. The table below provides an overview.

Guest Operating System	Oracle-Provided Image	Custom Image	Oracle-Tested
Oracle Linux 9.x	Y	Y	Y
Oracle Linux 8.x	Y	Y	Y
Oracle Linux 7.x	Y	Y	Y
Oracle Solaris 11.x	Y	Y	Y
Red Hat Enterprise Linux 9.x		Y	Y
Red Hat Enterprise Linux 8.x		Y	Y
Red Hat Enterprise Linux 7.x		Y	Y
CentOS Linux 8.x		Y	
CentOS Linux 7.x		Y	
SUSE Linux Enterprise Server 15 (latest)		Y	
SUSE Linux Enterprise Server 12 SP4		Y	
Ubuntu 20.04 and later		Y	
Ubuntu 18.04 and later		Y	
AlmaLinux OS 9.2		Y	Y
Kali Linux		Y	Y
Microsoft Windows Server 2025		Y	Y
Microsoft Windows Server 2022		Y	Y
Microsoft Windows Server 2019		Y	Y
Microsoft Windows Server 2016		Y	Y

Viewing a PDF of this Documentation

You can view and download a copy of the Private Cloud Appliance documentation set provided here.

Use the following link to display the Private Cloud Appliance in PDF format. You can save it to your local computer for convenient offline use.

Private Cloud Appliance Documentation

> **Note:**
>
> You can view the PDF file offline. However, it contains links to other Oracle Cloud Infrastructure documentation, which require an internet connection to view.

Installing and Configuring Private Cloud Appliance

Before you can use Private Cloud Appliance, you must prepare your site, install the appliance rack, and complete the initial configuration.

The following table describes the steps you perform to install Private Cloud Appliance at your site.

Step	Task	References
1	Review the Release Notes for any late-breaking information.	Oracle Private Cloud Appliance Release Notes
2	Before performing the installation, review important safety information.	• Oracle Private Cloud Appliance Safety and Compliance Guide • Important Safety Information for Oracle's Hardware Systems
3	Review information about Private Cloud Appliance features and hardware components.	• Architecture and Design of Private Cloud Appliance on page 6 • Overview of Private Cloud Appliance Hardware Components on page 14
4	Complete the site preparation procedures.	Preparing the Installation Site on page 55
5	Complete the site network preparation procedures.	Preparing the Data Center Network Environment on page 62
6	Complete the installation checklist.	Initial System Installation Checklist on page 74
7	Install the Private Cloud Appliance rack at your site.	Installing the Private Cloud Appliance Rack in the Data Center on page 82
8	Power on Private Cloud Appliance for the first time.	Powering On Private Cloud Appliance for the First Time on page 90

Step	Task	References
9	Perform the initial configuration of the system.	Configuring the Private Cloud Appliance System on page 95

Preparing the Installation Site

Before the arrival of your Private Cloud Appliance, prepare the installation site and ensure it meets the requirements outlined in this section.

To track progress as you prepare your site, use the Site Checklists for Private Cloud Appliance on page 68.

Follow these steps prior to unpacking and situating your Private Cloud Appliance:

1. Thoroughly clean and vacuum the area in preparation for the installation.
2. Note problems or peculiarities at the site that require special equipment.
3. Verify that the installation site flooring has a strength rating to withstand the combined weight of the Private Cloud Appliance and any other installed equipment.

   For more information, see Flooring Requirements on page 58.
4. Install all necessary electrical equipment and ensure that sufficient power is provided.

   See the Oracle Rack Cabinet 1242 Power Distribution Units User's Guide for the system Power Distribution Unit (PDU) power requirements.
5. Ensure that the installation site provides adequate air conditioning.

   For details, see Environmental Requirements on page 61 (Ventilation and Cooling).
6. Operate the air conditioning system for 48 hours to bring the room temperature to the appropriate level.

Review Safety Guidelines

Review the following safety precautions to ensure that the site is safe and ready for delivery. Failure to observe these precautions can result in personal injury, equipment damage, or malfunction.

- Do not block ventilation openings.
- Do not install the system in a location that is exposed to direct sunlight or near a device that may become hot.
- Do not install the system in a location that is exposed to excessive dust, corrosive gases, or air with high salt concentrations.
- Do not install the system in a location that is exposed to frequent vibrations. Install the system on a flat, level surface.
- Use a power outlet that uses proper grounding. When using shared grounding, the grounding resistance must not be greater than 10 ohms. Ensure that your facility administrator or a qualified electrical engineer verifies the grounding method for the building, and performs the grounding work.
- Ensure that each grounding wire is used exclusively for the Private Cloud Appliance system. Also observe the precautions, warnings, and notes about handling that appear on labels on the equipment.
- Do not place cables under the equipment or stretch the cables too tightly.
- Do not disconnect power cords from the equipment while its power is on.
- Do not place anything on top of the system or perform any work directly above it.
- If you cannot reach the connector lock when disconnecting LAN cables, then press the connector lock with a flathead screwdriver to disconnect the cable. You could damage the system board if you force your fingers into the gap rather than using a flathead screwdriver.
- Do not let the room temperature rise sharply, especially in winter. Sudden temperature changes can cause condensation to form inside the system. Allow for a sufficient warm-up period prior to server operation.
- Do not install the system near a photocopy machine, air conditioner, welding machine, or any other equipment that generates loud, electronic noises.
- Avoid static electricity at the installation location. Static electricity transferred to the system can cause malfunctions. Static electricity is often generated on carpets.

- Confirm that the supply voltage and frequency match the electrical ratings indicated on your Private Cloud Appliance.
- Do not insert anything into any Private Cloud Appliance opening, unless doing so is part of a documented procedure. The system contains high-voltage parts. If a metal object or other electrically conductive object enters an opening in the system, then it could cause a short circuit. This could result in personal injury, fire, electric shock, and equipment damage.

See also:

- “Important Safety Information for Sun Hardware Systems” (816-7190) included with the rack
- Oracle Rack Cabinet 1242 Power Distribution Units User's Guide
- Oracle Engineered System Safety and Compliance Guide
- Oracle Engineered System Safety and Compliance Guide (Nordic)

Site Requirements

Ensure the appliance installation site meets the requirements described in the following sections:

Space Requirements on page 56

- Product dimensions and weight
- Area surrounding the rack
- Receiving area and access route
- Maintenance access

Flooring Requirements on page 58

Electrical Power Requirements on page 58

- Power distribution units
- Facility power
- Circuit breakers and grounding

Environmental Requirements on page 61

- Temperature and humidity
- Ventilation and cooling

Space Requirements

Ensure that your site meets the space requirements for each Oracle Private Cloud Appliance rack.

Use the information in this section along with the Logistics Checklist on page 72, Data Center Room Checklist on page 69, and Access Route Checklist on page 70 to prepare your site.

Dimensions and Weight Specifications

Specification	Metric	Standard
Height	2000 mm	78.74 "
Width	600 mm	23.62 "
Depth (front door handle to rear door handle)	1197 mm	47.12 "
Depth (doors removed)	1112 mm	43.78 "

Specification	Metric	Standard
Weight (base rack, fully populated)	1000 kg	2204 lbs

Space Requirements

Minimum Ceiling Height

The minimum ceiling height for the cabinet is 2914 mm (114.72 inches), measured from the true floor or raised floor, whichever is higher. This includes an additional 914 mm (36 inches) of space required above the rack height for maintenance access.

Surrounding Space

The space surrounding the cabinet must not restrict the movement of cool air between the air conditioner and the front of the systems within the cabinet, or the movement of hot air coming out of the rear of the cabinet.

Receiving Specifications

Before the Private Cloud Appliance rack arrives, ensure that the receiving area is large enough for the package. The following are the package dimensions and weights for a rack:

- Shipping height: 2159 mm (85 inches)
- Shipping width: 1118 mm (44 inches)
- Shipping depth: 1543 mm (60.75 inches)
- Shipping weight (base rack, fully populated): 1118 kg (2465 lbs)

If your loading dock meets the height and ramp requirements for a standard freight carrier truck, then you can use a pallet jack to unload the rack. If the loading dock does not meet the requirements, then you must provide a standard forklift or other means to unload the rack. You can also request that the rack be shipped in a truck with a lift gate.

When the rack arrives, leave the rack in its shipping packaging until it arrives at its installation site. Use a conditioned space to remove the packaging material to reduce particles before entering the data center. The entire access route to the installation site should be free of raised-pattern flooring that can cause vibration.

Allow enough space for unpacking the system from its shipping cartons. Ensure that there is enough clearance and clear pathways for moving the rack from the unpacking location to the installation location. The following table lists the access route requirements for the rack.

Access Route Item	With Shipping Pallet	Without Shipping Pallet
Minimum door height	2159 mm (85 inches)	2000 mm (78.74 inches)
Minimum door width	1118 mm (44 inches)	600 mm (23.62 inches)
Minimum elevator depth	1543 mm (60.75 inches)	1058.2 mm (41.66 inches)
Maximum incline	6 degrees	6 degrees
Maximum elevator, pallet jack, and floor loading capacity	1145 kg (2520 lbs)	1145 kg (2520 lbs)

Maintenance Access Requirements

The maintenance area must be large enough for the rack, and have the required access space. For example, the required space to remove the side panels is 675.64 mm (26.6 inches).

The following table lists the maintenance access requirements for the Private Cloud Appliance rack.

Location	Maintenance Access Requirement
Rear maintenance	914 mm (36 inches)
Front maintenance	1232 mm (48.5 inches)
Top maintenance	914 mm (36 inches)

Flooring Requirements

Oracle allows the installation of the Private Cloud Appliance rack on solid flooring and on raised flooring (recommended). Review important flooring requirements before the system arrives at your site.

Use the information in this section along with the Data Center Room Checklist on page 69 to prepare your site.

The site floor and the optional raised flooring must be able to support the total weight of the system as specified in Space Requirements on page 56. The following table lists the floor load requirements.

Description	Requirement
Maximum allowable weight of installed rack equipment	952.54 kg (2100 lbs)
Maximum allowable weight of installed power distribution units	52.16 kg (115 lbs)
Maximum dynamic load (maximum allowable weight of installed equipment including power distribution units)	1004.71 kg (2215 lbs)

Electrical Power Requirements

Follow these important electrical power guidelines for Private Cloud Appliance.

Use the information in this section along with the Facility Power Checklist on page 71 to prepare your site.

The Private Cloud Appliance can operate effectively over a wide range of voltages and frequencies. However, it must have a reliable power source. Damage might occur if the ranges are exceeded. Electrical disturbances such as the following might damage the system:

- Fluctuations caused by brownouts
- Wide and rapid variations in input voltage levels or in input power frequency
- Electrical storms
- Faults in the distribution system, such as defective wiring

To protect your system from such disturbances, you should have a dedicated power distribution system, power-conditioning equipment, as well as lightning arresters or power cables to protect from electrical storms.

Power Distribution Units (PDUs)

Each rack has two preinstalled power distribution units. The PDUs accept different power sources. You must specify the type of PDU that is correct for your data center. The following tables list PDU requirements.

Global PDU Specifications

Item	15kVA - type 6440A	15kVA - type 6441A	15kVA - type 6442A	22kVA - type 7100873	22kVA - type 7100874	24kVA - type 6444A	24kVA - type 6445A
Region	North and South America, Japan and Taiwan	EMEA and APAC (except Japan and Taiwan)	North and South America, Japan and Taiwan	North and South America, Japan and Taiwan	EMEA and APAC (except Japan and Taiwan)	North and South America, Japan and Taiwan	EMEA and APAC (except Japan and Taiwan)
Number of inputs	1	1	3	3	3	2	2
Source voltage	200-220V	220/380 - 240/415VAC	200-240V	200-240V	200-240V	200-220V	220/380 - 240/415VAC
Frequency	50/60 Hz	50/60 Hz	50/60 Hz	50/60 Hz	50/60 Hz	50/60 Hz	50/60 Hz
Max. line current	40A	21A	24A	36.8A	32A	34.6A	16A
Phase current	23A	21A	N/A	N/A	32A	20A	16A
Recommended circuit breaker	50A	25A	30A	50A	32A	50A	25A
Outlets	42 x C13 6 x C19	42 x C13 6 x C19	42 x C13 6 x C19	42 x C13 6 x C19	42 x C13 6 x C19	42 x C13 6 x C19	42 x C13 6 x C19
Outlet groups	6	6	6	6	6	6	6
Max. group current	20A	20A	20A	20A	20A	20A	18A
Output voltage	200-240V	220-240V	200-240V	200-240V	200-240V	220-240V	200-240V
Data center receptacle	3ph 3W + ground 60A IEC60309	3ph 4W + ground 32A IEC60309 5P 6h 400V	1ph 2W + ground 30A NEMA L6-30	1ph 2W + ground 50A Hubbell CS8265C	1ph 2W + ground 32A IEC60309 3P 6H 230V	3ph 3W + ground 60A IEC60309	3ph 4W + ground 32A IEC60309 5P 6h 400V

PDU Specifications for Korea

Item	15 kVA - type 7600785	22KVA - type 7600788	24KVA - type 7600787
Region	South Korea only	South Korea only	South Korea only
Nominal Power Rating	15KVA	22KVA	24KVA

Item	15 kVA - type 7600785	22KVA - type 7600788	24KVA - type 7600787
Number of inputs	1	3	2
Source voltage	220/380 - 240/415VAC 3Ø	220-240V	220/380 - 240/415VAC 3Ø
Frequency	50/60 Hz	50/60 Hz	50/60 Hz
Max. line current	21A	32A	18A
Phase current	21A	32A	18A
Recommended circuit breaker	25A	32A	25A
Outlets	42 x C13 6 x C19	42 x C13 6 x C19	42 x C13 6 x C19
Outlet groups	6	6	6
Max. group current	20A	16A	20A
Output voltage	220-240V	220-240V	220-240V
Data center receptacle	3ph 4W + ground 32A IEC60309 5P 6h 400V	1ph 2W + ground 32A IEC60309 3P 6H 230V	3ph 4W + ground 32A IEC60309 5P 6h 400V

Facility Power Requirements

Electrical work and installations must comply with applicable local, state, or national electrical codes. Contact your facilities manager or qualified electrician to determine what type of power is supplied to the building.

To prevent catastrophic failures, design the input power sources to ensure adequate power is provided to the power distribution units (PDUs). When planning for power distribution requirements, balance the power load between available AC supply branch circuits. In the United States and Canada, ensure that the overall system AC input current load doesn't exceed 80 percent of the branch circuit AC current rating.

Provide a stable power source, such as an uninterruptable power supply (UPS) to reduce the possibility of component failures. If computer equipment is subjected to repeated power interruptions and fluctuations, then it's susceptible to a higher rate of component failure.

PDU power cords are 4 meters (13.12 feet) long, and 1 to 1.5 meters (3.3 to 4.9 feet) of the cord will be routed within the rack cabinet. The installation site AC power receptacle must be within 2 meters (6.6 feet) of the rack.

Circuit Breaker Requirements

Use dedicated AC breaker panels for all power circuits that supply power to the rack PDUs. Electrical work and installations must comply with applicable local, state, or national electrical codes. Rack PDUs require electrical circuits to be grounded to the Earth.

> **Note:**
>
> Circuit breakers are supplied by the customer. One circuit breaker is required for each power cord.

Grounding Guidelines

Private Cloud Appliance racks are shipped with grounding-type power cords (three-wire). Always connect the cords to grounded power outlets. Because different grounding methods are used, depending on location, check the

grounding type, and refer to documentation, such as IEC documents, for the correct grounding method. Ensure that the facility administrator or qualified electrical engineer verifies the grounding method for the building, and performs the grounding work.

Environmental Requirements

Ensure that the Private Cloud Appliance installation site meets these requirements for best environmental operating conditions.

Use the information in this section along with the Data Center Environment Checklist on page 69 to prepare your site.

Temperature and Humidity Requirements

Airflow through the Private Cloud Appliance rack is from front to back. For information, see Ventilation and Cooling Requirements on page 62.

Studies have shown that temperature increases of 10 degrees Celsius (18 degrees Fahrenheit) above 20 degrees Celsius (68 degrees Fahrenheit) reduce long-term electronics reliability by 50 percent. Excessive internal temperatures might result in full or partial shutdown of the rack.

Condition	Operating Requirement	Non-operating Requirement	Optimum
Temperature	5 ° to 32 ° Celsius (41 ° to 89.6 ° Fahrenheit)	-40 ° to 68 ° Celsius (-40 ° to 154 ° Fahrenheit)	For optimal rack cooling, data center temperatures from 21 ° to 23 ° Celsius (69.8 ° to 73.4 ° Fahrenheit)
Relative humidity	10 to 90 percent relative humidity, non-condensing	Up to 93 percent relative humidity	For optimal data center rack cooling, 45 to 50 percent non-condensing
Altitude	3,000 meters (9,840 feet) maximum	12,000 meters (39,370 feet)	Ambient temperature is reduced by 1 ° Celsius per 300 meters above 900 meters altitude above sea level

Set conditions to the optimal temperature and humidity ranges to minimize the chance of downtime because of component failure. Operating a rack for extended periods at or near the operating range limits, or installing it in an environment when it remains at or near non-operating range limits could significantly increase hardware component failure.

The ambient temperature range of 21 ° to 23 ° Celsius (69.8 ° to 73.4 ° Fahrenheit) is optimal for server reliability and operator comfort. Most computer equipment can operate in a wide temperature range, but near 22 ° Celsius (71.6 ° Fahrenheit) is preferable because it is easier to maintain safe humidity levels. Operating in this temperature range provides a safety buffer in the event that the air conditioning system goes down for a period of time.

The ambient relative humidity range of 45 to 50 percent is suitable for safe data processing operations. Most computer equipment can operate in a wide range (20 to 80 percent), but the range of 45 to 50 percent is recommended for the following reasons:

- Optimal range helps protect computer systems from corrosion problems associated with high humidity levels.
- Optimal range provides the greatest operating time buffer in the event of air conditioner control failure.
- This range helps to avoid failures or temporary malfunctions caused by intermittent interference from static discharges that may occur when relative humidity is too low.

> **Note:**
>
> Electrostatic discharge (ESD) is easily generated, and hard to dissipate in areas of low relative humidity, such as below 35 percent. ESD becomes critical when humidity drops below 30 percent. It is not difficult to maintain humidity in a data center because of the high-efficiency vapor barrier and low rate of air changes normally present.

Ventilation and Cooling Requirements

Always provide adequate space in front of and behind the rack to allow for proper ventilation. Do not obstruct the front or rear of the rack with equipment or objects that might prevent air from flowing through the rack. Rack-mountable servers and equipment typically draw cool air in through the front of the rack and let warm air out the rear of the rack. There is no airflow requirement for the left and right sides because of front-to-back cooling.

If the rack is not completely filled with components, then cover the empty sections will filler panels. Gaps between components can adversely affect airflow and cooling within the rack.

Relative humidity is the percentage of the total water vapor that can exist in the air without condensing, and is inversely proportional to air temperature. Humidity goes down when the temperature rises, and goes up when the temperature drops. For example, air with a relative humidity of 45 percent at a temperature of 24 ° Celsius (75.2 ° Fahrenheit) has a relative humidity of 65 percent at a temperature of 18 ° Celsius (64.4 ° Fahrenheit). As the temperature drops, the relative humidity rises to more than 65 percent, and water droplets are formed.

Air conditioning facilities usually don't precisely monitor or control temperature and humidity throughout an entire computer room. Generally, monitoring is done at individual points corresponding to multiple exhaust vents in the main unit, and other units in the room. Special consideration should be paid to humidity when using underfloor ventilation. When underfloor ventilation is used, monitoring is done at each point close to an exhaust vent. Distribution of the temperature and humidity across the entire room is uneven.

The rack has been designed to function while installed in a natural convection air flow. The following requirements must be followed to meet the environmental specification:

- Ensure that there is adequate airflow through the system.
- Ensure that the system has front-to-back cooling. The air intake is at the front of the system, and the air outlet is at the rear of the system.
- Allow a minimum clearance of 1219.2 mm (48 inches) at the front of the system, and 914 mm (36 inches) at the rear of the system for ventilation.

Preparing the Data Center Network Environment

The network architecture of Oracle Private Cloud Appliance relies on physical high speed Ethernet connectivity. Prepare the data center network configuration so it meets the requirements for incorporating the appliance.

The networking infrastructure in Private Cloud Appliance is integral to the system and must not be altered. The networking does not integrate into any data center management or provisioning frameworks such as Cisco ACI, Network Director, or the like.

However, Private Cloud Appliance can communicate with the Cisco ACI fabric in your data center using the L3Out functionality (static routes or eBGP) provided by Cisco ACI. For more information about this Cisco feature, see the Cisco ACI Fabric L3Out Guide.

> **Caution:**
>
> No changes to the networking switches in Private Cloud Appliance are supported, unless when instructed by Oracle Support or through a knowledge article on the My Oracle Support website.

For important conceptual information about appliance networking, see Private Cloud Appliance Network Infrastructure on page 28. It describes the different networks and their roles, the uplinks that connect to the data center network, and the network resources reserved for appliance operation.

Appliance Uplink Requirements

When preparing for the installation of Private Cloud Appliance, select an uplink configuration and set up the data center network to accept this configuration.

On each spine switch, ports 1 to 4 can be used for uplinks to the data center network. For speeds of 10Gbps or 25Gbps, the spine switch port must be split using a 4-way splitter or breakout cable. For higher speeds of 40Gbps or 100Gbps each switch port uses a single direct cable connection.

The uplinks are configured during system initialization, based on information you provide as part of the Initial System Installation Checklist on page 74. Unused spine switch uplink ports, including unused breakout ports, are disabled for security reasons.

It is critical that **both** spine switches have the same connections to the next-level data center switches. This configuration provides redundancy and load distribution at the level of the spine switches, the ports, and the data center switches. This outbound cabling depends on the network topology you deploy. The cabling pattern plays a key role in the continuation of service during failover scenarios.

For detailed information about choosing the appropriate configuration, and about the available topologies (Mesh, Square, Triangle) see Uplinks on page 29. Configuration guidance and reference examples are also provided.

- Before installation, you must run network cables from your existing network infrastructure to the Private Cloud Appliance installation site. For instructions see Connecting Private Cloud Appliance to the Data Center Network on page 89.
- Plan to connect at least 1 high-speed Ethernet port on each *spine* switch to your data center public Ethernet network.
- Configuring the optional Administration network requires 2 additional cable connections (one each from port 5 on the two spine switches) to a pair of next-level data center switches.
- Uplink connectivity is based on layer 3 of the OSI model.
- When upgrading from appliance software version 3.0.2-b892153 or earlier, and running vPC/HSRP, if you want to change the network configuration to support new features, contact Oracle for assistance.

Data Center DNS Configuration for Private Cloud Appliance

To integrate the data of the dedicated Private Cloud Appliance DNS zone into the data center DNS configuration, two options are supported: zone delegation or manual configuration.

The preferred approach is to configure zone delegation. However, if you select manual configuration, it is good practice to register network host names and IP addresses for the management network, client network, and additional public networks in the data center Domain Name System (DNS) prior to initial configuration. In particular, all public addresses, VIP addresses and infrastructure services endpoints should be registered in DNS prior to installation.

All addresses registered in DNS must be configured for forward resolution; reverse resolution is not supported in the Private Cloud Appliance services zone.

Zone Delegation (preferred)

For zone delegation to work, it is required that the data center's recursive caches are able to reach TCP/UDP port 53 on the virtual IP address shared by the appliance management nodes. It may be necessary to change your firewall configuration.

Configure the data center DNS server so that it operates as the parent zone of the appliance DNS zone. Thus, all DNS requests for the child zone are delegated to the appliance internal DNS server. In the data center DNS configuration, add a name server record for the child zone and an address record for the authoritative server of that zone.

In the example it is assumed that the data center DNS domain is `example.com`, that the appliance is named `mypca`, and that the management node cluster virtual IP address is 192.0.2.102. The appliance internal DNS server host name is `ns1`.

```
$ORIGIN example.com.
[...]
mypca       IN    NS    ns1.mypca.example.com.
```

```
ns1.mypca   IN    A     192.0.2.102
```

> **Caution:**
>
> DNS lookup for services endpoints has changed in controller software version 3.0.2-b1483396. Individual address records per service have been consolidated into CNAME records referencing a common *services* or *adminservices* record. With zone delegation of the appliance subdomain, lookups for a defined RTYPE return the CNAME record and the RTYPE. Lookups for undefined RTYPEs that previously returned no answer, now return a CNAME record only.

Manual Configuration

Manually add DNS records for all labels or host names required by the appliance.

In the examples it is assumed that the data center DNS domain is `example.com`, that the appliance is named `mypca`, and that the management node cluster virtual IP address is 192.0.2.102 in the data network and 203.0.113.12 in the (optional) administration network.

Note:

For object storage you must point the DNS label to the Object Storage Public IP. This is the public IP address you assign specifically for this purpose when setting up the data center public IP ranges during Initial Setup.

Appliance Infrastructure Service and DNS Label	Data Center DNS Record	Data Center DNS Record with Admin Network Enabled
Admin service ***admin***.mypca.example.com	admin A 192.0.2.102	admin A 203.0.113.12
Networking, Compute, Block Storage, Work Requests services ***iaas***.mypca.example.com	iaas A 192.0.2.102	iaas A 192.0.2.102
Identity and Access Management service ***identity***.mypca.example.com	identity A 192.0.2.102	identity A 192.0.2.102
DNS service ***dns***.mypca.example.com	dns A 192.0.2.102	dns A 192.0.2.102
Object storage ***objectstorage***.mypca.example.com Use the Object Storage Public IP from the Appliance Initial Setup.	objectstorage A 198.51.100.33	objectstorage A 198.51.100.33
File storage ***filestorage***.mypca.example.com	filestorage A 192.0.2.102	filestorage A 192.0.2.102
Alert manager ***alertmanager***.mypca.example.com	alertmanager A 192.0.2.102	alertmanager A 203.0.113.12
API ***api***.mypca.example.com	api A 192.0.2.102	api A 203.0.113.12
OKE service ***containerengine***.mypca.example.com	containerengine A 192.0.2.102	containerengine A 192.0.2.102
Resource principal service ***rps***.mypca.example.com	rps A 192.0.2.102	rps A 203.0.113.12
Grafana ***grafana***.mypca.example.com	grafana A 192.0.2.102	grafana A 203.0.113.12
Prometheus ***prometheus***.mypca.example.com	prometheus A 192.0.2.102	prometheus A

Data Center Network Configuration Guidelines

Follow these important guidelines for a smooth integration of Private Cloud Appliance into the data center network.

Data Center Switch Notes

- All uplinks, default and customer, are configured to use link aggregation (LACP). All switch ports included in an uplink configuration must belong to the same link aggregation group (LAG). The switch ports on the data center side of the uplinks must be configured accordingly.
- The spine switches operate with the Virtual Port Channel (vPC) feature enabled in static routing configurations.
- Private Cloud Appliance supports layer 3 based uplink connectivity to the customer data center. Static routing and BGP4-based dynamic routing are supported in layer 3.
- Autonegotiation isn't available for uplink ports. Transfer speed must be specified on the customer switches' end.

For more information, see Uplinks on page 29 and Uplink Protocols on page 33.

Administration Network Guidelines

If you choose to segregate administrative appliance access from the data traffic, ensure that the data center network is configured accordingly, so that all traffic can be routed to the appropriate destinations in both the administration and the data network.

Access to Service Endpoints

When the administration network is enabled, some appliance infrastructure services are accessed through the Admin Management VIP, instead of the regular Management Node VIP. These service endpoints are:

- 'admin'
- 'adminconsole'
- 'prometheus-gw'
- 'prometheus'
- 'grafana'
- 'api'
- 'alertmanager'
- 'rps'

The following service endpoints are always accessed through the Management Node VIP in the data network:

- 'console'
- 'iaas'
- 'identity'
- 'filestorage'
- 'objectstorage'
- 'dns'
- 'containerengine'

Ensure that the data center firewall is configured to allow this traffic. If you manage the DNS records required by the appliance in the data center DNS configuration, ensure that they point to the correct network and address, as shown in Data Center DNS Configuration for Private Cloud Appliance on page 63 (Manual Configuration).

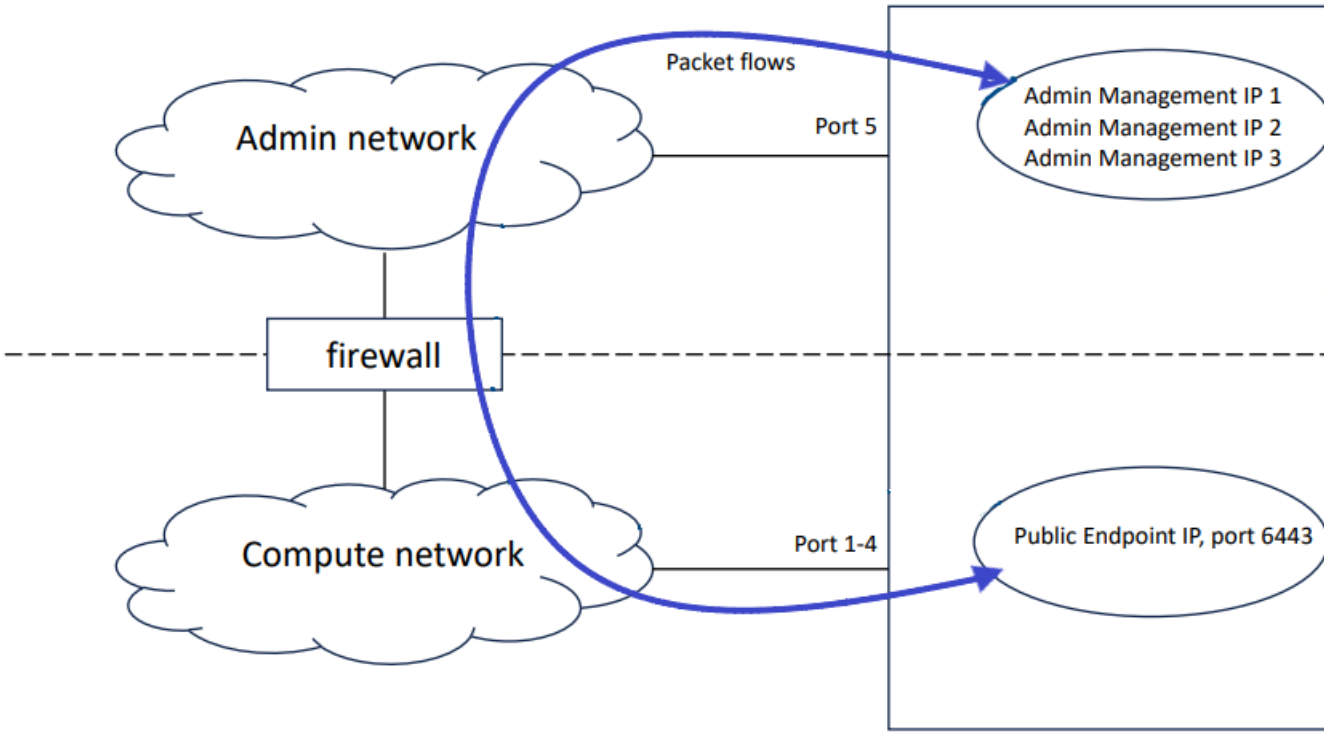
OKE Cluster Management

When Kubernetes Engine is used on a system configured with a separate administration network, the data center firewall must be configured to allow traffic between the OKE control plane and the OKE clusters deployed by Compute Enclave users.

The OKE control plane runs on the management nodes in the administration network, while the OKE clusters are deployed in the data network. The management interface of an OKE cluster is port 6443 on its

load balancer public IP address. This address is assigned from the data center IP range you reserved and configured as public IPs during initial appliance setup.

Because of the network segregation, traffic from the OKE control plane must exit the appliance through the administration network, and reenter through the data network to reach the OKE cluster. The data center network infrastructure must allow traffic in both directions. Without the necessary firewall and routing rules, users cannot deploy OKE clusters.



Default System IP Addresses

The management IP address represents a component's connection to the internal administration network.

> **Caution:**
>
> For hardware management, Private Cloud Appliance uses a network internal to the system. It is not recommended to connect the management ports or the internal administration network switches to the data center network infrastructure.

The table in this section lists the default management IP addresses assigned to servers and other hardware components in a Private Cloud Appliance base configuration.

Rack Unit	Rack Component	Management IP Address Assigned During Manufacturing
32	Spine Switch	`100.96.2.21`
31	Spine Switch	`100.96.2.20`

Rack Unit	Rack Component	Management IP Address Assigned During Manufacturing
26	Management Switch	`100.96.2.1` `100.96.0.1`
25	Leaf/Data Switch	`100.96.2.23`
24	Leaf/Data Switch	`100.96.2.22`
	Management Node VIP	`100.96.2.32` ILOM: `100.96.0.32`
7	Management Node	`100.96.2.35` ILOM: `100.96.0.35`
6	Management Node	`100.96.2.34` ILOM: `100.96.0.34`
5	Management Node	`100.96.2.33` ILOM: `100.96.0.33`
	Storage VIPs	Performance pool `100.96.2.5` Capacity pool `100.96.2.4`
3-4	Oracle ZFS Storage Appliance Controller Server (2 rack units)	`100.96.2.3` ILOM: `100.96.0.3`
1-2	Oracle ZFS Storage Appliance Controller Server (2 rack units)	`100.96.2.2` ILOM: `100.96.0.2`

Compute nodes are assigned an IP address in the internal administration network during the provisioning process. The system IP address is DHCP-based; the ILOM is assigned the system IP, where the third octet is changed from 2 to 0. For example: if a compute node receives IP `100.96.2.64` , then its ILOM has IP `100.96.0.64` . When assigned to a host, these IP addresses are stored and persisted in the DHCP database.

Site Checklists for Private Cloud Appliance

To determine your readiness for the installation of the Private Cloud Appliance rack in your data center, work with your Oracle representative to review and complete the checklists before the rack arrives.

Preparation is crucial to ensure a fluent installation process. These site checklists are designed to guide you through each phase and remind you of all aspects you need to consider.

System Components Checklist

For specifications, see Space Requirements on page 56.

Item	Comment
How many racks do you plan to install?	

Item	Comment
Is the rack installation a new system or an addition to an existing system?	
Are all machine racks adjacent to each other?	
If the connecting racks are not within the specified proximity, have you Oracle Support Services to provide and schedule the custom multirack cabling service with the installation?	

Data Center Room Checklist

For specifications, see Space Requirements on page 56 and Flooring Requirements on page 58.

Item	Comment
Has a vacant rack location been allocated for each rack?	
Does the floor layout meet the equipment maintenance access requirements?	
Will the rack be positioned so that the exhaust air of one rack doesn't enter the air inlet of another rack?	
Have cabinet stabilization measures been considered?	
If the data center has a raised floor: • Does the raised floor satisfy the weight requirements for the rack? • Is permission required to remove floor tiles for cabling and servicing below the floor? • Is the depth of the raised floor a minimum of 46 cm (18 inches)? • Are there cable routing channels or cutouts?	
Are you providing any additional hardware?	
Is the hardware you are providing fully compatible with the Private Cloud Appliance system?	
Will the new hardware location require any nonstandard cable lengths?	
Is the floor to ceiling height a minimum of 3214 mm or 3.2 m (10.5 feet)?	

Data Center Environment Checklist

For specifications, see Environmental Requirements on page 61.

Item	Comment
Does the computer room air conditioning meet temperature and humidity requirements?	

Item	Comment
Does the installation floor layout satisfy the ventilation requirements?	
Will the rack be positioned so that the exhaust air of one rack doesn't enter the air inlet of another rack?	
If the room cooling is from a raised floor: • Are the perforated floor tiles each rated at 400 CFM or greater? • Can additional perforated floor tiles be obtained if required for additional cooling?	
Does the data center air conditioning provide sufficient front-to-back airflow?	
Is airflow adequate to prevent hot spots?	
Can the data center continuously satisfy the environmental requirements?	

Access Route Checklist

For specifications, see Space Requirements on page 56 and Flooring Requirements on page 58.

Item	Comment
Has the access route been checked for clearance of the rack, including the minimum width and height requirements for all doors on the route?	
Do all the doors and entry ways conform to the width and height requirements for transportation, including the width of the unpacked unit?	
Do all the doors meet the height requirement of minimum 218 cm (86 inches) for packaged delivery?	
Does the access route provide sufficient space for transport of the packed devices?	
Are there any stairs, ramps, or thresholds that are of concern? If yes, provide details.	
Are all access route incline angles within the permitted range (under 6 degrees)?	
Have you confirmed that the access route is free of any obstacles that would expose the device to shock?	
Can the complete access route support the weight of the rack?	

Item	Comment
Is the access route onto the raised floor rated for dynamic loading of the rack?	
Are all the surfaces acceptable for rolling the new unpacked and packed equipment?	
If a pallet jack is to be used: • Can the pallet jack support the weight of the rack? • Are the pallet jack tines compatible with the shipping pallet?	
If there are stairs, is a loading elevator available for the equipment?	
If an elevator is to be used: • Does the elevator car meet the height, width, and depth requirements for carrying the rack? • Do the elevator doors meet the height and width requirements for moving the rack? • Does the elevator meet the weight requirements for transporting the rack?	

Facility Power Checklist

For specifications, see Electrical Power Requirements on page 58.

Item	Comment
Have the operating voltage and electric current requirements of the device and peripherals been reviewed?	
What type of power supply will be used? • Single-phase or 3-phase. • Low-voltage or High-voltage.	
Are enough power outlets provided within 2 meters (6.5 feet) for each rack?	
Do the power outlets have appropriate socket receptacles for the planned Power Distribution Units (PDUs)?	
Will optional ground cables be attached to the rack?	
Are the electrical circuits suitable in terms of voltage and current-carrying capacities?	
Does the power frequency meet the equipment specifications?	
Are power outlets available for the new equipment at the designated location?	
Will system power be delivered from two separate grids?	

Item	Comment
Is there a UPS to power the equipment?	
Are the minimum required power sources available to support the power load (kW or kVA) for the new hardware?	

Safety Checklist

Complete this checklist to ensure that the safety requirements are met. For information about safety, see Emergency Information for Private Cloud Appliance Installations on page 94 and Environmental Requirements on page 61.

Item	Comment
Is there an emergency power shut off?	
Is there a fire protection system in the data center room?	
Is the computer room adequately equipped to extinguish a fire?	
Is antistatic flooring installed?	
Is the area below the raised floor free of obstacles and blockages?	

Logistics Checklist

Complete this checklist to ensure that the logistics requirements are met. For information about receiving and unpacking, see Space Requirements on page 56.

Item	Comment
Is contact information for the data center personnel available?	
Is there security or access control for the data center?	
Are there any security background checks or security clearances required for Oracle personnel to access the data center? If yes, then provide the process for Oracle to follow.	
How many days in advance must background checks be completed?	
Are there any additional security access issues?	
Is computer room access available for installation personnel?	
Are laptops allowed in the data center?	
Are cell phones allowed in the data center?	
Are cameras allowed in the data center?	
Does the building have a delivery dock?	

Item	Comment
Is the site prepared for uncrating and packaging removal? Package removal should take place outside the data center room.	
Does the building have adequate receiving space?	
Is there a delivery / unpacking / staging area?	
Is the delivery / unpacking / staging area protected from the elements?	
Is inside delivery planned (direct to the final rack location in the data center room)?	
Is the unpacking area air-conditioned to avoid thermal shock for various hardware components?	
Will sufficient moving personnel be available to transport the rack?	
Is union labor required for any part of the delivery or installation?	
Is uncrating of cabinet and packaging removal required?	
Are there any restrictions on delivery truck length, width, or height?	
Is there storage space (cabinet) for the ride along spares? If not, does the customer allow cardboard boxes and other packing material in the computer room, since the spares are packed in cardboard boxes?	
Is there a time constraint on dock access? If yes, provide time constraints.	
Is a tail or side lift required on the delivery carrier to unload the equipment at the delivery dock?	
Will any special equipment be required to place the rack in the data center room? For example: • Stair walkers • Lifters • Ramps • Steel plates • Floor covers	
Does the delivery carrier require any special equipment, such as non-floor-damaging rollers, transport dollies, pallet jacks, or fork lifts?	

Final Preinstallation Checklist

Complete the following checklist to ensure that the network specification requirements are met. For more information about network requirements and IP addresses, see Preparing the Data Center Network Environment on page 62.

Network Specification Considerations	Yes, No, or N/A	Comment
Did you complete the networking worksheet, which is included in the “Initial System Installation Checklist”?		
Have you received the site-specific installation template from your Oracle technical representative?		
Did you review the installation template and consult with your Oracle technical representative regarding any site-specific changes, if any?		
Did you verify that IP addresses in the installation template are currently not in use?		
Have you performed the required configuration within your network infrastructure to allow the Private Cloud Appliance to use the IP addresses specified in the template?		
Have you registered IP addresses in the installation template with DNS?		
Did you run the required network cables from your network equipment to the location where the Private Cloud Appliance will be installed?		
Did you label the network cables that will connect to the Private Cloud Appliance?		

Initial System Installation Checklist

Complete this checklist to plan for the initial installation wizard, which walks you through the system initialization parameters and first boot process of your Private Cloud Appliance.

Work with your network and site administrators to plan the integration of the appliance into your data center. For initial configuration you can select either a dynamic network or static network configuration. This checklist contains information for both options, but you only need data for the implementation you select.

For more information about network requirements, see Preparing the Data Center Network Environment on page 62.

Note:

Checklist Legend

***** -- required fields for all configurations.

† -- required for static network configurations

‡ -- required for dynamic network configurations

General Configuration Parameters

Item	Your Configuration	Description and Example
Administrator user name and password		The rack doesn't ship with a default administrative user account. You create an administrator account during the initial installation. Passwords must contain at least 12 characters with at least one of each: uppercase character, lowercase character, digit, punctuation character, and no double quote ("").
Logical connection		Choose ***static*** or ***dynamic*** routing. Both routing options are supported by all three physical topologies (Mesh, Square, Triangle). Static routing implies that all egress traffic goes through a single default data center gateway IP address. It must be in the same subnet as the rack uplink IP addresses, so it is reachable from the spine switches. Dynamic routing implies that BGP (Border Gateway Protocol) is used to establish a TCP connection between two Autonomous Systems: the rack network and the data center network.
Availability Domain*	`AD-1`	The Private Cloud Appliance is considered an availability domain and is assigned the value `AD-1`.

Private Cloud Appliance Rack Details

Item	Your Configuration	Description and Example
System Name*		Name for the system, and used as the Private Cloud Appliance short name or Realm. This attribute has a maximum length of 24 characters. Acceptable characters are "a" to "z", "A" to "Z", "0" to "9", and "-". **Once set, this parameter can't be changed.**
Domain*		Domain name for your system which is used as the base domain for the internal network, and by Private Cloud Appliance public facing services. This attribute has a maximum length of 190 characters. Acceptable characters are "a" to "z", "A" to "Z", "0" to"9", "-" Example: `us.example.com` **Once set this parameter can't be changed.**
Rack Name		Provide a unique name for this Private Cloud Appliance system.

Item	Your Configuration	Description and Example
Description		Provide an optional description for Private Cloud Appliance. This is a free text field with a 256 character limit.

Static Routing Network Details

Item	Your Configuration	Description and Example
Uplink Gateway†		IP address for the uplink switch to the default gateway in your data center. Chose a valid IP address in the customer data center subnet that the rack uplink switches are attached to. Example: `10.68.48.86`
Spine switch virtual IP†		Virtual IP address which acts as the public VIP for the spine switches in your company network. Chose a valid IP address in the data center subnet that the rack uplink switches are attached to. Example: `10.68.49.103`
Uplink VLAN		VLAN used to connect to an uplink switch. Chose a VLAN value between 2 and 3899. VLANs 3900 to 4095 are reserved. Example: `322`
Uplink HSRP Group		Assign an HSRP group number to the rack. Acceptable values are 0 to 255, and the default value is 151. If there are multiple racks connected to the same data center infrastructure switches, ensure they use *different* HSRP groups.
Management node 1 IP addresses and host name*	10/25/40/100G*: 1G: host name:	Static IP addresses for management node 1 in your on-premises network. Provide a valid host name. An appended domain name is used if you don't provide a host name.
Management node 2 IP addresses and host name*	10/25/40/100G*: 1G: host name:	Static IP addresses for management node 2 in your on-premises network. Provide a valid host name. An appended domain name is used if you don't provide a host name.
Management node 3 IP addresses and host name*	10/25/40/100G*: 1G: host name:	Static IP addresses for management node 3 in your on-premises network. Provide a valid host name. An appended domain name is used if you don't provide a host name.

Item	Your Configuration	Description and Example
Management virtual IP address and host name*	10/25/40/100G*: 1G: host name*:	Virtual IP addresses of the management node cluster in your on-premises network. This is the IP used to DNAT to the internal management node VIP. Private Cloud Appliance DNS will resolve Private Cloud Appliance endpoints to this IP. Chose a valid IP address in the data center CIDR. Example: `10.68.49.170` Provide a valid host name. An appended domain name is used if you don't provide a host name.
Spine switch 1 IP address*		IP address for the spine switch 1 in your on-premises network. Chose a valid IP address in the data center subnet that the rack uplink switches are attached to. Example: `10.68.49.101`
Spine switch 2 IP address*		IP address for the spine switch 2 in your on-premises network. Chose a valid IP address in the data center subnet that the rack uplink switches are attached to. Example: `10.68.49.102`

Dynamic Routing Network Details

Item	Your Configuration	Description and Example
Peer 1 IP‡		Neighbor BGP IP - IP address of customer data center router-1 that the rack uplink switches are attached to. Example: `10.68.48.86`
Peer 1 ASN‡		BGP ASN of customer data center router-1. Example: `64512-65533 or 4200000000-4294967294` (Note: Oracle Spine switch default ASN is 136025.)

Item	Your Configuration	Description and Example
Peer 2 IP‡		Neighbor BGP IP - IP address of customer data center router-2 that the rack uplink switches are attached to. Example: `10.68.48.88`
Peer 2 ASN‡		BGP ASN of customer data center router-2 Example: `64512-65533 or 4200000000-4294967294` (Note: Oracle Spine switch default ASN is 136025.)
Uplink Gateway*		IP address for the uplink switch to the default gateway in your data center. Chose a valid IP address in customer data center subnet that the rack uplink switches are attached to. Example: `10.68.48.86`
Oracle ASN		The default is 136025.
BGP Topology		Options are triangle, square, or mesh. The default is mesh.
BGP Authentication (Optional) Admin BGP Authentication		Enable BGP authentication for your network, and admin network if used.
BGP Authentication Password (Required)		Enter the BGP authentication password for your network, and admin network if used.
Admin BGP Authentication Password (Required if admin network used)		Enter password in hex format. Password must be less than 64 characters (<64).
BGP KeepAlive Timer		The default is 60.
BGP Holddown Timer		The default is 180.
Enable MDA Authentication		The default is false.

Uplink Details

Item	Your Configuration	Description and Example
Uplink Port Speed*		All uplink ports must have the identical speed. The options are 10, 25, 40, 100. The default is 100.

Item	Your Configuration	Description and Example
Uplink Port Count*		The number of uplink ports per spine switch. Connectivity must be identical on both spine switches to provide redundancy and load-splitting. For 100G and 40G speeds, valid values are 1, 2 and 4. For 10G and 25G ports, valid values are 1, 2, 4, 8 and 16. The default for both is 4.
Uplink VLAN MTU*		MTU size determines the maximum packet size that can be transmitted over your uplink connection. The valid range is 576 to 9216. The default is 9216.
Uplink Netmask*		Netmask of the subnet rack is connected to in your data center. Example: `255.255.252.0`
Uplink Port FEC		Configure Forward Error Correction (FEC) for the uplink port. The default option is auto, with on and off as other valid options.

NTP and DNS Server Details

Item	Your Configuration	Description and Example
NTP servers*		At least one valid IP address for an NTP server in your data center. You can enter multiple IP addresses in a comma-separated list. Example: `10.147.24.1,10.211.17.1`
DNS server 1		IP address for primary DNS server. Example: `10.68.50.60`
DNS server 2		IP address for secondary DNS server. Example: `10.147.36.60`
DNS server 3		IP address for tertiary DNS server. Example: `206.233.27.1`

Administration Network Details (Optional)

This information is required only if you decide to implement a physically segregated appliance administration network.

Item	Your Configuration	Description and Example
Admin Networking	enable	Enable a separate Administration network to transport administrative traffic.
Management virtual IP address and host name*	10/25/40/100G*: 1G: host name*:	Virtual IP address of the management node cluster in the administration network. This is the IP used to DNAT to the internal management node VIP. Private Cloud Appliance DNS will resolve Private Cloud Appliance endpoints to this IP. Chose a valid IP address in the administration network CIDR. Example: `10.168.141.170` Provide a valid host name. An appended domain name is used if you don't provide a host name.
Management node 1 IP addresses and host name	10/25/40/100G: 1G: host name:	Static IP addresses for management node 1 in the administration network. Provide a valid host name. An appended domain name is used if you don't provide a host name.
Management node 2 IP addresses and host name	10/25/40/100G: 1G: host name:	Static IP addresses for management node 2 in the administration network. Provide a valid host name. An appended domain name is used if you don't provide a host name.
Management node 3 IP addresses and host name	10/25/40/100G: 1G: host name:	Static IP addresses for management node 3 in the administration network. Provide a valid host name. An appended domain name is used if you don't provide a host name.
Admin DNS servers		IP addresses for 1-3 DNS servers providing name resolution in the administration network. Example: `10.168.20.31,10.168.20.32,10.147.36.60`
Admin Port Speed		Port speed options are 10, 25, 40, 100. The port speed of your administration port can be different from the data port speed.
Admin Port Count		For 100G and 40G speeds, valid value is 1. For 10G and 25G ports, valid values are 1 to 4.

Item	Your Configuration	Description and Example
Admin HSRP Group		Assign an HSRP group number to your Administration network. Acceptable values are 0 to 255, and the default value is 152. If there are multiple racks connected to the same data center infrastructure switches, ensure they use *different* HSRP groups.
Admin VLAN		VLAN used to connect to the Administration network (only access mode supported). The valid range for users to select is 2 to 3899. The default is 3915, in the Oracle-reserved range.
Admin VLAN MTU		MTU size determines the maximum packet size that can be transmitted over your administration connection. The valid range is 576 to 9216. The default is 9216.
Admin Port FEC		Configure Forward Error Correction (FEC) for the administration port. The default option is auto, with on and off as other valid options.
Admin Gateway IP		IP address of the default gateway in your Administration network. Example: `10.168.141.1`
Admin Netmask		Netmask of the subnet the Administration network is connected to.
Admin CIDR		CIDR range for which the default route is the Administration gateway IP.
Admin Spine 1 IP		Public IP address of spine switch 1. Needed for HSRP configuration or the spine virtual IP.
Admin Spine 2 IP		Public IP address of spine switch 2. Needed for HSRP configuration or the spine virtual IP.
Admin Spine VIP		Public virtual IP of the spine switches.

Private Cloud Appliance Public IP Addresses

Item	Your Configuration	Description and Example
Public IP Addresses		A range of customer data center IP addresses can be used for Private Cloud Appliance components that require public IP addresses. Note: In this context, public IP addresses refer to IP addresses that have access to the data center network from the Private Cloud Appliance subnet. You must specify IP addresses or ranges that are routed to the Private Cloud Appliance from the data center. Route tables must be correct and consistent. Enter a string containing a comma separated list of valid CIDRs. Example: `"10.68.49.249","10.68.50.32/32","10.68.51.4/31"` Partial CIDR deletion isn't supported.
Object Storage IP address		Valid IP address for an `objectstorage` endpoint that is outside the public IP range.

Installing the Private Cloud Appliance Rack in the Data Center

When the Private Cloud Appliance hardware has arrived at the installation site, follow the instructions in this section to unpack the rack, install it in the chosen location, connect all cables, and power on the system.

Before starting the installation tasks, ensure that you have completed all preparations:

- Preparing the Installation Site on page 55
- Preparing the Data Center Network Environment on page 62

Review important safety and emergency information. See Emergency Information for Private Cloud Appliance Installations on page 94.

Follow the installation instructions in these sections:

- Unpacking the Private Cloud Appliance Rack on page 82
- Placing the Appliance Rack in the Allocated Space on page 84
- Connecting Private Cloud Appliance to the Data Center Network on page 89
- Powering On Private Cloud Appliance for the First Time on page 90

Unpacking the Private Cloud Appliance Rack

The unpacking location for the Private Cloud Appliance should be determined during the site planning process. When the shipment has arrived, verify that the required tools are present, and follow the instructions on the shipping carton.

Tools Required for Installation

The following tools and equipment are required for installation and service:

- Unpacking tools (provided in rack ship kit)
  - 17mm and 13mm open-ended double-sided wrench
  - Allen L-key 6mm hex wrench, 5-3/4 inches in length
  - Allen L-key 8mm hex wrench, 6-3/8 inches in length
- Rack setup tools and accessories (provided in rack ship kit)

  - 32 M6 cage nuts and washers
  - 32 M6 pan head screws
  - 8 M5 screws
  - 8 self-tapping Torx screws
  - 4 PDU brackets
  - 4 M5 cage nuts
  - 4 grounded power cables
  - 18mm and 16mm open-ended double-sided wrench
  - T-40 Torx screwdriver
  - T-25 Torx screwdriver
  - T-15 Torx screwdriver
  - 8mm and 6mm Allen wrench
  - Cage nut tool
  - Keys to the front door, rear door, and side panel locks
  - 17mm and 13mm open-ended double-sided wrench
- Customer-supplied tools (not provided in rack ship kit)
  - No. 2 Phillips screwdriver
  - Diagonal cutter
  - Antistatic wrist strap

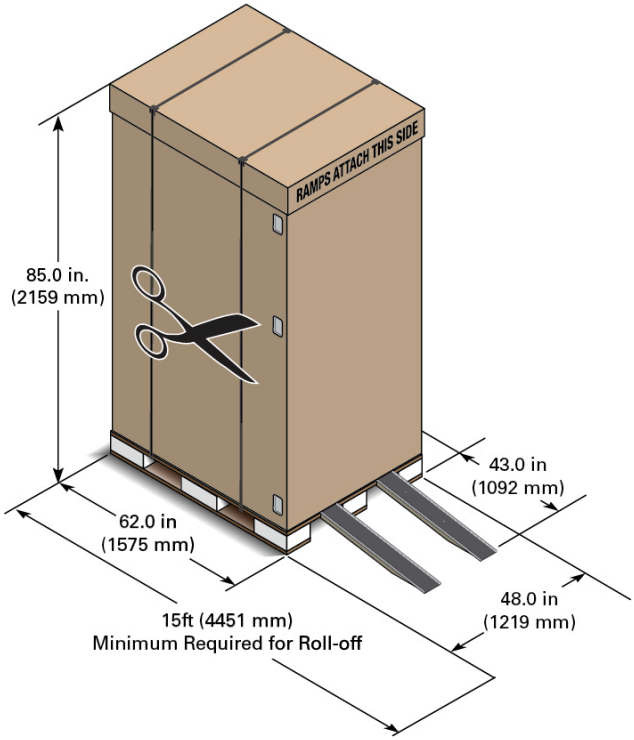
Unpack the System

Before unpacking the rack from the shipping carton, refer to the labels on the carton and to the instructions that they provide. After unpacking the rack, follow local laws and guidelines to recycle the packaging properly.

> **Caution:**
>
> Carefully unpack the rack from the packaging and shipping pallet. Rocking or tilting the rack can cause it to fall over and cause serious injury or death. You should always use professional movers when unpacking and installing this rack.

> **Note:**
>
> After unpacking the rack from the packaging, save the shipping brackets used to secure the rack to the shipping pallet. You can use these shipping brackets to secure the rack permanently to the installation site floor. Do not dispose of these brackets, because you cannot order replacement brackets.

> **Caution:**
>
> Shipping brackets are not for use for bracing or anchoring the rack during seismic events.



Placing the Appliance Rack in the Allocated Space

After unpacking, move the Private Cloud Appliance to the data center room. Position the rack in its allocated space and stabilize it.

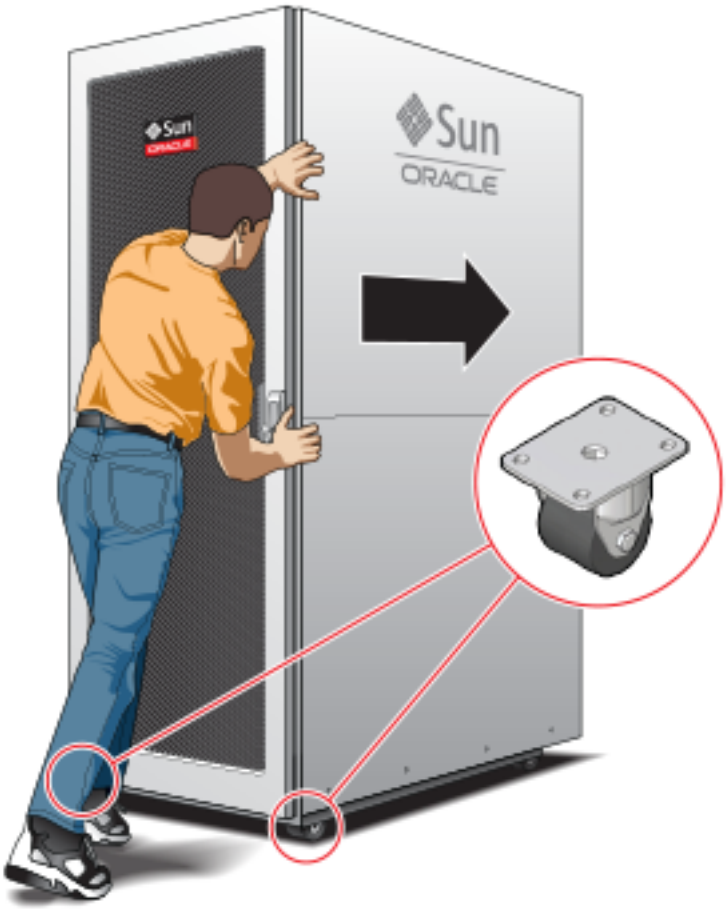
Moving the System

Follow these instructions to move the rack.

1. Ensure that the rack doors are closed and secured.
2. Ensure that the leveling and stabilizing feet on the rack are raised and out of the way.
3. Push the system from the back of the rack to the installation site.

> **Attention:**
>
> Never attempt to move the rack by pushing on the rack side panels. Pushing on the rack side panels can tip over the rack. This action can cause serious personal injury or death, as well as damage to the equipment.

The front casters of the rack are fixed; they do not pivot. When moving the appliance rack to the installation site, you must steer the unit using the rear casters. You can safely maneuver the system by carefully pushing it from behind.



It is preferred that at least three people push and guide the rack: one person in front and two persons in back to help guide the rack and keep people out of the path of the moving rack. When transporting configured racks from one location to another, take care to move them slowly, 0.65 meters per second (2.13 feet per second) or slower.

Carefully examine the transportation path. Avoid obstacles such as doorways or elevator thresholds that can cause abrupt stops or shocks. Go around obstacles by using ramps or lifts to enable smooth transport.

> **Caution:**
>
> Never tip or rock the appliance rack because it can fall over.

4. When the rack is at the installation site, verify that no components or connections have become dislodged or disconnected during transport. If necessary, reattach components and cables properly.

Stabilizing the Rack

When the appliance rack is in position, stabilize it to ensure that it does not move or tip over. You can stabilize the rack permanently by extending the rack leveling feet, and optionally using the shipping brackets to secure the rack permanently to the floor.

> **Caution:**
>
> Shipping brackets are not for use for bracing or anchoring the rack during seismic events.

To secure the rack to the installation floor using the shipping brackets, you must drill the appropriate holes in the floor, reattach the shipping brackets to the rack, position the rack over the mounting holes, and attach the shipping brackets to the floor firmly with bolts and washers that suit the specific environment. Mounting bolts and washers

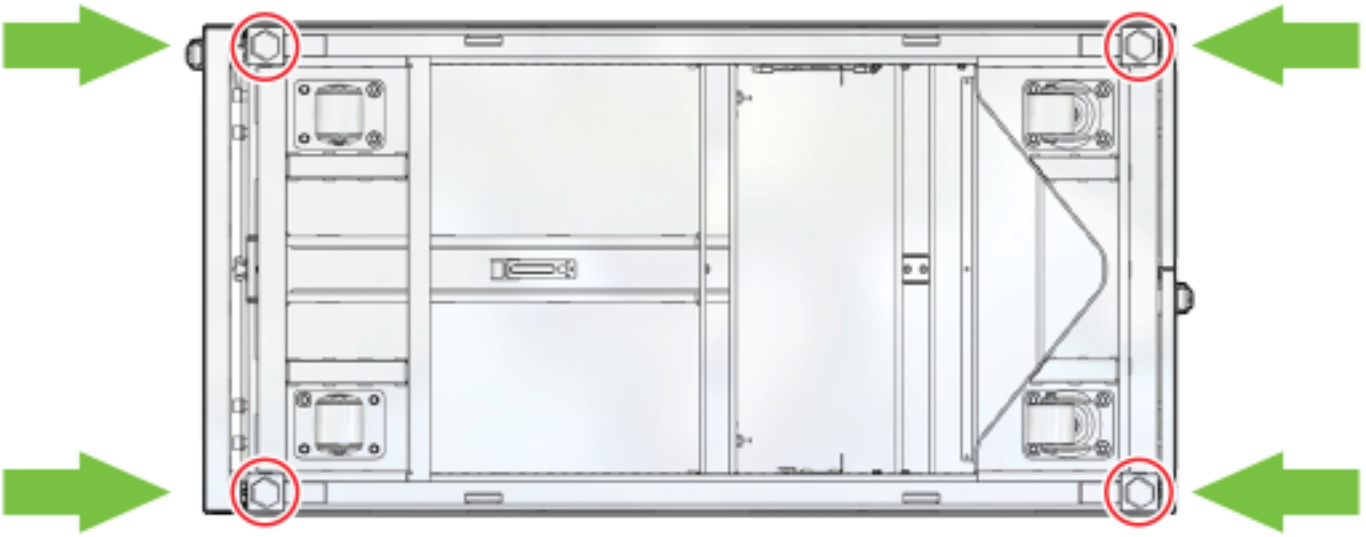
for the shipping brackets are not included in the shipment, because different floors require different bolt types and strengths.

(Optional) If you plan to route data or power distribution unit (PDU) power cords down through the bottom of the rack, you will need to cut a hole in the installation site floor. Cut a rectangular hole below the rear portion of the rack, between the two rear casters and behind the RETMA (Radio Electronics Television Manufacturers Association) rails.

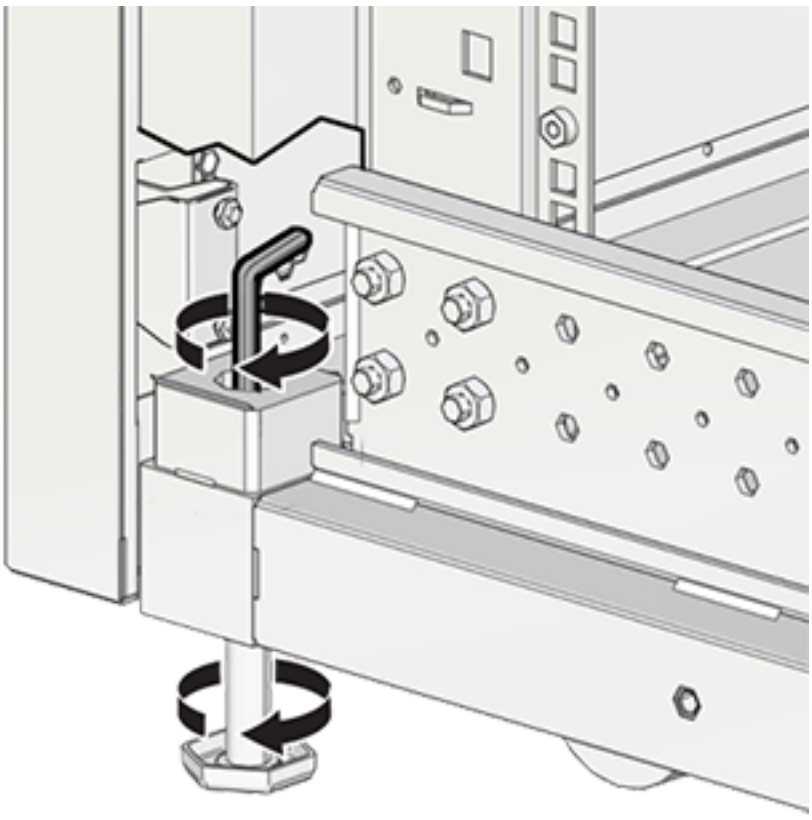
> **Caution:**
>
> Do not create a hole where the rack casters or leveling feet brackets will be placed.

When the rack is in position, the leveling feet must be deployed. The rack contains four leveling feet that can be lowered to share the load with the casters. This increases the footprint of the rack, which improves stability and helps prevent rack movement. The leveling feet must be used even when the rack is permanently secured to the floor. To adjust the leveling feet, do the following:

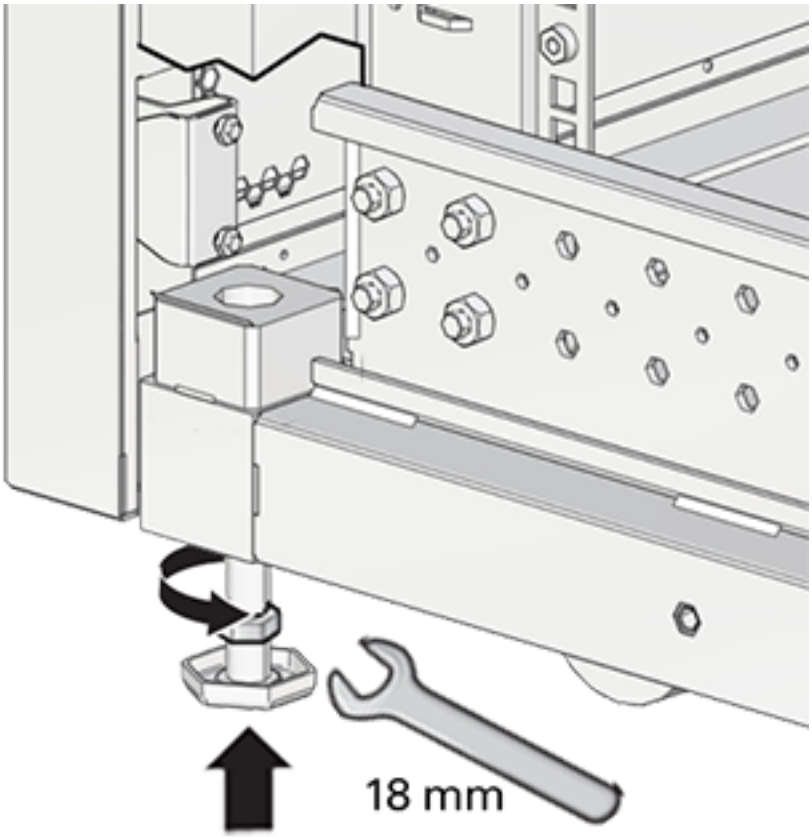
1. Locate the four leveling feet at the bottom four corners of the rack.



2. Using a 6mm hex wrench, lower the leveling feet to the floor.



3. Lock the four leveling feet using an 18mm open wrench.



When lowered correctly, the four leveling feet share the load with the casters to increase footprint, improve stability, and help support the full weight of the rack.

> **Caution:**
>
> When the rack needs to be moved to a different location, including repacking, verify that the leveling feet have been retracted before moving the rack. Otherwise the leveling feet may become bent, or the rack could tip over.

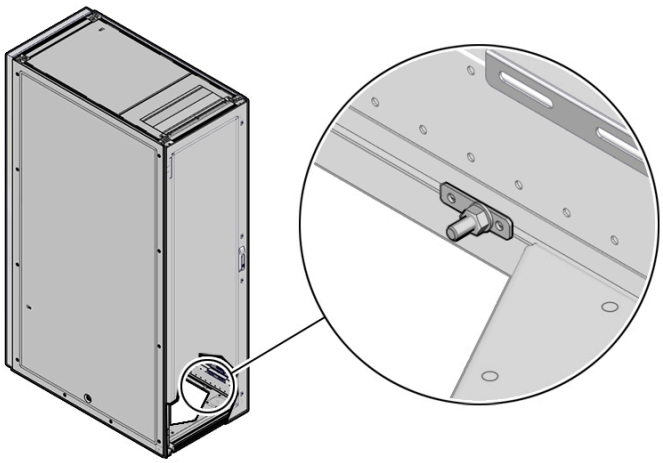
Attaching a Ground Cable (Optional)

The rack power distribution units (PDUs) achieve earth ground through their power cords. Final chassis ground is achieved by way of the ground prong when you connect the power cord to a socket. For additional grounding, attach a chassis earth ground cable to the system. The additional ground point enables electrical current leakage to dissipate more efficiently.

> **Caution:**
>
> The PDU power input lead cords and the ground cable must reference a common earth ground. If they do not, then a difference in ground potential can be introduced. If you are unsure of your facility's PDU receptacle grounding, then do not install a ground cable until you confirm that there is a proper PDU receptacle grounding. If a difference in ground potential is apparent, then you must take corrective action.

> **Note:**
>
> A grounding cable is not shipped with the Private Cloud Appliance.

1. Ensure that the installation site has a properly grounded power source in the data center. The facility PDU must have earth ground.
2. Ensure that all grounding points, such as raised floors and power receptacles, reference the facility ground.
3. Ensure that direct, metal-to-metal contact is made for this installation. During manufacturing, the ground cable attachment area might have been painted or coated.
4. Attach the ground cable to one of the attachment points located at the bottom rear of the system frame. The attachment point is an adjustable bolt that is inside the rear of the system cabinet on the right side.



Connecting Private Cloud Appliance to the Data Center Network

Before powering on the Private Cloud Appliance for the first time, ensure that the necessary network cable connections have been installed between the appliance rack and the data center network devices.

For external connectivity, 5 ports are reserved on each spine switch. Four ports are available to establish the uplinks between the appliance and the data center network; one port is reserved to optionally segregate the administration network from the data traffic.

If administration traffic is physically segregated, the administration network and the data network can be configured at different speeds. For example, you can configure your administration network to operate at 10Gbit, and your data network to operate at 40Gbit.

> **Note:**
>
> Peer connections between Private Cloud Appliance systems are not covered here. See Establishing a Peer Connection on page 300.

On each spine switch, ports 1 to 4 can be used for uplinks to the data center network. For speeds of 10Gbps or 25Gbps, the spine switch port must be split using a 4-way splitter or breakout cable. For higher speeds of 40Gbps or 100Gbps each switch port uses a single direct cable connection. For more information, see Uplinks on page 29.

At a minimum, you must connect 1 port on each spine switch, which provides a single high bandwidth, high availability network for the administration and data traffic.

Use the following table to determine the correct configuration for your environment.

> **Caution:**
>
> It is critical that **both** spine switches have a connection to a pair of next-level data center switches. This configuration provides redundancy and load splitting at the level of the spine switches and the data center switches. The cabling pattern is essential to the continuation of service during failover scenarios.

Use Case	Network Speed	Cables Required	Ports	Administration Network Port
Minimum Configuration	10Gbit or 25Gbit	2 splitter cables	Port 1 on each spine switch	Not Used
Minimum Configuration	40Gbit or 100Gbit	2 direct cables	Port 1 on each spine switch	Not Used
Minimum Configuration with Administration Network	10Gbit or 25Gbit	4 splitter cables	Port 1 on each spine switch	Port 5 on each spine switch
Minimum Configuration with Administration Network	40Gbit or 100Gbit	4 direct cables	Port 1 on each spine switch	Port 5 on each spine switch
Maximum Configuration	10Gbit or 25Gbit	8 splitter cables	Ports 1-4 on each spine switch	Not Used
Maximum Configuration	40Gbit or 100Gbit	8 direct cables	Ports 1-4 on each spine switch	Not Used

Use Case	Network Speed	Cables Required	Ports	Administration Network Port
Maximum Configuration with Administration Network	10Gbit or 25Gbit	10 splitter cables	Ports 1-4 on each spine switch	Port 5 on each spine switch
Maximum Configuration with Administration Network	40Gbit or 100Gbit	10 direct cables	Ports 1-4 on each spine switch	Port 5 on each spine switch

Powering On Private Cloud Appliance for the First Time

When the rack is in its allocated space and connected to the data center network, follow the instructions in this section to power on Private Cloud Appliance for the first time.

Before you begin, review the safety guidelines in Preparing the Installation Site on page 55.

Inspecting the Appliance

Follow these steps to visually inspect your Private Cloud Appliance after it is physically installed at your site, and prior to power being applied to the system.

1. Check for rack damage.
2. Check the rack for loose or missing screws.
3. Check your Private Cloud Appliance for the ordered configuration. Refer to the Customer Information Sheet (CIS) on the side of the packaging. Retain this information so you can confirm all of the components are discovered during system initialization.

   > **Note:**
   >
   > Private Cloud Appliance is preconfigured by Oracle as a self-contained system. You should not move any equipment or add any unsupported hardware to the system.

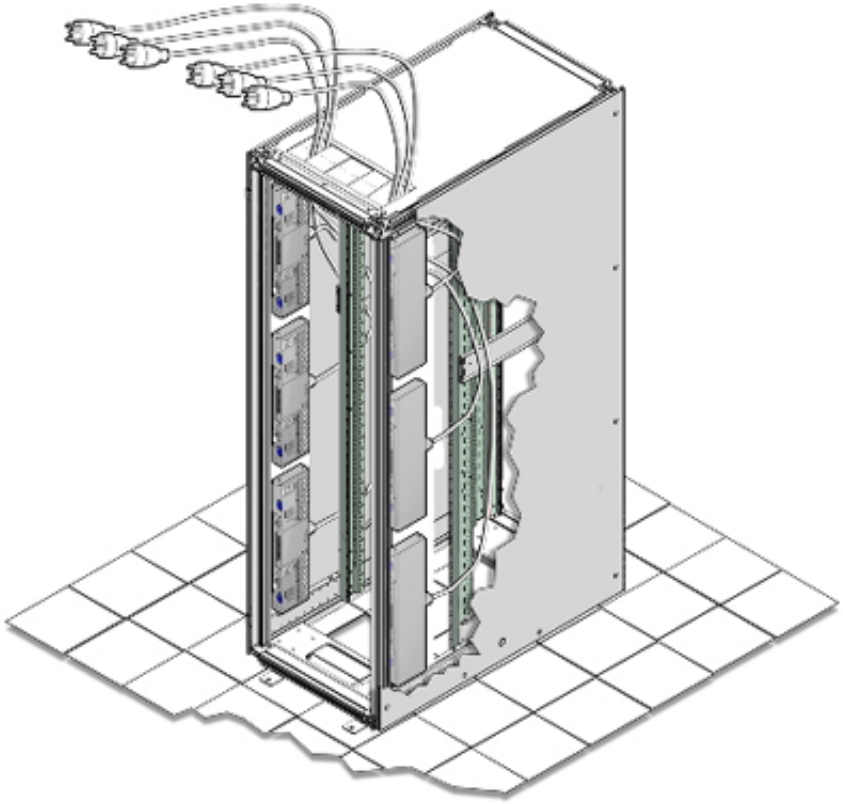
4. Check that all cable connections are secure and firmly in place as follows:
   a. Check the power cables. Ensure that the correct connectors have been supplied for the data center facility power source.
   b. Check the network data cables.
5. Check the site location tile arrangement for cable access and airflow
6. Check the data center airflow that leads in to the front of the system.

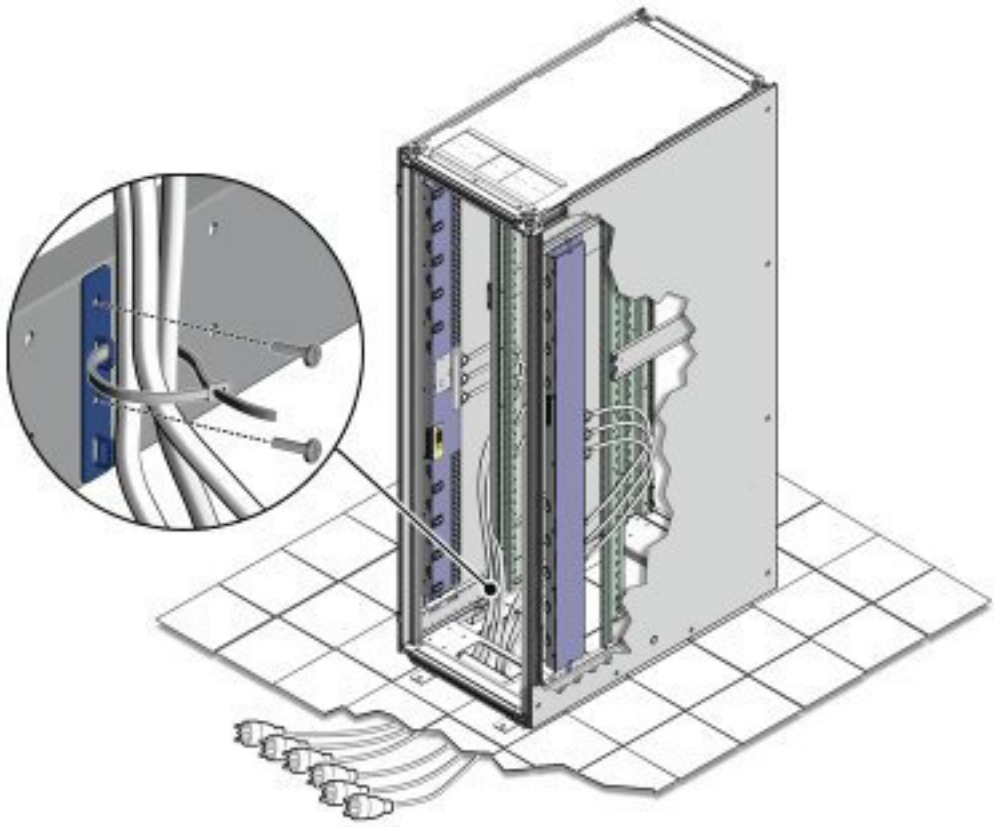
   For more information, see Environmental Requirements on page 61 (Ventilation and Cooling).

Connecting the Power Cords

Follow these steps to connect power cords to your Private Cloud Appliance.

1. Open the rear cabinet door.
2. Ensure that the correct power connectors have been supplied.
3. Unfasten the power cord cable ties. The ties are for shipping only and are no longer needed.
4. Route the power cords to the facility receptacles either above the rack or below the flooring.





5. Secure the power cords in bundles.
6. Plug the power distribution unit (PDU) power cord connectors into the facility receptacles. Ensure the breaker switches are in the OFF position before connecting the power cables.

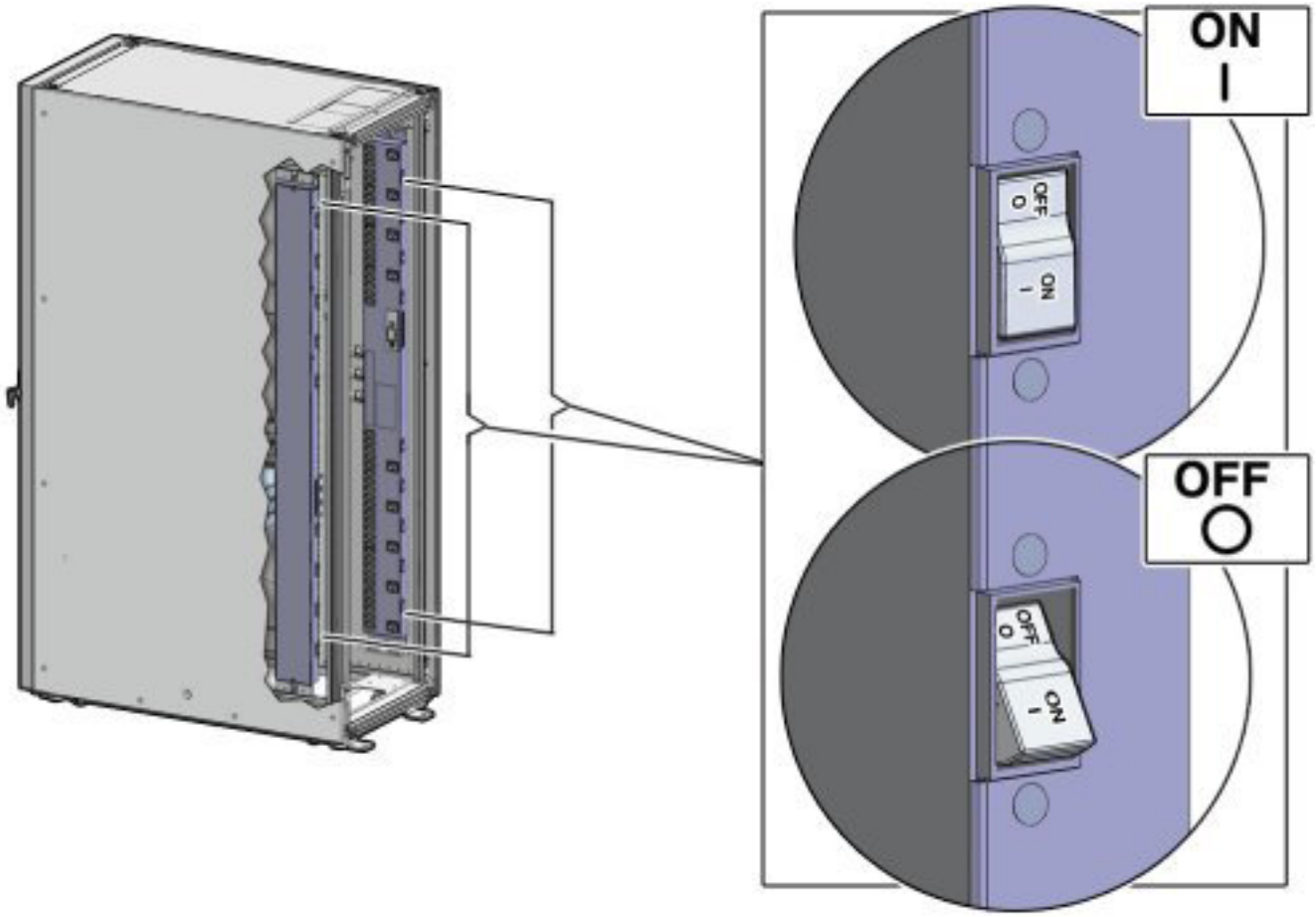
Applying Power to the System

Follow these steps to power on your Private Cloud Appliance.

> **Note:**
>
> You can connect to the appliance using a network connection to monitor the system power-on procedure. For instructions, see Connecting a Workstation to the System on page 97.

1. Ensure that the power switches located on the rear left and right side power supplies of the Oracle Storage Drive Enclosure DE3-24C and Oracle Storage Drive Enclosure DE3-24P are in the ON (|) position.

2. Switch on the power distribution unit (PDU) circuit breakers located on the rear of PDU A and B inside the rack. The circuit breakers are on the rear of the system cabinet as shown below. Press the ON (|) side of the toggle switch.



After power is applied, the LEDs on the all of the compute nodes and storage server heads will start to blink after approximately two minutes. From the rear of the rack, you can see the green LEDs on the power supply units (PSUs) on the compute nodes turn on instantly after power is applied. In addition, from the rear of the rack, you can see the display on the power distribution units (PDUs) illuminate once power is available.

> **Note:**
>
> Allow 20 minutes for the storage controllers to come online before powering on each management node.

3. Press the Power button located on the front of each management node.

   The first management node is located in rack unit 5 (U5). The second management node is located in rack unit 6 (U6), and the third management node is located in rack unit 7 (U7). A management nodes take approximately five to ten minutes to power on completely. Once complete, the Power/OK LED illuminates and remains a steady green.

   The management nodes will verify all components within the system. The management nodes ensure that the correct networking switches and storage devices are installed in the system, and search for compute nodes to power on and add to the compute fabric.

   Depending on your system configuration, powering on the compute nodes and bringing them to the ready-to-provision state should take approximately 10 minutes *per compute node*. Do not power cycle the management nodes during the discovery period. Proceed to configuring the appliance.

> **Caution:**
>
> Once powered on, do not power down the management nodes until you have completed the initial configuration process described in Completing the Initial Setup and Configuration on page 97.

Emergency Information for Private Cloud Appliance Installations

Review this Private Cloud Appliance emergency information to minimize the risk of injury and damage in the event of an emergency.

Cautions and Warnings

The following cautions and warnings apply to Private Cloud Appliance:

> **Warning:**
>
> Do not touch the parts of this product that use high-voltage power. Touching them might result in serious personal injury.

> **Warning:**
>
> Do not power off the system unless there is an emergency.

> **Caution:**
>
> Keep the front and rear cabinet doors closed. Failure to do so might cause system failure or result in damage to hardware components.

> **Caution:**
>
> Keep the top, front, and back of cabinets clear to allow proper airflow and prevent overheating of components.

> **Caution:**
>
> Use only the supplied hardware.

Emergency Power-off Considerations

If there is an emergency, then power to Private Cloud Appliance should be halted immediately. The following emergencies might require powering off the system:

- Natural disasters such as an earthquake, flood, hurricane, tornado or cyclone
- Abnormal noise, smell or smoke coming from the system
- Threat to human safety

> **Note:**
>
> If you use the compute instance high availability feature, it's advised to disable it before a system shutdown and enable it again after restart, when the compute nodes are back in normal operation. This requires access to the Service CLI, which is available on Private Cloud Appliance, but restricted to the cloud operations team on Compute Cloud@Customer.
>
> ```
> PCA-ADMIN> disableVmHighAvailability
> PCA-ADMIN> enableVmHighAvailability
> ```

Emergency Power-off Procedure

To perform an emergency power-off procedure, turn off power at the circuit breaker or pull the emergency power-off switch in the computer room.

Emergency Power-off Switch

Emergency power-off (EPO) switches are required in a computer room when computer equipment contains batteries capable of supplying more than 750 volt-amperes for more than five minutes. Systems that have these batteries include internal EPO hardware for connection to a site EPO switch or relay. Use of the EPO switch will remove power from the Private Cloud Appliance installation.

Restarting the Private Cloud Appliance System After a Power Outage

In case the Private Cloud Appliance was powered off or experienced an unexpected power outage, system components need to be restarted in a particular order.

If Private Cloud Appliance is shut down in a controlled manner, verify the compute instance high availability configuration. If it's active, you must disable it before shutting down the system. From the Service CLI, run this command: `disableVmHighAvailability`. Remember to enable this feature again when the system is back online.

When power has been reapplied to the system power distribution units, restart the components as follows:

1. Power on the switches.
2. The ZFS Storage Appliance should power on automatically when power is restored to the PDUs.

   Wait for the storage appliance to fully boot before continuing.
3. Power on the management nodes from their ILOMs using the `start /SYS` command, or by pressing the power button on the front of the node.

   Wait until the management nodes are in ready state and running. Ensure that the Kubernetes microservices pods are running on all 3 management nodes before you proceed.
4. Power on the compute nodes from their ILOMs using the `start /SYS` command.

   Wait until the compute nodes are in ready state and running. Ensure that the Kubernetes microservices pods are running on all compute nodes before you proceed.
5. If you use the VM High Availability feature, and it was disabled, you must enable this feature again when restarting the appliance. From the Service CLI, run this command: `enableVmHighAvailability`.
6. Virtual machines on the compute nodes restart on their own, as a result of the VM HA features or the Restore Instance Lifestyle State feature.

   If an instance remains in the STOPPED state after Power On, select an option to restart it manually:

   - From the OCI CLI, run this command:

     ```
     oci compute instance action --instance-id <Instance-ID> --action START
     ```

   - In the Compute Web UI, find the stopped instance in its compartment, click the Actions menu, and select Start.

Configuring the Private Cloud Appliance System

When the Private Cloud Appliance rack hardware has been installed, and the system has been powered on, the initial setup and configuration procedure must be performed.

Ensure that the Initial System Installation Checklist on page 74 has been completed in advance. It contains the information required to configure the system.

Before you connect to the Private Cloud Appliance for the first time, ensure that you have made the necessary preparations for external network connections. See Preparing the Data Center Network Environment on page 62.

Initial Setup

Follow these setup instructions:

- Connecting a Workstation to the System on page 97
- Completing the Initial Setup and Configuration on page 97
- Connecting a Bastion Host to Private Cloud Appliance (Optional) on page 120

Optional Expansions

This section also provides installation information for optional system expansions:

- Expanding Private Cloud Appliance with GPU Capacity on page 122
- Expanding Private Cloud Appliance Capacity with a Compute Expansion Rack on page 123
- Connecting External Systems to Private Cloud Appliance Using Flex Networks on page 124

What's Next?

When the initial setup and configuration of your Private Cloud Appliance is complete, you can begin to customize the appliance for use.

> **Note:**
>
> Ensure that the compute nodes are provisioned before you hand off a newly created tenancy to the tenancy administrator. Unprovisioned compute nodes can cause VCN creation to fail. Ensure that platform images have been imported.

Task	Directions	Background Information
Configuring ASR	Registering Private Cloud Appliance for Oracle Auto Service Request on page 272.	Oracle Auto Service Request (ASR) on page 272.
Creating a new administrator account	#unique_84.	Private Cloud Appliance Infrastructure Administration on page 145.
Provision compute nodes	Performing Administrative Operations on Compute Nodes on page 156.	Server Nodes Installed in Private Cloud Appliance on page 25".
Provide platform images	Providing OCI Images on page 194	Images
Create tenancies	Managing Private Cloud Appliance Tenancies on page 191	Enclaves and Interfaces on page 8
Install the OCI CLI in the Compute Enclave	Installing and Using the OCI CLI on page 397	Enclaves and Interfaces on page 8
Create an Internal-Only VNIC and Subnet	Managing VCNs and Subnets on page 478	Virtual Cloud Networks on page 465
Create a Network-Accessible VNIC and Subnet	Managing VCNs and Subnets on page 478 Configuring VCN Gateways on page 505	Virtual Cloud Networks on page 465
Import a compute image	Managing Custom Images on page 622	Images
Launch a compute instance	Tutorial: Launching Your First Instance on page 640	Compute Instances on page 634

Task	Directions	Background Information
Connect to a compute instance	Tutorial Task 7: Connect to Your Instance on page 645	Compute Instances on page 634
Get the status of a submitted job	`PCA-ADMIN> show Job id=<jobid>`	Using the Service CLI on page 150

Connecting a Workstation to the System

Connect a laptop or workstation to the Private Cloud Appliance in order to start system configuration with the initial installation process.

> **Note:**
>
> You access the initial configuration wizard through the Service Web UI using a web browser. For support information, see the Oracle Software Web Browser Support Policy.

1. Connect a workstation with a web browser directly to the management network using an Ethernet cable connected to port 2 in the management switch.
2. Configure the wired network connection of the workstation to use the static IP address `100.96.3.254/23`. You can also add `100.96.1.254/23` as another IP address if needed.

   Using the web browser on the workstation, connect to the Private Cloud Appliance initial configuration interface on the active management node at `https://100.96.2.32:30099`. This `100.96.2.32` address is the predefined virtual IP of the management node cluster for configuring Private Cloud Appliance

   Complete the initial system configuration using the GUI (preferred) or the Service CLI. See Configuring the Private Cloud Appliance System on page 95

Completing the Initial Setup and Configuration

The Private Cloud Appliance initial configuration wizard creates an administrator account, binds the system to your Oracle Cloud Infrastructure environment, and configures network connections for the appliance. When the initialization process is complete, the cloud services come online and you can begin to build your cloud.

The GUI is the preferred method to perform the initial configuration procedure. The Service CLI could be used as a secondary option.

Use the information from your completed Initial System Installation Checklist on page 74 to fill out the required parameters as you walk through the steps of the initial configuration procedure. Ensure the web browser on your workstation is connected to the Private Cloud Appliance initial configuration interface on the active management node at `https://100.96.2.32:30099`.

> **Caution:**
>
> Do not power down the management nodes during the initial configuration process.

> **Note:**
>
> If you configure a separate administration network, read and understand the administration network guidelines described in Data Center Network Configuration Guidelines on page 66.

Operating Modes

The relationship between Oracle Cloud Infrastructure and the OCI compatible compute infrastructure installed on the edge of the cloud is defined by the system operating mode. The parameters for the operating mode are locked in

during the first few steps of the initial system setup. These don't provide installation choices; each type of edge cloud system has its specific combination of operating mode parameters.

- **Link to OCI**

  When a system is linked to OCI, it's represented in the home OCI tenancy by an infrastructure resource. Linked infrastructures enable standard OCI billing and cost management using Universal Credits, even if they're not connected to OCI, or air-gapped.

  Systems not linked to OCI operate in standalone mode. They are not associated with a home OCI tenancy, and all infrastructure and resource management is performed locally.
- **Connection Mode**

  A system linked to a home OCI tenancy can be configured to operate in connected or disconnected mode. The connection mode refers to the presence of an active network connection between OCI and the system, which must be enabled or disabled during initial setup.
- **IAM Data Source**

  Identity and access management (IAM) happens either locally on the system or within the home OCI tenancy. On systems using OCI as the IAM data source, the configuration is synchronized with OCI at regular intervals.

Completing the Initial Setup and Configuration from the Graphical User Interface

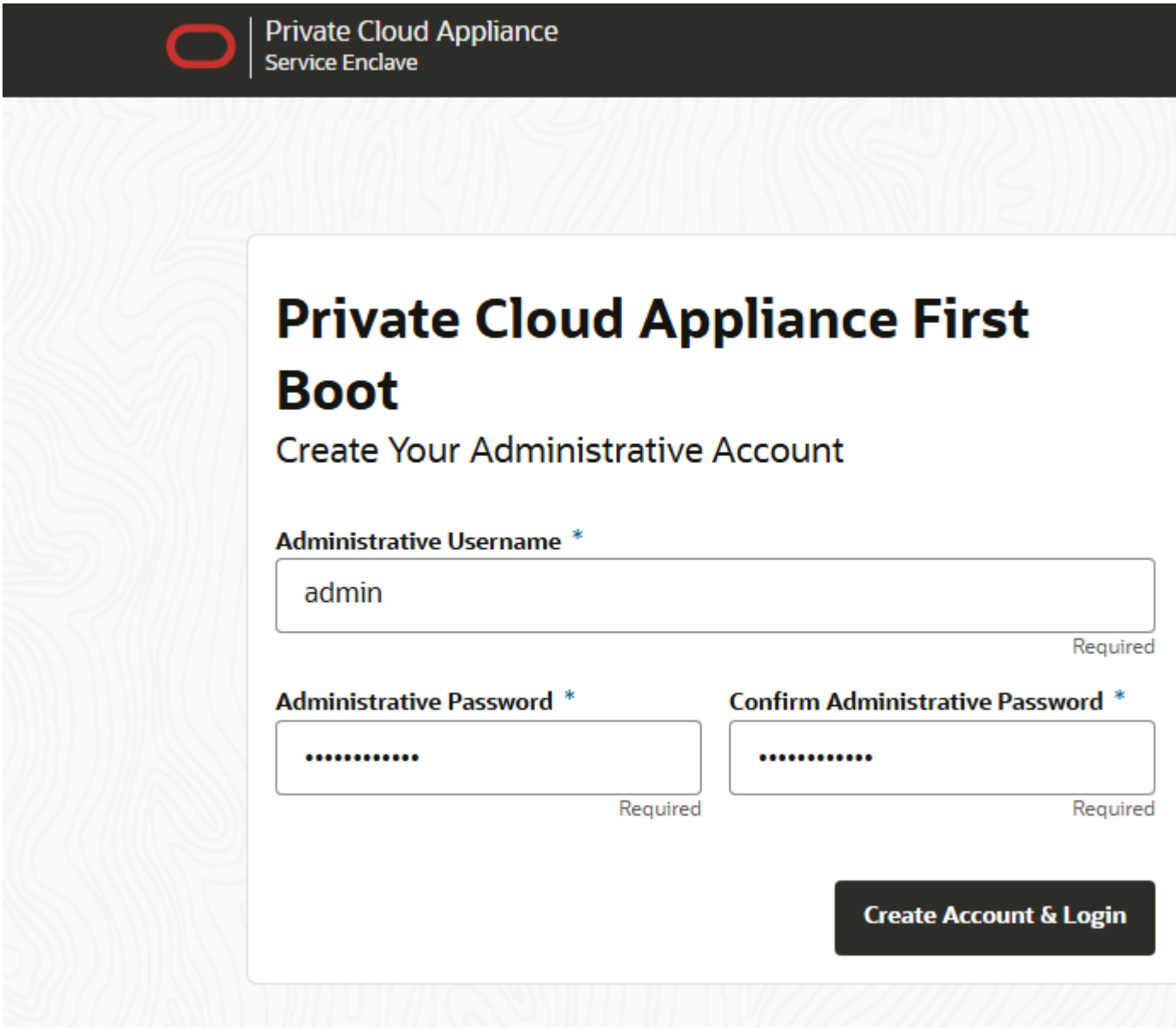
Using the GUI during first boot is the preferred way to perform the Private Cloud Appliance initial setup and configuration.

This entire procedure is best performed as one workflow. It's broken down into sections to improve navigation and clarity.

Setting the System Operating Mode

Connect to the Private Cloud Appliance for the first time to create a primary administration account, unlock the system, and set the parameters that determine the operating mode. Private Cloud Appliance operates disconnected from OCI.

1. Log in to the workstation you connected to the appliance, and browse to `https://100.96.2.32:30099`.
2. From the First Boot page, create the primary administrative account for your appliance, which is used for initial configuration and will persist after the first boot process. Additional accounts can be added later.



a. Enter an Administrative Username.

b. Enter and confirm the Administrative Password.

> **Note:**
>
> Passwords must contain a minimum of 12 characters with at least one of each: uppercase character, lowercase character, digit, and any punctuation character – except for double quote ("") characters, which are not allowed.

c. Click Create Account & Login.

> **Caution:**
>
> At the Service Enclave Sign In page, **Do not sign in and do not refresh your browser**.

3. Unlock the system. This step must be performed from the Service CLI.

a. Open a terminal window and log in to one of the management nodes using the primary administrative account.

> **Note:**
>
> Management nodes are named `pcamn01`, `pcamn02` and `pcamn03` by default.

```
$ ssh admin@pcamn01 -p 30006
Password authentication
Password:
PCA-ADMIN>
```

b. Enter the `systemStateunlock` command.

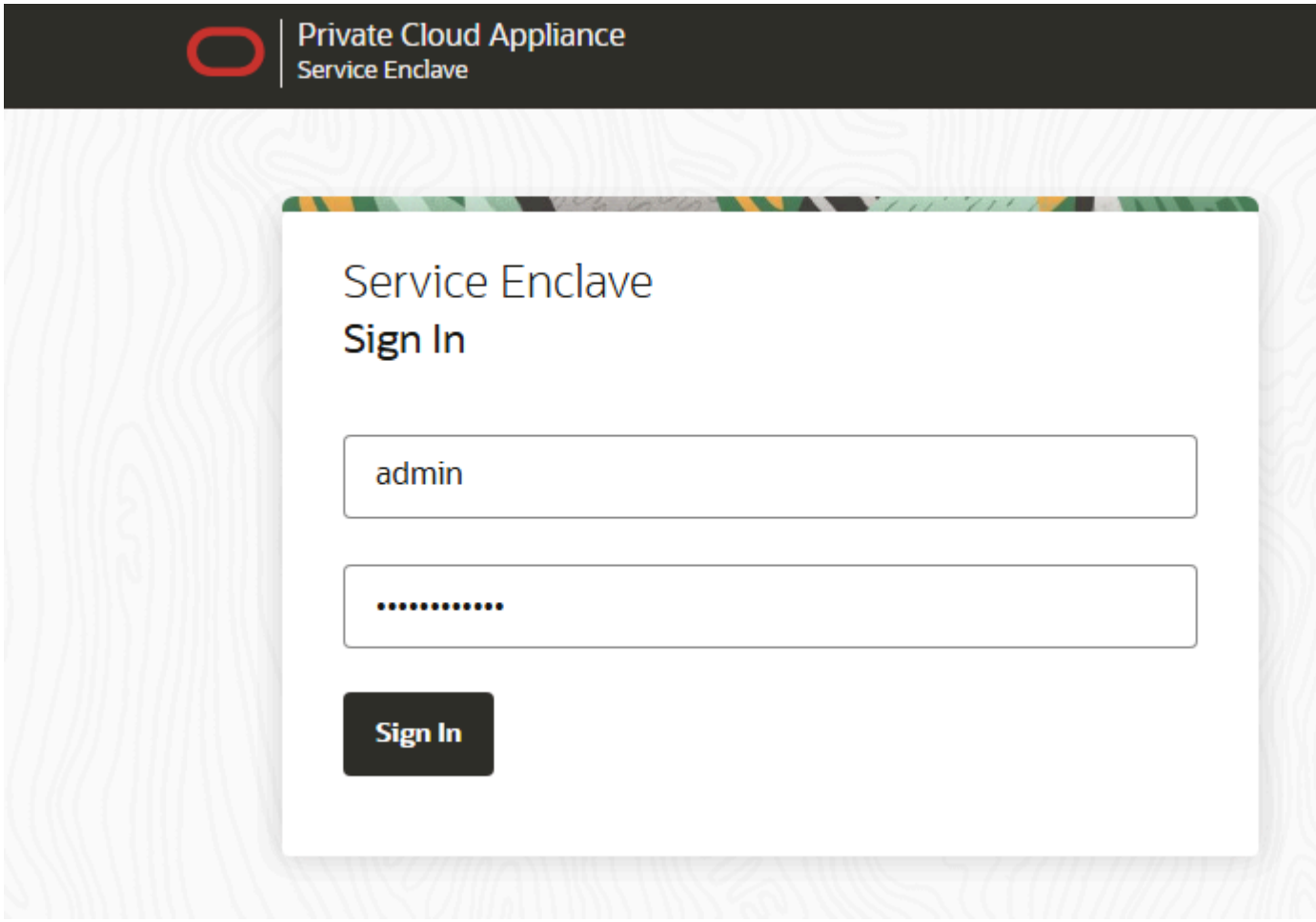
```
PCA-ADMIN> systemStateunlock
```

c. Verify the system is unlocked.

```
PCA-ADMIN> show pcaSystem
Data:
  Id = 5709f72b-c439-4c3a-8959-758df94eff25
  Type = PcaSystem
  System Config State = Config System Params
  system state locked = false
```

d. Enter the `exit` command or close the terminal window.

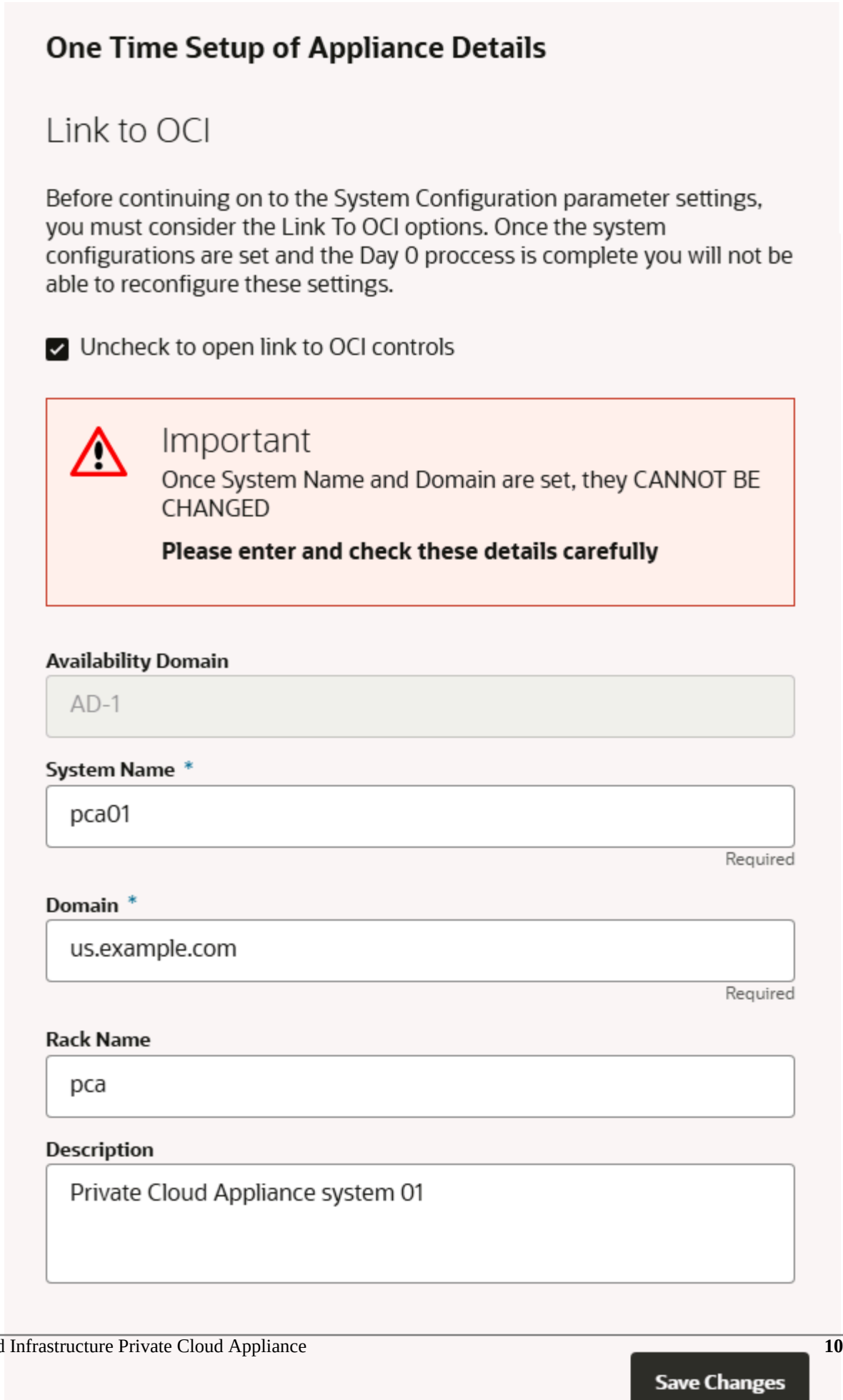
4. Return to the browser window displaying the Service Enclave Sign In page. Use the primary administrative account to sign in.



> **Note:**
>
> You might need to accept the self-signed SSL certificate again before signing in.

5. In the One Time Setup page, configure the operating mode and related system parameters.
   a. Private Cloud Appliance must not be connected to OCI.

      Keep the checkbox under Link to OCI selected. The controls to link the system to OCI must not be used.


One Time Setup of Appliance Details
Link to OCI
Before continuing on to the System Configuration parameter settings, you must consider the Link To OCI options. Once the system configurations are set and the Day 0 proccess is complete you will not be able to reconfigure these settings.
Uncheck to open link to OCI controls
Important
Once System Name and Domain are set, they CANNOT BE CHANGED
Please enter and check these details carefully
Availability Domain
AD-1
System Name *
pca01
Required
Domain *
us.example.com
Required
Rack Name
pca
Description
Private Cloud Appliance system 01
Save Changes

b. Provide the following appliance details.

The Availability Domain cannot be modified. Required entries are marked with an asterisk.

- System Name*
- Domain*
- Rack Name
- Description

> **Note:**
>
> The system name and domain fields support only lowercase characters. If you enter uppercase characters in these fields, they are converted to lowercase when changes are saved.

6. **Confirm the parameters you entered are correct. After this step they can no longer be changed.** Click Save Changes when you are ready to proceed.

   Keep the browser window open. Proceed to the next section of this initial setup procedure.

Configuring the System Network

When the operating mode and base system parameters are locked in, you must configure the system network. The Network Configuration wizard guides you through the sets of parameters you must provide. Refer to the information you gathered in the Initial System Installation Checklist on page 74 to complete the system configuration. It's helpful to enter all this information in a text file.

> **Important:**
>
> Enter the IP addresses for the uplinks exactly as they appear in the network configuration spreadsheet you filled out in preparation. The order of entry is important, especially when multiple IPs are added in the same field, because they map to specific spine and data center switches in the uplink topology.

1. Return to the browser window you left open at the end of the preceding section. Refresh the page and sign in to the system with the primary administrative account.

   > **Note:**
   >
   > You might need to accept the self-signed SSL certificate again before signing in.

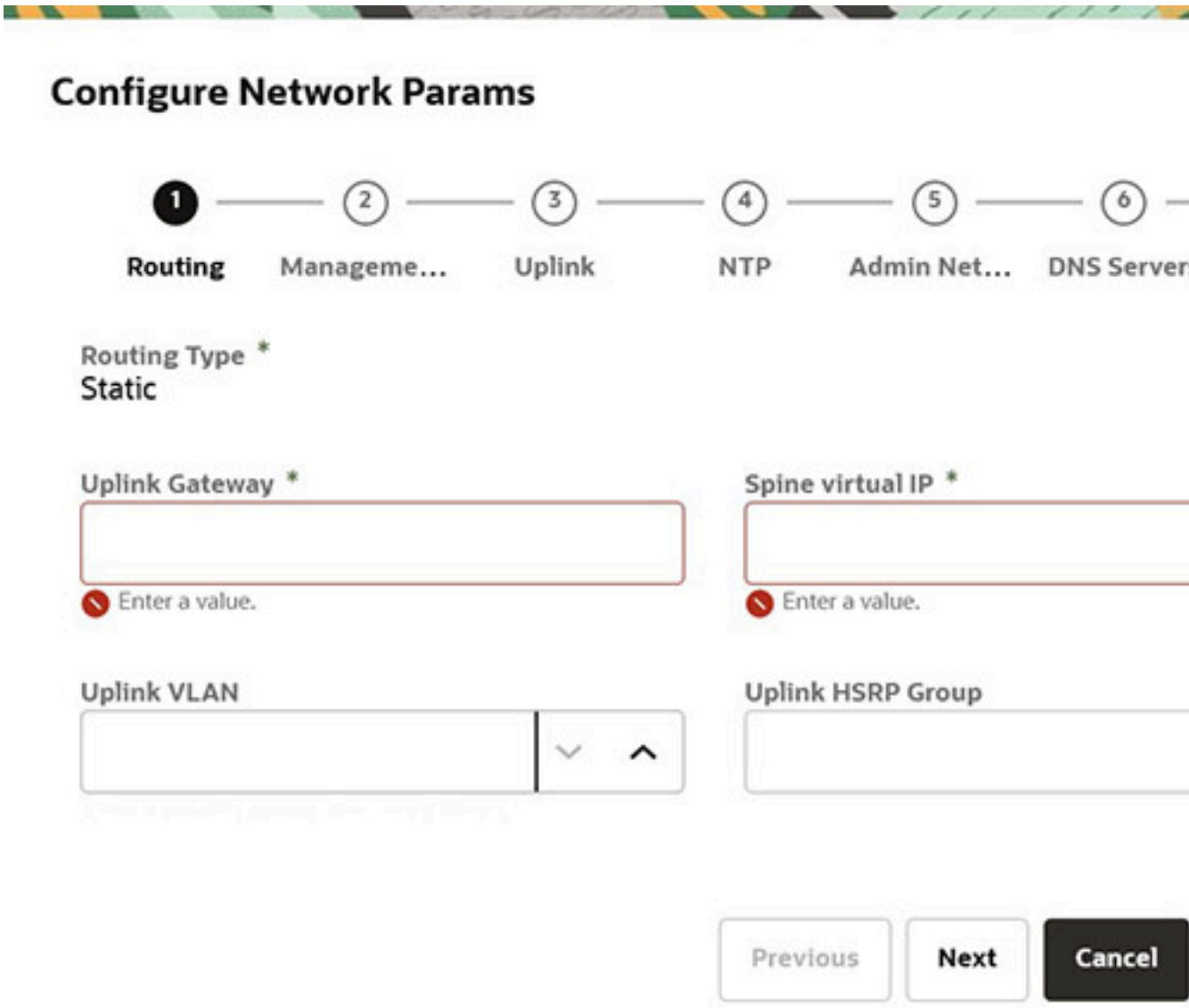
2. In the Configure Network Params wizard, select the routing design for the logical connection between Private Cloud Appliance and the data center network.

   The options are (A) static routing or (B) dynamic routing. The required parameters are different depending on the routing design you selected.

   Option A: Static Routing

   Enter the following data center information:

   - Routing Type: Static*
   - Uplink gateway IP Address*
   - Spine virtual IP* (comma-separated values if using the 4 port dynamic mesh topology)
   - Uplink VLAN
   - Uplink HSRP Group



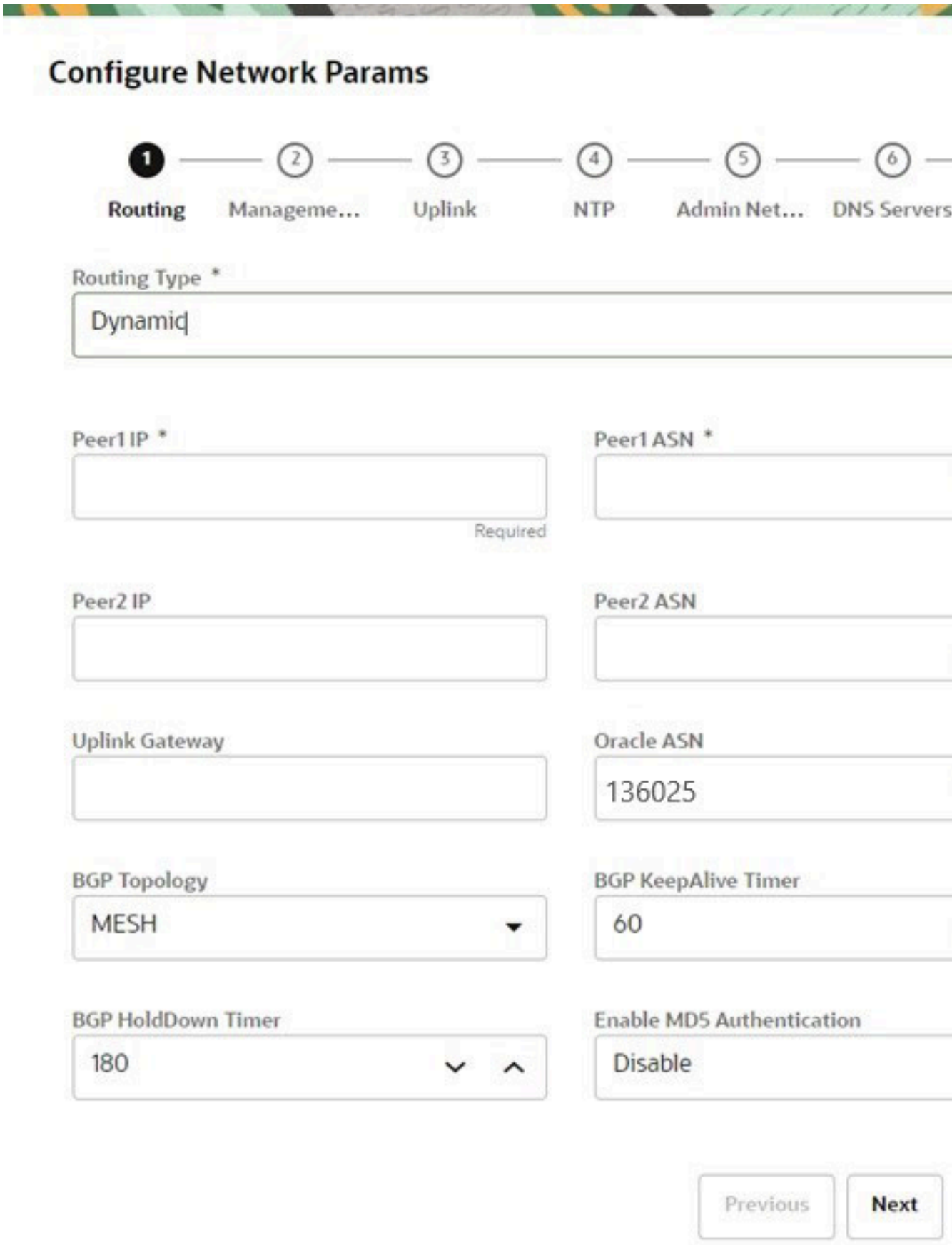
Option B: Dynamic Routing

Enter the following data center information:

- Routing Type: Dynamic*
- Peer1 IP and ASN*
- Peer2 IP and ASN
- Uplink Gateway
- Oracle ASN

> **Note:**
>
> The default Oracle ASN is 136025. The default BGP ASN inbound path filtering does not accept your own ASN in the AS path. If there is more than one Private Cloud Appliance site, then customers must adjust the Oracle ASN field to be unique in their data center network.

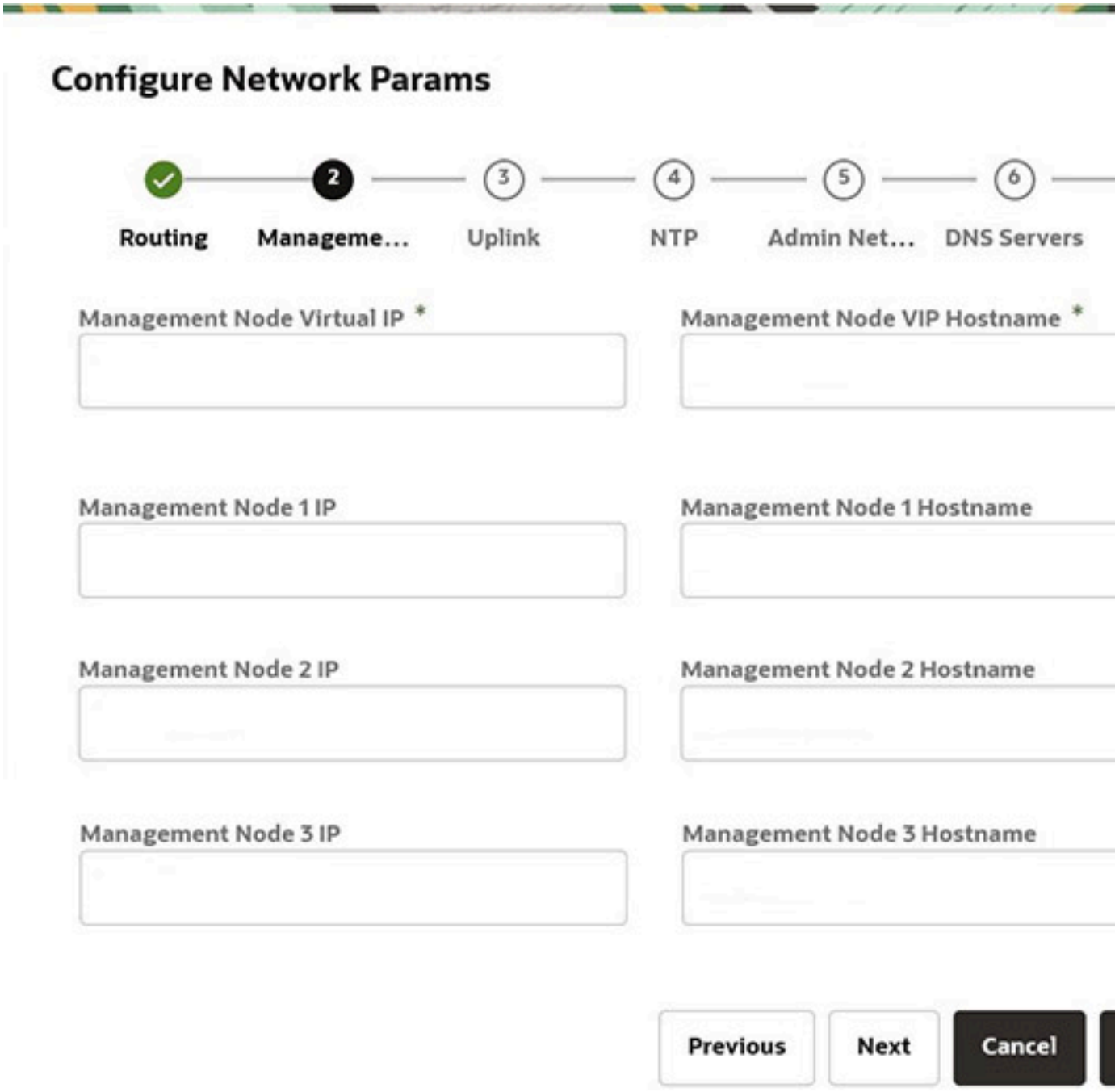
- BGP Topology (square, mesh, triangle), KeepAlive Timer and HoldDown Timer
- MD5 Authentication: enable or disable


Configure Network Params
1 Routing
2 Manageme...
3 Uplink
4 NTP
5 Admin Net...
6 DNS Servers
Routing Type *
Dynamiq
Peer1 IP *
Required
Peer1 ASN *
Peer2 IP
Peer2 ASN
Uplink Gateway
Oracle ASN
136025
BGP Topology
MESH
BGP KeepAlive Timer
60
BGP HoldDown Timer
180
Enable MD5 Authentication
Disable
Previous
Next

> **Note:**
>
> Communication between VCNs on different DRGs within a Private Cloud Appliance environment is possible if route entries and firewall access are provided on the data center network that connects the two VCNs.

3. Click Next to proceed to the wizard page for management node network parameters.

   Enter a shared virtual IP and associated host name for the management node cluster. Add an IP address and host name for each of the three individual management nodes.

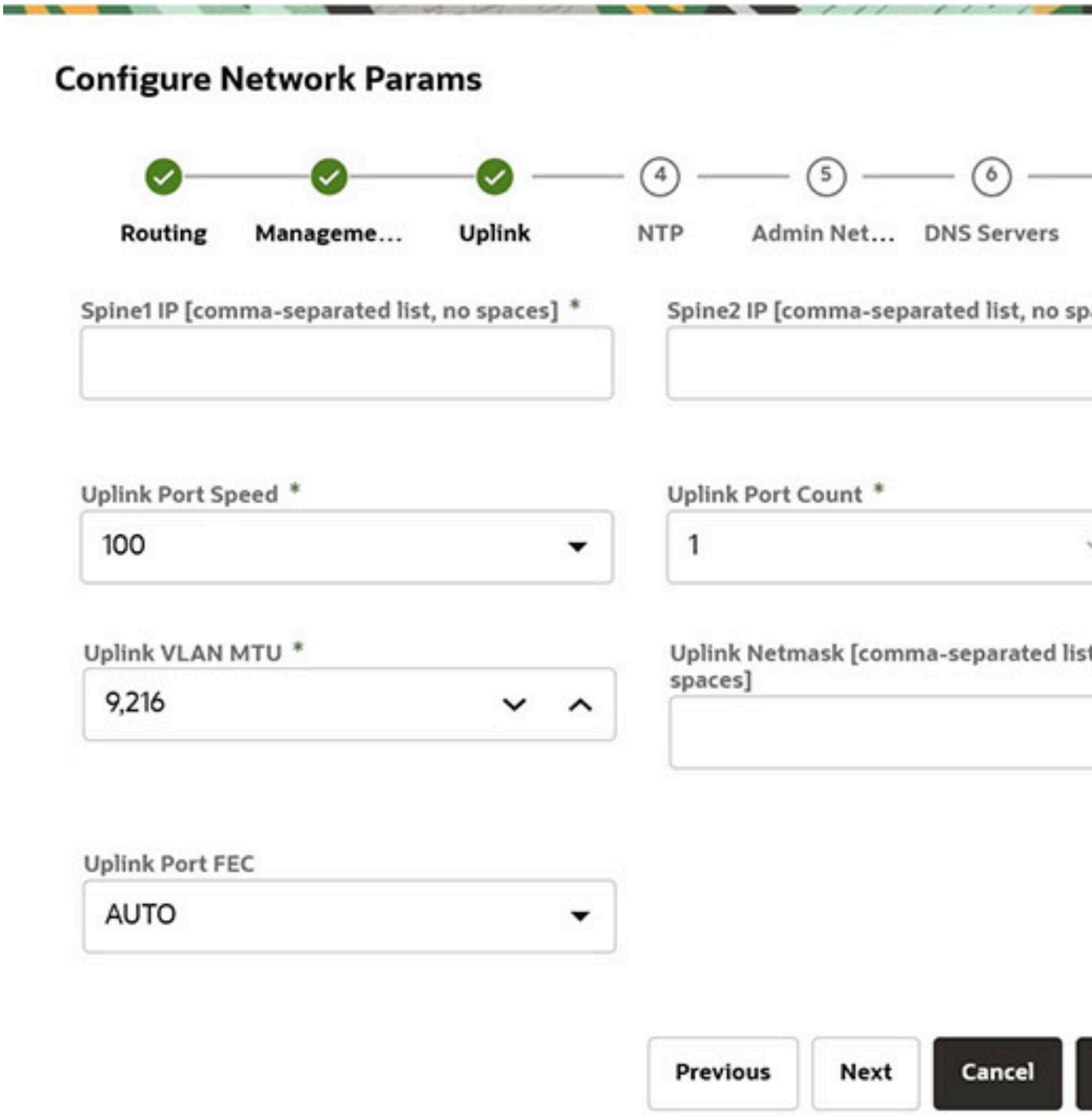


4. Click Next to proceed to the wizard page for uplink parameters.

Enter the following data center uplink information:

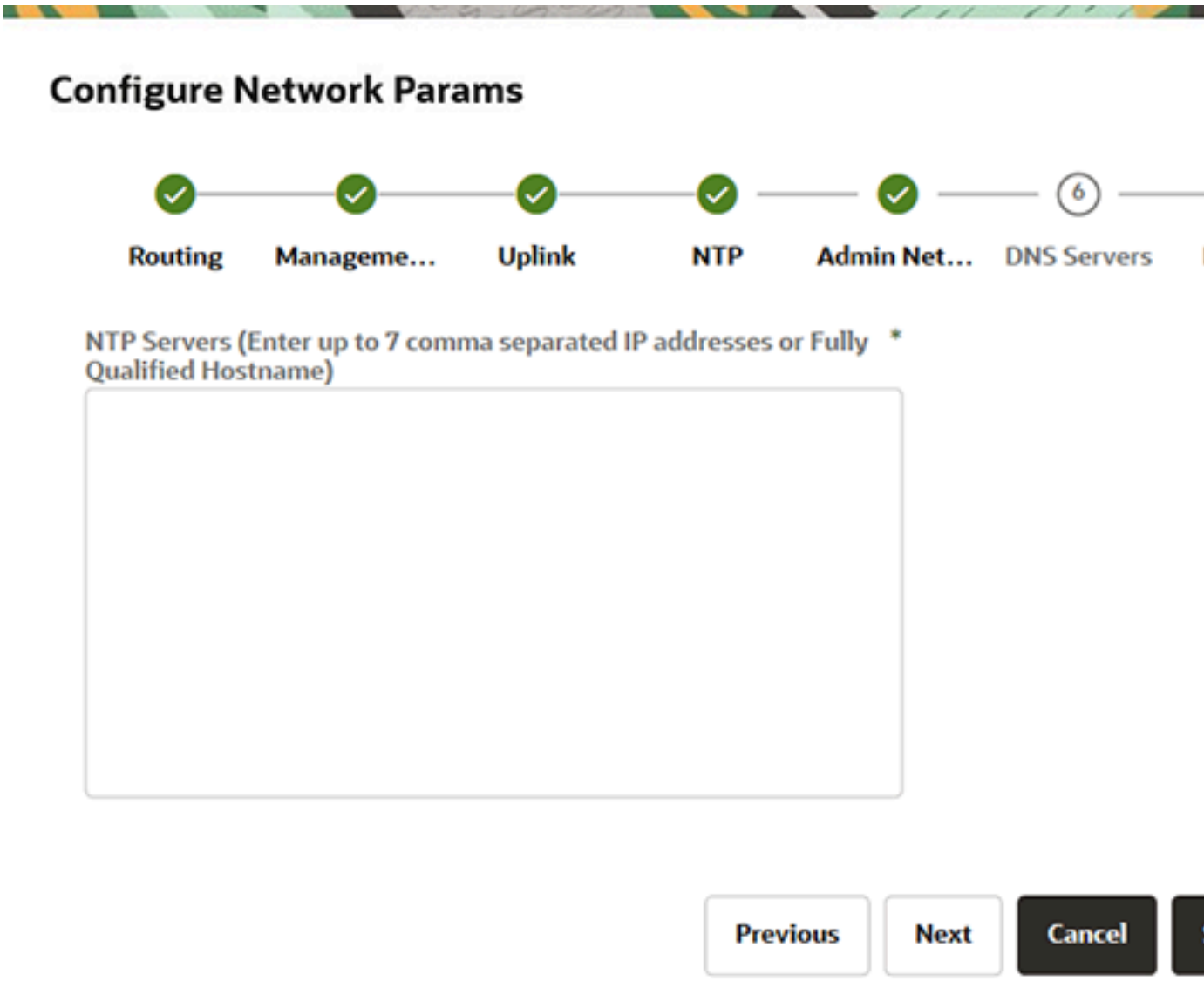
- IP Address for Spine Switch 1 and 2*
- Uplink Port Speed and Port Count*
- Uplink VLAN MTU and Netmask*
- Uplink Port FEC

> **Note:**
>
> Uplinks ports are always configured as port channels, even when only one uplink port count is configured. Port channel is configured with LACP mode with LACP rate as fast.



5. Click Next to proceed to the wizard page for NTP parameters.

   Enter the IP address or fully qualified domain name of at least one NTP server. Use a comma separated list to specify multiple NTP servers.



Note:

When the network configuration has been applied, you can check the NTP connection status.

In the Service Web UI, it's displayed in the NTP tab of the Network Environment Information page. In the Service CLI you can run this command:

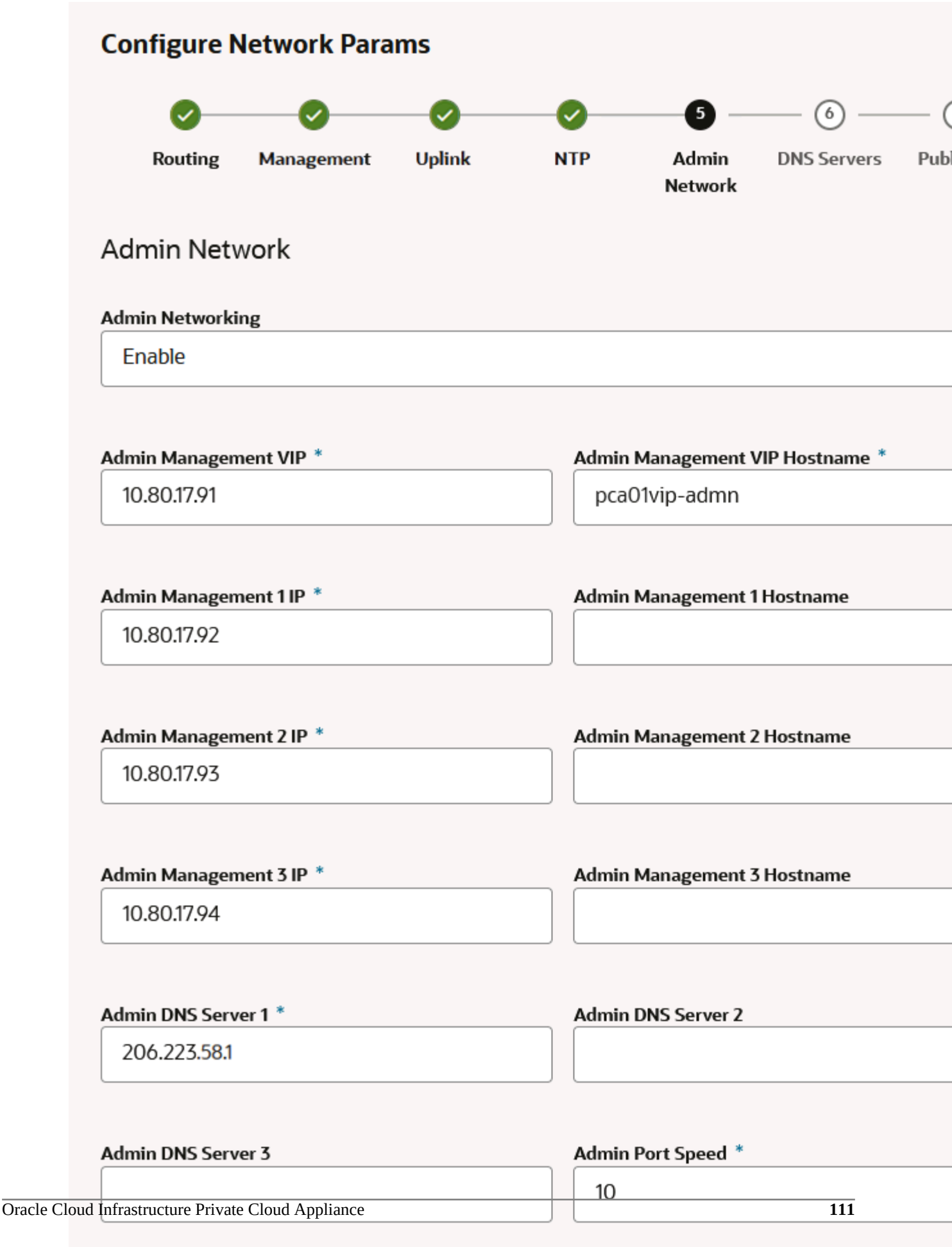
```
PCA-ADMIN> checkNtpServers
Data:
  id              Accessible
  --              ----------
  10.64.0.252     true
  192.0.2.2       true
```

6. Click Next to proceed to the wizard page for Admin Network parameters.

   (Optional) If you don't intend to configure a segregated appliance administration network, leave it disabled and skip this step.

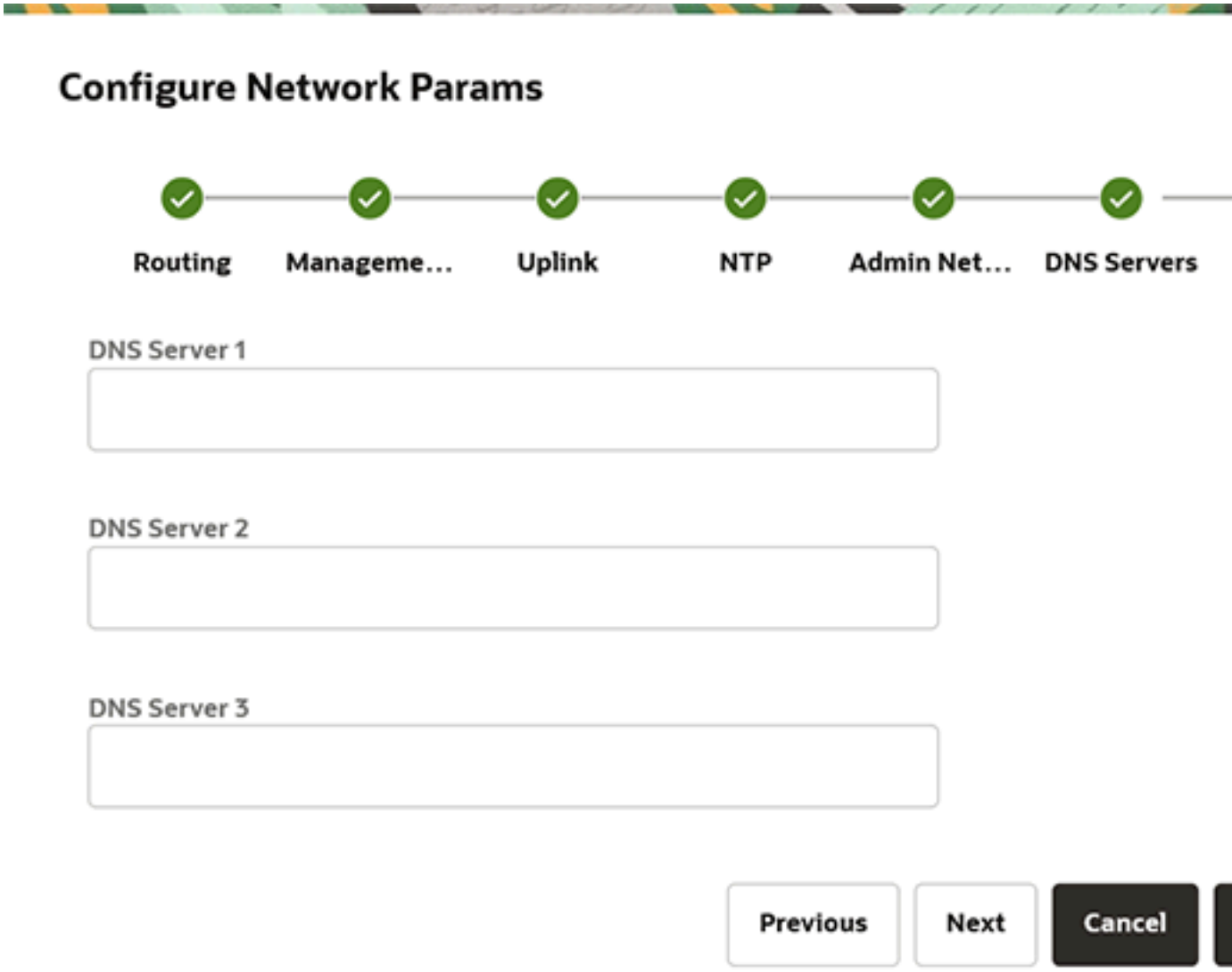
If you elected to segregate administrative appliance access from the data traffic, configure the administration network by entering the following mandatory and optional information.

- Admin Networking: Enable
- Admin Management VIP address and Hostname
- Admin Management 1 and 2 and 3 IP address and Hostname
- At least 1, but up to 3, Admin DNS Server IP addresses
- Admin Port Speed, Port Count, and Admin Router Group
- Admin VLAN, MTU, Port FEC, and Gateway IP
- Admin IP Address for Spine Switch 1 and 2, and a shared Virtual IP
- Admin Peer1 and Peer2 IP address and ASN for BGP
- Admin Topology (Triangle, Square, Mesh)
- Admin BGP Authentication (Enable/Disable). Keepalive Timer, Hold Down Timer, and Static Routing


Configure Network Params
Routing
Management
Uplink
NTP
5
Admin Network
6
DNS Servers
Admin Network
Admin Networking
Enable
Admin Management VIP *
10.80.17.91
Admin Management VIP Hostname *
pca01vip-admn
Admin Management 1 IP *
10.80.17.92
Admin Management 1 Hostname
Admin Management 2 IP *
10.80.17.93
Admin Management 2 Hostname
Admin Management 3 IP *
10.80.17.94
Admin Management 3 Hostname
Admin DNS Server 1 *
206.223.58.1
Admin DNS Server 2
Admin DNS Server 3
Admin Port Speed *
10

7. Click Next to proceed to the wizard page for DNS parameters.

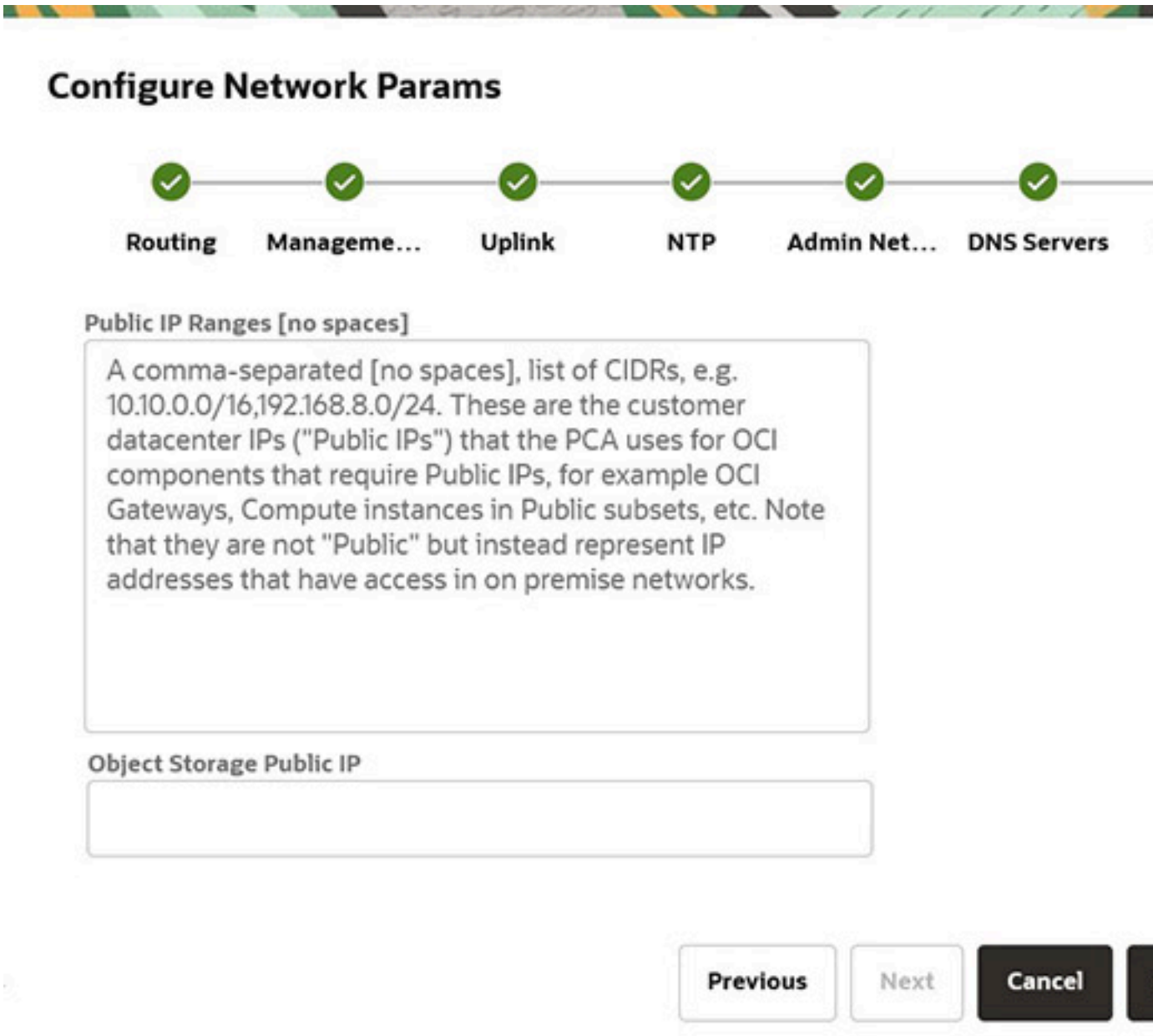
   Enter at least one and up to three DNS servers in the respective fields.



8. Click Next to proceed to the wizard page for public IP addresses.

   Enter the data center IP addresses that the system controller software can assign to resources as public IPs.

   - Public IP list of CIDRs in a comma-separated list
   - Object Storage Public IP (must be outside the public IP range)



9. Keep the browser window open. Proceed to the next section of this initial setup procedure.

Verifying and Applying the Network Configuration

All parameters in all pages of the Network Configuration wizard have been entered. However, you must still verify and apply the configuration.

1. Verify the parameters you entered in each of the wizard. Use the Previous/Next buttons to scroll through the pages.

   The network configuration information is persisted when a test procedure is completed successfully. Testing starts when you save the network configuration in the next step. If you need to change any parameters after testing begins, you must reenter all information.
2. Save the network configuration parameters by clicking Save Changes.

   A network configuration test begins, and can take up to 15 minutes. **Do not** close the browser window before it's completed.

If a problem is encountered, the Configure Network Params wizard reopens and the error is displayed.

3. In the Testing Network Parameters page, select the appropriate option:

   - Click Reenter Network Configuration.

     You are returned to a blank Configure Network Params wizard where you must enter all your information again.
   - Click Commit Changes.

     The network parameters are locked. When locked, the routing type and public IPs cannot be changed.

     > **Caution:**
     >
     > When you click Commit Changes, system initialization begins and can take up to 15 minutes. **Do not** close the browser window during this time.

   If a problem is encountered, the Configure Network Params wizard reopens and the error is displayed. Otherwise, a Configuration Complete message is displayed.
4. Click Sign Out. You are returned to the Service Enclave sign in page.
5. Verify the system configuration.

   a. Connect to the Service Web UI at the new virtual IP address of the management node cluster: `https://<virtual_ip>:30099`.
   b. From the Dashboard, click Appliance to view the system details and click Network Environment to view the network configuration.

      > **Note:**
      >
      > Alternatively, you can log in to the Service CLI as an administrator and run the following commands to confirm your entries.

      ```
      # ssh 100.96.2.32 -l admin -p 30006

      PCA-ADMIN> show pcaSystem
      PCA-ADMIN> show networkConfig
      ```

Completing the Initial Setup and Configuration from the Service CLI

If necessary, you can perform the Private Cloud Appliance initial setup and configuration from the Service CLI. However, using the GUI is the preferred method.

This entire procedure is best performed as one workflow. It's broken down into sections to improve navigation and clarity.

Setting the System Operating Mode

Connect to the Private Cloud Appliance for the first time to create a primary administration account, unlock the system, and set the parameters that determine the operating mode. Private Cloud Appliance operates disconnected from OCI.

1. Log in to the workstation you connected to the appliance, and sign in to the Private Cloud Appliance management node cluster for initial configuration.

   The management cluster has a predefined virtual IP: `100.96.2.32`. When prompted for a password, press enter.

   ```
   # ssh 100.96.2.32 -l "" -p 30006
   Password authentication
   Password:
   ```
2. Confirm initial user sign-in, where `System Config State = Config User`.

   ```
   PCA-ADMIN> show pcaSystem
   Data:
   ```

```
  Id = o780c522-fkl5-43b1-8g30-eea90263f2e9
  Type = PcaSystem
  System Config State = Config User
```

3. Create the primary administrative account for the appliance.

   a. Provide a username and password.

   > **Note:**
   >
   > Passwords must contain a minimum of 12 characters with at least one of each: uppercase character, lowercase character, digit, and any punctuation character – except for double quote ("") characters, which are not allowed.

   ```
   PCA-ADMIN> createadminaccount name=admin password=password
    confirmpassword=password
   Status: Success
   JobId: 302a6h99-fh7y-41sd-8i30-ea28581dcw9e
   ```

   b. Enter the `exit` command to log out of the CLI.

4. Unlock the system.

   a. Open a terminal window and log in to one of the management nodes using the primary administrative account.

   > **Note:**
   >
   > Management nodes are named `pcamn01`, `pcamn02` and `pcamn03` by default.

   ```
   $ ssh admin@pcamn01 -p 30006
   Password authentication
   Password:
   PCA-ADMIN>
   ```

   b. Enter the `systemStateunlock` command.

   ```
   PCA-ADMIN> systemStateunlock
   ```

   c. Verify the system is unlocked.

   ```
   PCA-ADMIN> show pcaSystem
   Data:
     Id = 5709f72b-c439-4c3a-8959-758df94eff25
     Type = PcaSystem
     System Config State = Config System Params
     system state locked = false
   ```

   d. Enter the `exit` command to log out of the CLI.

5. Use the primary administrative account to sign back in.

   ```
   # ssh admin@100.96.2.32 -p 30006
   Password authentication
   Password:
   PCA-ADMIN>
   ```

6. Confirm the system is ready for configuration, where `System Config State = Config System Params`.

   ```
   PCA-ADMIN> show pcaSystem
   Data:
     Id = o780c522-fkl5-43b1-8g30-eea90263f2e9
     Type = PcaSystem
     Product Type = PCA
   ```

```
  System Mode = Standalone
  Connected State = false
  System Config State = Config System Params
[...]
```

7. Configure the operating mode and related system parameters.

   a. Private Cloud Appliance must not be connected to OCI. Do not run the `linkToOCI` command. These operating mode parameters will be persisted:

   ```
   System Mode = Standalone
   Connected State = false
   IAM Data Source = local
   ```

   b. Configure the system name and domain name, then verify the settings.

   ```
   PCA-ADMIN> setDay0SystemParameters systemName=pca01
    domainName=us.example.com
   Status: Success
   JobId: 2e5c08cc-3fe3-4631-ab86-8c60e30468ad

   PCA-ADMIN> show pcasystem
   Data:
     Id = o780c522-fkl5-43b1-8g30-eea90263f2e9
     Type = PcaSystem
   [...]
     System Name = pca01
     Domain Name = us.example.com
     Availability Domain = AD-1
   ```

   c. Keep the CLI connection open. Proceed to the next section of this initial setup procedure.

Configuring the System Network

When the operating mode and base system parameters are locked in, you must configure the system network. Refer to the information you gathered in the Initial System Installation Checklist on page 74 to complete the system configuration. It's helpful to enter all this information in a text file.

> **Important:**
>
> Enter the IP addresses for the uplinks exactly as they appear in the network configuration spreadsheet you filled out in preparation. The order of entry is important, especially when multiple IPs are added in the same field, because they map to specific spine and data center switches in the uplink topology.

1. Configure the network parameters according to the selected routing design for the logical connection between Private Cloud Appliance and the data center network.

   The options are (A) static routing or (B) dynamic routing. The required parameters are different depending on the routing design you selected.

   > **Note:**
   >
   > Uplinks ports are always configured as port channels, even when only one uplink port count is configured. Port channel is configured with LACP mode with LACP rate as fast.

   > **Attention:**
   >
   > If you elected to segregate administrative appliance access from the data traffic, include the admin network parameters in the system network configuration command. This applies to both static and dynamic routing, which are shown in the steps that follow.

When the Admin Network is enabled, some services will be accessed through the Admin Management VIP instead of the Management VIP. Any address record for those services should reference the Admin Network IP instead of the standard Network IP. The list of services is:

- 'admin'
- 'adminconsole'
- 'prometheus-gw'
- 'prometheus'
- 'grafana'
- 'api'
- 'alertmanager'
- 'rps'

2. For dynamic routing, ignore this step and proceed to the next step.

   If you selected **Option A: Static Routing**, enter the following parameters on a single line.

```
PCA-ADMIN> setDay0StaticRoutingParameters
uplinkPortCount=2
uplinkPortSpeed=40
uplinkMtu=9216

mgmtVipHostname=name
mgmtVip=10.nn.nn.22
ntpIps=10.nn.nn.1,10.nn.nn.105,nn.nn.17.1
spine1Ip=10.nn.nn.18
spine2Ip=10.nn.nn.19
spineVip=10.nn.nn.20
uplinkNetmask=255.255.255.248
uplinkGateway=10.nn.nn.1
uplinkVlan=678
uplinkRouterGroup=116
objectStorageIp=10.nn.nn.241
mgmt01Ip=10.nn.nn.7
mgmt02Ip=10.nn.nn.8
mgmt03Ip=10.nn.nn.9
mgmt01Hostname=mn1
mgmt02Hostname=mn2
mgmt03Hostname=mn3
dnsIp1=10.1nn.nn.200
dnsIp2=206.nn.nn.1
dnsIp3=206.nn.nn.2
```

Note:

After static routing parameters are configured, monitor the process using the `show networkConfig` command. When the process is complete, the `Network Config Lifecycle State` is `Creating` rather than `Active`. The `Network Config Lifecycle State` isn't `Active` until the `lockDay0NetworkParameters` command is issued.

Example: square topology with static routing

This example shows a square topology with ECMP static routing. For more information about this configuration, see Square Topology with ECMP Static Routing on page 41.

```
setDay0StaticRoutingParameters
uplinkPortCount=1 uplinkPortSpeed=100
uplinkMtu=9216
```

```
uplinkNetmask=255.255.255.252
peer1Ip=10.nn.nn.34
peer2Ip=10.nn.nn.38
uplinkTopology=square
mgmtVipHostname=pca1vip
mgmtVip=10.nn.nn.43
ntpIps=10.nn.nn.105
spine1Ip=10.nn.nn.33
spine2Ip=10.nn.nn.37
objectStorageIp=10.nn.nn.49
mgmt01Ip=10.nn.nn.44
mgmt02Ip=10.nn.nn.45
mgmt03Ip=10.nn.nn.46
mgmt01Hostname=pac1mn1
mgmt02Hostname=pca1mn2
mgmt03Hostname=pca1mn3
dnsIp1=206.nn.nn.1
dnsIp2=206.nn.nn.2
```

Example: mesh topology with static routing

This example shows a mesh topology with ECMP static routing. For more information about this configuration, see Mesh Topology with ECMP Static Routing on page 34.

```
setDay0StaticRoutingParameters
uplinkPortCount=4 uplinkPortSpeed=100
uplinkMtu=9216

uplinkNetmask=255.255.255.254,255.255.255.254
peer1Ip=10.nn.nn.96,10.nn.nn.98
peer2Ip=10.nn.nn.100,10.nn.nn.102
uplinkTopology=mesh
mgmtVipHostname=pca2vip
mgmtVip=10.nn.nn.107
ntpIps=10.nn.nn.105
spine1Ip=10.nn.nn.97,10.nn.nn.103
spine2Ip=10.nn.nn.101,10.nn.nn.99
objectStorageIp=10.nn.nn.113
mgmt01Ip=10.nn.nn.108
mgmt02Ip=10.nn.nn.109
mgmt03Ip=10.nn.nn.110
mgmt01Hostname=pca2mn1
mgmt02Hostname=pca2mn2
mgmt03Hostname=pca2mn3
dnsIp1=206.nn.nn.1
dnsIp2=206.nn.nn.2
```

3. If you configured static routing, ignore this step and proceed to the next step.

   If you selected **Option B: Dynamic Routing**, enter the following parameters on a single line.

```
PCA-ADMIN> setDay0DynamicRoutingParameters
uplinkPortSpeed=100
uplinkPortCount=2
uplinkMtu=9216

spine1Ip=10.nn.nn.17
spine2Ip=10.nn.nn.21
uplinkNetmask=255.255.255.252
mgmtVipHostname=example-vip
mgmtVip=10.nn.nn.8
ntpIps=10.nn.nn.1
peer1Asn=50000
peer1Ip=10.nn.nn.18
```

```
peer2ASN=50000
peer2Ip=10.nn.nn.22
objectStorageIp=10.nn.nn.1
mgmt01Ip=10.nn.nn.nn
mgmt02Ip=10.nn.nn.nn
mgmt03Ip=10.nn.nn.nn
dnsIp1=10.1nn.nn.200
bgpTopology=topology-type
BGPAuthentication=true
BGPAuthenticationPassword=<bgp-password>
adminBGPAuthentication=true
adminBGPAuthenticationPassword=<admin-bgp-password>
```

> **Note:**
>
> BGP authentication isn't enabled if you don't specify a password. Because BGP often involves separate administrative domains, password coordination is necessary between those responsible for both ends of the BGP links.

The `adminBGPpassword` must be established and changed on both ends of the BGP links at the same time. This might require careful coordination between different administrators. If one BGP authentication password is changed and the other isn't, the link fails.

To verify success in BGP operation, run the command `show bgp sessions`.

Example: mesh topology with dynamic routing and admin network

The following example shows a dynamic mesh topology configuration using BGP authentication on the uplinks, where a separate administration network is set up:

```
setDay0DynamicRoutingParameters
uplinkPortSpeed=40 uplinkPortCount=4
uplinkMtu=9216

spine1Ip=10.nn.nn.29,10.nn.nn.35
spine2Ip=10.nn.nn.33,10.nn.nn.31
uplinkNetmask=255.255.255.252,255.255.255.252
mgmtVipHostname=example_vip
mgmtVip=10.nn.nn.46
ntpIps=10.nn.nn.1,10.nn.nn.2
peer1Asn=50000
peer1Ip=10.nn.nn.30,10.nn.nn.32
peer2Asn=50000
peer2Ip=10.nn.nn.34,10.nn.nn.36
objectStorageIp=10.nn.nn.72 mgmt01Ip=10.nn.nn.44
mgmt02Ip=10.nn.nn.45
mgmt03Ip=10.nn.nn.46
dnsIp1=10.1nn.nn.200
bgpTopology=mesh
BGPAuthentication=true
BGPAuthenticationPassword=bgp-password
adminBGPAuthentication=true
adminBGPAuthenticationPassword=admin-bgp-password
```

4. Configure the management node IP addresses and host names. Add at least one and up to three DNS servers. The `dnsIp1` field is required. Enter the command parameters on a single line.

```
PCA-ADMIN> edit NetworkConfig \
mgmt01Ip=10.nn.nn.9 \
mgmt02Ip=10.nn.nn.10 \
mgmt03Ip=10.nn.nn.11 \
mgmt01Hostname=apac01-mn1 \
```

```
mgmt02Hostname=apac01-mn2 \
mgmt03Hostname=apac01-mn3 \
dnsIp1=206.nn.nn.1 \
dnsIp2=206.nn.nn.2 \
dnsIp3=10.nn.nn.197
```

5. Enter the data center IP addresses that the appliance can assign to resources as public IPs. Enter the data as a comma separated list on a single line.

```
edit NetworkConfig publicIps=10.nn.nn.2/31,10.nn.nn.4/30,10.nn.nn.8/29, \
10.nn.nn.16/28,10.nn.nn.32/27,10.nn.nn.64/26,10.nn.nn.128/26,10.nn.nn.192/27,
 \
10.nn.nn.224/28,10.nn.nn.240/29,10.nn.nn.248/30,10.nn.nn.252/31,10.nn.nn.254/32
```

6. Keep the CLI connection open. Proceed to the next section of this initial setup procedure.

Verifying and Applying the Network Configuration

All network configuration parameters have been entered. However, you must still verify and apply the configuration.

1. Verify the network parameters are configured. You can monitor the process using the `show NetworkConfig` command. When the process is complete, the `Network Config Lifecycyle State` changes to `ACTIVE`.

> **Note:**
>
> After static routing parameters are configured, the `Network Config Lifecycle State` is `Creating` rather than `Active`. The `Network Config Lifecycle State` isn't `Active` until the `lockDay0NetworkParameters` command is issued in the next step.

```
PCA-ADMIN> show networkConfig
Data:
    Uplink Port Speed = 40
    Uplink Port Count = 2
[...]
    Uplink Router Group = 116
    Network Config Lifecycle State = ACTIVE
```

When this process is complete, the `System Config State` changes from `Wait for Networking Service` to `Test for Network Params`.

```
PCA-ADMIN> show pcasystem
Data:
  Id = 1e79d401-4a4a-44d2-9e60-57ec223b5418
  Type = PcaSystem
  System Config State = Test for Network Params
[...]
```

2. Apply the configuration by locking the network parameters.

```
PCA-ADMIN> lockDay0NetworkParameters
```

System network configuration is now complete.

Connecting a Bastion Host to Private Cloud Appliance (Optional)

In addition to the public Ethernet connection, it is allowed to connect the Private Cloud Appliance management switch to a management or machine administration network at your installation site. From this network, a bastion host provides direct administrator access to the appliance components.

If you choose to use such an uplink, consider it as a long-term alternative to the temporary workstation connection described in Connecting a Workstation to the System on page 97. Configure the administration uplink *after* the initialization of the appliance, when the appliance network settings have been applied.

A connection to the appliance internal management network, either directly into the management switch or through an additional Ethernet switch in the data center, is **not required** to access the appliance management functionality of the Private Cloud Appliance user interfaces. The primary role of the appliance internal management network is to allow the controller software on the management nodes to interact with the compute nodes and other rack components. Connecting to this network from outside the appliance allows you to gain direct administrator access to each component, for example to control the ILOMs.

> **Caution:**
>
> Do not make any changes to anything on this network unless directed to do so by Oracle Support.

Bastion Host Configuration

Follow these guidelines when configuring a bastion host.

> **Caution:**
>
> Connect **port 2** on the management switch.

Ensure that the data center Ethernet switch used in this connection is configured to prevent DHCP leakage to the `100.96.0.0/22` subnet used by the Private Cloud Appliance. Do not connect to any network with any kind of broadcast services in addition to DHCP.

For the *bastion host*, which is the name used to describe the machine that is permanently connected to the data center administration network, use the IP address `100.96.3.254/23` and assign it statically to its network interface. Ensure there is no other machine on the same subnet using the same IP address and causing IP conflicts.

Both the ILOM and internal management network are configured on the same management switch. In order to communicate with both networks, you must configure the bastion host with two paths to the switch. You can choose one of two configuration options:

- Configure two IP addresses on the bastion host.

  For example, add `100.96.1.254/23` as a second IP address.

  ```
  # cat ifcfg-eth1
  NAME=eth1
  DEVICE=eth1
  BOOTPROTO=static
  ONBOOT=yes
  NM_CONTROLLED=no
  USERCTL=no
  DEFROUTE=no
  IPV6INIT=no
  IPADDR1=100.96.3.254
  PREFIX1=23
  IPADDR2=100.96.1.254
  PREFIX2=23
  ```
- Or, add a route to the existing networks.

  On the `100.96.0.0/23` network, if the bastion host is configured with the IP `100.96.3.254` for subnet `100.96.2.0/23`, add this route:

  ```
  ip route add 100.96.0.0/23 via 100.96.2.1 dev eth1
  ```

  and on the `100.96.2.0/23` network, if the bastion host is configured with the IP `100.96.1.254` for subnet 100.96.0.0/23, add this route:

  ```
  ip route add 100.96.2.0/23 via 100.96.0.1 dev eth1
  ```

Expanding Private Cloud Appliance with GPU Capacity

To enable GPU-accelerated workloads, Private Cloud Appliance can be expanded with server nodes that have GPUs installed.

X10-2c GPU nodes are delivered in an expansion rack, with 1 to 6 factory-installed nodes. More nodes can be installed after initial deployment. Up to two expansion racks can be connected to a base rack, for a maximum of 12 GPU nodes. For more information, see GPU Expansion Rack Configuration on page 21.

Installation Requirements

Site Preparation

If you have decided to expand your Private Cloud Appliance environment with GPU nodes, carefully plan ahead for the installation of the additional hardware. The X10-2c GPU expansion rack has the same external dimensions as the base rack, and contains the same type of hardware. Therefore, the base rack site requirements also apply for the expansion rack. They are described in detail in Preparing the Installation Site on page 55.

Rack Cabling

The cable connections between the base rack and the GPU expansion rack must not exceed 25 meters. Allocate a space for the expansion rack near the base rack, ensuring that the inter-rack cabling is within the specified maximum length when routed through the floor or ceiling. The required cable length must be specified with the order. The default length of cables included in the shipment is 10 meters.

High-Performance Storage

The GPU compute shapes are optimized for high speed and low latency. They use high-performance storage exclusively, meaning the system's ZFS Storage Appliance must provide a high performance storage pool consisting of one or more *performance* disk trays. In case no performance tray is present in the existing installation, one is added to the GPU expansion order. If the base rack has no rack units available to add the performance tray, it will be installed in a storage expansion rack. The high performance storage pool must be configured before the GPU expansion rack is activated.

Connecting the Expansion Rack to the Base Rack

When the X10-2c GPU expansion rack is in its allocated space, it must be connected to the base rack. The expansion rack leaf switches are cross-connected to the base rack spine switches to extend the data network into the expansion rack. Similarly, the expansion rack components are added to the internal management network through a cable connection between the management switches in the racks. The ports required for this setup have been reserved on all connected switches.

Connecting the Management Switches

A GPU expansion rack management switch must be connected to the base rack management switch using two cable links with 25Gbps SFP28 transceivers.

- On the side of the GPU expansion rack, use management switch ports 49 and 50.
- Connect the cables from the first GPU expansion rack to base rack management switch ports 49 and 50. Connect the cables from the second GPU expansion rack to base rack management switch ports 51 and 52.

Connecting the Leaf Switches

Each GPU expansion rack leaf switch has redundant cross-connections to the two spine switches of the base rack. There are 8 total cable links between a GPU expansion rack and the base rack, using 100Gbps QSFP28 transceivers.

- On the side of the GPU expansion rack, use ports 33-36 in each leaf switch.
- Connect the cables from the first GPU expansion rack to the base rack spine switches using this pattern:
  - Leaf1 (RU18/pcaswlf03) ports 33 and 34 to Spine1 (RU29/pcaswsp01) ports 11 and 12

    - Leaf1 (RU18/pcaswlf03) ports 35 and 36 to Spine2 (RU30/pcaswsp02) ports 11 and 12
    - Leaf2 (RU19/pcaswlf04) ports 33 and 34 to Spine1 (RU29/pcaswsp01) ports 13 and 14
    - Leaf2 (RU19/pcaswlf04) ports 35 and 36 to Spine2 (RU30/pcaswsp02) ports 13 and 14
- Connect the cables from the second GPU expansion rack to the base rack spine switches using this pattern:
    - Leaf1 (RU18/pcaswlf05) ports 33 and 34 to Spine1 (RU29/pcaswsp01) ports 15 and 16
    - Leaf1 (RU18/pcaswlf05) ports 35 and 36 to Spine2 (RU30/pcaswsp02) ports 15 and 16
    - Leaf2 (RU19/pcaswlf06) ports 33 and 34 to Spine1 (RU29/pcaswsp01) ports 17 and 18
    - Leaf2 (RU19/pcaswlf06) ports 35 and 36 to Spine2 (RU30/pcaswsp02) ports 17 and 18

When the physical connections are in place, the GPU expansion rack must be activated. For more information, see Integrating GPU Expansion Nodes on page 166.

Expanding Private Cloud Appliance Capacity with a Compute Expansion Rack

To increase the compute capacity beyond the physical limits of the base rack, Private Cloud Appliance can be expanded with compute nodes installed in a compute expansion rack.

A compute expansion rack contains 3, 6, 9, or 12 factory-installed nodes. More nodes can be installed after initial deployment, but always in sets of 3. Up to two expansion racks can be connected to a base rack, for a maximum of 24 compute nodes in addition to those already installed in the base rack. For more information, see Compute Expansion Rack Configuration on page 23.

Installation Requirements

Site Preparation

If you have decided to expand your Private Cloud Appliance environment with a compute expansion rack, carefully plan ahead for the installation of the additional hardware. The expansion rack has the same external dimensions as the base rack, and contains the same type of hardware. Therefore, the base rack site requirements also apply for the expansion rack. They are described in detail in Preparing the Installation Site on page 55.

Rack Cabling

The cable connections between the base rack and the compute expansion rack must not exceed 25 meters. Allocate a space for the expansion rack near the base rack, ensuring that the inter-rack cabling is within the specified maximum length when routed through the floor or ceiling. The required cable length must be specified with the order. The default length of cables included in the shipment is 10 meters.

Connecting the Expansion Rack to the Base Rack

When the compute expansion rack is in its allocated space, it must be connected to the base rack. The expansion rack leaf switches are cross-connected to the base rack spine switches to extend the data network into the expansion rack. Similarly, the expansion rack components are added to the internal management network through a cable connection between the management switches in the racks. The ports required for this setup have been reserved on all connected switches.

Connecting the Management Switches

A compute expansion rack management switch must be connected to the base rack management switch using two cable links with 25Gbps SFP28 transceivers.

- On the side of the compute expansion rack, use management switch ports 49 and 50.
- Connect the cables from the first compute expansion rack to base rack management switch ports 49 and 50. Connect the cables from the second compute expansion rack to base rack management switch ports 51 and 52.

Connecting the Leaf Switches

Each compute expansion rack leaf switch has redundant cross-connections to the two spine switches of the base rack. There are 8 total cable links between a GPU expansion rack and the base rack, using 100Gbps QSFP28 transceivers.

- On the side of the compute expansion rack, use ports 33-36 in each leaf switch.
- Connect the cables from the first compute expansion rack to the base rack spine switches using this pattern:
  - Leaf1 (RU18/pcaswlf03) ports 33 and 34 to Spine1 (RU29/pcaswsp01) ports 11 and 12
  - Leaf1 (RU18/pcaswlf03) ports 35 and 36 to Spine2 (RU30/pcaswsp02) ports 11 and 12
  - Leaf2 (RU19/pcaswlf04) ports 33 and 34 to Spine1 (RU29/pcaswsp01) ports 13 and 14
  - Leaf2 (RU19/pcaswlf04) ports 35 and 36 to Spine2 (RU30/pcaswsp02) ports 13 and 14
- Connect the cables from the second compute expansion rack to the base rack spine switches using this pattern:
  - Leaf1 (RU18/pcaswlf05) ports 33 and 34 to Spine1 (RU29/pcaswsp01) ports 15 and 16
  - Leaf1 (RU18/pcaswlf05) ports 35 and 36 to Spine2 (RU30/pcaswsp02) ports 15 and 16
  - Leaf2 (RU19/pcaswlf06) ports 33 and 34 to Spine1 (RU29/pcaswsp01) ports 17 and 18
  - Leaf2 (RU19/pcaswlf06) ports 35 and 36 to Spine2 (RU30/pcaswsp02) ports 17 and 18

When the physical connections are in place, the compute expansion rack must be activated. For more information, see Integrating a Compute Expansion Rack on page 167.

Connecting External Systems to Private Cloud Appliance Using Flex Networks

Optionally, Private Cloud Appliance can be integrated with external systems in the on-premises network using cable connections to reserved ports on the spine switches. *Flex networks* operate in two modes, edge mode and hub mode.

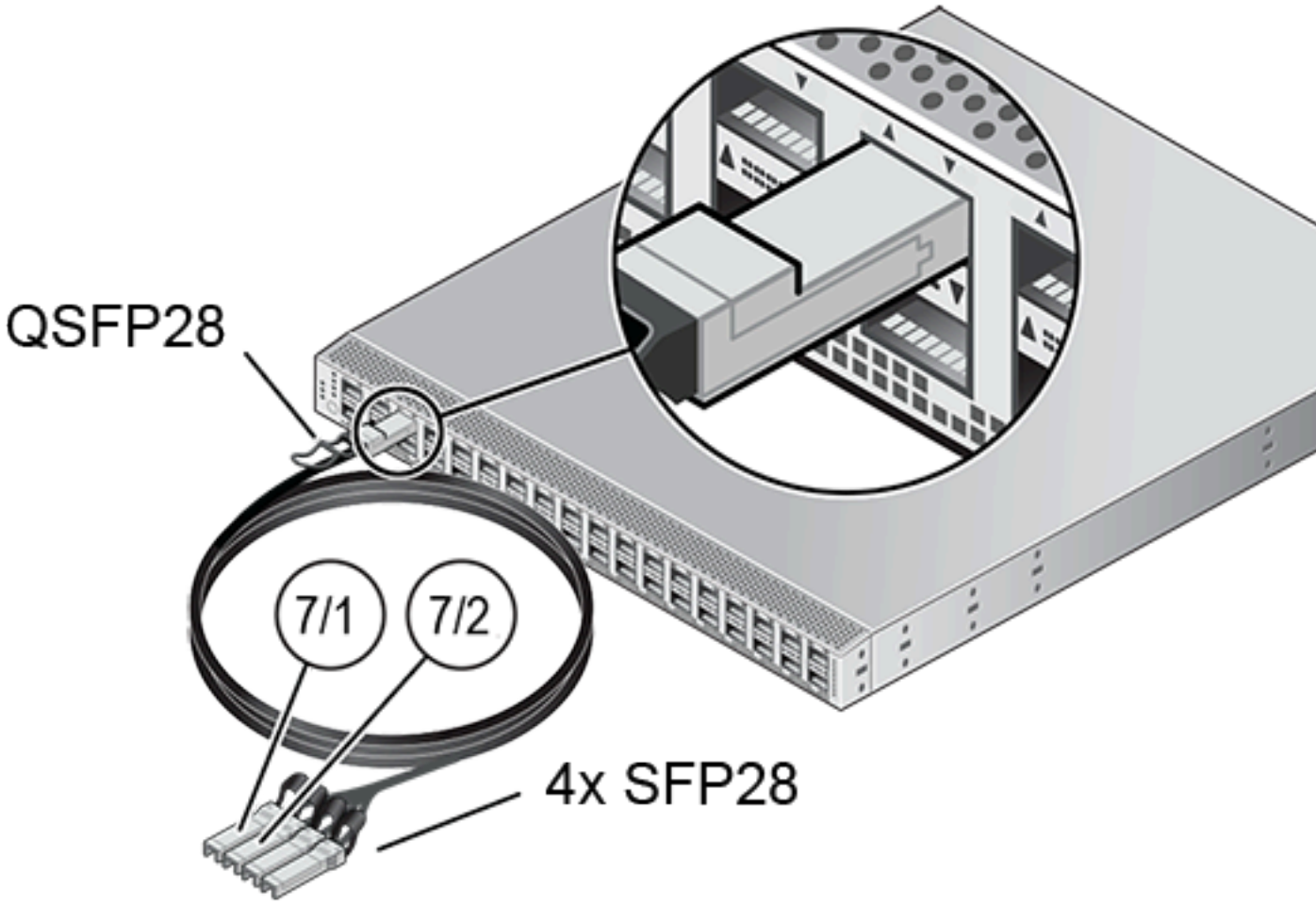
In edge mode, Private Cloud Appliance can be integrated with external systems such as Oracle Exadata, Oracle Database Appliance, and Oracle ZFS Storage Appliance, with the external components directly connected to reserved ports on the spine switches of the Private Cloud Appliance. Four 100 Gbps ports per spine switch are reserved for this purpose and in edge mode, they are split into four 25 Gbps breakout connections per port, providing a maximum of thirty two (32) 25 Gbps connections. Each externally connected component is cabled directly using two connections to both spine switch for redundancy, which allows for as many as sixteen (16) external connections. This feature was formerly known as Exadata Networks.

In hub mode, a flex network connection uses a gateway on the outside of Private Cloud Appliance to route traffic between external devices in an on-premises network and compute instances in a VCN. This mode allows for connectivity to devices that are not directly connected to the Private Cloud Appliance and supports an increased number of network speeds, for example: 10, 20, 25, 40, 50, and 100 Gbps.

Flex networks can support a mix of edge mode and hub mode across different ports on the Private Cloud Appliance spine switches.

For more information on the use of flex networks, see Integrating External Systems with Flex Networks on page 47.

> **Note:**
>
> The 25 Gbps breakout configuration applies to connectivity with Oracle Exadata nodes or Oracle Database Appliance nodes and does not indicate that flex networks are limited to 25 Gbps.



When the cable connections are in place, the administrator must configure the flex network in the Service Enclave, to enable traffic between the connected system and the Private Cloud Appliance environment. See Creating and Managing Flex Networks on page 207.

Cabling Reference for Private Cloud Appliance

Components in a Private Cloud Appliance are cabled according to detailed specifications. Use the cabling reference in case you need to verify cabling or reconnect components that have become disconnected.

Different rack configurations exist for Private Cloud Appliance. For more information, see Overview of Private Cloud Appliance Hardware Components on page 14. Reference tables are provided for each configuration.

- Private Cloud Appliance X11 Management Switch Ethernet Connections on page 126
- Private Cloud Appliance X10 Management Switch Ethernet Connections on page 127
- Private Cloud Appliance X9 Management Switch Ethernet Connections on page 129
- Private Cloud Appliance X11 Spine and Leaf Switch Data Network Connections on page 130
- Private Cloud Appliance X10 Spine and Leaf Switch Data Network Connections on page 132
- Private Cloud Appliance X9 Spine and Leaf Switch Data Network Connections on page 136
- Private Cloud Appliance X10-2c GPU Expansion Rack Cabling on page 139
- Private Cloud Appliance Compute Expansion Rack Cabling on page 142

Private Cloud Appliance X11 Management Switch Ethernet Connections

This section applies to the Private Cloud Appliance X11 rack configuration, which uses Oracle Server X11 2U compute nodes. The table lists the cabling connections for the management switch. These are Ethernet connections intended for the internal administration network.

From Rack Unit	From Component and Port	To Management Switch Rack Unit and Port
N/A	bastion/gateway	U23: port 2
U40-41 (^)	N/A	N/A
U38-39 (*)	N/A	N/A
U36-37 (*)	compute node: NET MGMT	U23: port 34
	compute node: NET0	U23: port 33
U34-35 (*)	compute node: NET MGMT	U23: port 32
	compute node: NET0	U23: port 31
U32-33 (*)	compute node: NET MGMT	U23: port 30
	compute node: NET0	U23: port 29
U30-31 (*)	compute node: NET MGMT	U23: port 28
	compute node: NET0	U23: port 27
U29	spine switch: NET MGMT	U23: port 48
U28	spine switch: NET MGMT	U23: port 47
U22	leaf switch: NET MGMT	U23: port 46
U21	leaf switch: NET MGMT	U23: port 45
U19-20 (*)	compute node: NET MGMT	U23: port 26
	compute node: NET0	U23: port 25
U17-18 (*)	compute node: NET MGMT	U23: port 24
	compute node: NET0	U23: port 23
U15-16	compute node: NET MGMT	U23: port 22
	compute node: NET0	U23: port 21
U13-14	compute node: NET MGMT	U23: port 20
	compute node: NET0	U23: port 19
U11-12	compute node: NET MGMT	U23: port 18
	compute node: NET0	U23: port 17

From Rack Unit	From Component and Port	To Management Switch Rack Unit and Port
U09-10	management node: NET MGMT	U23: port 16
	management node: NET0	U23: port 15
U07-08	management node: NET MGMT	U23: port 14
	management node: NET0	U23: port 13
U05-06	management node: NET MGMT	U23: port 12
	management node: NET0	U23: port 11
U03	ZFS Storage Appliance controller: NET MGMT	U23: port 10
	ZFS Storage Appliance controller: NET0	U23: port 9
	ZFS Storage Appliance controller: PCIE6-2	U23: port 6
U01	ZFS Storage Appliance controller: NET MGMT	U23: port 8
	ZFS Storage Appliance controller: NET0	U23: port 7
	ZFS Storage Appliance controller: PCIE6-2	U23: port 5
N/A	Oracle Support	U23: port 1

(*) – Indicates a rack unit designated as a flex bay. These rack units can contain compute nodes or storage nodes. The rack unit and port numbers assigned to flex bays only apply when a compute node is installed in that location.

(^) – Indicates a rack unit designated as a flex bay. These rack units can only contain storage nodes because of power constraints.

Private Cloud Appliance X10 Management Switch Ethernet Connections

This section applies to the Private Cloud Appliance X10 rack configuration, which uses Oracle Server E5-2L 2U compute nodes. The table lists the cabling connections for the management switch. These are Ethernet connections intended for the internal administration network.

From Rack Unit	From Component and Port	To Management Switch Rack Unit and Port
N/A	bastion/gateway	U24: port 2
U39-40 (^)	compute node: NET0	U24: port 19
U37-38 (*)	compute node: NET0	U24: port 39
U35-36 (*)	compute node: NET0	U24: port 38
U33-34 (*)	compute node: NET0	U24: port 37

From Rack Unit	From Component and Port	To Management Switch Rack Unit and Port
U31-32 (*)	compute node: NET0	U24: port 36
U30	spine switch: NET MGMT	U24: port 14
U29	spine switch: NET MGMT	U24: port 13
U23	leaf switch: NET MGMT	U24: port 34
U22	leaf switch: NET MGMT	U24: port 33
U20-21 (*)	compute node: NET0	U24: port 35
U18-19 (*)	compute node: NET0	U24: port 21
U16-17 (*)	compute node: NET0	U24: port 17
U14-15 (*)	compute node: NET0	U24: port 15
U12-13	compute node: NET0	U24: port 12
U10-11	compute node: NET0	U24: port 10
U08-09	compute node: NET0	U24: port 8
U07	management node: NET0	U24: port 5
U06	management node: NET0	U24: port 32
U05	management node: NET0	U24: port 31
U03-04	ZFS Storage Appliance ZS9-2 controller: NET0	U24: port 28
U03-04	ZFS Storage Appliance ZS9-2 controller: PCIe 6-2	U24: port 29
U01-02	ZFS Storage Appliance ZS9-2 controller: NET0	U24: port 25
U01-02	ZFS Storage Appliance ZS9-2 controller: PCIe 6-2	U24: port 26
N/A	Oracle Support	U24: port 1

(*) – Indicates a rack unit designated as a flex bay. These rack units can contain compute nodes or storage nodes. The rack unit and port numbers assigned to flex bays only apply when a compute node is installed in that location.

(^) – Indicates a rack unit reserved for an expansion compute node. To maintain balanced availability domains, compute nodes must be installed in multiples of 3. The reserved slot allows for 6 compute nodes to be installed without sacrificing a second flex bay.

Private Cloud Appliance X9 Management Switch Ethernet Connections

This section applies to the Private Cloud Appliance X9 rack configuration, which uses Oracle Server X9-2 1U compute nodes. The table lists the cabling connections for the management switch. These are Ethernet connections intended for the internal administration network.

From Rack Unit	From Component and Port	To Management Switch Rack Unit and Port
--	bastion	U26: port 2
U41*	compute node: NET0	U26: port 42
U40*	compute node: NET0	U26: port 41
U39*	compute node: NET0	U26: port 40
U38*	compute node: NET0	U26: port 39
U37*	compute node: NET0	U26: port 38
U36*	compute node: NET0	U26: port 37
U35*	compute node: NET0	U26: port 36
U34*	compute node: NET0	U26: port 35
U32	leaf switch: NET MGMT	U26: port 14
U31	leaf switch: NET MGMT	U26: port 13
U25	spine switch: NET MGMT	U26: port 34
U24	spine switch: NET MGMT	U26: port 33
U23*	compute node: NET0	U26: port 23
U22*	compute node: NET0	U26: port 22
U21*	compute node: NET0	U26: port 21
U20*	compute node: NET0	U26: port 20
U19*	compute node: NET0	U26: port 19
U18*	compute node: NET0	U26: port 18
U17*	compute node: NET0	U26: port 17
U16*	compute node: NET0	U26: port 16
U15	compute node: NET0	U26: port 15
U14	compute node: NET0	U26: port 12
U13	compute node: NET0	U26: port 11

From Rack Unit	From Component and Port	To Management Switch Rack Unit and Port
U12	compute node: NET0	U26: port 10
U11	compute node: NET0	U26: port 9
U10	compute node: NET0	U26: port 8
U09	compute node: NET0	U26: port 7
U08	compute node: NET0	U26: port 6
U07	management node: NET0	U26: port 5
U06	management node: NET0	U26: port 32
U05	management node: NET0	U26: port 31
U03	ZFS Storage Appliance ZS9-2 controller: NET0	U26: port 28
U03	ZFS Storage Appliance ZS9-2 controller: PCIe 6-2	U26: port 29
U01	ZFS Storage Appliance ZS9-2 controller: NET0	U26: port 25
U01	ZFS Storage Appliance ZS9-2 controller: PCIe 6-2	U26: port 26
N/A	PDU-A	U26: port 24
N/A	PDU-B	U26: port 43
N/A	Oracle Support	U26: port 1

* Indicates a rack unit designated as a flex bay. These rack units can contain compute nodes or storage nodes. The rack unit and port numbers assigned to flex bays only apply when a compute node is installed in that location.

Private Cloud Appliance X11 Spine and Leaf Switch Data Network Connections

This section applies to the Private Cloud Appliance X11 rack configuration, which uses Oracle Server X11 2U compute nodes. The tables list the internal 100Gbit data network cabling connections for the spine and leaf switches.

Component Connections

The compute nodes, management nodes, and storage controllers are connected to the data network as follows:

Rack Unit	From Node Port 1	To Switch Rack Unit and Port	From Node Port 2	To Switch Rack Unit and Port
U36-37	PCIe5 port 1	U21: port 25	PCIe5 port 2	U22: port 25
compute node	PCIe1 port 1	U21: port 24	PCIe1 port 2	U22: port 24

Rack Unit	From Node Port 1	To Switch Rack Unit and Port	From Node Port 2	To Switch Rack Unit and Port
U34-35	PCIe5 port 1	U21: port 23	PCIe5 port 2	U22: port 23
compute node	PCIe1 port 1	U21: port 22	PCIe1 port 2	U22: port 22
U32-33	PCIe5 port 1	U21: port 21	PCIe5 port 2	U22: port 21
compute node	PCIe1 port 1	U21: port 20	PCIe1 port 2	U22: port 20
U30-31	PCIe5 port 1	U21: port 19	PCIe5 port 2	U22: port 19
compute node	PCIe1 port 1	U21: port 18	PCIe1 port 2	U22: port 18
U19-20	PCIe5 port 1	U21: port 17	PCIe5 port 2	U22: port 17
compute node	PCIe1 port 1	U21: port 16	PCIe1 port 2	U22: port 16
U17-18	PCIe5 port 1	U21: port 15	PCIe5 port 2	U22: port 15
compute node	PCIe1 port 1	U21: port 14	PCIe1 port 2	U22: port 14
U15-16	PCIe5 port 1	U21: port 13	PCIe5 port 2	U22: port 13
compute node	PCIe1 port 1	U21: port 12	PCIe1 port 2	U22: port 12
U13-14	PCIe5 port 1	U21: port 11	PCIe5 port 2	U22: port 11
compute node	PCIe1 port 1	U21: port 10	PCIe1 port 2	U22: port 10
U11-12	PCIe5 port 1	U21: port 9	PCIe5 port 2	U22: port 9
compute node	PCIe1 port 1	U21: port 8	PCIe1 port 2	U22: port 8
U09-10 management node	PCIe1 port 1	U21: port 7	PCIe1 port 2	U22: port 7
U07-08 management node	PCIe1 port 1	U21: port 6	PCIe1 port 2	U22: port 6
U05-06 management node	PCIe1 port 1	U21: port 5	PCIe1 port 2	U22: port 5
U03-04	PCIe9 port 1	U28: port 27 B	PCIe9 port 2	U29: port 27 B
ZS11-2 storage controller	PCIe1 port 1	U28: port 35	PCIe5 port 2	U29: port 35
U01-02	PCIe9 port 1	U28: port 27 A	PCIe9 port 2	U29: port 27 A
ZS11-2 storage controller	PCIe1 port 1	U28: port 33	PCIe5 port 2	U29: port 33

Data Network Switch Interconnects

The leaf and spine switches at the core of the data network are interconnected as follows:

From Rack Unit	From Component and Port	To Rack Unit	To Component and Port
U28	spine switch: port 23	U21	leaf switch: port 33
U28	spine switch: port 24	U21	leaf switch: port 34
U28	spine switch: port 25	U22	leaf switch: port 35
U28	spine switch: port 26	U22	leaf switch: port 36
U29	spine switch: port 23	U22	leaf switch: port 33
U29	spine switch: port 24	U22	leaf switch: port 34
U29	spine switch: port 25	U21	leaf switch: port 35
U29	spine switch: port 26	U21	leaf switch: port 36

From Rack Unit	From Component and Port	To Rack Unit	To Component and Port
U21	leaf switch: port 3	U22	leaf switch: port 3
U21	leaf switch: port 4	U22	leaf switch: port 4

From Rack Unit	From Component and Port	To Rack Unit	To Component and Port
U28	spine switch: port 28	U29	spine switch: port 28
U28	spine switch: port 29	U29	spine switch: port 29
U28	spine switch: port 30	U29	spine switch: port 30
U28	spine switch: port 31	U29	spine switch: port 31
U28	spine switch: port 32	U29	spine switch: port 32

Private Cloud Appliance X10 Spine and Leaf Switch Data Network Connections

This section applies to the Private Cloud Appliance X10 rack configuration, which uses Oracle Server E5-2L 2U compute nodes. The tables list the internal 100Gbit data network cabling connections for the spine and leaf switches.

Component Connections

The compute nodes, management nodes, and storage controllers are connected to the data network as follows:

From Rack Unit	From Component and Port	To Spine or Leaf Switch Rack Unit and Port
U39-40	compute node: PCIe5 port 1	U22: port 19
U39-40	compute node: PCIe5 port 2	U23: port 19
U39-40	compute node: PCIe1 port 1	U22: port 18

From Rack Unit	From Component and Port	To Spine or Leaf Switch Rack Unit and Port
U39-40	compute node: PCIe1 port 2	U23: port 18
U37-38	compute node: PCIe5 port 1	U22: port 31
U37-38	compute node: PCIe5 port 2	U23: port 31
U37-38	compute node: PCIe1 port 1	U22: port 30
U37-38	compute node: PCIe1 port 2	U23: port 30
U35-36	compute node: PCIe5 port 1	U22: port 29
U35-36	compute node: PCIe5 port 2	U23: port 29
U35-36	compute node: PCIe1 port 1	U22: port 28
U35-36	compute node: PCIe1 port 2	U23: port 28
U33-34	compute node: PCIe5 port 1	U22: port 27
U33-34	compute node: PCIe5 port 2	U23: port 27
U33-34	compute node: PCIe1 port 1	U22: port 26
U33-34	compute node: PCIe1 port 2	U23: port 26
U31-32	compute node: PCIe5 port 1	U22: port 25
U31-32	compute node: PCIe5 port 2	U23: port 25
U31-32	compute node: PCIe1 port 1	U22: port 24
U31-32	compute node: PCIe1 port 2	U23: port 24
U20-21	compute node: PCIe5 port 1	U22: port 23
U20-21	compute node: PCIe5 port 2	U23: port 23
U20-21	compute node: PCIe1 port 1	U22: port 22
U20-21	compute node: PCIe1 port 2	U23: port 22
U18-19	compute node: PCIe5 port 1	U22: port 21
U18-19	compute node: PCIe5 port 2	U23: port 21
U18-19	compute node: PCIe1 port 1	U22: port 20
U18-19	compute node: PCIe1 port 2	U23: port 20
U16-17	compute node: PCIe5 port 1	U22: port 17

From Rack Unit	From Component and Port	To Spine or Leaf Switch Rack Unit and Port
U16-17	compute node: PCIe5 port 2	U23: port 17
U16-17	compute node: PCIe1 port 1	U22: port 16
U16-17	compute node: PCIe1 port 2	U23: port 16
U14-15	compute node: PCIe5 port 1	U22: port 15
U14-15	compute node: PCIe5 port 2	U23: port 15
U14-15	compute node: PCIe1 port 1	U22: port 14
U14-15	compute node: PCIe1 port 2	U23: port 14
U12-13	compute node: PCIe5 port 1	U22: port 13
U12-13	compute node: PCIe5 port 2	U23: port 13
U12-13	compute node: PCIe1 port 1	U22: port 12
U12-13	compute node: PCIe1 port 2	U23: port 12
U10-11	compute node: PCIe5 port 1	U22: port 11
U10-11	compute node: PCIe5 port 2	U23: port 11
U10-11	compute node: PCIe1 port 1	U22: port 10
U10-11	compute node: PCIe1 port 2	U23: port 10
U08-09	compute node: PCIe5 port 1	U22: port 9
U08-09	compute node: PCIe5 port 2	U23: port 9
U08-09	compute node: PCIe1 port 1	U22: port 8
U08-09	compute node: PCIe1 port 2	U23: port 8
U07	management node: port 1	U22: port 7
U07	management node: port 2	U23: port 7
U06	management node: port 1	U22: port 6
U06	management node: port 2	U23: port 6
U05	management node: port 1	U22: port 5
U05	management node: port 2	U23: port 5
U03-04	ZFS Storage Appliance ZS9-2 controller: PCIe3 port 1	U29: port 35

From Rack Unit	From Component and Port	To Spine or Leaf Switch Rack Unit and Port
U03-04	ZFS Storage Appliance ZS9-2 controller: PCIe10 port 1	U30: port 35
U01-02	ZFS Storage Appliance ZS9-2 controller: PCIe3 port 1	U29: port 33
U01-02	ZFS Storage Appliance ZS9-2 controller: PCIe10 port 1	U30: port 33

Data Network Switch Interconnects

The leaf and spine switches at the core of the data network are interconnected as follows:

From Rack Unit	From Component and Port	To Rack Unit	To Component and Port
U29	spine switch: port 23	U22	leaf switch: port 33
U29	spine switch: port 24	U22	leaf switch: port 34
U29	spine switch: port 25	U23	leaf switch: port 35
U29	spine switch: port 26	U23	leaf switch: port 36
U30	spine switch: port 23	U23	leaf switch: port 33
U30	spine switch: port 24	U23	leaf switch: port 34
U30	spine switch: port 25	U22	leaf switch: port 35
U30	spine switch: port 26	U22	leaf switch: port 36

From Rack Unit	From Component and Port	To Rack Unit	To Component and Port
U22	leaf switch: port 3	U23	leaf switch: port 3
U22	leaf switch: port 4	U23	leaf switch: port 4

From Rack Unit	From Component and Port	To Rack Unit	To Component and Port
U29	spine switch: port 28	U30	spine switch: port 28
U29	spine switch: port 29	U30	spine switch: port 29
U29	spine switch: port 30	U30	spine switch: port 30
U29	spine switch: port 31	U30	spine switch: port 31

From Rack Unit	From Component and Port	To Rack Unit	To Component and Port
U29	spine switch: port 32	U30	spine switch: port 32

Private Cloud Appliance X9 Spine and Leaf Switch Data Network Connections

This section applies to the Private Cloud Appliance X9 rack configuration, which uses Oracle Server X9-2 1U compute nodes. The tables list the internal 100Gbit data network cabling connections for the spine and leaf switches.

Component Connections

The compute nodes, management nodes, and storage controllers are connected to the data network as follows:

From Rack Unit	From Component and Port	To Spine or Leaf Switch Rack Unit and Port
U41	compute node: port 1	U24: port 31
U41	compute node: port 2	U25: port 31
U40	compute node: port 1	U24: port 30
U40	compute node: port 2	U25: port 30
U39	compute node: port 1	U24: port 29
U39	compute node: port 2	U25: port 29
U38	compute node: port 1	U24: port 28
U38	compute node: port 2	U25: port 28
U37	compute node: port 1	U24: port 27
U37	compute node: port 2	U25: port 27
U36	compute node: port 1	U24: port 26
U36	compute node: port 2	U25: port 26
U35	compute node: port 1	U24: port 25
U35	compute node: port 2	U25: port 25
U34	compute node: port 1	U24: port 24
U34	compute node: port 2	U25: port 24
U23	compute node: port 1	U24: port 23
U23	compute node: port 2	U25: port 23
U22	compute node: port 1	U24: port 22

From Rack Unit	From Component and Port	To Spine or Leaf Switch Rack Unit and Port
U22	compute node: port 2	U25: port 22
U21	compute node: port 1	U24: port 21
U21	compute node: port 2	U25: port 21
U20	compute node: port 1	U24: port 20
U20	compute node: port 2	U25: port 20
U19	compute node: port 1	U24: port 19
U19	compute node: port 2	U25: port 19
U18	compute node: port 1	U24: port 18
U18	compute node: port 2	U25: port 18
U17	compute node: port 1	U24: port 17
U17	compute node: port 2	U25: port 17
U16	compute node: port 1	U24: port 16
U16	compute node: port 2	U25: port 16
U15	compute node: port 1	U24: port 15
U15	compute node: port 2	U25: port 15
U14	compute node: port 1	U24: port 14
U14	compute node: port 2	U25: port 14
U13	compute node: port 1	U24: port 13
U13	compute node: port 2	U25: port 13
U12	compute node: port 1	U24: port 12
U12	compute node: port 2	U25: port 12
U11	compute node: port 1	U24: port 11
U11	compute node: port 2	U25: port 11
U10	compute node: port 1	U24: port 10
U10	compute node: port 2	U25: port 10
U09	compute node: port 1	U24: port 9

From Rack Unit	From Component and Port	To Spine or Leaf Switch Rack Unit and Port
U09	compute node: port 2	U25: port 9
U08	compute node: port 1	U24: port 8
U08	compute node: port 2	U25: port 8
U07	management node: port 1	U24: port 7
U07	management node: port 2	U25: port 7
U06	management node: port 1	U24: port 6
U06	management node: port 2	U25: port 6
U05	management node: port 1	U24: port 5
U05	management node: port 2	U25: port 5
U03	ZFS Storage Appliance ZS9-2 controller: PCIE3 port 1	U31: port 35
U03	ZFS Storage Appliance ZS9-2 controller: PCIE10 port 1	U32: port 35
U01	ZFS Storage Appliance ZS9-2 controller: PCIE3 port 1	U31: port 33
U01	ZFS Storage Appliance ZS9-2 controller: PCIE10 port 1	U32: port 33

Data Network Switch Interconnects

The leaf and spine switches at the core of the data network are interconnected as follows:

From Rack Unit	From Component and Port	To Rack Unit	To Component and Port
U31	spine switch: port 23	U24	leaf switch: port 33
U31	spine switch: port 24	U24	leaf switch: port 34
U31	spine switch: port 25	U25	leaf switch: port 35
U31	spine switch: port 26	U25	leaf switch: port 36
U32	spine switch: port 23	U25	leaf switch: port 33
U32	spine switch: port 24	U25	leaf switch: port 34
U32	spine switch: port 25	U24	leaf switch: port 35
U32	spine switch: port 26	U24	leaf switch: port 36

From Rack Unit	From Component and Port	To Rack Unit	To Component and Port
U24	leaf switch: port 3	U25	leaf switch: port 3
U24	leaf switch: port 4	U25	leaf switch: port 4

From Rack Unit	From Component and Port	To Rack Unit	To Component and Port
U31	spine switch: port 28	U32	spine switch: port 28
U31	spine switch: port 29	U32	spine switch: port 29
U31	spine switch: port 30	U32	spine switch: port 30
U31	spine switch: port 31	U32	spine switch: port 31
U31	spine switch: port 32	U32	spine switch: port 32

Private Cloud Appliance X10-2c GPU Expansion Rack Cabling

This section applies to the X10-2c GPU expansion rack, containing 3U X10-2c GPU L40S Compute Server nodes. Inter-rack cabling information is critical to connect the GPU expansion rack to the appliance base rack correctly. The internal component cabling information can be used as a reference, for example in case of hardware maintenance or when installing additional GPU nodes.

Inter-Rack Connections

The table lists the cabling connections between the Private Cloud Appliance base rack and up to two X10-2c GPU expansion racks. These are Ethernet connections between the management and data network switches installed in each rack.

Base Rack Unit	From Switch/Port	GPU Rack 1	GPU Rack 2	To Switch/Port
U24 (pcaswmn01)	MgmtBase: port 49	U22 (pcaswmn02)	--	MgmtExp1: port 49
U24 (pcaswmn01)	MgmtBase: port 50	U22 (pcaswmn02)	--	MgmtExp1: port 50
U24 (pcaswmn01)	MgmtBase: port 51	--	U22 (pcaswmn03)	MgmtExp2: port 49
U24 (pcaswmn01)	MgmtBase: port 52	--	U22 (pcaswmn03)	MgmtExp2: port 50
U29 (pcaswsp01)	Spine1: port 11	U18 (pcaswlf03)	--	Leaf1Exp1: port 33
U29 (pcaswsp01)	Spine1: port 12	U18 (pcaswlf03)	--	Leaf1Exp1: port 34
U29 (pcaswsp01)	Spine1: port 13	U19 (pcaswlf04)	--	Leaf2Exp1: port 33
U29 (pcaswsp01)	Spine1: port 14	U19 (pcaswlf04)	--	Leaf2Exp1: port 34
U29 (pcaswsp01)	Spine1: port 15	--	U18 (pcaswlf05)	Leaf1Exp2: port 33
U29 (pcaswsp01)	Spine1: port 16	--	U18 (pcaswlf05)	Leaf1Exp2: port 34

Base Rack Unit	From Switch/Port	GPU Rack 1	GPU Rack 2	To Switch/Port
U29 (pcaswsp01)	Spine1: port 17	--	U19 (pcaswlf06)	Leaf2Exp2: port 33
U29 (pcaswsp01)	Spine1: port 18	--	U19 (pcaswlf06)	Leaf2Exp2: port 34
U30 (pcaswsp02)	Spine2: port 11	U18 (pcaswlf03)	--	Leaf1Exp1: port 35
U30 (pcaswsp02)	Spine2: port 12	U18 (pcaswlf03)	--	Leaf1Exp1: port 36
U30 (pcaswsp02)	Spine2: port 13	U19 (pcaswlf04)	--	Leaf2Exp1: port 35
U30 (pcaswsp02)	Spine2: port 14	U19 (pcaswlf04)	--	Leaf2Exp1: port 36
U30 (pcaswsp02)	Spine2: port 15	--	U18 (pcaswlf05)	Leaf1Exp2: port 35
U30 (pcaswsp02)	Spine2: port 16	--	U18 (pcaswlf05)	Leaf1Exp2: port 36
U30 (pcaswsp02)	Spine2: port 17	--	U19 (pcaswlf06)	Leaf2Exp2: port 35
U30 (pcaswsp02)	Spine2: port 18	--	U19 (pcaswlf06)	Leaf2Exp2: port 36

Management Switch Internal Connections

The table lists the internal cabling connections for the management switch in the GPU expansion rack. These are Ethernet connections intended for the appliance administration network.

From Rack Unit	From Component and Port	To Management Switch Rack Unit and Port
U28-30	X10-2c GPU node: Bay11 port 0	U22: port 28
U28-30	X10-2c GPU node: NETMGT	U22: port 27
U24-26	X10-2c GPU node: Bay11 port 0	U22: port 26
U24-26	X10-2c GPU node: NETMGT	U22: port 25
U21	PDU: NETMGT	U22: port 4
U20	PDU: NETMGT	U22: port 3
U19	Leaf switch: NET MGMT	U22: port 46
U18	Leaf switch: NET MGMT	U22: port 45
U14-16	X10-2c GPU node: Bay11 port 0	U22: port 24
U14-16	X10-2c GPU node: NETMGT	U22: port 23
U10-12	X10-2c GPU node: Bay11 port 0	U22: port 22
U10-12	X10-2c GPU node: NETMGT	U22: port 21
U06-08	X10-2c GPU node: Bay11 port 0	U22: port 20

From Rack Unit	From Component and Port	To Management Switch Rack Unit and Port
U06-08	X10-2c GPU node: NETMGT	U22: port 19
U02-04	X10-2c GPU node: Bay11 port 0	U22: port 18
U02-04	X10-2c GPU node: NETMGT	U22: port 17
N/A	Oracle Support	U22: port 1

Leaf Switch Internal Connections

The table lists the internal cabling connections for the leaf switches. These are high speed (100Gbps) Ethernet connections intended for the appliance data network.

From Rack Unit	From Component and Port	To Leaf Switch Rack Unit and Port
U28-30	X10-2c GPU node: Bay14 port 1	U18: port 19
U28-30	X10-2c GPU node: Bay14 port 2	U19: port 19
U28-30	X10-2c GPU node: Bay12 port 1	U18: port 18
U28-30	X10-2c GPU node: Bay12 port 2	U19: port 18
U24-26	X10-2c GPU node: Bay14 port 1	U18: port 17
U24-26	X10-2c GPU node: Bay14 port 2	U19: port 17
U24-26	X10-2c GPU node: Bay12 port 1	U18: port 16
U24-26	X10-2c GPU node: Bay12 port 2	U19: port 16
U14-16	X10-2c GPU node: Bay14 port 1	U18: port 15
U14-16	X10-2c GPU node: Bay14 port 2	U19: port 15
U14-16	X10-2c GPU node: Bay12 port 1	U18: port 14
U14-16	X10-2c GPU node: Bay12 port 2	U19: port 14
U10-12	X10-2c GPU node: Bay14 port 1	U18: port 13
U10-12	X10-2c GPU node: Bay14 port 2	U19: port 13
U10-12	X10-2c GPU node: Bay12 port 1	U18: port 12
U10-12	X10-2c GPU node: Bay12 port 2	U19: port 12
U06-08	X10-2c GPU node: Bay14 port 1	U18: port 11
U06-08	X10-2c GPU node: Bay14 port 2	U19: port 11

From Rack Unit	From Component and Port	To Leaf Switch Rack Unit and Port
U06-08	X10-2c GPU node: Bay12 port 1	U18: port 10
U06-08	X10-2c GPU node: Bay12 port 2	U19: port 10
U02-04	X10-2c GPU node: Bay14 port 1	U18: port 9
U02-04	X10-2c GPU node: Bay14 port 2	U19: port 9
U02-04	X10-2c GPU node: Bay12 port 1	U18: port 8
U02-04	X10-2c GPU node: Bay12 port 2	U19: port 8

Private Cloud Appliance Compute Expansion Rack Cabling

This section applies to the Private Cloud Appliance compute expansion rack, containing Oracle Server X11 2U compute nodes. Inter-rack cabling information is critical to connect the compute expansion rack to the appliance base rack correctly. The internal component cabling information can be used as a reference, for example in case of hardware maintenance or when installing additional compute nodes.

Inter-Rack Connections

The table lists the cabling connections between the Private Cloud Appliance base rack and up to two compute expansion racks. These are Ethernet connections between the management and data network switches installed in each rack.

Base Rack Unit	From Switch/Port	Exp Rack 1	Exp Rack 2	To Switch/Port
U24 (pcaswmn01)	MgmtBase: port 49	U18 (pcaswmn02)	--	MgmtExp1: port 49
U24 (pcaswmn01)	MgmtBase: port 50	U18 (pcaswmn02)	--	MgmtExp1: port 50
U24 (pcaswmn01)	MgmtBase: port 51	--	U18 (pcaswmn03)	MgmtExp2: port 49
U24 (pcaswmn01)	MgmtBase: port 52	--	U18 (pcaswmn03)	MgmtExp2: port 50
U29 (pcaswsp01)	Spine1: port 11	U19 (pcaswlf03)	--	Leaf1Exp1: port 33
U29 (pcaswsp01)	Spine1: port 12	U19 (pcaswlf03)	--	Leaf1Exp1: port 34
U29 (pcaswsp01)	Spine1: port 13	U20 (pcaswlf04)	--	Leaf2Exp1: port 33
U29 (pcaswsp01)	Spine1: port 14	U20 (pcaswlf04)	--	Leaf2Exp1: port 34
U29 (pcaswsp01)	Spine1: port 15	--	U19 (pcaswlf05)	Leaf1Exp2: port 33
U29 (pcaswsp01)	Spine1: port 16	--	U19 (pcaswlf05)	Leaf1Exp2: port 34
U29 (pcaswsp01)	Spine1: port 17	--	U20 (pcaswlf06)	Leaf2Exp2: port 33
U29 (pcaswsp01)	Spine1: port 18	--	U20 (pcaswlf06)	Leaf2Exp2: port 34
U30 (pcaswsp02)	Spine2: port 11	U19 (pcaswlf03)	--	Leaf1Exp1: port 35

Base Rack Unit	From Switch/Port	Exp Rack 1	Exp Rack 2	To Switch/Port
U30 (pcaswsp02)	Spine2: port 12	U19 (pcaswlf03)	--	Leaf1Exp1: port 36
U30 (pcaswsp02)	Spine2: port 13	U20 (pcaswlf04)	--	Leaf2Exp1: port 35
U30 (pcaswsp02)	Spine2: port 14	U20 (pcaswlf04)	--	Leaf2Exp1: port 36
U30 (pcaswsp02)	Spine2: port 15	--	U19 (pcaswlf05)	Leaf1Exp2: port 35
U30 (pcaswsp02)	Spine2: port 16	--	U19 (pcaswlf05)	Leaf1Exp2: port 36
U30 (pcaswsp02)	Spine2: port 17	--	U20 (pcaswlf06)	Leaf2Exp2: port 35
U30 (pcaswsp02)	Spine2: port 18	--	U20 (pcaswlf06)	Leaf2Exp2: port 36

Management Switch Internal Connections

The table lists the internal cabling connections for the management switch in the expansion rack. These are Ethernet connections intended for the appliance administration network.

From Rack Unit	From Component and Port	To Management Switch Rack Unit and Port
U32-33	compute node: NETMGT	U18: port 40
U32-33	compute node: NET0	U18: port 39
U30-31	compute node: NETMGT	U18: port 38
U30-31	compute node: NET0	U18: port 37
U28-29	compute node: NETMGT	U18: port 36
U28-29	compute node: NET0	U18: port 35
U26-27	compute node: NETMGT	U18: port 34
U26-27	compute node: NET0	U18: port 33
U24-25	compute node: NETMGT	U18: port 32
U24-25	compute node: NET0	U18: port 31
U22-23	compute node: NETMGT	U18: port 30
U22-23	compute node: NET0	U18: port 29
U20	Leaf switch: NET MGMT	U18: port 46
U19	Leaf switch: NET MGMT	U18: port 45
U16-17	compute node: NETMGT	U18: port 28
U16-17	compute node: NET0	U18: port 27

From Rack Unit	From Component and Port	To Management Switch Rack Unit and Port
U14-15	compute node: NETMGT	U18: port 26
U14-15	compute node: NET0	U18: port 25
U12-13	compute node: NETMGT	U18: port 24
U12-13	compute node: NET0	U18: port 23
U10-11	(reserved)	
U10-11	(reserved)	
U08-09	(reserved)	
U08-09	(reserved)	
U06-07	compute node: NETMGT	U18: port 22
U06-07	compute node: NET0	U18: port 21
U04-05	compute node: NETMGT	U18: port 20
U04-05	compute node: NET0	U18: port 19
U02-03	compute node: NETMGT	U18: port 18
U02-03	compute node: NET0	U18: port 17
N/A	Oracle Support	U18: port 1

Leaf Switch Internal Connections

The table lists the internal cabling connections for the leaf switches. These are high speed (100Gbps) Ethernet connections intended for the appliance data network.

From Rack Unit	From Node Slot 1 and Port	To Leaf Switch Rack Unit and Port	From Node Slot 2 and Port	To Leaf Switch Rack Unit and Port
U32-33	compute node: PCIe5 port 1	U19: port 31	compute node: PCIe5 port 2	U20: port 31
U32-33	compute node: PCIe1 port 1	U19: port 30	compute node: PCIe1 port 2	U20: port 30
U30-31	compute node: PCIe5 port 1	U19: port 29	compute node: PCIe5 port 2	U20: port 29
U30-31	compute node: PCIe1 port 1	U19: port 28	compute node: PCIe1 port 2	U20: port 28
U28-29	compute node: PCIe5 port 1	U19: port 27	compute node: PCIe5 port 2	U20: port 27
U28-29	compute node: PCIe1 port 1	U19: port 26	compute node: PCIe1 port 2	U20: port 26
U26-27	compute node: PCIe5 port 1	U19: port 25	compute node: PCIe5 port 2	U20: port 25

From Rack Unit	From Node Slot 1 and Port	To Leaf Switch Rack Unit and Port	From Node Slot 2 and Port	To Leaf Switch Rack Unit and Port
U26-27	compute node: PCIe1 port 1	U19: port 24	compute node: PCIe1 port 2	U20: port 24
U24-25	compute node: PCIe5 port 1	U19: port 23	compute node: PCIe5 port 2	U20: port 23
U24-25	compute node: PCIe1 port 1	U19: port 22	compute node: PCIe1 port 2	U20: port 22
U22-23	compute node: PCIe5 port 1	U19: port 21	compute node: PCIe5 port 2	U20: port 21
U22-23	compute node: PCIe1 port 1	U19: port 20	compute node: PCIe1 port 2	U20: port 20
U16-17	compute node: PCIe5 port 1	U19: port 19	compute node: PCIe5 port 2	U20: port 19
U16-17	compute node: PCIe1 port 1	U19: port 18	compute node: PCIe1 port 2	U20: port 18
U14-15	compute node: PCIe5 port 1	U19: port 17	compute node: PCIe5 port 2	U20: port 17
U14-15	compute node: PCIe1 port 1	U19: port 16	compute node: PCIe1 port 2	U20: port 16
U12-13	compute node: PCIe5 port 1	U19: port 15	compute node: PCIe5 port 2	U20: port 15
U12-13	compute node: PCIe1 port 1	U19: port 14	compute node: PCIe1 port 2	U20: port 14
U10-11	(reserved)	U19: port 6	(reserved)	U20: port 6
U08-09	(reserved)	U19: port 5	(reserved)	U20: port 5
U06-07	compute node: PCIe5 port 1	U19: port 13	compute node: PCIe5 port 2	U20: port 13
U06-07	compute node: PCIe1 port 1	U19: port 12	compute node: PCIe1 port 2	U20: port 12
U04-05	compute node: PCIe5 port 1	U19: port 11	compute node: PCIe5 port 2	U20: port 11
U04-05	compute node: PCIe1 port 1	U19: port 10	compute node: PCIe1 port 2	U20: port 10
U02-03	compute node: PCIe5 port 1	U19: port 9	compute node: PCIe5 port 2	U20: port 9
U02-03	compute node: PCIe1 port 1	U19: port 8	compute node: PCIe1 port 2	U20: port 8

Private Cloud Appliance Infrastructure Administration

The Private Cloud Appliance administrators manage and configure the infrastructure from the Service Enclave. This highly restricted part of the system provides tools for hardware and capacity management, tenancy control, and centralized monitoring of components at all system layers.

An appliance administrator is a highly privileged user with access to the physical components of Private Cloud Appliance. There is no functional relationship between an appliance administrator account and a tenancy administrator account; these are entirely separate entities. While appliance administrators may be authorized to create and delete tenancies, their account does not grant any permission to access a tenancy or use its resources. An appliance administrator has no access whatsoever to user data or instances.

A policy framework is provided to define the access privileges granted to administrator accounts. It includes predefined authorization groups and permissions, which can be extended and customized. See Controlling Administrator Access Privileges on page 178.

For more information about the Service Enclave, see Enclaves and Interfaces on page 8. General usage is described in the following topics:

- Using the Service Web UI on page 146
- Using the Service CLI on page 150

For information and instructions related to appliance hardware administration and external connections, see these topics:

- Performing Administrative Operations on Compute Nodes on page 156
- Integrating GPU Expansion Nodes on page 166
- Integrating a Compute Expansion Rack on page 167
- Creating and Managing Flex Networks on page 207
- Reconfiguring the Private Cloud Appliance Network Environment on page 235

For information and instructions related to other administration functions of the Service Enclave, see these topics:

- Managing Private Cloud Appliance Administrator Accounts on page 174
- Managing Private Cloud Appliance Tenancies on page 191
- Viewing and Setting Resource Limits on page 195
- Status and Health Monitoring on page 250
- System Backup and Restore on page 292
- Disaster Recovery on page 297

Using the Service Web UI

The Service Web UI is the graphical interface to the Service Enclave. Administrators can use the Service Web UI on its own or together with the Service CLI to complete tasks.

The Service Web UI provides the same core functionality as the Service CLI. However, the CLI does have some additional operations that don't have a UI equivalent. This topic provides general guidelines for using the Service Web UI. The actual operations and functions are documented throughout the administrator area of the documentation, as part of step-by-step instructions.

Signing In

To sign in to the Service Web UI, complete the following steps:

1. In a supported browser, enter the Private Cloud Appliance administration URL.

   For example: `https://adminconsole.`***`pcasys1`***`.`***`example.com`*** – where **`pcasys1`** is the name of your Private Cloud Appliance and **`example.com`** is your domain.

   The Sign In page is displayed.
2. Enter your Username and Password, and then click Sign In.

   The Service Enclave dashboard is displayed. It has a navigation menu and a set of quick action tiles.

> **Note:**
>
> If this is the first sign in after initial setup and configuration, the dashboard displays the ASR Phone Home page so you can register your system with My Oracle Support.
>
> For more information, see Registering Private Cloud Appliance for Oracle Auto Service Request on page 272.

Navigating the Dashboard

When you sign in to the Service Web UI, the dashboard is displayed with a Quick Actions area containing clickable tiles for common tasks, such as viewing rack unit, tenancy, and appliance details and managing users and the network environment.

In the Observability & Management part of the dashboard, there is a quick action tile for Monitoring. When you click Monitoring, the Grafana console opens. For more information, see #unique_118.

In the top bar of the dashboard you can locate the realm and the system and domain names for your Private Cloud Appliance. Your user name appears in the top bar as well, with links to your profile information, hardware data sync, oracle.com, and the option to sign out.

> **Note:**
>
> The dashboard is static and not configurable.

The navigation menu, which you can click or tab to, lists appliance components and resources that you can manage within the Service Enclave. When you click an item in the navigation menu, a page is displayed that contains information about the component or resource. The following table provides details about what you can expect to find on these component and resource pages.

Component or Resource	Information Provided
Appliance Details	Read-only appliance configuration details such as realm, region and domain. The page contains an option to edit rack name and description. The Rack Units list provides an overview of installed hardware components, and lets you drill down into more detailed component information. See further in this table.
Network Environment	Read-only network configuration information and an Edit button that opens a Configure Network Params wizard where you can modify: • Routing uplink gateway, VLAN, and HSRP group, and spine virtual IP • Management nodes IPs and hostnames • Uplink port speed, count, port FEC, VLAN MTU, and netmask and spine IPs • NTP servers IP addresses • Admin network status • DNS servers IP addresses • Public IP ranges and object storage IP For more information, see Reconfiguring the Private Cloud Appliance Network Environment on page 235.

Component or Resource	Information Provided
Rack Units	Read-only list of all hardware components installed in the rack and detected by the appliance software and the following information for each: • Name • Rack unit type • State • Rack elevation Each component also has an Actions menu (three dots) with a View Details link to a component's detail page. For management nodes, switches, and storage controllers, the detail pages provide read-only rack unit and system information. For each compute node in the list, you can see additional information: • Provisioning state • Maintenance lock • Provisioned lock A compute node's detail page provides read-only compute node, rack unit, and system information. Additionally, from either its detail page or the Actions menu, you can perform several actions on a compute node, such as locking for maintenance, migrating all virtual machines, stopping, deprovisioning. For more information, see Performing Administrative Operations on Compute Nodes on page 156.
Tenancies	Read-only list of all tenancies in the system and the following information for each: • Name • Description • Action menu Contains options to view a tenancy's details page, edit a tenancy's description, or delete a tenancy. You can also edit or delete a tenancy from its details page. A Create Tenancy button. For more information, see Managing Private Cloud Appliance Tenancies on page 191.

Component or Resource	Information Provided
Identity Providers	Read-only list of identity providers and the following information for each: • Name • Force Authentication • Encrypt Assertion • Action menu Contains options to view an identity provider's details page and edit or delete the identity provider. You can also edit or delete an identity provider from its details page. A Create Identity Provider button. For more information, see #unique_119.
IDP Group Mappings	Read-only list of IDP group mappings in the system and the following information for each: • Name • IDP Group Name • Admin Group Name • Description • Action menu Contains options to view read-only information on an IDP group mapping details page. A Create Group Mapping button. For more information, see #unique_119.
Users	Read only list of users in the system and the following information for each: • Name • Authorization Group • Default User • Action menu Contains options to view read-only information on a user's details page, change a user password, or delete a user. You can also change a user password or delete a user from its details page. A Create User button. For more information, see #unique_84.

Component or Resource	Information Provided
Jobs	Read-only list of jobs that ran and the following information for each: • Object type • Start and end times • Run status - Active, Succeeded, Failed, or Aborted • Action menu Contains an option to view read-only information on a job's details page, which includes the user account that the job ran from.
Upgrade & Patching	Read-only list of upgrade and patching jobs that ran and the following information for each: • Job name • Request and job IDs • Start and end times • Command name • Result - Passed, Failed, Not Run, Canceled, or None A Create Upgrade or Patch button, where you can select: • Upgrade Request - includes several types of upgrades, such as compute node, host, ILOM, Kubernetes, and platform. • Patch Request - includes several types of patches, such as compute node, host, ILOM, Kubernetes, OCI Images, and platform. For more information, see these publications: • Oracle Private Cloud Upgrade Guide • Oracle Private Cloud Appliance Patching Guide
ASR Phone Home	Read-only auto service request information and a Register button where you can register your Private Cloud Appliance. For more information, see Oracle Auto Service Request (ASR) on page 272.

Using the Service CLI

The command line interface to the Private Cloud Appliance Service Enclave, which we refer to as the *Service CLI* in the documentation, is available through the Oracle Linux shell on the management nodes. There is no additional installation or configuration required.

The CLI provides access to all the functionality of the Service Web UI, and several additional operations that don't have a UI equivalent. This topic provides general guidelines for using the Service CLI. The actual commands and their functions are documented throughout the administrator area of the documentation, as part of step-by-step instructions.

> **Note:**
>
> The CLI only accepts characters from the 7-bit ASCII table of letters, numerics, and other characters. Not all are valid for all fields, but A-Z, a-z, and 0-9 are generally accepted. The CLI ignores accented characters or

> those from the UTF table. Special care is needed to screen values provided in certificates or other sources.

Accessing the CLI

To access the Service CLI, establish an SSH connection to TCP port 30006 on a management node, and log in as an authorized administrator.

- Connect to the primary node in the management cluster (preferred):

```
$ ssh admin@admin.pca_hostname.example.com -p 30006
Password authentication
PCA-ADMIN>
```

- Connect to a specific management node:

```
$ ssh admin@pcamn02 -p 30006
Password authentication
PCA-ADMIN>
```

After successful authentication, you are in an interactive, closed shell environment where you perform administrative operations by entering commands at the `PCA-ADMIN>` prompt.

To terminate your CLI session, enter the `exit` command.

Command Syntax

In general, commands entered in the Service CLI have the following syntax:

```
PCA-ADMIN> command objectType <attributes> [options]
```

where:

- **`command`** is the command type to be initiated, for example: `list` or `create`.
- **`objectType`** is the target component or process affected by the command, for example: `list ComputeNode` or `create Tenant`.
- **`attributes`** are properties used to identify a specific object of the selected type to which the command must be applied, for example: `show ManagementNode` ***`name=pcamn02`***.
- **`options`** are additional parameters that may be provided to affect the behavior of the command.

  For example, you can add sorting and filtering options to the `list` command and select which data columns (fields) to display: `list RackUnit` ***`fields`*** `ipAddress,name,rackElevation,serialNumber,firmwareVersion` ***`where`*** `state` ***`eq`*** `running`.

The main elements of a command are separated by a space. Attributes are specified as "`type=value`". Lists are entered as a comma-separated series of values (such as `fields` *`ipAddress,name,rackElevation,serialNumber,firmwareVersion`*).

Help and Command Completion

The Service CLI includes a `help` command. It shows how the most common types of commands are used, which helps you get familiar with the basics of the CLI.

```
PCA-ADMIN> help
For Most Object Types:
    create <objectType> [(attribute1)="value1"] ... [on <objectType>
 <instance>]
    delete <objectType> <instance>
    edit <objectType> <instance>  (attribute1)="value1" ...
```

```
    list <objectType> [fields (attribute1,attribute2)]where
 [(filterableAttribute1)  \
        <filterComparator> "value1" [AND|OR] [(filterableAttribute2)
 <filterComparator> "value2"
    show <objectType> <instance>
For Most Object Types with Children:
    add <objectType> <instance> to <objectType> <instance>
    remove <objectType> <instance> from <objectType> <instance>
Other Commands:
    exit
    showallcustomcmds
    showcustomcmds <objectType>
    showobjtypes
```

The easiest way to learn which commands and object types are available, is to use the question mark ("?"). After signing in, you start by entering "?" at the CLI prompt, in order to display the set of base commands.

```
PCA-ADMIN> ?
           add
           clear
           count
           create
           delete
           edit
           [...]
```

You can drill down into the commands, object types and other elements by adding the "?" to see the available parameters at that cursor position.

> **Note:**
>
> Mind the position of the question mark: it's separated from the command by a space. If you omit the space, the CLI displays the parameters allowed at the level of that command, instead of the parameters that might follow *after* the command.

For example, if you want to see which object types you can list, type `list ?` and press Enter.

Next, assume that you want to find compute nodes that have not yet been provisioned. To achieve this, you can display a list of compute nodes filtered by their provisioning state. The "?" allows you to navigate through the command parameters, as shown below.

Each time you type "?" the CLI displays the parameters you can use at the cursor position. Press the Up arrow key to bring back the part of the command you already typed at the prompt, then add the next part of your command, and type "?" again to display the next set of parameters.

When your command is complete, press Enter.

```
PCA-ADMIN> list ?
                AuthorizationGroup
                ComputeNode
                Event
                Fault
                [...]
PCA-ADMIN> list ComputeNode ?
                            fields
                            limit
                            orderby
                            where
PCA-ADMIN> list ComputeNode where ?
                                  id
                                  provisioningState
                                  provisioningStateLastChangedTime
```

```
                                  provisioningType
                                  faultDomain
                                  [...]
PCA-ADMIN> list ComputeNode where provisioningState ?
                                                      EQ
                                                      NE
                                                      LIKE
                                                      [...]
PCA-ADMIN> list ComputeNode where provisioningState EQ ?
                                                         READYTOPROVISION
                                                         PROVISIONED
PCA-ADMIN> list ComputeNode where provisioningState EQ READYTOPROVISION
Command: list ComputeNode where provisioningState EQ READYTOPROVISION
Status: Success
Time: 2021-06-25 14:04:16,837 UTC
Data:
  id                                     name       provisioningState
  --                                     ----       -----------------
  bb940637-9825-4f7c-a5f4-1b49bcf6a5c9   pcacn005   Ready To Provision
  76df44a9-6980-4242-a3a2-e1614b3d44d1   pcacn008   Ready To Provision
  8fc0d06f-c64a-40ea-8a20-89680f03eb5e   pcacn011   Ready To Provision
```

The Service CLI also provides a form of tab completion. When you start to type a command and press the Tab key, the CLI autocompletes the part it can predict. If more than one possible value remains, you should add at least one more letter and press the Tab key again. The following examples illustrate how the CLI performs tab completion.

- Tab completion with one possible match

```
PCA-ADMIN> list Com<Tab>
PCA-ADMIN> list ComputeNode
```

- Tab completion with more than one possible match

```
PCA-ADMIN> list Ra<Tab>
PCA-ADMIN> list Rack

PCA-ADMIN> list RackU<Tab>
PCA-ADMIN> list RackUnit
```

Base and Custom Commands

When you enter the `help` command or type the question mark (`"?"`) at the `PCA-ADMIN>` prompt, the CLI returns information about its base commands, such as `create`, `edit`, `add`, `remove`, `delete`, `list`, `show`, and so on. However, there is another set of less commonly used *custom commands*. You can display them all as a single list, or only those available for a particular object type. You can use the `"?"` to navigate through the commands.

```
PCA-ADMIN> showallcustomcmds
Operation Name: <Related Object(s)>
    ----------------------------------
    asrClientDisable:  ASRPhonehome
    asrClientEnable:  ASRPhonehome
    asrClientRegister:  ASRPhonehome
    [...]
    changeIlomPassword:  ComputeNode, ManagementNode
    changePassword:  ComputeNode, LeafSwitch, ManagementNode,
 ManagementSwitch, SpineSwitch, User, ZFSAppliance
    clearFirstBootError:  NetworkConfig
    configZFSAdDomain:  ZfsAdDomain
    configZFSAdWorkgroup:  ZfsAdDomain
    createAdminAccount:
    createUserInGroup:  User
    deletePlatformImage:  PlatformImage
```

```
    deprovision:  ComputeNode
    disableVmHighAvailability:  PcaSystem
    drAddComputeInstance:  ComputeInstance
    drAddSiteMapping:  DrSiteMapping
    drConfigCleanupPrimary:  DrConfig
    [...]
    maintenanceLock:  ComputeNode
    maintenanceUnlock:  ComputeNode
    migrateVm:  ComputeNode
    patchCN:  PatchRequest
    patchEtcd:  PatchRequest
    patchHost:  PatchRequest
    patchIlom:  PatchRequest
    patchKubernetes:  PatchRequest
    patchMySQL:  PatchRequest
    patchOCIImages:  PatchRequest
    patchPlatform:  PatchRequest
    patchSwitch:  PatchRequest
    patchVault:  PatchRequest
    patchZfssa:  PatchRequest
    [...]
    start:  CnUpdateManager, ComputeNode, Day0NetworkConfigManager,
 FaultManager, PurgeManager, ZfsPoolManager
    stop:  CnUpdateManager, ComputeNode, Day0NetworkConfigManager,
 FaultManager, PurgeManager, ZfsPoolManager
    syncHardwareData:
    syncUpstreamUlnMirror:  PatchRequest
    systemStateLock:  PcaSystem
    systemStateUnlock:  PcaSystem
    updateSauronCredentials:
    upgradeCN:  UpgradeRequest
    upgradeEtcd:  UpgradeRequest
    upgradeFullMN:  UpgradeRequest
    upgradeHost:  UpgradeRequest
    upgradeIlom:  UpgradeRequest
    upgradeKubernetes:  UpgradeRequest
    upgradeMySQL:  UpgradeRequest
    upgradePlatform:  UpgradeRequest
    upgradePreConfig:  UpgradeRequest
    upgradeSwitch:  UpgradeRequest
    upgradeVault:  UpgradeRequest
    upgradeZfssa:  UpgradeRequest
```

```
PCA-ADMIN> showcustomcmds ?
    ASRBundle
    ASRPhonehome
    BackupJob
    CnUpdateManager
    ComputeInstance
    ComputeNode
    Day0NetworkConfigManager
    DrConfig
    DrJob
    DrSiteMapping
    Event
    ExadataNetwork
    FaultDomainInfo
    FaultManager
    Job
    LeafSwitch
    ManagementNode
    ManagementSwitch
    NetworkConfig
```

```
    PatchRequest
    PcaSystem
    PlatformImage
    PurgeManager
    SpineSwitch
    UpgradeJob
    UpgradeJobList
    UpgradeRequest
    User
    Vcn
    ZfsAdDomain
    ZFSAppliance
    ZfsPoolManager
```

```
PCA-ADMIN> showcustomcmds ComputeNode
    provisioningLock
    provisioningUnlock
    maintenanceLock
    maintenanceUnlock
    provision
    deprovision
    migrateVm
    reset
    start
    stop
    changePassword
    changeIlomPassword
    getRunningInstances
    getRunningInstancesCount
```

Changing Passwords for Hardware Components

You can change the password for compute nodes and management nodes, switches, ILOMs, or the ZFS Storage Appliance. These passwords are managed from the Service CLI.

> **Important:**
>
> The following password rules apply:
>
> - Passwords for compute nodes, leaf switches, management nodes, management switches, or spine switches must contain at least 8 but no more than 20 characters.
> - Passwords for ZFS Storage Appliance or ILOMs must contain at least 8 but no more than 16 characters.
> - All passwords must contain at least 1 uppercase letter (A-Z), 1 lowercase letter (a-z), 1 digit (0-9), and 1 of the following symbols: @$!%*#&.
> - Enclose the password in double quotes "*password*" when entering it as part of the CLI command.

Listing Components with Configurable Passwords

To view the components for which you can change passwords, use the `changepassword ?` or the `changeilompassword ?` command.

```
PCA-ADMIN> changepassword ?
ComputeNode
LeafSwitch
ManagementNode
ManagementSwitch
SpineSwitch
```

```
ZFSAppliance
```

```
PCA-ADMIN> changeilomPassword ?
ComputeNode
ManagementNode
```

Changing Component Passwords

To change the password for a hardware component, use the `changepassword` command.

Syntax (entered on a single line):

```
changepassword
ComputeNode|LeafSwitch|ManagementNode|ManagementSwitch|SpineSwitch|
ZFSAppliance
id=<component_id> OR name=<component_name>
password="<new_password>" confirmPassword="<repeat_new_password>"
```

Example:

```
PCA-ADMIN> changePassword id=21ad5b60-d30d-4a95-b39f-5bf152005f0f
password=************* confirmPassword=*************
Status: Success
JobId: fe772781-d0af-47cc-af87-2059f8e70b63
```

Changing Component ILOM Passwords

To change the ILOM password for a compute node or management node, use the `changeilompassword` command.

Syntax (entered on a single line):

```
changeilompassword ComputeNode|ManagementNode
id=<component_id> OR name=<component_name>
password="<new_password>" confirmPassword="<repeat_new_password>"
```

Example:

```
PCA-ADMIN> changeilomPassword id=21ad5b60-d30d-4a95-b39f-5bf152005f0f
password="*************" confirmPassword="*************"
Status: Success
JobId: fe772781-d0af-47cc-af87-2059f8e70b63
```

Performing Administrative Operations on Compute Nodes

The Private Cloud Appliance compute nodes are the core components for hosting cloud workloads, and require careful management in particular planned or unplanned circumstances. An administrator must perform operations described in this section when compute nodes are provisioned, when maintenance or upgrades take place, when an outage occurs, and so on.

The Rack Units list of the Service Web UI, provides access to extra functions and details through the Actions menu: the button with three vertical dots on the right hand side of each table row. In practice, only the View Details and Copy ID operations are available for all component types. For compute nodes, however, several more options are available in the Actions menu and the compute node detail page. All operations can also be performed from the Service CLI.

> **Note:**
>
> When compute nodes are in the discovery state or coming up, their status is 'Failed' until the hardware process transitions them to 'Ready to Provision'. This process typically takes under five minutes. If the failed state persists,

use the Service CLI command `list ComputeNode` to determine the provisioning state of the compute nodes and take appropriate action.

See these topics for detailed instructions:

- Compute Node Provisioning and Life Cycle State on page 157
- Compute Node Maintenance Operations on page 160

Compute Node Provisioning and Life Cycle State

The compute nodes must be provisioned to contribute to the compute capacity of Private Cloud Appliance, and host compute instances. Provisioning is an administrator task performed from the Service Enclave, which is also used to deprovision a node, and to control the life cycle state.

Provisioning a Compute Node

The appliance software detects the compute nodes that are installed in the rack and cabled to the switches, meaning they appear in the Rack Units list as *Ready to Provision*. You can provision them from the Service Web UI or Service CLI.

Using the Service Web UI

1. In the navigation menu, click Rack Units.
2. In the Rack Units table, click the host name of the compute node you want to provision.

   The compute node detail page appears.
3. In the top-right corner of the page, click Controls and select the Provision command.

Using the Service CLI

1. Display the list of compute nodes.

   Copy the ID of the compute node you want to provision.

```
PCA-ADMIN> list ComputeNode
Data:
  id                                     name
 provisioningState   provisioningType
  --                                     ----
 -----------------   ----------------
  29f68a0e-4744-4a92-9545-7c48fa365d0a   pcacn001   Ready to
 Provision  Unspecified
  7a0236f4-b00e-461d-93a0-b22673a18d9c   pcacn003   Ready to
 Provision  Unspecified
  dc8ae567-b07f-48e0-89bd-e57069c20010   pcacn002   Ready to
 Provision  Unspecified
```

2. Provision the compute node with this command:

```
PCA-ADMIN> provision id=7a0236f4-b00e-461d-93a0-b22673a18d9c
JobId: ea93cac4-4430-4663-aafd-d70701593fb2
```

   Use the job ID to check the status of your provision command.

```
PCA-ADMIN> show Job id=ea93cac4-4430-4663-aafd-d70701593fb2
[...]
  Done = true
  Name = MODIFY_TYPE
  Run State = Succeeded
```

3. Repeat the provision command for any other compute nodes you want to provision at this time.
4. Confirm that the compute nodes have been provisioned.

```
PCA-ADMIN> list ComputeNode
```

```
Data:
  id                                     name
 provisioningState   provisioningType
  --                                     ----
 -----------------   ----------------
  29f68a0e-4744-4a92-9545-7c48fa365d0a   pcacn001   Provisioned
    KVM
  7a0236f4-b00e-461d-93a0-b22673a18d9c   pcacn003   Provisioned
    KVM
  dc8ae567-b07f-48e0-89bd-e57069c20010   pcacn002   Provisioned
    KVM
```

Starting, Resetting, or Stopping a Compute Node

Using the Service Web UI

1. Ensure that the compute node is locked for maintenance.

   See Locking a Compute Node for Maintenance on page 164.
2. In the navigation menu, click Rack Units.
3. In the Rack Units table, locate the compute node you want to start, reset or stop.
4. Click the Action menu (three vertical dots) and select the appropriate action: Start, Reset, or Stop.
5. When the confirmation window appears, click the appropriate action button to proceed.

   A pop-up window appears for a few seconds to confirm that the compute node is starting, stopping, or restarting.
6. When the compute node is up and running again, release the maintenance and provisioning locks.

Using the Service CLI

1. Display the list of compute nodes.

   Copy the ID of the compute node that you want to start, reset or stop.

   ```
   PCA-ADMIN> list ComputeNode
   Data:
     id                                     name
    provisioningState   provisioningType
     --                                     ----
    -----------------   ----------------
     3e62bf25-a26c-407e-ab8b-df01a4ad98b6   pcacn002   Provisioned
       KVM
     f7b8356b-052f-4911-babb-447e6ab9c78d   pcacn003   Provisioned
       KVM
     4e06ebdf-faed-484e-996d-d77af786f123   pcacn001   Provisioned
       KVM
   ```
2. Ensure that the compute node is locked for maintenance.

   See Locking a Compute Node for Maintenance on page 164.
3. Start, reset or stop the compute node using the corresponding command:

   ```
   PCA-ADMIN> start ComputeNode id=f7b8356b-052f-4911-babb-447e6ab9c78d
   PCA-ADMIN> reset id=f7b8356b-052f-4911-babb-447e6ab9c78d
   PCA-ADMIN> stop ComputeNode id=f7b8356b-052f-4911-babb-447e6ab9c78d
   ```
4. When the compute node is up and running again, release the maintenance and provisioning locks.

Deprovisioning a Compute Node

If you need to take a compute node out of service, for example to replace a defective one, you must deprovision it first, so that its data is removed cleanly from the system databases.

Using the Service Web UI

1. In the navigation menu, click Rack Units.
2. In the Rack Units table, click the host name of the compute node you want to deprovision.

   The compute node detail page appears.
3. In the top-right corner of the page, click Controls and select the Provisioning Lock command.

   When the confirmation window appears, click Lock to proceed.

   After successful completion, the Compute Node Information tab shows Provisioning Locked = Yes.
4. Ensure that no more compute instances are running on the compute node.

   Click Controls and select the Migrate All Vms command. The system migrates the instances to other compute nodes.
5. To deprovision the compute node, click Controls and select the Deprovision command.

   When the confirmation window appears, click Deprovision to proceed.

   After successful completion, the Compute Node Information tab shows Provisioning State = Ready to Provision.

Using the Service CLI

1. Display the list of compute nodes.

   Copy the ID of the compute node you want to deprovision.

```
PCA-ADMIN> list ComputeNode
Data:
  id                                     name
 provisioningState   provisioningType
  --                                     ----
 -----------------   ----------------
  29f68a0e-4744-4a92-9545-7c48fa365d0a   pcacn001   Provisioned
    KVM
  7a0236f4-b00e-461d-93a0-b22673a18d9c   pcacn003   Provisioned
    KVM
  dc8ae567-b07f-48e0-89bd-e57069c20010   pcacn002   Provisioned
    KVM
```

2. Set a provisioning lock on the compute node.

```
PCA-ADMIN> provisioningLock id=7a0236f4-b00e-461d-93a0-b22673a18d9c
JobId: ed4a4646-6d73-41f9-9cb0-73ea35e0d766
```

   Use the job ID to check the status of your command.

```
PCA-ADMIN> show Job id=ed4a4646-6d73-41f9-9cb0-73ea35e0d766
[...]
  Done = true
  Name = MODIFY_TYPE
  Run State = Succeeded
```

3. Confirm that the compute node is under provisioning lock.

```
PCA-ADMIN> show ComputeNode id=7a0236f4-b00e-461d-93a0-b22673a18d9c
[...]
  Provisioning Locked = true
```

4. Migrate all running compute instances off the compute node you want to deprovision.

```
PCA-ADMIN> migrateVm id=7a0236f4-b00e-461d-93a0-b22673a18d9c
JobId: 6f1e94bc-7d5b-4002-ada9-7d4b504a2599
```

Use the job ID to check the status of your command.

```
PCA-ADMIN> show Job id=6f1e94bc-7d5b-4002-ada9-7d4b504a2599
Data:
[...]
  Done = true
  Name = MODIFY_TYPE
  Run State = Succeeded
```

5. Deprovision the compute node with this command:

```
PCA-ADMIN> deprovision id=7a0236f4-b00e-461d-93a0-b22673a18d9c
JobId: 9868fdac-ddb6-4260-9ce1-c018cf2ddc8d
```

Use the job ID to check the status of your deprovision command.

```
PCA-ADMIN> show Job id=9868fdac-ddb6-4260-9ce1-c018cf2ddc8d
[...]
  Done = true
  Name = MODIFY_TYPE
  Run State = Succeeded
```

6. Confirm that the compute node has been deprovisioned.

```
PCA-ADMIN> list ComputeNode
Data:
  id                                     name
 provisioningState   provisioningType
  --                                     ----
 -----------------   ----------------
  29f68a0e-4744-4a92-9545-7c48fa365d0a   pcacn001   Provisioned
    KVM
  7a0236f4-b00e-461d-93a0-b22673a18d9c   pcacn003   Ready to
 Provision  Unspecified
  dc8ae567-b07f-48e0-89bd-e57069c20010   pcacn002   Provisioned
    KVM
```

Compute Node Maintenance Operations

For maintenance operations, including Private Cloud Appliance software upgrade or patching, and when hardware repair is needed, compute nodes must be placed in maintenance mode. This requires evacuation of running compute instances, and locking the node out of any other system operations.

Evacuating a Compute Node

Some compute node operations can only be performed if the compute node has no running compute instances. Administrators can migrate all running instances away from a compute node, also known as evacuating the compute node.

By default, if enough resources are available, running instances are live migrated to other compute nodes in the same fault domain.

> **Important:**
>
> Before you perform a compute node evacuation, check what the behavior will be for any instances that cannot be live migrated to another compute node in the same fault domain.
>
> This topic, and High Availability Configuration for Compute Instances on page 168, describe how to check settings and how instances are handled for different settings.

> Live migration between different types of compute nodes is not supported. For example, you can't migrate compute instances from an Oracle Server X10 to an Oracle Server X11.

Check whether strict fault domain enforcement is set.

- When strict fault domain enforcement is disabled (Strict FD is set to Disabled in the Service Web UI or `Strict FD Enabled` is `false` in the Service CLI), instances that cannot be live migrated to another compute node in the same fault domain are live migrated to a different fault domain if enough resources are available in that fault domain.
- When strict fault domain enforcement is enabled (Strict FD is set to Enabled in the Service Web UI or `Strict FD Enabled` is `true` in the Service CLI), instances that cannot be live migrated to another compute node in the same fault domain do not migrate; those instances are still running in the compute node that you want to evacuate.

Enable or disable strict fault domain enforcement to set whether instances that cannot live migrate to other compute nodes in the same fault domain will be live migrated to a different fault domain or still running in the same compute node after you attempt to evacuate the compute node.

If some instances cannot be live migrated, either because the current fault domain is not able to accommodate them and strict fault domain enforcement is enabled, or because strict fault domain enforcement is disabled but other fault domains also cannot accommodate the instances, then you can re-run the migrate operation with the force option specified. When the force option is specified, the Compute service soft stops any instances that fail to live migrate, allowing the evacuation to proceed.

Restart stopped instances. If instances were stopped by the Compute service (not manually stopped by an administrator) and you want them to be automatically restored to running when resources become available, check that the Auto Recovery property of the Compute service is enabled and the instance availability recovery action is set to `RESTORE_INSTANCE`. See Viewing and Setting Compute Service Configuration on page 170 and Configuring the Recovery State for a Stopped Instance on page 173.

Instances can be stopped by the Compute service if the force option is used when an administrator evacuates a compute node, or in response to an unplanned compute node outage. You can change the Auto Recovery setting at any time before or after resources become available after an administrative maintenance or unplanned outage to restart instances that were stopped by the Compute service. If the instance availability recovery action is set to `STOP_INSTANCE`, the instance remains stopped even though the Auto Recovery property is enabled. If the instance availability recovery action is later changed to `RESTORE_INSTANCE`, a subsequent Auto Recovery pass will restart the instance.

Return relocated instances. If instances are live migrated to a different fault domain (displaced), and you want them returned to their selected fault domain (the fault domain that is specified in the instance configuration) when resources become available, check that the Auto Resolve property of the Compute service is enabled. See Viewing and Setting Compute Service Configuration on page 170 and Compute Service Configuration Commands on page 171. You can set the Auto Resolve property at any time before or after the compute node evacuation completes to relocate any displaced instances.

Use the following procedures to perform the migrate operation.

Compute Node Evacuation: Before You Begin

1. Check fault domain and compute node resources. See #unique_125. Based on this information, decide whether to do any of the following:
   - Terminate instances that are no longer needed.
   - Reconfigure some instances to use fewer resources. For example, specify a different shape.
   - Reconfigure some instances to specify a different fault domain.
   - Stop some instances while you perform the compute node evacuation.
   - Shutdown non migratable instances. See next step.
   - Specify the force option on the migration operation to soft stop any instances that cannot be live migrated. See the discussion above of instance availability recovery action and Auto Recovery configuration.

2. While it's possible to specify the `force` option on the `vmMigrate` operation to soft stop any instances that cannot be live migrated, the best practice is to gracefully shut down non migratable instances before migration so any workloads running on the instance will be in a good state.

   a. Display the list of non migratable instances.

      Copy the ID of the running instances, so you can shut them down.

      ```
      PCA-ADMIN> getNonMigratableInstances
      Data:
        id                              Display Name  Compute Node Id
       Domain State
        --                              ------------  ---------------
       ------------
        ocid1.instance.unique_ID     instance202   CN_ID
       running
        ocid1.instance.unique_ID     kqh027        CN_ID            shut
       off
      ```

   b. Shut down the running instances.

      See Stopping, Starting, and Resetting an Instance on page 661.

3. Disable provisioning on the compute node.

   See Disabling Compute Node Provisioning on page 163.

Using the Service Web UI

1. In the navigation menu, click Rack Units.
2. In the Rack Units table, find the host name of the compute node that you want to evacuate. Click the Actions menu for that host, and click the Migrate All Vms option.

   Alternatively, in the Rack Units table, click the host name of the compute node that you want to evacuate to display the details page for that compute node. Click the Controls menu, and click the Migrate All Vms option.
3. On the Confirm Migrating VMs dialog, choose whether to force stop any instances that cannot be migrated.

   By default, the force stop option is not enabled, and instances that cannot be migrated will still be running on the node after the migrate operation completes. To force stop instances that cannot be migrated, enable the force stop option in the Confirm Migrating VMs dialog.
4. On the Confirm Migrating VMs dialog, click the Migrate button.

   The Compute service live migrates the running instances to other compute nodes if enough resources are available and High Availability settings are configured to allow it. If the Force option was specified, any instances that could not be migrated are soft stopped. If any instances could not be migrated and Force was not specified, those instances remain running in the compute node that you are attempting to evacuate.

Using the Service CLI

1. Display the list of compute nodes.

   Copy the ID of the compute node that you that you want to evacuate.

   ```
   PCA-ADMIN> list ComputeNode
   Data:
     id                                     name
    provisioningState   provisioningType
     --                                     ----
    -----------------   ----------------
     3e62bf25-a26c-407e-ab8b-df01a4ad98b6   pcacn002   Provisioned
       KVM
   ```

```
  f7b8356b-052f-4911-babb-447e6ab9c78d   pcacn003   Provisioned
    KVM
  4e06ebdf-faed-484e-996d-d77af786f123   pcacn001   Provisioned
    KVM
```

2. Use the `migrateVm` command to live migrate all running compute instances off the compute node. To soft stop any instances that fail to migrate, set the `force` option:

```
PCA-ADMIN> migrateVm id=7a0236f4-b00e-461d-93a0-b22673a18d9c
 force=true
JobId: 6f1e94bc-7d5b-4002-ada9-7d4b504a2599
```

The Compute service live migrates the running instances to other compute nodes if enough resources are available and High Availability settings are configured to allow it. If `force=true` was specified, any instances that could not be migrated are soft stopped. If any instances could not be migrated and `force=true` was not specified, those instances remain running in the compute node that you are attempting to evacuate.

Use the job ID to check the status of the `migrateVm` command.

```
PCA-ADMIN> show Job id=6f1e94bc-7d5b-4002-ada9-7d4b504a2599
[...]
  Done = true
  Name = MODIFY_TYPE
  Run State = Succeeded
```

Disabling Compute Node Provisioning

Several compute node operations can only be performed on condition that provisioning has been disabled. Follow these instructions to impose and release a provisioning lock.

Using the Service Web UI

1. In the navigation menu, click Rack Units.
2. In the Rack Units table, click the host name of the compute node you want to make changes to.

   The compute node detail page appears.
3. In the top-right corner of the page, click Controls and select the Provisioning Lock command.

   When the confirmation window appears, click Lock to proceed.

   After successful completion, the Compute Node Information tab shows Provisioning Locked = Yes.
4. To release the provisioning lock, click Controls and select the Provisioning Unlock command.

   When the confirmation window appears, click Unlock to proceed.

   After successful completion, the Compute Node Information tab shows Provisioning Locked = No.

Using the Service CLI

1. Display the list of compute nodes.

   Copy the ID of the compute node for which you want to disable provisioning operations.

```
PCA-ADMIN> list ComputeNode
Data:
  id                                     name
 provisioningState   provisioningType
  --                                     ----
 -----------------   ---------------
  3e62bf25-a26c-407e-ab8b-df01a4ad98b6   pcacn002   Provisioned
    KVM
  f7b8356b-052f-4911-babb-447e6ab9c78d   pcacn003   Provisioned
    KVM
```

```
  4e06ebdf-faed-484e-996d-d77af786f123   pcacn001   Provisioned
    KVM
```

2. Set a provisioning lock on the compute node.

```
PCA-ADMIN> provisioningLock id=f7b8356b-052f-4911-babb-447e6ab9c78d
JobId: 6ee78c8a-e227-4d31-a770-9b9c96085f3f
```

   Use the job ID to check the status of your command.

```
PCA-ADMIN> show Job id=6ee78c8a-e227-4d31-a770-9b9c96085f3f
[...]
  Done = true
  Name = MODIFY_TYPE
  Run State = Succeeded
```

3. When the job has completed, confirm that the compute node is under provisioning lock.

```
PCA-ADMIN> show ComputeNode id=f7b8356b-052f-4911-babb-447e6ab9c78d
[...]
  Provisioning State = Provisioned
  [...]
  Provisioning Locked = true
  Maintenance Locked = false
```

   All provisioning operations are now disabled until the lock is released.

4. To release the provisioning lock, use this command:

```
PCA-ADMIN> provisioningUnlock id=f7b8356b-052f-4911-
babb-447e6ab9c78d
JobId: 523892e8-c2d4-403c-9620-2f3e94015b46
```

   Use the job ID to check the status of your command.

```
PCA-ADMIN> show Job id=523892e8-c2d4-403c-9620-2f3e94015b46
[...]
  Done = true
  Name = MODIFY_TYPE
  Run State = Succeeded
```

5. When the job has completed, confirm that the provisioning lock has been released.

```
PCA-ADMIN> show ComputeNode id=f7b8356b-052f-4911-babb-447e6ab9c78d
[...]
  Provisioning State = Provisioned
  [...]
  Provisioning Locked = false
  Maintenance Locked = false
```

Locking a Compute Node for Maintenance

For maintenance operations, compute nodes must be placed in maintenance mode. Follow these instructions to impose and release a maintenance lock. Before you can lock a compute node for maintenance, you must disable provisioning first. Maintenance operations can only be performed if the compute node has no running compute instances.

> **Caution:**
>
> Depending on the high-availability configuration of the Compute service, automatic instance migrations can prevent you from successfully locking a compute node. See High Availability Configuration for Compute Instances

on page 168. This situation is more likely to occur when available compute capacity is limited.

- Instance recovery or migration operations after a compute node outage can cause a maintenance lock to fail. Compute nodes involved in instance migrations will reject the maintenance lock until the migrations are complete.
- Displaced instances could be migrated back to their original fault domain when a compute node maintenance lock is released. A compute node from where a displaced instance is migrated back will reject the maintenance lock until the migration is complete.
- Migrating an instance typically takes no more than 30 seconds. However, large instances and heavy workloads increase the time required.
- In the event that an instance gets stuck in moving state and migration fails to complete, its host compute node cannot be locked for maintenance. Contact Oracle for assistance.

Using the Service Web UI

1. Ensure that provisioning has been disabled on the compute node.
2. Ensure that the compute node has no active instances. They must be migrated or shut down.
3. In the navigation menu, click Rack Units.
4. In the Rack Units table, click the host name of the compute node that requires maintenance.

   The compute node detail page appears.
5. In the top-right corner of the page, click Controls and select the Maintenance Lock command.

   When the confirmation window appears, click Lock to proceed.

   After successful completion, the Compute Node Information tab shows Maintenance Locked = Yes.
6. To release the maintenance lock, click Controls and select the Maintenance Unlock command.

   When the confirmation window appears, click Unlock to proceed.

   After successful completion, the Compute Node Information tab shows Maintenance Locked = No.

Using the Service CLI

1. Display the list of compute nodes.

   Copy the ID of the compute node that requires maintenance.

```
PCA-ADMIN> list ComputeNode
Data:
  id                                     name
 provisioningState   provisioningType
  --                                     ----
 -----------------   ----------------
  3e62bf25-a26c-407e-ab8b-df01a4ad98b6   pcacn002   Provisioned
    KVM
  f7b8356b-052f-4911-babb-447e6ab9c78d   pcacn003   Provisioned
    KVM
  4e06ebdf-faed-484e-996d-d77af786f123   pcacn001   Provisioned
    KVM
```

2. Ensure that provisioning has been disabled on the compute node.
3. Lock the compute node for maintenance.

```
PCA-ADMIN> maintenanceLock id=f7b8356b-052f-4911-babb-447e6ab9c78d
JobId: e46f6603-2af2-4df4-a0db-b15156491f88
```

Use the job ID to check the status of your command.

```
PCA-ADMIN> show Job id=e46f6603-2af2-4df4-a0db-b15156491f88
[...]
  Done = true
  Name = MODIFY_TYPE
  Run State = Succeeded
```

4. When the job has completed, confirm that the compute node has been locked for maintenance.

```
PCA-ADMIN> show ComputeNode id=f7b8356b-052f-4911-babb-447e6ab9c78d
[...]
  Provisioning State = Provisioned
  [...]
  Provisioning Locked = true
  Maintenance Locked = true
```

The compute node is now ready for maintenance.

5. To release the maintenance lock, use this command:

```
PCA-ADMIN> maintenanceUnlock id=f7b8356b-052f-4911-babb-447e6ab9c78d
JobId: 625af20e-4b49-4201-879f-41d4405314c7
```

Use the job ID to check the status of your command.

```
PCA-ADMIN> show Job id=625af20e-4b49-4201-879f-41d4405314c7
[...]
  Done = true
  Name = MODIFY_TYPE
  Run State = Succeeded
```

6. When the job has completed, confirm that the provisioning lock has been released.

```
PCA-ADMIN> show ComputeNode id=f7b8356b-052f-4911-babb-447e6ab9c78d
[...]
  Provisioning State = Provisioned
  [...]
  Provisioning Locked = true
  Maintenance Locked = false
```

Integrating GPU Expansion Nodes

The GPU nodes must be discovered and provisioned before their hardware resources are available for use within Private Cloud Appliance compute instances. Unlike standard compute nodes, which are added to the base rack and automatically integrated and prepared for provisioning, GPU nodes go through a more strictly controlled process.

GPU nodes are installed in an expansion rack. Its networking components must be connected to the base rack so the new hardware can be integrated into the hardware administration and data networks. In this section we assume the GPU expansion rack has been installed and connected to the Private Cloud Appliance base rack. For installation requirements, physical hardware installation information, and cabling details, see Expanding Private Cloud Appliance with GPU Capacity on page 122.

The GPU expansion rack is activated by running a script from one of the management nodes. With precise timing and orchestration based on a static mapping, this script powers on and configures each component in the GPU expansion rack. The required ports on the switches are enabled so that all hardware can be discovered and registered in the component database. When the scripted operations are completed, the data and management networks are operational across the interconnected racks. The operating system and additional software are installed on the new nodes, after which they are ready to provision.

Installation and activation of the expansion rack and GPU nodes are performed by Oracle. From this point forward, the system treats GPU nodes the same way as all other compute nodes. After provisioning, appliance administrators

can manage and monitor them from the Service Enclave UI or CLI. See Performing Administrative Operations on Compute Nodes on page 156.

> **Note:**
>
> Live migration is not supported for GPU instances. This impacts some compute node operations.
>
> - Evacuating a GPU node will fail. Instances must be stopped manually.
> - The high availability configuration of the Compute Service applies to GPU instances, but is further restricted by limited hardware resources.
>
> When a GPU node goes offline and returns to normal operation, the Compute Service restarts instances that were stopped during the outage. An instance might be restarted, through cold migration, on another GPU node with enough hardware resources are available.

> **Caution:**
>
> For planned maintenance or upgrade, best practice is to issue a shut down command from the instance OS, then gracefully stop the instance from the Compute Web UI or OCI CLI.

GPU nodes are added to the 3 existing fault domains, which is consistent with the overall Oracle cloud architecture. The fault domains might become unbalanced because, unlike standard compute nodes, GPU nodes can be added one at a time. This has no functional impact on the fault domains because the server families operate separately from each other. The GPU nodes can only host compute instances based on a GPU shape, and migrations between different server families in the same fault domain are not supported.

In the Compute Enclave, consuming resources provided by a GPU node is straightforward. Users deploy compute instances with a dedicated shape to allocate 1-4 GPUs. Instances based on a GPU shape always run on a GPU node.

Integrating a Compute Expansion Rack

Private Cloud Appliance compute capacity can be increased by adding a compute expansion rack. Its compute nodes must be discovered and provisioned before their hardware resources are available for use within compute instances.

The compute expansion rack networking components must be connected to the base rack so the new hardware can be integrated into the hardware administration and data networks. In this section we assume the compute expansion rack has been installed and connected to the Private Cloud Appliance base rack. For installation requirements, physical hardware installation information, and cabling details, see Expanding Private Cloud Appliance Capacity with a Compute Expansion Rack on page 123.

The compute expansion rack is activated by running a script from one of the management nodes. With precise timing and orchestration based on a static mapping, this script powers on and configures each component in the compute expansion rack. The required ports on the switches are enabled so that all hardware can be discovered and registered in the component database. When the scripted operations are completed, the data and management networks are operational across the interconnected racks. The operating system and additional software are installed on the new nodes, after which they are ready to provision.

Installation and activation of the expansion rack and expansion compute nodes are performed by Oracle. From this point forward, the system treats the expansion compute nodes the same way as all other compute nodes. After provisioning, appliance administrators can manage and monitor them from the Service Enclave UI or CLI. See Performing Administrative Operations on Compute Nodes on page 156.

> **Caution:**
>
> For planned maintenance or upgrade, best practice is to issue a shut down command from the instance OS, then gracefully stop the instance from the Compute Web UI or OCI CLI.

High Availability Configuration for Compute Instances

For compute instances, high availability means automated recovery in case the underlying infrastructure fails, or a component is brought down for maintenance. The state of the compute nodes, hypervisors, and compute instances is monitored continually.

High availability (HA) of compute instances is configurable. The behavior described in this section is based on standard settings. For information about configurable HA settings – such as reboot migration, fault domain placement, and automatic recovery – see Configuring High Availability in the Compute Service on page 169.

By default, the system attempts to live-migrate or restart instances in their selected fault domain, but it can also restart instances in other fault domains if insufficient resources are available in the selected fault domain. The selected fault domain is the one specified in the instance configuration.

Compute Node Outage

If a compute node goes down because of an unplanned reboot, its instances are restarted when the compute node successfully returns to normal operation. However, this behavior is configurable. At the next polling interval, by default if instances are found that should be running but are in a different state, the start command is issued again. If any instances have crashed and remain in that state, the hypervisor attempts to restart them up to 5 times. Instances that were not running before the compute node became unavailable, remain shut down when the compute node is up and running again.

If a compute node is lost because of a failure, by default the system attempts to live migrate running compute instances from the failed compute node to other compute nodes. Actual behavior depends on how you have configured the Compute service high availability parameters.

A compute node is considered failing when it has been disconnected from the data network or has been in powered-off state for about 5 minutes. This 5-minute timeout is the threshold for placing the compute node in `FAIL` state and its agent in `EVACUATING` state. This condition is required before the *reboot migration* can start.

Reboot Migration

Reboot migration implies that all compute instances from the failing compute node are stopped and restarted on another compute node. When migration is complete, the failing compute node's agent indicates that instances have been evacuated. If the compute node eventually reboots successfully, it must go through a cleanup process that removes all stale instance configurations and associated virtual disks. After cleanup, the compute node can host compute instances again.

During the entire reboot migration, the instances remain in "moving" configuration state. When migration is completed, the instance configuration state is changed to "running". Instances that were stopped before the failure are not migrated, because they are not associated with any compute node.

Fault Domain Settings

Fault domain preference is not strictly enforced with instance migration, meaning the Compute service by default can stop instances if their selected fault domain has insufficient resources, and restart them on a compute node in another fault domain. If strict fault domain enforcement is configured in the Compute service, instances that cannot be migrated to another compute node in the selected fault domain must be stopped.

If automatic fault domain resolution is enabled in the Compute service, instances that were migrated to a different fault domain can be migrated back to their selected fault domain.

Planned Maintenance

In case of planned maintenance, the administrator must first disable provisioning for the compute node in question, and apply a maintenance lock. When the compute node is under a provisioning lock, the administrator can live-migrate all running compute instances to another compute node. Maintenance mode can only be activated when there are no more running instances on the compute node. You can specify the force option to stop any instances that cannot be migrated. All compute instance operations on this compute node are disabled. A compute node in maintenance mode cannot be provisioned or deprovisioned.

Configuring High Availability in the Compute Service

Administrators can set parameters to control how the Compute service tries to keep instances available in response to planned or unplanned compute node outages. Compute service high availability settings affect all compute instances, and interact with individual instance recovery settings.

When planned maintenance needs to be performed, a compute node is evacuated. If possible, the Compute service live migrates all running instances to other compute nodes in the same fault domain. If this default scenario cannot be performed, the high availability (HA) parameters, at the level of the Compute service and the individual instance, determine further options to migrate, stop, and recover affected compute instances.

In the case of an unplanned compute node outage, the Compute service stops the instances, and if the outage persists, attempts to evacuate the compute node by restarting the affected instances on other compute nodes. This automated form of cold migration is called *reboot migration*.

Instance and Compute Service High Availability Configuration

The high availability (HA) configuration of the Compute service enables you to manage outcomes for different types of compute node outages. Instance availability recovery action is the only high availability configuration that is set for each instance. All other high availability configuration is set on the Compute service and affects all instances.

The *selected* fault domain is the fault domain that is specified in the instance configuration. A *displaced* instance is in a fault domain that is not its selected fault domain.

Planned Maintenance Outage

See Compute Node Maintenance Operations on page 160 for information about how to evacuate a compute node. If possible, the Compute service live migrates running instances to other compute nodes in the same fault domain. The section also describes how to use the instance availability recovery action (set on each instance), and the Auto Recovery and Auto Resolve properties of the Compute service when performing a compute node evacuation.

Unplanned Outages

The Compute service attempts to stop instances and reboot migrate the instances under the following compute node outage conditions:

- Power down from HW status
- Inability to reach the compute node data network

A compute node could experience an outage where the Compute service cannot migrate the instances. For example, if the Compute service cannot reach the compute node at all, then the Compute service cannot stop and reboot migrate the instances.

Unplanned Outage Shorter than Five Minutes

In an unplanned outage, the Compute service stops the affected instances. If the outage lasts less than five minutes, by default the Compute service attempts to restart instances that were running before the outage. Actual behavior depends on how the instances and the Compute service are configured. The following decision flow describes how you can control this behavior.

Do you want the Compute service to attempt to restart instances that were running prior to the outage? This is the default.

- Yes. Check that Auto Recovery is enabled and the instance availability recovery action is set to `RESTORE_INSTANCE`. See Configuring the Recovery State for a Stopped Instance on page 173.

  If some instances can no longer be accommodated in their selected fault domain, Auto Recovery will continue to poll and attempt to restart the instances. See also `getForcedStoppedInstances`.

  If the instance availability recovery action is set to `STOP_INSTANCE`, the instance will remain stopped, even if Auto Recovery is enabled.
- No. Disable Auto Recovery. Instances that had been running prior to the outage will remain stopped.

The instance availability recovery action setting and Auto Recovery setting can be changed at any time, and the changes will be effective at the next polling time.

Unplanned Outage Longer than Five Minutes

In an unplanned outage, the Compute service stops the affected instances. If the outage lasts more than five minutes, by default the Compute service attempts to reboot migrate (cold migrate) instances off the compute node. Instances that cannot be accommodated on other compute nodes in the same fault domain are reboot migrated to other fault domains. Actual behavior depends on how the Compute service is configured. The following decision flow describes how you can control this behavior.

Do you want running instances to be reboot migrated? Reboot migration is stopping and starting each running instance on a given compute node. See also High Availability Configuration for Compute Instances on page 168.

- Yes. Check that VM High Availability is enabled.

  If some instances cannot be accommodated on another compute node in the same fault domain, do you want those instances to be reboot migrated to a different fault domain?

  - Yes. Check that Strict FD is disabled. Instances that cannot be accommodated in any fault domain remain stopped by the Compute service.

    After reboot migration, do you want instances that are running in a fault domain that is not their selected fault domain to be automatically live migrated to their selected fault domain when resources become available?

    - Yes. Check that Auto Resolve is enabled. See also `getDisplacedInstances`.
    - No. Disable Auto Resolve.
  - No. Enable Strict FD. Instances that were running prior to the outage and cannot be migrated to another compute node in the current fault domain remain stopped by the Compute service.
- No. Disable VM High Availability. Instances that were running prior to the outage are stopped by the Compute service.

Do you want instances that were stopped by the Compute service to be automatically restored to running in their selected fault domain? If yes, check that Auto Recovery is enabled and the instance availability recovery action is set to `RESTORE_INSTANCE`. See Configuring the Recovery State for a Stopped Instance on page 173.

Viewing and Setting Compute Service Configuration

For information about how these configuration settings work, see Compute Service Configuration Commands on page 171.

Using the Service Web UI

On the navigation menu, click FD Instances and then click Compute Service Detail.

The Compute Service Information page shows the current settings for Auto Recovery, Auto Resolve Displaced Instances, VM High Availability, and Strict FD. All of these settings are enabled by default except for Strict FD, which is disabled by default. By default, fault domain placement is not strictly enforced when the Compute service migrates instances.

Use the Controls menu on the Compute Service Information page to change the values of these configuration settings between Enabled and Disabled.

Using the Service CLI

Use the `show computeservice` command to show the current Compute service configuration settings. In the following example, the default values are set for the four high availability configuration settings: `Auto Recovery Action Enabled`, `Auto-Resolve Displaced Instances Enabled`, `VM High Availability`

`Enabled`, and `Strict FD Enabled`. All of these settings are true by default except for `Strict FD Enabled`, which is false by default.

```
PCA-ADMIN> show computeservice
Data:
 Id = unique_ID
 Type = ComputeService
 total CN cpu usage percent = 23.3
 total CN memory usage percent = 16.2
 Auto Recovery Action Enabled = true
 Auto-Resolve Displaced Instances Enabled = true
 VM High Availability Enabled = true
 Strict FD Enabled = false
 Name = Compute Service
 Work State = Normal
```

To change these settings, use the commands in the following list. The `showcustomcmds computeservice` command lists all high availability configuration commands in the Compute service.

```
PCA-ADMIN> showcustomcmds computeservice
    enableAutoRecoveryAction
    disableAutoRecoveryAction
    enableAutoResolveDisplacedInstances
    disableAutoResolveDisplacedInstances
    enableVmHighAvailability
    disableVmHighAvailability
    enableStrictFD
    disableStrictFD
    getForcedStoppedInstances
    getDisplacedInstances
```

For example, to disable `Auto Recovery Action Enabled`, run the `disableAutoRecoveryAction` command. To enable strict fault domain enforcement, run the `enableStrictFD` command.

Compute Service Configuration Commands

The Service CLI commands for Compute service HA configuration are shown in the list that follows. To access the equivalent Service Web UI settings, click the navigation menu and click FD Instances. See Viewing and Setting Compute Service Configuration on page 170.

In these descriptions, the *selected* fault domain is the fault domain that is specified in the instance configuration. A *displaced* instance is in a fault domain that is not its selected fault domain.

`enableAutoRecoveryAction`

Enables the automatic restart of instances that were stopped by the Compute service. This is the default. If the instance availability recovery action is set to `RESTORE_INSTANCE`, this command causes instances that were stopped by the Compute service to be automatically restarted in their selected fault domain when resources are available. See also Configuring the Recovery State for a Stopped Instance on page 173 and `getForcedStoppedInstances`.

Instances could have been stopped by the Compute service for the following reasons:

- As a result of specifying the force option on a migrate all operation and some instances were not able to be migrated. See Compute Node Maintenance Operations on page 160.
- As a result of an unplanned compute node outage.

You can set this Auto Recovery property at any time before or after an administrative maintenance outage or an unplanned outage to restart instances that were stopped by the Compute service. If the instance availability recovery action is set to `STOP_INSTANCE`, the instance remains stopped even though

the Auto Recovery property is enabled. If the instance availability recovery action is later changed to `RESTORE_INSTANCE`, a subsequent Auto Recovery pass will restart the instance.

`disableAutoRecoveryAction`

Disables the automatic restart of stopped instances. Instances that were stopped by the Compute service are not automatically restarted when resources are available.

`enableAutoResolveDisplacedInstances`

Enables the return of running instances to their selected fault domain. This is the default. If instances were moved to a different fault domain (displaced) during compute node evacuation, this command enables those instances to be automatically live migrated to their selected fault domain once sufficient resources are available in that fault domain. See also `getDisplacedInstances`.

You can set this Auto Resolve configuration at any time before or after an outage to relocate any displaced instances. Instances that are stopped are not migrated.

`disableAutoResolveDisplacedInstances`

Disables the return of instances to their selected fault domain. Instances that were moved to a different fault domain during compute node evacuation remain in the fault domain to which they were moved.

`enableVmHighAvailability`

Enables High Availability (reboot migration) off of an unreachable compute node. This is the default.

`disableVmHighAvailability`

Disables reboot migration.

`enableStrictFD`

Enables strict fault domain enforcement. During compute node evacuation, any instance that cannot be moved to a different compute node in the same fault domain is stopped if the force option was specified. If the force option was not specified, the migrate operation fails.

`disableStrictFD`

Disables strict fault domain enforcement. This is the default. During compute node evacuation, any instance that cannot be moved to a different compute node in the same fault domain is moved to a different fault domain. This move to a different fault domain is temporary if the Auto Resolve property of the Compute service is enabled: If Auto Resolve is enabled, then when resources become available, the moved instances are live migrated back to their selected fault domain. See also `getDisplacedInstances`.

`getForcedStoppedInstances`

Lists all instances that were stopped via the use of the force option on the migrate operation or that were stopped by the Compute service in response to an unplanned outage.

```
PCA-ADMIN> getForcedStoppedInstances
Data:
 id                          displayName  compartmentId
 --                          -----------  -------------
 ocid1.instance.unique_ID  inst-name    ocid1.compartment.unique_ID
```

In the Service Web UI, click the navigation menu, click FD Instances, and then click Forced Stopped Instances. Use the Actions menu to copy the OCIDs.

`getDisplacedInstances`

Lists instances that are currently running in a fault domain that is not their selected fault domain. Instances that are not running are not shown.

In the following example, running instances are being migrated away from fault domain 1. One instance has been placed in fault domain 2 and one has been placed in fault domain 3.

```
PCA-ADMIN> getDisplacedInstances
Data:
 id                          displayName  compartmentId
 faultDomain     faultDomainSelected
 --                          -----------  -------------
 -----------     -------------------
 ocid1.instance.unique_ID  inst-name    ocid1.compartment.unique_ID
 FAULT-DOMAIN-3  FAULT-DOMAIN-1
 ocid1.instance.unique_ID  inst-name    ocid1.compartment.unique_ID
 FAULT-DOMAIN-2  FAULT-DOMAIN-1
```

In the Service Web UI, click the navigation menu, click FD Instances, and then click Displaced Instances. Use the Actions menu to copy the OCIDs.

Configuring the Recovery State for a Stopped Instance

If the Compute service stopped an instance, you can configure how that stopped instance will be treated when resources are again available by setting the instance availability recovery action and the Auto Recovery property of the Compute service.

See the description of the `enableAutoRecoveryAction` command in Compute Service Configuration Commands on page 171 for reasons that an instance can be stopped by the Compute service. See also the descriptions of `disableAutoRecoveryAction` and `getForcedStoppedInstances`.

During instance launch or in a subsequent instance update, set the instance recovery action in the instance availability configuration.

In the Compute Web UI, see the "Availability configuration" section in the dialog to create or edit an instance or create or edit an instance configuration. To restart instances that were stopped by the Compute service, check the box labeled "Restore instance lifecycle state after infrastructure maintenance". This is the default. To keep stopped instances stopped, uncheck the "Restore instance" box.

In the OCI CLI, use the `--availability-config` option or the `availabilityConfig` property in the compute instance `launch` or `update` command or the instance configuration `create` or `update` command. Set the `recoveryAction` to `RESTORE_INSTANCE` or `STOP_INSTANCE`. The default behavior is `RESTORE_INSTANCE`.

```
"availabilityConfig": {"recoveryAction": "STOP_INSTANCE"}
```

Enabling Strict Fault Domain Enforcement

To enable strict fault domain enforcement, do one of the following:

- In the Service Web UI, click the navigation menu, click FD Instances, and click Compute Service Detail. On the Compute Service Information page, click the Controls menu, and click Enable Strict FD.
- In the Service CLI, run the `enableStrictFD` command.

For more information about the effect of fault domain enforcement, see Compute Service Configuration Commands on page 171.

In case the current fault domain does not have enough resources to accommodate all instances that need to be migrated, do the following:

- If you are performing a planned compute node evacuation, specify the force option on the migration operation to stop the instances in their current fault domain.
- Run the `enableAutoRecoveryAction` command or select Enable Auto Recovery in the Service Web UI.
- Ensure that the instance availability recovery action for each instance is set to `RESTORE_INSTANCE`, which is the default. See Configuring the Recovery State for a Stopped Instance on page 173.

See the example in Evacuating a Compute Node on page 160.

Managing Private Cloud Appliance Administrator Accounts

A primary administrative account is configured during the initial setup of Private Cloud Appliance. With this primary account, which cannot be deleted, other administrator accounts must be created. The Service Enclave provides control over the privileges, preferences, and passwords of the administrator accounts.

Appliance administrator accounts can be created locally, but Private Cloud Appliance also supports federating with an existing identity provider, so people can sign in with their existing id and password. A single federated identity provider is supported for appliance administrator accounts. The process of establishing a federation trust with the identity provider is the same as for identity federation at the tenancy level. For more information, see Identity Federation with Microsoft Active Directory for Administrator Accounts on page 185.

The Service Enclave functions available to an administrator are determined by the authorization group to which the account belongs. For each authorization group, access control is configured using policies. A policy regulates the actions explicitly allowed on resources. Policy statements can also apply to authorization families, which are logical groups of resources or functions.

To create and manage administrator accounts, you must understand how to use the policy framework for access control. For more information, see Controlling Administrator Access Privileges on page 178.

Creating a New Administrator Account

Privileges granted to the administrator account depend on authorization group membership.

Using the Service Web UI

1. Open the navigation menu and click Users.
2. Click Create User to open the Create User window.
3. Enter the following details:
   - **Name:** Enter a name for this administrator account. This name will be used to sign in.
   - **Authorization Group:** Select the authorization group to which the new administrator is added. This selection determines the access rights and privileges of the administrator account.
   - **Password:** Set a password for the new administrator account. Enter it a second time to confirm.
4. Click Create User. The new administrator account is displayed in the Users table.

Using the Service CLI

1. Display the list of authorization groups. Copy the ID of the authorization group in which you want to create the new administrator account.

```
PCA-ADMIN> list AuthorizationGroup
Command: list AuthorizationGroup
Data:
  id Name
 -- ----
 9e6fef47-6ba7-4123-b25d-f9406173a609 OracleServiceAdmin
 2652ac1a-aa9e-4edf-bae7-d434efb23052 OCIApp
 411ed79b-8f66-434b-862a-3c6e1b036fc4 SuperAdmin
 f5f9a82e-aa0a-4c31-a873-fae59fe20f38 Initial
```

2. Create a new administrator account using the command `createUserInGroup`.

   Required parameters are the user name, password, and authorization group.

```
PCA-ADMIN> createUserInGroup name=testadmin
 password=************ confirmPassword=************
 authGroup=365ece7b-0a09-4a04-853c-7a0f6c4789f0
JobId: 6dd5a542-4399-4414-ac3b-636968744f79
```

3. Verify that the new administrator account was created correctly. Use the `list` and `show` commands to display the account information.

```
PCA-ADMIN> list User
Data:
  id                                     name
  --                                     ----
  401fce73-5bee-48b1-b86d-fba1d85e049b   admin
  682ebc19-8493-4e9a-817c-148acea4b1d4   testadmin

PCA-ADMIN> show user name=testadmin
Data:
  Id = 682ebc19-8493-4e9a-817c-148acea4b1d4
  Type = User
  Name = testadmin
  Default User = false
  AuthGroupIds 1 = id:365ece7b-0a09-4a04-853c-7a0f6c4789f0
 type:AuthorizationGroup  name:InternalGroup
  UserPreferenceId = id:1321249c-0651-49dc-938d-7764b9638ea9
 type:UserPreference  name:
```

Changing Authorization Group Membership

When you create an administrator account, you select the authorization group to which the new administrator is added. However, you can change which authorization groups an administrator belongs to at any time.

Using the Service Web UI

To add an administrator to an additional authorization group:

1. Open the navigation menu and click Authorization Groups.
2. Click the authorization group to which you want to add an administrator.
3. Under Resources, click Users and then click Add User to Group.
4. From the Add User to Group form, select an administrator and then click OK.

To remove an administrator from an authorization group:

1. If the administrator belongs only to the authorization group you want to remove the account from, add the administrator to another authorization group first.
2. Open the navigation menu and click Authorization Groups.
3. Click the authorization group from which you want to remove an administrator.
4. Under Resources, click Users. The list of users in the authorization group is displayed.
5. From the list, click the Actions menu for the user you want to remove, and then click Remove User from Group.

Using the Service CLI

1. Gather the IDs of the administrator account you want to change, and the authorization groups involved in the configuration change.

```
PCA-ADMIN> list User
Data:
  id                                     name
  --                                     ----
  401fce73-5bee-48b1-b86d-fba1d85e049b   admin
  682ebc19-8493-4e9a-817c-148acea4b1d4   testadmin

PCA-ADMIN> list AuthorizationGroup
Data:
  id                                     name
  --                                     ----
  587fc90d-3312-41d9-8be3-1ce21b8d9b41   MonitorGroup
```

```
  c18cc6af-4ef8-4b1c-b85d-ee3b065f503e   DrAdminGroup
  8f03faf2-c321-4455-af21-75cbffc269ef   AdminGroup
  5ac65f5d-1f8c-42ea-a1de-95a1941f009f   Day0ConfigGroup
  365ece7b-0a09-4a04-853c-7a0f6c4789f0   InitialGroup
  7da8be67-758c-4cd6-8255-e9d2900c788e   SuperAdminGroup
```

2. To add an administrator to an authorization group, use the `add User` command.

```
PCA-ADMIN> add User id=682ebc19-8493-4e9a-817c-148acea4b1d4 to
 AuthorizationGroup id=587fc90d-3312-41d9-8be3-1ce21b8d9b41
JobId: 3facde6d-acb6-4fc4-84dc-93de88eea25c
```

3. Display the administrator account details to verify the changes you made.

```
PCA-ADMIN> show User name=testadmin
Data:
  Id = 682ebc19-8493-4e9a-817c-148acea4b1d4
  Type = User
  Name = testadmin
  Default User = false
  AuthGroupIds 1 = id:365ece7b-0a09-4a04-853c-7a0f6c4789f0
 type:AuthorizationGroup  name:InternalGroup
  AuthGroupIds 2 = id:587fc90d-3312-41d9-8be3-1ce21b8d9b41
 type:AuthorizationGroup  name:MonitorGroup
  UserPreferenceId = id:1321249c-0651-49dc-938d-7764b9638ea9
 type:UserPreference  name:
```

4. To remove an administrator from an authorization group, use the `remove User` command.

```
PCA-ADMIN> remove User name=testadmin from AuthorizationGroup
 id=587fc90d-3312-41d9-8be3-1ce21b8d9b41
JobId: 44110d28-70af-4a42-8eb7-7d59a3bc8295
```

Changing Administrator Credentials

The administrator's password is set during account creation. You can always change the password for your own account. Depending on privileges, you may be authorized to change the password of another administrator as well.

Using the Service Web UI

1. Open the navigation menu and click Users.
2. Click the administrator account for which you want to change the password. The user detail page is displayed.

   Alternatively, to display your own user detail page, click your name in the top-right corner of the page and select My Profile.
3. Click Change Password to open the Change Password window.
4. Enter the new account password. Enter it a second time for confirmation. Click Save Changes to apply the new password.

Using the Service CLI

1. Display the list of administrator accounts. Copy the ID of the account for which you want to change the password.

```
PCA-ADMIN> list User
Data:
  id                                     name
  --                                     ----
  401fce73-5bee-48b1-b86d-fba1d85e049b   admin
  682ebc19-8493-4e9a-817c-148acea4b1d4   testadmin
```

2. Set a new password for the selected administrator account using the `changePassword` command.

```
PCA-ADMIN> changePassword id=682ebc19-8493-4e9a-817c-148acea4b1d4
 password=************ confirmPassword=************
JobId: 35710cd9-26ac-4be9-8b73-c4cf634cc121
```

Changing Administrator Account Preferences

When logged in to the Service CLI you can change certain settings for your own administrator account. Those changes take effect immediately and are persisted for all your future CLI connections.

However, you can also change settings temporarily for just your current CLI session. To do so, replace the object `UserPreference` with `CliSession` in the command examples below.

Setting	Options	Description
alphabetizeMode	YES, NO	Enable this setting to display any managed object's attributes in alphabetical order. The default setting is "No".
attributeDisplay	DISPLAYNAME, ATTRIBUTENAME	Use this setting to control whether the name of each object's attribute is displayed. The default setting is "displayName".
endLineCharsDisplayValue	CRLF, CR, LF	Specify the end-of-line character to be used when the CLI output consists of multiple lines. The default setting is "CRLF".
outputMode	VERBOSE, SPARSE, XML	Specify the CLI output format. The default setting is "Sparse".
wsCallMode	SYNCHRONOUS, ASYNCHRONOUS	Use this setting to determine whether the CLI output from a command is invoked synchronously or asynchronously. The default setting is "Asynchronous".
wsTimeoutInSeconds	<value>	When the CLI is set to "Synchronous" call mode, use this setting to determine how many seconds the CLI waits for a job returned by an operation to complete.

Proceed as follows:

1. Display your current account preferences.

```
PCA-ADMIN> show UserPreference
Data:
  Id = ec433c0f-4208-4e92-859e-498218d0f5c9
  Type = UserPreference
  WS Call Mode = Asynchronous
  Alphabetize Mode = No
  Attribute Display = Display Name
  End Line Characters Display Value = CRLF
  Output Mode = Verbose
  Command Wait Timeout In Seconds = 240
  UserId = id:401fce73-5bee-48b1-b86d-fba1d85e049b  type:User  name:admin
```

2. Change the setting of your choice using the `edit userPreference` command.

```
PCA-ADMIN> edit UserPreference outputMode=XML
JobId: 9d312d9b-6169-47cb-97d4-6a8984237fa0
```

3. Execute the edit command for any other settings you want to change.

4. Display your current account preferences again to verify the changes you made.

```
PCA-ADMIN> show UserPreference
Data:
  Id = ec433c0f-4208-4e92-859e-498218d0f5c9
  Type = UserPreference
  WS Call Mode = Asynchronous
  Alphabetize Mode = No
  Attribute Display = Display Name
  End Line Characters Display Value = CRLF
  Output Mode = Xml
  Command Wait Timeout In Seconds = 180
  UserId = id:401fce73-5bee-48b1-b86d-fba1d85e049b  type:User  name:admin
```

Deleting an Administrator Account

Using the Service Web UI

1. Open the navigation menu and click Users.
2. Click the administrator account you want to delete. The user detail page is displayed.
3. Click Delete. Confirm the operation when prompted.

Using the Service CLI

1. Look up the name and ID of the administrator account you want to delete.

```
PCA-ADMIN> list User
Data:
  id                                     name
  --                                     ----
  401fce73-5bee-48b1-b86d-fba1d85e049b   admin
  682ebc19-8493-4e9a-817c-148acea4b1d4   testadmin
```

2. To delete the administrator account, use the `delete User` command followed by the account name or ID.

```
PCA-ADMIN> delete User name=testadmin
JobId: 56e9dfcb-6b64-4f9d-b137-171f538029d3
```

3. Verify that the deleted account is no longer displayed in the user list.

```
PCA-ADMIN> list User
Data:
  id                                     name
  --                                     ----
  401fce73-5bee-48b1-b86d-fba1d85e049b   admin
```

Controlling Administrator Access Privileges

An administrator is granted access privileges based on the authorization groups the account is a member of. The policy attached to an authorization group defines access to resources and functions. Without a valid policy, the members of the authorization group have no access privileges.

Working with Authorization Groups

When configuring administrative access, you can use the default authorization groups or create a new one. The default groups are:

- Initial

  Users have limited access to the Service Enclave. They are authorized to create the initial administrator account and view information about the appliance, but do not have read access to any other resources.

- OCIApp

  Users have specific access to operations related to the use of OCI APIs and apps.
- OracleServiceAdmin

  Users have specific access to operations related to the use of Private Cloud Appliance services.
- SuperAdmin

  Users have unrestricted access to the Service Enclave. They are authorized to perform all available operations, including the setup of other administrator accounts and management of authorization groups and families.

> **Note:**
>
> Other internal authorization groups exist. For example, the *Day0* group provides specific access to operations related to the initial setup of the appliance.
>
> On existing systems upgraded from an older release, legacy authorization groups are not removed. For continuity, authorization families and policies are created during the upgrade process to ensure the same access privileges are retained.

Using the Service Web UI

1. Open the navigation menu and click Authorization Group.
2. Click Create Group.
3. Enter a name using 1 to 255 characters, and then click Create Authorization Group.

   The new authorization group's details page displays.
4. Click Add Policy Statement. The Authorization Policy Statement Form window displays.

   > **Note:**
   >
   > For information, see Writing Policy Statements on page 184.
5. Enter a name using 1 to 255 characters.
6. Select an action: Inspect, Read, Use, or Manage.
7. Select a policy application:
   - Resources - Enter the resources you want the policy to apply to.
   - Function Family - Select one from the drop down.
   - Resource Family - Select one from the drop down.

   > **Note:**
   >
   > For information, see Working with Authorization Families on page 182.
8. Click Create Policy Statement.

   The new policy statement displays on the details page. Add up to 100 additional policy statements.

Using the Service CLI

1. Create a new authorization group.

```
PCA-ADMIN> create AuthorizationGroup name=authors
JobId: 14ea4d22-acf1-455d-a7a1-ec0a30f29671
Data:
id:c672d9c6-90ec-4776-bccb-caae128e86db name:authors
```

2. View the help for the `create authpolicyStatement` command.

```
PCA-ADMIN> create authpolicyStatement ?
*action
```

```
activeState
functionFamily
resourceFamily
resources
*on
```

3. Enter `showcustomcmds ?` to see options for resources, or enter `showallcustomcmds` to view options for functions, for example:

```
PCA-ADMIN> showcustomcmds ?
                          ASRBundle
                          ASRPhonehome
                          BackupJob
                          CnUpdateManager
                          ComputeInstance
                          ComputeNode
                          [...]

PCA-ADMIN> showallcustomcmds
    Operation Name: <Related Object(s)>
    ----------------------------------
    [...]
    backup:  BackupJob
    changeIlomPassword:  ComputeNode, ManagementNode
    changePassword:  ComputeNode, LeafSwitch, ManagementNode,
 ManagementSwitch, SpineSwitch, User, ZFSAppliance
    clearFirstBootError:  NetworkConfig
    configZFSAdDomain:  ZfsAdDomain
    configZFSAdWorkgroup:  ZfsAdDomain
    createAdminAccount:
    createUserInGroup:  User
    deletePlatformImage:  PlatformImage
    deprovision:  ComputeNode
    disableVmHighAvailability:  PcaSystem
    drAddComputeInstance:  ComputeInstance
    drAddSiteMapping:  DrSiteMapping
    [...]
```

> **Note:**
>
> For more information on commands and syntax, see Using the Service CLI on page 150.

4. Create a policy statement using `resources`, `functionFamily` or `resourceFamily`.

> **Note:**
>
> For more information, see Writing Policy Statements on page 184 and Working with Authorization Families on page 182.

```
PCA-ADMIN> create authpolicyStatement action=manage
 resources=ComputeNode on authorizationGroup id=c672d9c6-90ec-4776-
bccb-caae128e86db
```

```
PCA-ADMIN> create authpolicyStatement action=manage
 authresourceFamily=rackops on authorizationGroup
 id=c672d9c6-90ec-4776-bccb-caae128e86db
```

```
PCA-ADMIN> create authpolicyStatement action=manage
 authfunctionFamily=computeops on authorizationGroup
 id=c672d9c6-90ec-4776-bccb-caae128e86db
```

5. View the details for the authorization group.

```
PCA-ADMIN> show authorizationGroup name=authors
Data:
Id = c672d9c6-90ec-4776-bccb-caae128e86db
Type = AuthorizationGroup
Name = authors
Policy Statements 1 = dea601bf-9bfc-4b2c-a135-d98378e69c87(ACTIVE)-
Allow authors to MANAGE ComputeNode
Is Predefined Authorization Group = false
AuthPolicyStatementIds 1 = id:4adde579-1f6a-49eb-a783-9478465f135e
 type:AuthPolicyStatement name:
AuthPolicyStatementIds 2 = id:be498a4e-3e0a-4cfa-9013-188542adb8e3
 type:AuthPolicyStatement name:
```

To inactivate a policy statement:

1. View the help for the `edit authpolicyStatement` command.

```
PCA-ADMIN> edit authpolicyStatement ?
id=<object identifier>
```

2. Find the policy statement's ID using the `show authorizationGroup name=`*`group-name`* command.

```
PCA-ADMIN> show authorizationGroup name=authors
[...]
Policy Statements 1 = dea601bf-9bfc-4b2c-a135-d98378e69c87(ACTIVE)-
Allow authors to MANAGE ComputeNode
Is Predefined Authorization Group = false
AuthPolicyStatementIds 1 = id:4adde579-1f6a-49eb-a783-9478465f135e
 type:AuthPolicyStatement name:
AuthPolicyStatementIds 2 = id:be498a4e-3e0a-4cfa-9013-188542adb8e3
 type:AuthPolicyStatement name:
```

3. Using the ID of the policy statement (`AuthPolicyStatementIds` *`Number`* `= id:`*`unique-identifier`*) view the command to activate or inactivate the policy statement.

```
PCA-ADMIN> edit authpolicyStatement
 id=be498a4e-3e0a-4cfa-9013-188542adb8e3 ?
activeState
```

4. Inactivate the policy statement.

```
PCA-ADMIN> edit authpolicyStatement
 id=be498a4e-3e0a-4cfa-9013-188542adb8e3 activeState=inactive
JobId: 842c444e-060d-461d-a4e0-c9cdd9f1d3c3
```

5. Verify the policy statement is inactive.

```
PCA-ADMIN> show authorizationGroup name=authors
Data:
Id = c672d9c6-90ec-4776-bccb-caae128e86db
Type = AuthorizationGroup
Name = authors
Policy Statements 1 = 4adde579-1f6a-49eb-a783-9478465f135e(ACTIVE)-
Allow authors to MANAGE ComputeNode
Policy Statements 2 =
 be498a4e-3e0a-4cfa-9013-188542adb8e3(INACTIVE)-Allow authors to
 MANAGE ComputeNode
Is Predefined Authorization Group = false
AuthPolicyStatementIds 1 = id:4adde579-1f6a-49eb-a783-9478465f135e
 type:AuthPolicyStatement name:
```

```
AuthPolicyStatementIds 2 = id:be498a4e-3e0a-4cfa-9013-188542adb8e3
 type:AuthPolicyStatement name:
```

Working with Authorization Families

Using an authorization family allows you to create policies that you can reuse across authorization groups. The default authorization groups use predefined policies, which are created using authorization families. There are two types of authorization families you can use in policy statements:

- *Resource families* are used to define appliance resources, such as servers, storage, and network infrastructure.
- *Function families* are used to define appliance functions, such as compartment, user, and compute management.

The following table lists the predefined authorization families and how they are used in the default authorization group policies.

Authorization Family	Type	Used In Policies For...	Users In Group Can...
Day0	Function Family	SuperAdmin authorization group	• set Day0 system, static routing, dynamic routing, and network parameters • get management node, compute node, and ZFS health from ILOM • unlock and lock the appliance
Initial	Function Family	Initial authorization group	create the initial admin account
OCIApp	Function Family	SuperAdmin authorization group	create OCI apps account
OracleServiceAdmin	Function Family	SuperAdmin authorization group	create Oracle services account
SuperAdmin	Function Family	SuperAdmin authorization group	manage all appliance functions
Day0	Resource Family	SuperAdmin authorization group	read system information and networking configuration
Initial	Resource Family	Initial authorization group	read system information
OCIApp	Resource Family	SuperAdmin authorization group	manage OCI apps
OracleServiceAdmin	Resource Family	SuperAdmin authorization group	manage Oracle services
SuperAdmin	Resource Family	SuperAdmin authorization group	manage all resources on appliance

Using the Service Web UI

1. Open the navigation menu and click Authorization Families.
2. Click Create Authorization Family.
3. Select either authorization family type: Function Family or Resources Family.
4. Enter a name.

5. Enter the resources to include in the family.

> **Note:**
>
> For information on how to find the resource and function options, see the CLI instructions.

6. Click Create Family.

Using the Service CLI

To create an authorization function family:

1. Display the options for the `create authfunctionFamily` command.

```
PCA-ADMIN> create authfunctionFamily ?
*name
*resources
```

2. Enter `showallcustomcmds` to view options for functions, for example:

```
PCA-ADMIN> showallcustomcmds
    Operation Name: <Related Object(s)>
    ----------------------------------
    [...]
    backup:  BackupJob
    changeIlomPassword:  ComputeNode, ManagementNode
    changePassword:  ComputeNode, LeafSwitch, ManagementNode,
 ManagementSwitch, SpineSwitch, User, ZFSAppliance
    clearFirstBootError:  NetworkConfig
    configZFSAdDomain:  ZfsAdDomain
    configZFSAdWorkgroup:  ZfsAdDomain
    createAdminAccount:
    createUserInGroup:  User
    deletePlatformImage:  PlatformImage
    deprovision:  ComputeNode
    disableVmHighAvailability:  PcaSystem
    drAddComputeInstance:  ComputeInstance
    drAddSiteMapping:  DrSiteMapping
    [...]
```

3. Create the authorization function family.

```
PCA-ADMIN> create authfunctionFamily name=cnops
 resources=ComputeNode.reset,ComputeNode.start,ComputeNode.stop
Command: create authfunctionFamily name=cnops
 resources=ComputeNode.reset,ComputeNode.start,ComputeNode.stop
JobId: 4cd37ea7-161f-4b11-952f-ffa992a37d5f
Data:
id:ae0216da-20d1-4e03-bf65-c7898c6079b2 name:cnops
```

4. List the authorization function families.

```
PCA-ADMIN> list authfunctionFamily
Data:
id name
-- ----
7f1ac922-571a-4253-a120-e5d15a877a1e Initial
2185058a-3355-48be-851c-2fa0e5a896bd SuperAdmin
7f092ddd-1a51-4a17-b4e2-96c4ece005ec Day0
ae0216da-20d1-4e03-bf65-c7898c6079b2 cnops
```

To create an authorization resource family:

1. Display the options for the `create authresourceFamily` command.

```
PCA-ADMIN> create authresourceFamily ?
*name
*resources
```

2. Enter `showcustomcmds ?` to see options for resources, for example:

```
PCA-ADMIN> showcustomcmds ?
                          ASRBundle
                          ASRPhonehome
                          BackupJob
                          CnUpdateManager
                          ComputeInstance
                          ComputeNode
                          [...]
```

> **Note:**
>
> For more information on commands and syntax, see Using the Service CLI on page 150.

3. Create the authorization resource family.

```
PCA-ADMIN> create authresourceFamily name=rackops
 resources=ComputeNode,RackUnit
JobId: eb49ac48-e3f3-4c2f-bf11-d5d18a066788
Data:
id:b54e4413-15bd-440e-b399-e2ab75f17c35 name:rackops
```

4. List the authorization resource families.

```
PCA-ADMIN> list authresourceFamily
Data:
id name
-- ----
9aefc9c8-556d-42a4-9369-d7cdf0bf0c52 SuperAdmin
b591cc7b-b117-449e-af35-cb4fc6f0c213 Day0
87633db2-d724-45b6-97a5-30babb6c4869 cnops
b54e4413-15bd-440e-b399-e2ab75f17c35 rackops
a45c08b4-f895-4da8-87f4-c81ca0b2bf27 Initial
```

Writing Policy Statements

Policies are required for authorization groups to work. You can create individual policies or use authorization families. You can create policy statements from the Service Web UI or Service CLI. Each policy statement must contain the following:

- Name - 1 to 255 characters
- Action - Inspect, Read, Use, or Manage
- Resource/Authorization Family - One or more resources or one authorization family
- (**Service CLI only**) Authorization Group - the ID of the group

> **Note:**
>
> You cannot modify a policy statement. If you need to make changes to a policy statement, you must delete it and then recreate it.

The following table contains information about the actions you can take on a resource.

Action	Type of Access
`inspect`	Ability to list resources, without access to any confidential information or user-specified metadata that may be part of that resource.
`read`	Includes `inspect` plus the ability to get user-specified metadata and the actual resource itself.
`use`	Includes `read` plus the ability to work with existing resources. The actions vary by resource type.
`manage`	Includes all permissions for the resource.

Identity Federation with Microsoft Active Directory for Administrator Accounts

An administrator can configure identity federation between Private Cloud Appliance and Microsoft Active Directory. Authentication for Private Cloud Appliance is then delegated to the identity provider (Active Directory), enabling users to sign in with their existing accounts.

Many organizations use an identity provider to manage user logins and passwords and to authenticate users for access to secure websites, services, and resources. Federation involves setting up a trust relationship between the identity provider and Private Cloud Appliance. When an administrator has established this relationship, federated users are prompted with a *single sign-on* when accessing the Service Web UI.

> **Tip:**
>
> We document Microsoft Active Directory as a reference implementation, but Private Cloud Appliance works with other SAML 2.0-compliant identity providers, including Oracle Access Manager. For information and instructions, see Using Oracle Access Manager to Sign On.

You can federate several Microsoft Active Directory (AD) accounts with Private Cloud Appliance (for example, one for each division of the organization), but each federation trust that you set up must be for a *single* AD account. To set up a trust, you perform some tasks in the Service Web UI and some tasks in Active Directory Federation Services (ADFS).

Before you begin federating, ensure you already have:

- Installed and configured Microsoft Active Directory Federation Services for your organization.
- Set up groups in Active Directory that will map to groups in Private Cloud Appliance.
- Created users in Active Directory who will sign in to the Service Web UI.

> **Note:**
>
> Consider naming Active Directory groups that you intend to map to Private Cloud Appliance groups with a common prefix to make it easy to apply a filter rule – for example: PCA_Administrators, PCA_NetworkAdmins, PCA_InstanceLaunchers.

To set up a federated identity provider, follow these instructions:

- Enabling the Federation Trust Relationship on page 186
- Configuring Identity Federation on page 189

Enabling the Federation Trust Relationship

Identity federation requires a trust relationship between Private Cloud Appliance and its identity provider, Microsoft Active Directory. Specific details from the federation partner system are required to establish this mutual trust, so that data for user authentication can be exchanged.

Gathering Required ADFS Metadata

Identity federation requires the following information from the identity provider:

- The SAML metadata document from Active Directory Federation Services (ADFS)

  Its default location is: `https://<id-provider-name>/FederationMetadata/2007-06/FederationMetadata.xml`.
- The names of the Active Directory (AD) groups that you want to map to Private Cloud Appliance groups

  Make a note of all the AD groups that need mapping.

> **Caution:**
>
> Ensure that all the Private Cloud Appliance groups have been configured before you add AD as an identity provider.

Verifying Identity Provider Certificates

> **Caution:**
>
> If your ADFS certificate is signed by a known certificate authority it should already exist in the Private Cloud Appliance certificate bundle. In that case you can skip this section.

The Private Cloud Appliance Certificate Authority (CA) is based on self-signed, OpenSSL generated root and intermediate x.509 certificates. These CA certificates are used to issue x.509 server/client certificates, enabling you to add outside CA trust information to the appliance. If you use a self-signed certificate for ADFS, you need to add outside CA trust information from ADFS to the management nodes.

> **Note:**
>
> If you are using the metadataUrl property to create or update an identity provider, you need to add the identity provider web server certificate chain to the Private Cloud Appliance outside CA bundle. See your identity provider's documentation on how to find the web server certificate chain and then follow steps 3 to 8.

To add outside CA trust information, proceed as follows:

1. Download the SAML metadata document for your ADFS, which is available by default at:

   ```
   https://<id-provider-name>/FederationMetadata/2007-06/
   FederationMetadata.xml
   ```

2. Open the file in a text or XML editor and locate the signing certificate section. For example:

   ```
   <KeyDescriptor use="signing">
   <KeyInfo>
   <X509Data>
   <X509Certificate>
   <!--CERTIFICATE IS HERE-->
   </X509Certificate>
   </X509Data>
   </KeyInfo>
   </KeyDescriptor>
   ```

3. Log on to management node 1 (`pcamn01`).

4. Navigate to `/etc/pca3.0/vault` and create a new directory named `customer_ca`.

   > **Note:**
   >
   > You can use this directory for multiple files. For example, you can create a file for the identity provider certificate and one for the web server certificate chain.

5. In the `customer_ca` directory, create a new file in PEM format.
6. Copy the certificate from the `FederationMetadata.xml` file, which is located within the `<X509Certificate>` tag, and paste into the new PEM file. Be sure to include the `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----`, for example:

   ```
   -----BEGIN CERTIFICATE-----
   <CERTIFICATE CONTENT>
   -----END CERTIFICATE-----
   ```

7. Save the file and close.
8. Run the following command to update the `ca_outside_bundle.crt` on all management nodes:

   ```
   python3 /usr/lib/python3.11/site-packages/pca_foundation/secret_service/
   cert_generator/cert_generator_app.py -copy_to_mns
   ```

Adding Private Cloud Appliance As a Trusted Relying Party in ADFS

To complete the federation process, you must add Private Cloud Appliance as a trusted relying party in Microsoft Active Directory Federation Services (ADFS) and then add associated relying party claim rules.

Adding a Relying Party in ADFS

1. In the Service Web UI on the Identity Providers page, view the following text block:

   You need the Private Cloud Appliance Federation Metadata document when setting up a trust with Microsoft Active Directory Federation Services or with other SAML 2.0-compliant identity providers. This is an XML document that describes the Private Cloud Appliance endpoint and certificate information. Click Here

2. Click "Click Here".

   A metadata XML file opens in the browser with a URL similar to:

   ```
   https://adminconsole.system-name.domain-name/wsapi/rest/saml/
   metadata/
   ```

3. Copy the metadata XML file URL.
4. From the system installed with ADFS, open a browser window and paste the URL.
5. Save the file, making sure to use the `.xml` extension, for example, ***`my-sp-metadata`***`.xml`.
6. Go to the AD FS Management Console and sign in to the account you want to federate.
7. Add Private Cloud Appliance as a trusted relying party.

   a. Under AD FS, right-click Relying Party Trusts and the select Add Relying Party Trust.
   b. In the Add Relying Party Trust Wizard Welcome page, select Claims Aware and then click Start.
   c. On the Select Data Source page, select "Import data about the relying party from a file".
   d. Click Browse and navigate to your ***`my-sp-metadata`***`.xml` and then click Open.
   e. On the Specify Display Name page, enter a display name, add any optional notes for the relying party, and then click Next.
   f. On the Choose Access Control Policy page, select the type of access you want to grant and then click Next.
   g. On the Ready to Add Trust page, review the settings, and then click Next to save your relying party trust information.

h. On the Finish page, check "Configure claims issuance policy for this application" and then click Close.

The Edit Claim Issuance Policy dialog appears, which you can leave open for the next procedure.

Adding Relying Party Claim Rules

After adding Private Cloud Appliance as a trusted relying party, you must add claim rules so that the elements required (Name ID and groups) are added to the SAML authentication response.

To add a Name ID rule, proceed as follows:

1. In the Edit Claim Issuance Policy dialog, click Add Rule.

   The Select Rule Template dialog is displayed.
2. For Claim rule template, select Transform an Incoming Claim and then click Next.
3. Enter the following:

   - **Claim rule name**: Enter a name for this rule, for example, `nameid`.
   - **Incoming claim type**: Select Microsoft Windows account name.
   - **Outgoing claim type**: Select a claim type, for example, Name ID.
   - **Outgoing name ID format**: Select Persistent Identifier.
   - Select Pass through all claim values and then click Finish.

     The rule is displayed in the rules list.

The Issuance Transform Rules dialog displays the new rule.

If your Active Directory users are in no more than 100 groups, you simply add the groups rule. However, if your Active Directory users are in more than 100 groups, those users cannot be authenticated to use the Service Web UI. For these groups, you must apply a filter to the groups rule.

To add the groups rule, proceed as follows:

1. In the Issuance Transform Rules dialog, click Add Rule.

   The Select Rule Template dialog is displayed.
2. For Claim rule template, select Send Claims Using a Custom Rule and then click Next.
3. In the Add Transform Claim Rule Wizard, enter the following:

   - **Claim rule name**: Enter groups.
   - **Custom rule**: Enter the custom rule.

     ```
     c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/
     claims/windowsaccountname", Issuer == "AD AUTHORITY"]
      => issue(store = "Active Directory", types = ("https://
     auth.oraclecloud.com/saml/claims/groupName"), query =
      ";tokenGroups;{0}", param = c.Value);
     ```

   - Click Finish.

   The Issuance Transform Rules dialog displays the new rule.

Disabling the Certificate Revocation Check

For ADFS to work with SAML, you must disable the Certificate Revocation List (CRL) checking. Open Powershell on the ADFS system and enter the following command, where *`TRUST_NAME`* is the name of the relying party trust:

```
Get-AdfsRelyingPartyTrust -Name '<TRUST_NAME>' | Set-
AdfsRelyingPartyTrust -EncryptionCertificateRevocationCheck None -
SigningCertificateRevocationCheck None
```

Configuring Identity Federation

To set up an identity provider in Private Cloud Appliance, ensure that you have its metadata file and that CA certificate requirements have been verified. Add the required group mappings to enable federated user authentication.

Follow the step by step instructions in this section to manage identity providers and their group mappings.

Managing Identity Providers

Adding Active Directory as an Identity Provider

1. Sign in to the Service Web UI.
2. Open the navigation menu and click Identity Provider.
3. On the Identity Providers page, click Create Identity Provider.
4. On the Create an Identity Provider page, provide the following information:
   - **Display Name**

     The name that the federated users see when choosing which identity provider to use for signing in to the Service Web UI. This name must be unique across all identity providers and cannot be changed.
   - **Description**

     A friendly description of the identity provider.
   - **Authentication Contexts**

     Click Add Class Reference and select an authentication context from the list.

     When one or more values are specified, Private Cloud Appliance (the relying party), expects the identity provider to use one of the specified authentication mechanisms when authenticating the user. The returned SAML response from the identity provider must contain an authentication statement with that authentication context class reference. If the SAML response authentication context does not match what is specified here, the Private Cloud Appliance authentication service rejects the SAML response with a 400.
   - **Encrypt Assertion** (Optional)

     When enabled, the authorization service expects encrypted assertions from the identity provider. Only the authorization service can decrypt the assertion. When not enabled, the authorization service expects SAML tokens to be unencrypted, but protected, by SSL.
   - **Force Authentication** (Optional)

     When enabled, users are always asked to authenticate at their identity provider when redirected by the authorization service. When not enabled, users are not asked to re-authenticate if they already have an active login session with the identity provider.
   - **Metadata URL**

     Enter the URL for the `FederationMetadata.xml` document from the identity provider.

     By default, the metadata file for ADFS is located at `https://<id-provider-name>/FederationMetadata/2007-06/FederationMetadata.xml`.
5. Click Create Identity Provider.

   Your new identity provider is assigned an OCID and is displayed on the Identity Providers page

After the identity provider is added, you must set up the group mappings between Private Cloud Appliance and Active Directory. See Managing Group Mappings for an Identity Provider on page 190.

Updating an Identity Provider

1. Open the navigation menu and click Identity Providers.

   A list of the identity providers is displayed.
2. For the identity provider you want to update, click the Actions icon (three dots) and then click Edit.
3. Change any of the following information. However, be aware that changing this information can affect the federation.

- **Description**
- **Authentication Contexts**

  Add or delete a class reference.
- **Encrypt Assertion**

  Enable or disable encrypted assertions from the identity provider.
- **Force Authentication**

  Enable or disable redirect authentication from the identity provider.
- **Metadata URL**

  Enter the URL for a new FederationMetadata.xml document from the identity provider.

4. Click Update Identity Provider.

Viewing Identity Providers and Configuration Details

The identity provider details page displays general information such as authentication contexts. It also provides the identity provider's settings, which include the redirect URL. From this page, you can also edit the identity provider and manage the group mappings.

1. Open the navigation menu and click Identity Providers.

   A list of the identity providers is displayed.
2. For the identity provider whose details you want to view, click the Actions icon (three dots) and then click View Details.

   The identity provider details page is displayed.

Deleting an Identity Provider

If you want to remove the option for federated users to log into Private Cloud Appliance you must delete the identity provider, which also deletes all of the associated group mappings.

1. Open the navigation menu, click Identity and then click Federation.

   A list of the identity providers is displayed.
2. For the identity provider you want to delete, click the Actions icon (three dots) and then click Delete.
3. At the Delete Identity Provider prompt, click Confirm.

Managing Group Mappings for an Identity Provider

When working with group mappings, remember the following:

- A given Active Directory group is mapped to a single Private Cloud Appliance group.
- Private Cloud Appliance group names must not contain spaces and cannot be changed later. Allowed characters are letters, numerals, hyphens, periods, underscores, and plus signs (+).
- You cannot update a group mapping, but you can delete the mapping and add a new one.

> **Important:**
>
> Before federated users can sign in to the Service Web UI, you must provide them with the URL. Ensure that you have configured all required group mappings, otherwise a federated user cannot perform any operations in Private Cloud Appliance.

Creating a Group Mapping

Perform the following steps for each identity provider group you want to map:

1. Open the navigation menu and click IDP Group Mappings.

   A list of the identity provider group mappings is displayed.

2. Click Create Group Mapping.

   The IDP Group Mapping Form is displayed
3. In the Name field, enter a name for the IDP group mapping.
4. In the IDP Group Name field, enter the *exact* name of the identity provider group.
5. From the Admin Group Name list, select the Private Cloud Appliance group you want to map to the identity provider group.
6. Optionally, enter a Description of the group.
7. Click Create IDP Group Mapping.

   The new group mapping is displayed in the list.

Updating a Group Mapping

1. Open the navigation menu and click IDP Group Mappings.

   A list of the identity provider group mappings is displayed.
2. For the group mapping you want to update, click the Actions icon (three dots) and then click Edit.

   The IDP Group Mapping Form is displayed.
3. Modify any of the following fields. However, be aware that changing this information can affect the federation.
   - Name
   - IDP Group Name
   - Admin Group Name
   - Description
4. Click Modify IDP Group Mapping.

   The updated group mapping is displayed in the list.

Deleting a Group Mapping

1. Open the navigation menu and click IDP Group Mappings.

   A list of the identity provider group mappings is displayed.
2. For the group mapping you want to delete, click the Actions icon (three dots) and then click Delete.
3. At the Deleting IDP Group Mapping prompt, click Confirm.

Managing Private Cloud Appliance Tenancies

A tenancy is an environment where users create and manage cloud resources in order to build and configure virtualized workloads. A Private Cloud Appliance environment must contain at least one tenancy.

All the tenancies in the environment are collectively referred to as the Compute Enclave. However, tenancy management is a responsibility of the appliance administrator. Tenancies are created from the Service Enclave and subsequently handed over to the initial user in the tenancy: the primary tenancy administrator. For more information, see Enclaves and Interfaces on page 8.

Creating and Updating a Tenancy

An infrastructure administrator sets up a tenancy from the Service Enclave and provides access details to the primary tenancy administrator. Then the tenancy administrator can start configuring additional user accounts and cloud resources in the Compute Enclave.

Using the Service Web UI

1. In the navigation menu, click Tenancies.
2. In the top-right corner of the Tenancies page, click Create Tenancy.

   The Create Tenancy window appears.
3. Fill out the tenancy details:

- **Name:** Enter a name for the new tenancy.
- **Description:** Optionally, enter a description for the new tenancy.
- **Service Namespace:** Set a unique namespace for all resources created within this tenancy.
- **Authentication Credentials:** Set a user name and password for the primary tenancy administrator.

  This account must be used to log in to the tenancy for the first time. The tenancy administrator sets up additional user accounts, defines compartments, policies and other resources, and generally configures the cloud environment so that users can start deploying their required resources.

4. Click Save Changes to create the new tenancy.

   The new tenancy is displayed in the Tenancies list.

Using the Service CLI

1. Create a new tenancy with the `create Tenant` command.

   The name, namespace and admin account credentials are required parameters; a description is optional.

   Syntax (entered on a single line):

   ```
   create Tenant
   name=<tenancy_name>
   serviceNamespace=<tenancy_namespace>
   description=<tenancy_description>
   adminUserName=<tenancy_admin_user_name>
   adminPassword=<tenancy_admin_password>
   confirmPassword=<tenancy_admin_password>
   ```

   Example:

   ```
   PCA-ADMIN> create Tenant name=myTestTenancy serviceNamespace=test
    description="A tenancy for testing purposes" \
   adminUserName=testadmin adminPassword=************
    confirmPassword=************
   JobId: a0ee398f-5d44-4b3f-8b9c-e5a9692c36a4
   Data:
     id:ocid1.tenancy.....<uniqueID>  name:myTestTenancy
   ```

2. Use the job ID to check the status of your command.

   ```
   PCA-ADMIN> show Job id=a0ee398f-5d44-4b3f-8b9c-e5a9692c36a4
   Data:
     Id = a0ee398f-5d44-4b3f-8b9c-e5a9692c36a4
     Type = Job
     AssociatedObj = id:ocid1.tenancy.unique_ID  type:Tenant
    name:myTestTenancy
     AssociatedObj Type = Tenant
     AssociatedObj Id = ocid1.tenancy.unique_ID
     Done = true
     Name = CREATE_TYPE
     Run State = Succeeded
   [...]
   ```

3. Verify that the new tenancy was created correctly. Use the `list` and `show` commands to display the tenancy information.

   ```
   PCA-ADMIN> list Tenant
   Data:
     id                              name
     --                              ----
     ocid1.tenancy.unique_ID   myTenancy1
     ocid1.tenancy.unique_ID   myTenancy2
     ocid1.tenancy.unique_ID   myTenancy3
   ```

```
  ocid1.tenancy.unique_ID   myTestTenancy

PCA-ADMIN> show Tenant name=myTestTenancy
Data:
  Id = ocid1.tenancy.unique_ID
  Type = Tenant
  Name = myTestTenancy
  Description = A tenancy for testing purposes
  Service Namespace = test
```

4. Provide the Compute Web UI URL, tenancy name, user name and password to the primary tenancy administrator. The tenancy is now ready for use.

   The tenancy administrator sets up additional user accounts, defines compartments, policies and other resources, and generally configures the cloud environment so that users can start deploying their required resources.

Updating a Tenancy

The only tenancy property that an administrator can modify at this time is the description.

- **Service Web UI:** Open the tenancy detail page and click Edit.
- **Service CLI:** Use the command `edit Tenant name=<tenancy_name> description=<tenancy_description>`

Deleting a Tenancy

Ensure that tenancy users have removed all their resources. The tenancy can only be deleted if it is empty.

Using the Service Web UI

1. In the navigation menu, click Tenancies.
2. In the tenancies table, click the name of the tenancy you want to delete.

   The tenancy detail page is displayed.
3. In the top-right corner of the tenancy detail page, click Delete. Confirm the operation when prompted.

Using the Service CLI

1. Look up the name and ID of the tenancy you want to delete.

```
PCA-ADMIN> list Tenant
Data:
  id                            name
  --                            ----
  ocid1.tenancy.unique_ID   myTenancy1
  ocid1.tenancy.unique_ID   myTenancy2
  ocid1.tenancy.unique_ID   myTenancy3
  ocid1.tenancy.unique_ID   myTestTenancy
```

2. To delete the tenancy, use the `delete Tenant` command followed by the tenancy name or ID.

```
PCA-ADMIN> delete Tenant name=myTestTenancy
Status: Running
JobId: 92b84ac2-1f2c-41d7-980e-d7549957ef93
```

3. Verify that the deleted tenancy is no longer displayed in the tenancy list.

```
PCA-ADMIN> list Tenant
Data:
  id                            name
  --                            ----
  ocid1.tenancy.unique_ID   myTenancy1
  ocid1.tenancy.unique_ID   myTenancy2
```

```
  ocid1.tenancy.unique_ID   myTenancy3
```

Providing OCI Images

OCI Images, also called platform images, are loaded onto the system during Private Cloud Appliance installation, and new platform images might be provided during appliance upgrade or patching operations. The images are placed in the management cluster shared storage under `/nfs/shared_storage/oci_compute_images`.

To enable users to deploy instances from the platform images, these images must also be imported into the Compute Enclave. During appliance upgrade or patching, the import is part of the normal process. If the Oracle-provided OCI images were not correctly imported, or if you uploaded custom images to be used by all Compute Enclave users, run the `importPlatformImages` command as described in this section.

New versions of an image, added during upgrade and patching, never replace any existing versions. If earlier versions of an image are obsolete and should no longer be available to users, an administrator must delete those older versions manually as described in this section. If more than three versions of an image are available, only the three most recent versions are shown when images are listed in the Compute Enclave. Older platform images are still available, and can be selected by specifying the image OCID.

Importing OCI Images

From the Service CLI, run the `importPlatformImages` command to make all images that are in the management cluster shared storage (`/nfs/shared_storage/oci_compute_images`) also available in all tenancies in the Compute Enclave.

```
PCA-ADMIN> importPlatformImages
JobId: f21b9d86-ccf2-4bd3-bab9-04dc3adb2966
```

Use the `JobId` to get more detailed information about the job. In the following example, no new images have been delivered:

```
PCA-ADMIN> show job id=f21b9d86-ccf2-4bd3-bab9-04dc3adb2966
Data:
  Id = f21b9d86-ccf2-4bd3-bab9-04dc3adb2966
  Type = Job
  Done = true
  Name = OPERATION
  Progress Message = There are no new platform image files to import
  Run State = Succeeded
```

Listing OCI Images

In the Service CLI, use the `listplatformImages` command to list all platform images that have been imported from the management cluster shared storage.

```
PCA-ADMIN> listplatformImages
Data:
  id                       displayName
 lifecycleState
  --                       -----------
 --------------
  ocid1.image.unique_ID   uln-pca-Oracle-Linux-7.9-2024.05.29_0.oci
 AVAILABLE
  ocid1.image.unique_ID   uln-pca-Oracle-Linux-8-2024.05.29_0.oci
 AVAILABLE
  ocid1.image.unique_ID   uln-pca-Oracle-Linux-9-2024.05.29_0.oci
 AVAILABLE
  ocid1.image.unique_ID   uln-pca-Oracle-Linux8-OKE-1.26.6-20240611.oci
 AVAILABLE
  ocid1.image.unique_ID   uln-pca-Oracle-Linux8-OKE-1.27.7-20240602.oci
 AVAILABLE
  ocid1.image.unique_ID   uln-pca-Oracle-Linux8-OKE-1.28.3-20240602.oci
 AVAILABLE
```

```
  ocid1.image.unique_ID   uln-pca-Oracle-Solaris-11-2024.05.07_0.oci
 AVAILABLE
```

Compute Web UI users see the same `lifecycleState` that `listplatformImages` shows. Shortly after the import operation, new images are displayed with life cycle state `IMPORTING`. When the import job is complete, the life cycle state changes to `AVAILABLE`.

If you delete a platform image, both the Service CLI and the Compute Web UI list the image as `DELETING` or `DELETED`.

Deleting OCI Images

In the Service CLI, use the `deleteplatformImage` command to delete a platform image. The image shows as `DELETING` and then `DELETED` in the Service CLI output and in the Compute Enclave, and eventually is not listed at all.

> **Important:**
>
> The `deleteplatformImage` command does not delete the image file from shared storage. It must also be removed from the file system. Running the `importPlatformImages` command reimports the image and makes it available again in the Compute Enclave.

```
PCA-ADMIN> deleteplatformImage imageId=ocid1.image.unique_ID_7
JobId: 401567c3-3662-46bb-89d2-b7ad1541fa2d

PCA-ADMIN> listplatformImages
Data:
  id                          displayName
 lifecycleState
  --                          -----------
 --------------
  ocid1.image.unique_ID   uln-pca-Oracle-Linux-7.9-2024.05.29_0.oci
 AVAILABLE
  ocid1.image.unique_ID   uln-pca-Oracle-Linux-8-2024.05.29_0.oci
 AVAILABLE
[...]
  ocid1.image.unique_ID   uln-pca-Oracle-Solaris-2024.05.07_0.oci
 DELETED
```

Viewing and Setting Resource Limits

The Limit Service in Private Cloud Appliance enables you to view and change (override) limits that are currently set for cloud resources. You can create resource limit templates to easily apply a set of resource limits.

Cloud Services with Configurable Limits

The following procedures describe how to list the cloud services that are supported by the Limit Service. These are the services that allow you to override some resource limits and view the current limits.

Using the Service Web UI

1. On the navigation menu, select Limits, and then select Limit Service.
2. Select the Configuration tab to see the list of services that are supported by the Limit Service.

Using the Service CLI

Use the `show LimitService` command to list the services that support viewing and setting resource limits. The following example lists only two services that support viewing and setting resource limits: IAM and Load Balancer. More services are listed when you run this command.

```
PCA-ADMIN> show LimitService
Data:
  Id = 175b1cc5-bfad-4990-940f-632d9263fa3d
  Type = LimitService
```

```
  Supported Services 1 = identity
  Supported Services 2 = loadbalancer
  Overall Communication State = Okay
  Name = Limit Service
  Work State = Normal
```

Viewing Resource Limits

Learn how to list the available Private Cloud Appliance resource limit definitions, show the full specified resource limit definition, and show the default and current limit values of the specified resources.

Listing and Showing Limit Definitions

This section describes how to list all available limit definitions for all supported services and show a description of each limit. It also describes how to show the details of a specified limit definition, including the default value of the limit, supported tags, and whether quotas are supported. See Supported Tags and Quotas Supported on page 197.

Using the Service Web UI

1. On the navigation menu, select Limits, and then select Limit Definitions.

   For each limit definition, the table shows the name of the limit definition, the name of the service, and the description of the limit.
2. To show the details of a particular limit definition, either select the name of the limit definition in the list, or select the Actions menu for that limit definition and select View Details.

   The Resources section of the details page shows overrides that have already been created for this resource limit. You can edit or delete an override in that list, view details of an override, or use the button above the list to create a new override for this resource limit. See Creating Resource Limit Overrides on page 202.

Using the Service CLI

To list all available limit definitions, use the `list LimitDefinition` command. The following example lists only two limit definitions. Many more limit definitions are listed when you run this command.

```
PCA-ADMIN> list LimitDefinition
Data:
  id                      Name       Service Name   Description
  --                      ----       ------------   -----------
  identity~groups         groups     identity       Maximum number of
 user groups
  loadbalancer~backends   backends   loadbalancer   Maximum number of
 backends per load balancer (system-wide)
```

To show the details of a specified limit definition, use the `show LimitDefinition` command. Provide the resource limit `id` from the `list LimitDefinition` output.

```
PCA-ADMIN> show LimitDefinition id=identity~groups
Data:
  Id = identity~groups
  Type = LimitDefinition
  Name = groups
  Service Name = identity
  Description = Maximum number of user groups
  Supported Tags 1 = {TENANCY}
  Quotas Supported = false
  Value = 100

PCA-ADMIN> show LimitDefinition id=loadbalancer~backends
Data:
  Id = loadbalancer~backends
  Type = LimitDefinition
```

```
  Name = backends
  Service Name = loadbalancer
  Description = Maximum number of backends per load balancer
  Supported Tags 1 = SYSTEM
  Value = 512
```

The "Value" is the default limit for this resource. For the current limit value (if an override has been set), see the example in Viewing Current Resource Limits on page 197.

Supported Tags and Quotas Supported

Supported Tags are tags that you can use to override values. Curly braces indicate variable tags. For example, `{TENANCY}` indicates that you can set different limit values on different tenancies. In the preceding Service CLI example, you can set different `groups` limits for different tenancies. You cannot set different `backends` limits for different load balancers; all load balancers have the same `backends` limit.

If Quotas Supported is Yes or true, a Compute Enclave user with sufficient authorization can create additional compartment-specific limits within the specified tenancy.

Viewing Current Resource Limits

This section describes how to show the current limit values of the specified resources. If an override has been set, the current limit value could be different from the default value.

Using the Service Web UI

1. On the navigation menu, select Limits, and then select Service Limits.
2. On the Service Name menu at the top of the page, select the service of the resource that you are interested in.
3. On the Tag menu, select the tag.

   All resources that are supported in the Limits Service, that belong to the selected service, and that have the specified tag are listed.

   The "Value" column shows the current value of the limit.

   The "Assigned By" column shows whether the value is the default limit value from the limit definition or an override value set in a template or on a tag (see Setting Resource Limit Overrides on page 201). If the value is an override value set in a template, the name of the template is shown.

   Any of the columns can be sorted by selecting the arrow keys in the column header. Select the Refresh button to re-load the original list sorted alphabetically by limit name.

Using the Service CLI

The `getCurrentServiceLimits` command shows all current limits for the specified service and tag or shows the specified current limit.

The following example shows all current limits for the IAM service and the specified tenancy.

```
PCA-ADMIN> getCurrentServiceLimits serviceName=identity
 tag=ocid1.tenancy.unique_ID
Data:
  Limit Name   Service Name   Value   Assigned By    Tag
        Template Name
  ----------   ------------   -----   -----------    ---
        -------------
  users        identity       200     Template
 ocid1.tenancy.unique_ID   template_name
  groups       identity       200     Tag Override
 ocid1.tenancy.unique_ID   null
...
```

```
  api-keys     identity       3       Default
 ocid1.tenancy.unique_ID   null
```

The "Assigned By" column shows whether the "Value" is the default limit value from the limit definition or an override value set with `create LimitOverride` in a template or on a tag (see Setting Resource Limit Overrides on page 201). If the value is an override value set in a template, the name of the template is shown.

Provide the limit name to show only that particular limit for the specified service and tag.

```
PCA-ADMIN> getCurrentServiceLimits serviceName=identity
 tag=ocid1.tenancy.unique_ID limitName=users
Data:
  Limit Name   Service Name   Value   Assigned By   Tag
       Template Name
  ----------   ------------   -----   -----------   ---
       -------------
  users        identity       200     Template
 ocid1.tenancy.unique_ID   template_name
```

Working with Resource Limit Templates

Resource limit templates enable you to easily apply a set of limits for Private Cloud Appliance resources.

For example, you can create a template to apply one set of resource limits to two different tenancies and create another template to apply a second set of resource limits to three other tenancies. You can apply no more than one template to a tag.

Templates only need to specify limit values that are different from the default. If a limit is not specified, then its value is the default value from the limit definition if the template's default value behavior is set to `INHERIT`.

The commands in this topic create a resource limit template, list the available templates, show more information about the specified template, assign a template to a tag, unassign a template, and delete the template.

Viewing Resource Limit Templates

Using the Service Web UI

1. On the navigation menu, select Limits, and then select Limit Templates.

   For each limit template, the table shows the name of the template, whether the template allows tag overrides, and the default value behavior: Inherit or Zero.
2. To show the details of a particular limit template, either select the name of the limit override in the list, or select the Actions menu for that limit override and select View Details.

   The Resources section of the details page shows overrides that have already been created for this limit template. You can edit or delete an override in that list, view details of an override, or use the button above the list to create a new override for this template. See Creating Resource Limit Overrides on page 202.

Using the Service CLI

The `list LimitTemplate` command lists all available resource limit templates. Both templates shown in the following example allow you to override the limit values in the template.

```
PCA-ADMIN> list LimitTemplate
Data:
  id           Name         Allows Overrides   Default Value Behavior
  --           ----         ----------------   ----------------------
  template1    template1    true               INHERIT
  template2    template2    true               ZERO
```

The `show LimitTemplate` command shows more information for the specified template, particularly overrides. Provide the template `id` from the output of the `list LimitTemplate` command.

In the following example, the default value of 100 for `identity~users` was overridden by using `create LimitOverride` (see Setting Resource Limit Overrides on page 201).

```
PCA-ADMIN> show LimitTemplate id=template1
Data:
  Id = template1
  Type = LimitTemplate
  Name = template1
  Allows Overrides = true
  Default Value Behavior = INHERIT
  Overrides 1 - Limit Name = users
  Overrides 1 - Service Name = identity
  Overrides 1 - Value = 200
```

Creating Resource Limit Templates

This section describes how to create a resource limit template.

By default (Allow Overrides is True), if the same limit is set both in the template and directly on the tag, the limit that is set directly on the tag is used.

When Allow Overrides is false, limits set by the template are used; any limits set directly on the tag are ignored. If a limit for the tag to which the template is assigned is not explicitly set in the template, then the default value of that limit is used or 0 is used, depending on the value of Default Value Behavior.

By default (the value of Default Value Behavior is Inherit), any limit values that are not explicitly set in this template inherit their default value from their limit definition.

When the value of Default Value Behavior is Zero, any limit values that are not explicitly set in this template are set to 0: No new resources of that type can be created. Existing resources are not affected. Unassigning this template restores the ability to create new resources for the tag.

Using the Service Web UI

1. On the navigation menu, select Limits, and then select Limit Templates.
2. Select the Create Limit Template button above the list.
3. On the Create Limit Template dialog, enter the following information:
   - Name: Maximum 40 characters.
   - Default Value Behavior: Select either Inherit or Zero. Inherit is selected by default.
   - Allow Overrides: Select either True or False. True is selected by default.
4. Select the Create Limit Template button in the dialog.

   On the details page for this template, scroll to the Resources section to define limit overrides for this template. See Creating Resource Limit Overrides on page 202.

Using the Service CLI

Use the `create LimitTemplate` command to create a new template with the specified template name.

The following examples create resource limit templates. The `create LimitTemplate` command does not set any limit values. Use the `create LimitOverride` command, specifying a template instead of a tag, to set limit values for the specified template. See Setting Resource Limit Overrides on page 201.

In the following example, `defaultValueBehavior` is set to `INHERIT` by default:

```
PCA-ADMIN> create LimitTemplate name=template1
Status: Success
JobId: a2f6df33-e70e-408b-aee5-7583b24e53ce
```

In the following example, `defaultValueBehavior` is explicitly set to `ZERO`:

```
PCA-ADMIN> create LimitTemplate name=template2
 defaultValueBehavior=ZERO
Status: Success
JobId: a2f6df33-e70e-408b-aee5-7583b24e53ce
```

Assigning Resource Limit Templates

This section describes how to assign a limit template to a tag.

At most one limit template can be assigned to a given tag. If you assign a second template to a tag, the second template replaces the first template; you do not need to unassign the first template.

If you assign a template to a tag and a limit for that tag is directly applied to the tag, then the Allow Overrides setting determines which limit value is used:

- If Allow Overrides is false for the template, then the limit value specified in the template is used. If a limit value is not explicitly set in the template, then the default value of that limit is used or 0 is used, depending on the value of Default Value Behavior.
- If Allow Overrides is true for the template, then the limit value that was already specified on the tag is used.

Using the Service Web UI

1. On the navigation menu, select Limits, and then select Limit Template Assignments.

   For each limit template assignment, the template name and the tag are shown.
2. To assign a limit template, select the Assign Limit Template button above the template assignments list.
3. On the Create Limit Template Assignment dialog, enter the following information:
   - Template Name: Select the template name from the drop-down menu.
   - Tag: Select the tag from the drop-down menu.
   - Allow Overrides: Select either True or False. True is selected by default.
4. Select the Create Limit Template Assignment button on the dialog.

Using the Service CLI

Use the `assignLimitTemplate` command to assign the specified template to the specified tag.

```
PCA-ADMIN> assignLimitTemplate templateName=template1
 tag=ocid1.tenancy.unique_ID1
Status: Success
JobId: 4fe99f37-554d-4ce8-8084-639f6069cc44
```

Use the `list LimitTemplateAssignment` command to list all templates and the tags to which they are assigned. In the following example, template1 is assigned to two different tags:

```
PCA-ADMIN> list LimitTemplateAssignment
Data:
  Template Name   Tag
  -------------   ---
  template1       ocid1.tenancy.unique_ID1
  template1       ocid1.tenancy.unique_ID2
  template3       SYSTEM
```

Unassigning Resource Limit Templates

This section describes how to unassign a limit template from a tag.

When a template is unassigned, the limits set in that template are removed from the affected tags.

- If you specify only a template name, that template is unassigned from all tags to which it had been assigned.

- If you specify both a template name and a tag, the specified template is unassigned from the specified tag.
- If you specify only a tag, any template that had been assigned to that tag is unassigned from that tag.

Using the Service Web UI

1. On the navigation menu, select Limits, and then select Limit Template Assignments.

   For each limit template assignment, the template name and the tag are shown.
2. To unassign a limit template, select the Actions menu for that template and then select the Unassign option, or select the Unassign Limit Templates button above the template assignments list.
3. On the Unassign Limit Templates dialog, enter the following information:

   - Template Name: Select the template name from the drop-down menu.
   - Tag: Select the tag from the drop-down menu.

   You must specify at least one template name or tag. You can specify multiple template names or tags, or multiple template name and tag pairs. See the rules at the beginning of this topic.
4. Select the Unassign Limit Templates button on the dialog.

Using the Service CLI

Use the `unassignLimitTemplate` command to unassign a limit template.

```
PCA-ADMIN> unassignLimitTemplate templateName=template1
 tag=ocid1.tenancy.unique_ID2
Status: Success
JobId: 48fa692d-a997-48da-8b60-309bc4be7ad1
```

Deleting Resource Limit Templates

This topic describes how to delete a resource limit template. If the template overrides a limit that is set per tenancy, you might want to check which tenancies the template is assigned to. See Assigning Resource Limit Templates on page 200.

Using the Service Web UI

1. On the navigation menu, select Limits, and then select Limit Templates.
2. For the limit template that you want to delete, select the Actions menu, and select the Delete option.
3. Confirm the delete operation.

Using the Service CLI

Use the `delete LimitTemplate` command to delete the specified template.

```
PCA-ADMIN> delete LimitTemplate id=template2
Status: Success
JobId: 62ae5905-3150-4724-a921-727f61dcdcdc
```

Setting Resource Limit Overrides

In Private Cloud Appliance, a resource limit override is a value that replaces the limit value that is currently set in the specified template or for the specified tag.

A resource limit override is a value that replaces the limit value that is currently set in the specified template or for the specified tag. Before you create a limit override, see Viewing Resource Limits on page 196 to see the current limit value and the default limit value. See Planning Resource Limit Settings on page 204 to see the allowable limit values.

This section describes how to list all resource limit overrides, set and edit overrides, and delete overrides.

Listing Resource Limit Overrides

Using the Service Web UI

1. On the navigation menu, select Limits, and then select Limit Overrides.

   For each limit override, the table shows the name of the limit, the name of the service, the limit value set by the override, the tag, and the template name if applicable.
2. To show the details of a particular limit override, either select the name of the limit override in the list, or select the Actions menu for that limit override and select View Details.

Using the Service CLI

Use the `list LimitOverride` command to return all limit overrides, for both templates and tags.

```
PCA-ADMIN> list LimitOverride
Data:
  id                                              Limit Name      Service
 Name  Value  Tag                          Template Name
  --                                              ----------
 ------------  -----  ---                          -------------
  identity~users~TAG~ocid1.tenancy.unique_ID  users           identity
     200    ocid1.tenancy.unique_ID  null
  loadbalancer~load-balancers~TAG~SYSTEM      load-balancers
 loadbalancer  10     SYSTEM                       null
  loadbalancer~listeners~TEMPLATE~template3   listeners
 loadbalancer  10     null                         template3
```

Creating Resource Limit Overrides

This section describes how to create a resource limit override. The new limit value can be fewer than the current number of resources. Existing resources are not affected by setting a new limit. If the new limit is fewer than the current number, you will not be able to create a new one of those resources until their number drops below the new, lower, limit.

When you create an override, you specify either a tag or a template. If the same limit is set by specifying a tag name and also by specifying a template name, see Assigning Resource Limit Templates on page 200 for the rule for which limit value will be used.

See Planning Resource Limit Settings on page 204 for limit default values and allowable values. You might not be able to set a limit override to the maximum allowable value. The limit you can set depends on other configuration, including limit overrides that you have already set for related resources. See Listing Resource Limit Overrides on page 202 for the list of limits that are already overridden.

If creating a limit override fails because you have attempted to set the limit too high, you will receive an error message with information about why the limit is too high and what the limit can be. Using this information, you might decide to override a different limit as well as, or instead of, the limit you were attempting to override.

Using the Service Web UI

1. On the navigation menu, select Limits, and then select Limit Overrides.
2. Select the Create Limit Override button above the list.
3. On the Create Limit Override dialog, enter the following information:
   - Select the service name from the menu.
   - Select the limit name from the menu.
   - Enter the new value for the resource limit.
   - Select either a tag from the Tag menu or a template from the Template menu. If both are selected, the override defaults to the selected template.
4. Select the Create Limit Override button in the dialog.

Using the Service CLI

Use the `create LimitOverride` command to create a resource limit override for either a template or a tag.

Provide the name of the limit, the name of the service, and the new resource limit value. Provide either a template name or a tag name.

The following example sets a limit override for a tenancy tag:

```
PCA-ADMIN> create LimitOverride limitName=users serviceName=identity
 value=200 tag=ocid1.tenancy.unique_ID
Status: Success
JobId: a5d75c66-8708-4355-b5ef-1a3213f9b690
```

The following example sets a limit override for a `SYSTEM` tag:

```
PCA-ADMIN> create LimitOverride limitName=load-balancers
 serviceName=loadbalancer value=10 tag=SYSTEM
Status: Success
JobId: job_ID
```

The following example sets a limit override in a template:

```
PCA-ADMIN> create LimitOverride limitName=listeners
 serviceName=loadbalancer value=10 template=template3
Status: Success
JobId: job_ID
```

Updating Resource Limit Overrides

This section describes how to edit a limit override value. See Planning Resource Limit Settings on page 204 for limit default values and allowable values. You might not be able to set a limit override to the maximum allowable value. The limit you can set depends on other configuration, including limit overrides that you have already set for related resources.

If the current limit is set by a template, see Working with Resource Limit Templates on page 198 to see what other limits are set in that template and what tags the template is assigned to.

Using the Service Web UI

1. On the navigation menu, select Limits, and then select Limit Overrides.
2. For the limit override that you want to edit, select the Actions menu, and select the Edit option.
3. The Update Limit Override dialog displays the current limit value in a text field. You can change that displayed value.
4. Select the Update Limit Override button in the dialog.

Using the Service CLI

Use the `edit LimitOverride` command to reset the value of the specified resource limit. Use the `id` from the `list LimitOverride` command output. See Listing Resource Limit Overrides on page 202.

```
PCA-ADMIN> edit LimitOverride
 id=loadbalancer~listeners~TEMPLATE~template3 value=8
Status: Success
JobId: 37baa077-6d75-4901-86c5-82568c514b26
```

Deleting Resource Limit Overrides

This section describes how to delete a resource limit override. Deleting the override deletes the limit value for the tag or template.

- The following rules describe what the value of the limit will be when more than one limit value is set and one is deleted:
- If you delete the limit value of a tag, and a template with an override for the same limit is assigned to that tag, then the limit value from the template is used.
- If you delete the limit value of a tag, and no template is assigned to that tag, then the default limit value from the limit definition is used.
- If you delete the limit value of a tag, and a template is assigned to that tag that does not include an override for the same limit, then the default limit value from the limit definition is used if `defaultValueBehavior` is `INHERIT` for the template.
- If you delete the limit value of a template that is assigned to a tag, and an override for the same limit is assigned to that tag, then the limit value assigned to the tag is used if `allowsOverrides` is true for the template.
- If you delete the limit value of a template that is assigned to a tag, and no other override for the same limit is assigned to that tag, then the default limit value from the limit definition is used if `defaultValueBehavior` is `INHERIT` for the template.

Using the Service Web UI

1. On the navigation menu, select Limits, and then select Limit Overrides.
2. For the limit override that you want to delete, select the Actions menu, and select the Delete option.
3. Confirm the delete operation.

Using the Service CLI

Use the `delete LimitOverride` command to delete the limit value for the tag or template.

Use the `id` from the `list LimitOverride` command output to delete the limit value.

The following example deletes the value for the load-balancers limit that is set for the `SYSTEM` tag:

```
PCA-ADMIN> delete LimitOverride id=loadbalancer~load-
balancers~TAG~SYSTEM
Status: Success
JobId: 3bc59401-d8cd-41ba-ab75-140a8fcd087f
```

The following example deletes the value for the listeners limit that is set for `template3`:

```
PCA-ADMIN> delete LimitOverride
 id=loadbalancer~listeners~TEMPLATE~template3
Status: Success
JobId: job_ID
```

Planning Resource Limit Settings

The tables in this topic show the maximum allowable limit for each Private Cloud Appliance resource and tag.

> **Important:**
>
> You might not be able to set a limit override to the maximum allowable value. The limit you can set depends on other configuration, including limit overrides that you have already set for related resources.

If creating a limit override fails because you have attempted to set the limit too high, you will receive an error message with information about why the limit is too high and what the limit can be. Using this information, you might decide to override a different limit as well as, or instead of, the limit you were attempting to override.

In the following example, the maximum value for load-balancers is 12, and the user attempts to set a new value of 14.

```
PCA-ADMIN> create LimitOverride serviceName=loadbalancer limitName=load-
balancers value=14 tag=SYSTEM
Status: Failure
Error Msg: PCA_GENERAL_000037: Error returned from Limit service. Code:
 'InvalidParameter'.
```

```
Message: 'Rule 'lb' violated: {loadbalancer/load-balancers} <= 12'
```

Some of the limits in the following tables do not show a range in the Allowable Limit column. They show the same value in the Allowable Limit column that they show in the Default Limit column. These limit values cannot be overridden.

```
PCA-ADMIN> create LimitOverride serviceName=network limitName=subnet-count
 tag=SYSTEM value=30
Status: Failure
Error Msg: PCA_GENERAL_000037: Error returned from Limit service. Code:
 'NotSupportedError'.
Message: 'Limit 'network/subnet-count' does not support overrides.'
```

Use the procedures described in Viewing Resource Limits on page 196 with the service names and limit names shown in the following tables to get additional information such as the limit description and supported tags.

File System Limits

All File System service limits have service name `file-system`.

The maximum allowable limit might not be possible in your environment. See the Important note at the beginning of this topic.

Limit Name	Default Limit	Maximum Allowable Limit
file-system-count	800	0 - 800
file-system-sys-count	800	0 - 800
mount-target-count	80	0 - 80
mount-target-sys-count	80	0 - 80

IAM Service Limits

All IAM service limits have service name `identity`.

The maximum allowable limit might not be possible in your environment. See the Important note at the beginning of this topic.

Limit Name	Default Limit	Maximum Allowable Limit
users	100	0 - 800
groups	100	0 - 800
dynamic groups	50	0 - 400
compartments	50	0 - 400
policies	100	0 - 800
policy-statements	50	0 - 400
group-users	100	0 - 800
user-groups	100	0 - 800

Limit Name	Default Limit	Maximum Allowable Limit
idps	3	0 - 24
idp-group-mappings	100	0 - 800
tag-namespaces	100	0 - 800
tags	100	0 - 800
tag-defaults	5	0 - 40
cost-tracking-tags	10	0 - 80
api-keys	3	0 - 24

Load Balancer Service Limits

All Load Balancer service limits have service name `loadbalancer` and tag `SYSTEM`.

The maximum allowable limit might not be possible in your environment. See the Important note at the beginning of this topic.

Limit Name	Default Limit	Maximum Allowable Limit
listeners	16	0 - 16
backend-sets	16	0 - 16
backends	512	0 - 512
hostnames	4	0 - 4
path-route-sets	16	0 - 16
network-security-groups	5	0 - 5
certificates	16	0 - 16
cipher-suites	16	0 - 16
load-balancers	12	0 - 12
load-balancers-min-total	36	36
load-balancers-total	144	0 - 144

Note that the `load-balancers-min-total` limit cannot be overridden. The limit value of `load-balancers-min-total` can only be 36. The only exception is if you assign a template to SYSTEM with no limit value set for `load-balancers-min-total` and `defaultValueBehavior` set to `ZERO`.

You can set `load-balancers-total` as high as 144, but the effective limit depends on how many compute nodes the system has. For example, if you have six compute nodes, the effective `load-balancers-total` is 72, even if you set the limit higher.

Network Limits

All Network service limits have service name `network`.

The maximum allowable limit might not be possible in your environment. See the Important note at the beginning of this topic.

Limit Name	Default Limit	Maximum Allowable Limit
vcn-system-count	80	80
subnet-system-count	320	320
subnet-count	40	40
vcn-count	80	0 - vcn-system-count
nsg-count	100	0 - 100
nsg-rule-count	50	0 - 300

Storage Controller Limits

All Storage Controller limits have service name `block-storage` and tag `SYSTEM`.

The maximum allowable limit might not be possible in your environment. See the Important note at the beginning of this topic.

Limit Name	Default Limit	Maximum Allowable Limit
vol-attachment-per-instance-count	32	0 - 32
vol-group-volume-count	32	0 - 32

Creating and Managing Flex Networks

Direct connections between Private Cloud Appliance and external systems can be configured through *Flex Networks*. These also enable traffic for specific workloads to exit the rack through different physical ports on the spine switches.

You can create a Flex network in either Edge or Hub mode. The main difference between the modes is that you must configure a gateway for Hub mode.

Flex networks are managed from the Service Enclave. Before you can create a Flex network, you must physically connect your Private Cloud Appliance to an external system. For instructions, see Connecting External Systems to Private Cloud Appliance Using Flex Networks on page 124. For more general information about flex networks, see Integrating External Systems with Flex Networks on page 47.

In order to *use* a Flex network, the VCNs containing compute instances that connect to external system components must have a dynamic routing gateway (DRG) configured. The enabled subnet needs a route rule with the external system CIDR as destination and the DRG as target.

> **Note:**
>
> If a Flex network is in the **provisioning** or **updating** or **terminating** state, then a user cannot attach or detach a DRG or IGW, or create a NAT gateway. These operations need to be done when the Flex network is in the **available** or **terminated** state.

You can configure up to 128 Flex networks per appliance, with a limit of 32 Flex networks per port.

Attention:

This feature was formerly called *Exadata Networks*. The previous commands have been deprecated and replaced by Flex Network commands. The following table describes both sets of commands.

Flex Network Commands

Deprecated Commands	New Commands
PCA-ADMIN> exaDataCreateNetwork ?	PCA-ADMIN> create FlexNetwork ?
*cidr	*cidr
*ports	*ports
*spine1Ip	*spine1Ip
*spine2Ip	*spine2Ip
*spineVip	*spineVip
advertiseNetwork	advertiseNetwork
exadataSpeed	speed
gatewayIp	gatewayIp
vlan	vlan
PCA-ADMIN> exaDataGetNetwork ?	PCA-ADMIN> show FlexNetwork ?
*exadataNetworkId	id=<object identifier>
PCA-ADMIN> exaDataListNetwork	PCA-ADMIN> list FlexNetwork
PCA-ADMIN> exaDataUpdateNetwork ?	PCA-ADMIN> edit FlexNetwork id=123 ?
*exadataNetworkId	*ports
*ports	
PCA-ADMIN> exaDataDeleteNetwork ?	PCA-ADMIN> delete FlexNetwork ?
*exadataNetworkId	id=<object identifier>
PCA-ADMIN> exaDataEnableAccess ?	PCA-ADMIN> flexNetworkEnableAccess ?
*exadataNetworkId	*flexNetworkId
*subnetId	*subnetId
PCA-ADMIN> exaDataDisableAccess ?	PCA-ADMIN> flexNetworkDisableAccess ?
*exadataNetworkId	*flexNetworkId
*subnetId	*subnetId

Creating and Enabling a Flex Network

Learn about the system parameters you need to gather, and the required operations to set up a network connection between Private Cloud Appliance and an external system.

Flex Network Task Map

This task map describes the steps required to establish a Flex network between Private Cloud Appliance and an external system such as Oracle Exadata or Oracle Database Appliance.

No.	Task	Links
1	Identify the physical ports on the Spine switch you plan to use for the external connection, then cable the hardware together.	Connecting External Systems to Private Cloud Appliance Using Flex Networks on page 124
2	Create the Flex network from the Service enclave.	Adding the New Flex Network on page 211
3	From the Compute enclave, create a DRG to provide a way for compute instances to access the external system.	Creating a Dynamic Routing Gateway on page 511
4	From the Compute enclave, create VCNs, Subnets, Route Tables and Internet Gateways, if needed. See *Other Considerations* following this table.	Creating a VCN on page 479 Managing VCNs and Subnets on page 478
5	From the Compute enclave, create DRG attachments to enable VCN to use DRG.	Attaching VCNs to a Dynamic Routing Gateway on page 512
6	From the Service enclave, enable communication between the Flex network and the VM subnets.	Enabling Flex Network Access on page 212

Other Considerations

When implementing a Flex network, consider the following:

- If a VM connected to a Flex network must also be accessed from a domain controller, you need to configure a second VNIC for that VM. See Creating and Attaching a Secondary VNIC on page 520.
- Use an Internet Gateway (public subnet) for Domain Controller access using the primary VNIC.
- Use a Dynamic Routing Gateway (private subnet) to access the Flex network.
- Use separate Route Tables: one for the Internet Gateway with `0.0.0.0/0` and one for the DRG with a *specific* route rule for the Flex network.
- Update Security Lists as needed to enable ingress traffic.

Flex Network with Route Table

In its original design, a flex network provides direct connectivity between a VCN and external devices, for example Oracle Exadata nodes or a ZFS Storage Appliance. This design is known as *edge mode*, and allows you to optionally advertise the flex network from the spine switches to the on-premises network.

A flex network in *hub mode* uses a gateway on the outside of Private Cloud Appliance instead of direct connections, to route traffic between external devices in an on-premises network and compute instances in a VCN. Flex network access is controlled by a route table, which is configured through the `gatewayIp` parameter.

Traffic between VCNs can also be routed through a flex network, for example if a firewall connected to the flex network needs to inspect the traffic.

Up to 6 route table entries can be specified in a flex network configuration. The syntax is as follows:

- "`r:`" - route table entry (1 to 6)
- "`net_x`" - destination subnet
- "`gw_x`" - gateway IP for the associated subnet
- "`:`" (colon) - separator between route table entries

```
r:<net_1>,<gw_1>:<net_2>,<gw_2>:<net_n>,<gw_n>
```

For example:

```
r:10.25.0.0/26,10.212.3.8:172.16.48.0/22,10.212.3.30:0.0.0.0/0,10.212.3.15
```

adds route table entries for:

```
10.25.0.0/26 via 10.212.3.8
172.16.48.0/22 via 10.212.3.30
0.0.0.0/0 (default) via 10.212.3.15
```

> **Important:**
>
> In earlier Private Cloud Appliance software versions, these restrictions apply:
>
> - Only a single gateway IP address is accepted. It's configured as the default gateway, providing access to any connected subnet.
> - When a gateway is configured, the flex network cannot be advertised to the data center network. These settings are mutually exclusive.
> - When a VCN has access to a flex network, overlapping CIDRs are not allowed.
> - VCN-to-VCN traffic cannot flow through the flex ports on the spine switches, only through the uplink ports.

Required Parameters

To set up a network connection between Private Cloud Appliance and an external system, you need this set of parameters:

Parameter	Example Value	Description
cidr	10.*nn*.*nn*.0/24	Choose a valid CIDR range that is within the CIDR range of the external system.
spine1Ip	10.*nn*.*nn*.2	A valid IP address in the CIDR specified.
spine2Ip	10.*nn*.*nn*.3	A valid IP address in the CIDR specified.
spineVip	10.*nn*.*nn*.1	A valid IP address in the CIDR specified.
vlan	3062	Choose a VLAN from 2 to 3899 that isn't in use as the uplink VLAN or other Flex network VLANs. This parameter can be unspecified for attaching a device not supporting VLAN tagging.
speed	10	Speed of the aggregated switch links under the port-channel must be 10, 20, 25, 40, 50, or 100 Gbit.

Parameter	Example Value	Description
ports	7/1	Ports 7/1-4, 8/1-4, 9/1-4, or 10/1-4 are valid for 10G or 25G speeds. Ports 7, 8, 9, or 10 are valid for 40G or 100G speeds. For more detail, see the next table.
gateway IP or route table entry	10.*nn.nn.nn*	Valid IP address of gateway. Default is null.
	`r:172.16.0.0/24,10.nn.nn.nn`	Route to destination subnet (max. 6) through gateway, entered as `r:<subnet_cidr>,<gateway_ip>`
advertiseNetwork	True	True or False. Enables or disables the visibility of the Flex network to the customer data center servers.

Note:

When enabling a flex network, compute instance access to the uplink depends on the active Private Cloud Appliance controller software version. With version 3.0.2-b1483396 and later, uplink access is provided for flex networks in edge mode (without gateway) through the external system VRF. With older software versions, compute instances can connect using a separate interface through an internet or NAT gateway.

Valid speeds and valid port configurations are related. The following table shows the valid port configurations based on speed selected. Ports must be bonded on the external system side to match the Private Cloud Appliance configuration.

Speed	Valid Port Configurations
10 Gbit	7/1-4, 8/1-4, 9/1-4, or 10/1-4
20 Gbit	7/1-2, 8/1-2, 9/1-2, or 10/1-2 (20G bonds two 10G ports)
25 Gbit	7/1-4, 8/1-4, 9/1-4, or 10/1-4
40 Gbit	7, 8, 9, or 10
50 Gbit	7/1-2, 8/1-2, 9/1-2, or 10/1-2 (50G bonds two 25G ports)
100 Gbit	7, 8, 9, or 10

Note:

For 25G Flex networks, forward error correction (FEC) is always set to ***off***, with or without a gateway.

Adding the New Flex Network

Using the Service Web UI

1. Determine the Flex network parameters. See Required Parameters on page 210
2. In the Dashboard, click the Racks quick action tile.
3. In the PCA Config navigation menu on the Racks page, click Flex Networks.
4. In the top-right corner above the table, click Create Flex Network.
5. Fill out the Flex Network form using the parameters you collected in advance.

By default the network is not advertised to the data center network. You have to click the slider to set it to "on"/"true".

6. Click Submit to create the new network. It appears in the Flex Networks table and its Lifecycle State changes to Available when the configuration has been applied successfully.
7. Next, add a subnet to the Flex network. See Enabling Flex Network Access on page 212.

Using the Service CLI

1. Determine the Flex network parameters. See Required Parameters on page 210
2. Create the Flex network by entering the parameters.

```
PCA-ADMIN> create flexNetwork cidr=10.nn.nn.0/24  \
spine1Ip=10.nn.nn.1 spine2Ip=10.nn.nn.2 spinevip=10.nn.nn.3 \
vlan=900 gatewayIp=10.nn.nn.10 ports=7/1 advertiseNetwork=false
Status: Success
JobId: unique_id
```

3. Next, add a subnet to the Flex network. See Enabling Flex Network Access on page 212.

Enabling Flex Network Access

Enable access from a subnet to the Flex network through the Service CLI. For Flex network access from that subnet, ensure that the configured IP address ranges of Flex networks do not overlap.

> **Note:**
>
> Overlapping CIDRs can be used if the flex network configuration is considered safe and passes internal system tests. There must be clean separation between flex networks and spine switch VRFs.

Subnets that have been granted access, appear in the Flex network detail page under Access Lists, grouped by their parent VCN.

Using the Service CLI

1. Get the OCID of the Flex network you want to enable, using the `list FlexNetwork` command.
2. Enable access to a configured Flex network.

```
PCA-ADMIN> flexNetworkEnableAccess
 flexNetworkId=ocid1.exadata.unique_id \
subnetId=ocid1.subnet.unique_id
Status: Success
Data:
 id
 --
 ocid1.vcn.unique_id
```

3. If you are using a secondary VNIC to access the Flex network, you must add a route to the Flex network CIDR address range for interface `eth1` (the secondary VNIC). Sign in to the compute instance configured with the secondary VNIC to add the route.

```
[root@hostname]# Flex-CIDR-address-range via gateway dev vlan-
interface
```

For example, if the Flex address range is `192.168.0.0/24` and the gateway is `192.168.0.1` and the VLAN interface is `bond0.900`:

```
[root@hostname]# 192.168.0.0/24 via 192.168.0.1 dev bond0.900
```

This entry appears as a second interface in the IP routing table:

```
Destination Gateway      Genmask       Flags Metric Ref Use Iface
 . . . . . . . . . . . . . . . . . . . . . . . .
 192.168.1.0 192.168.1.1 255.255.255.0 0     0      0       eth0
 192.168.0.0 192.168.0.1 255.255.255.0 0     0      0       eth1
```

A ping from the secondary VNIC, `eth1`, now succeeds to the Flex network.

Verifying and Updating the Flex Network Configuration

Learn how to verify the setup of existing Private Cloud Appliance Flex Networks, and update their configurations.

Verifying Flex Network Configurations

Using the Service Web UI

1. In the Dashboard, click the Rack Units quick action tile.
2. In the PCA Config navigation menu on the Rack Units page, click Flex Networks. The table contains all configured Flex networks.
3. In the overview table, click the name (OCID) of the network for which you want to display details.

   The Flex Network detail page shows the configuration parameters, the state of the network, and the subnets that have been granted access.

Using the Service CLI

1. Use the `list FlexNetwork` command to display configured Flex networks, including their OCIDs.

```
PCA-ADMIN> list FlexNetwork
Data:
  id                        vlan    cidr                spine1Ip
      spine2Ip            spineVip        ports
  --                        ----    ----                --------
      --------            --------         -----
  ocid1.exadata.unique_id 2001  10.nn.nn.0/24   10.nn.nn.101
 10.nn.nn.102   10.nn.nn.1   7/1,7/2
```

2. Get the OCID of the Flex network for which you want details.
3. Use the `show flexNetwork` command to display details about a specific Flex network, including the state of the network, subnet and VCN IDs.

```
PCA-ADMIN> show flexNetwork flexNetworkId=ocid1.exadata.unique_id
Data:
  CIDR = 10.nn.nn.0/24
  Vlan = 2001
  Spine1Ip = 10.nn.nn.101
  Spine2Ip = 10.nn.nn.102
  SpineVip = 10.nn.nn.1
  Ports = 7/1,7/2
  advertiseNetwork = false
  Access List 1 - Vcn Id = ocid1.vcn.unique_id
  Access List 1 - Subnet Ids 1 = ocid1.subnet.unique_id
  Access List 1 - Subnet Ids 1 = ocid1.subnet.unique_id
  Access List 2 - Vcn Id = ocid1.vcn.unique_id
  Access List 2 - Subnet Ids 1 = ocid1.subnet.unique_id
  Lifecycle State = AVAILABLE
  gatewayIp = 10.nn.nn.21
  exaDataSpeed = 100
  name
```

Editing a Flex Network

Using the Service Web UI

1. In the Dashboard, click the Rack Units quick action tile.
2. In the PCA Config navigation menu on the Rack Units page, click Flex Networks. The table contains all configured Flex networks.
3. For the Flex network you want to edit, click the three dots in the Actions column, then click edit.
4. Enter the new ports and click Submit.

Using the Service CLI

Use the `edit flexNetwork` command to add or remove Flex network ports.

```
PCA-ADMIN> edit flexNetwork id=ocid1.exadata.unique_id  ports=7/1-4
Data:
  id                        vlan    cidr                spine1Ip
   spine2Ip           spineVip         ports
  --                        ----    ----                --------
   --------           --------         -----
  ocid1.exadata.unique_id 2001   10.nn.nn.0/24   10.nn.nn.101
 10.nn.nn.102  10.nn.nn.1   7/1-4
```

Disabling Flex Network Access

Disabling access from a subnet to the Flex network must be done through the Service CLI.

Subnets that have been granted access, appear in the Flex network detail page under Access Lists, grouped by their parent VCN. When you disable access for a given subnet, it is removed from the Access Lists.

Using the Service CLI

1. Get the OCID of the Flex network you want to disable, using the `list FlexNetwork` command.
2. Get the OCID of the subnet ID for the Flex network using the `list FlexNetwork` command.
3. Disable access to a configured Flex network.

```
PCA-ADMIN> flexNetworkDisableAccess
 flexNetworkId=ocid1.exadata.unique_id \
subnetId=ocid1.subnet.unique_id
Status: Success
```

Deleting a Flex Network

Using the Service Web UI

1. Ensure that, for the Flex network you intend to delete, access has been disabled first.
2. Navigate to the Flex Network page.
3. Select one of these options to delete the Flex network:
   - In the overview table, open the Actions menu on the right hand side of the row and select Delete. When prompted, click Confirm.
   - Open the Flex network detail page, then click the Delete button in the top-right corner.

Using the Service CLI

1. Ensure that, for the Flex network you intend to delete, access has been disabled first.
2. Get the OCID of the Flex network you want to delete, using the `list FlexNetwork` command.
3. Delete the Flex network.

```
PCA-ADMIN> delete FlexNetwork flexNetworkId=ocid1.exadata.unique_id
```

```
Status: Success
```

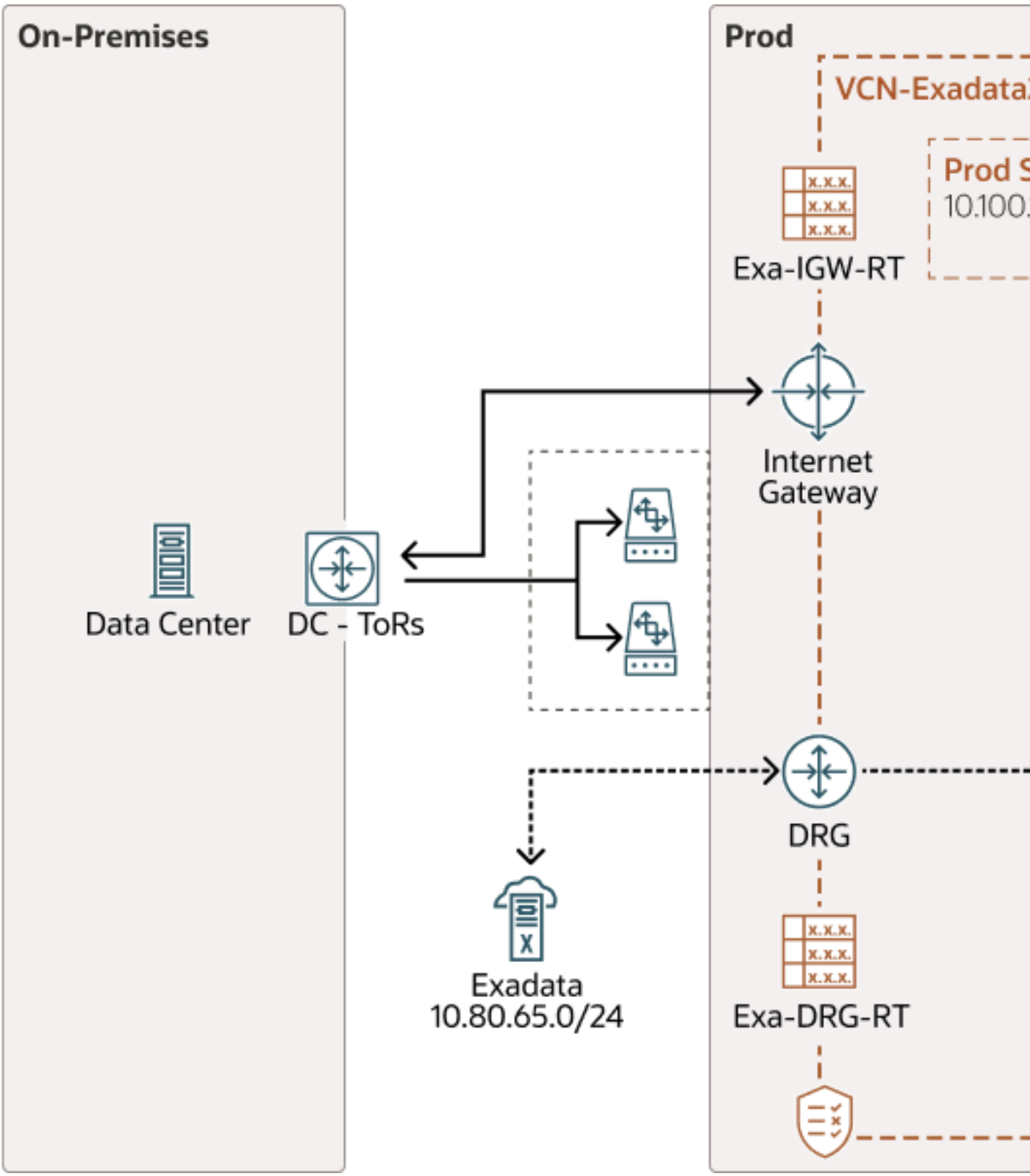
Flex Network Scenarios

This topic presents practical usage scenarios for Private Cloud Appliance Flex networks. Both Hub mode and Edge mode are taken into account.

Configuring a Flex Network in Edge Mode with Connection to Oracle Exadata

This example describes how to create a Flex network in Edge mode between Private Cloud Appliance and an Oracle Exadata.

This example, as shown in the diagram, creates a Flex network with a VCN that contains 3 virtual machines each connected to a private subnet, and also connected to one public subnet. The private subnets are routed through a dynamic routing gateway attached to the VCN, out to Exadata, which provides the compute instances access to Exadata. This example also includes a public subnet, accessible by the compute instances. This public subnet can be configured with an internet gateway to reach the data center ToR switches.



Before You Begin

1. Identify which physical ports on the Private Cloud Appliance spine switches will connect to the external system. See Valid Port Configurations in Creating and Enabling a Flex Network on page 209.
2. Identify the on-premises network subnet and reserve three IP addresses for the spine switches.

Configuration Procedure

1. Create the Flex network from the Service Enclave. See Adding the New Flex Network on page 211.

   To create a Flex network, at a minimum, you need the following parameters:

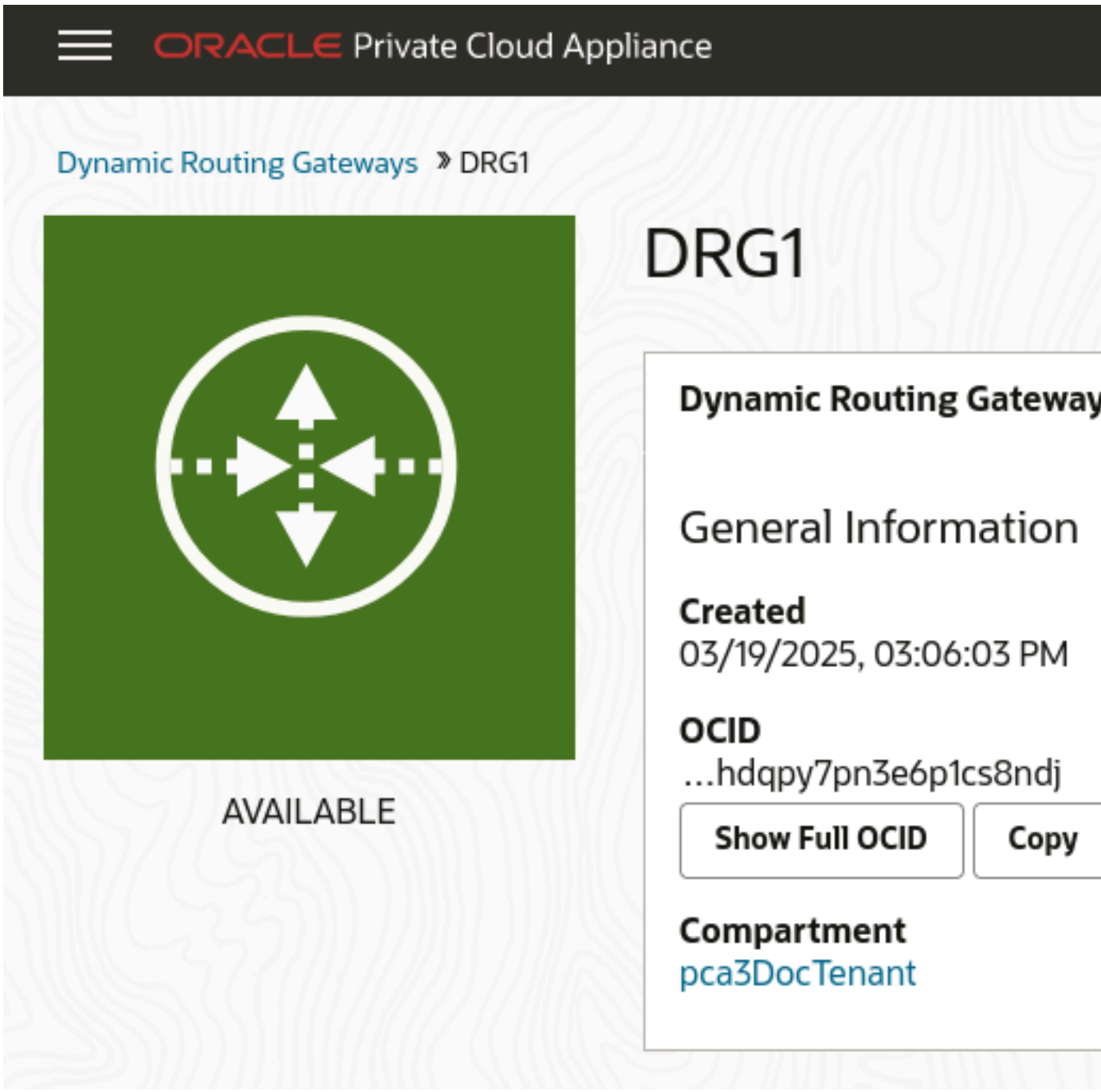
Parameter	Example Value
cidr	10.*nn.nn*.0/24
spine1Ip	10.*nn.nn*.101
spine2Ip	10.*nn.nn*.102
spineVip	10.*nn.nn*.1
vlan	2100
ports	7/1,7/2
advertiseNetwork	True

   Example:

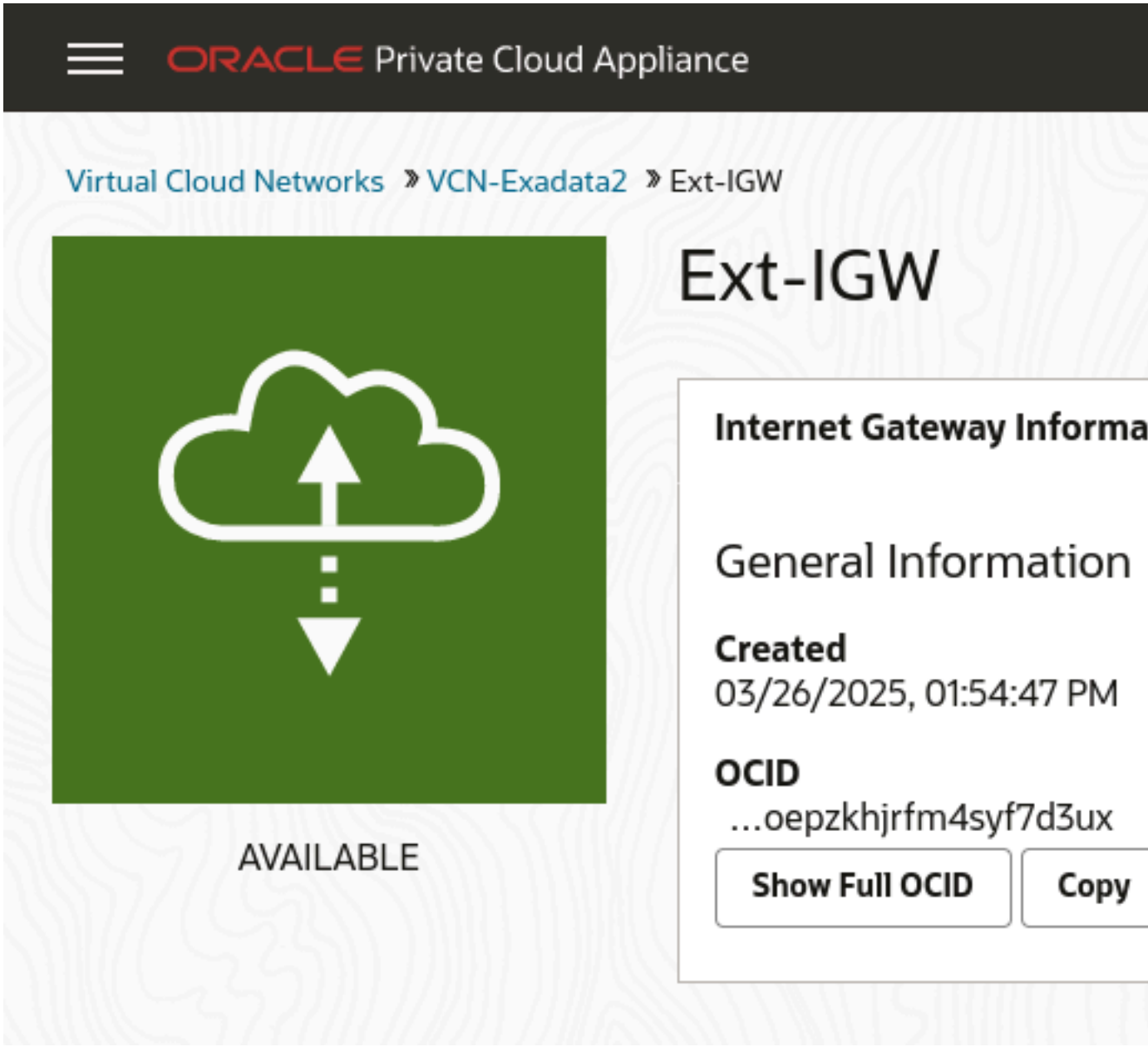
```
PCA-ADMIN> create flexNetwork cidr=10.nn.nn.0/24 \
spine1Ip=10.nn.nn.101 spine2Ip=10.nn.nn.102 spinevip=10.nn.nn.1 \
vlan=2100 ports=7/1 advertiseNetwork=true
Status: Success
JobId: 165f366-64c0-495e-sab1-34s8824b0da

PCA-ADMIN> list flexNetwork
Data:
  id                                    Vlan   CIDR            Spine1Ip
 Spine2Ip      SpineVip    Ports
  --                                    ----   ----            --------
 --------      --------    -----
  ocid1.cccexadata2.oc1.<unique_id> 2100  10.nn.nn.0/24   10.nn.nn.101
 10.nn.nn.102  10.nn.nn.1  7/1,7/2
```

   Note the OCID of the Flex network. You need this OCID to enable the Flex network in step 5.

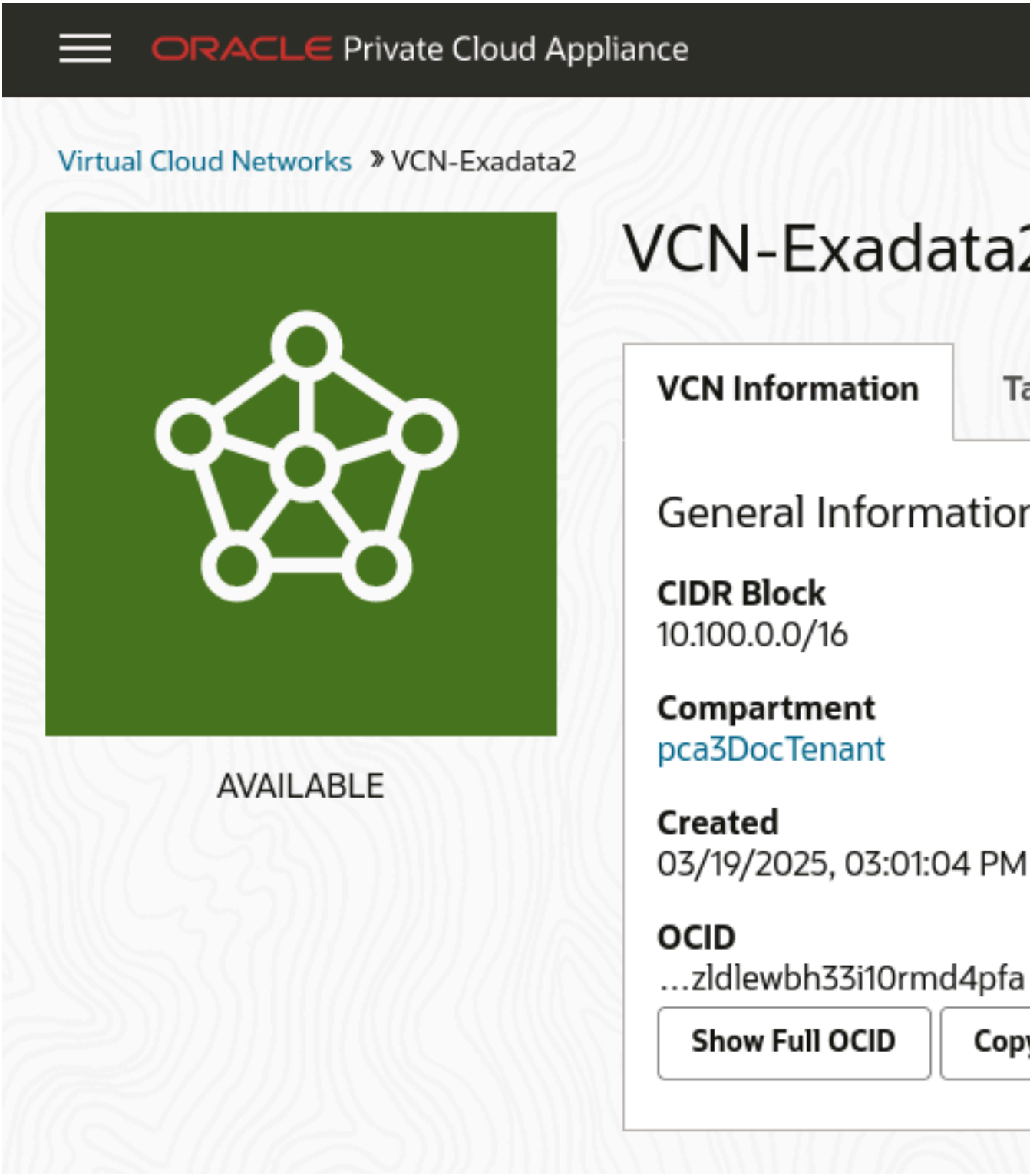
2. From the Compute Enclave, create a dynamic routing gateway (DRG) to provide a way for compute instances to access the external system. See Creating a Dynamic Routing Gateway on page 511.



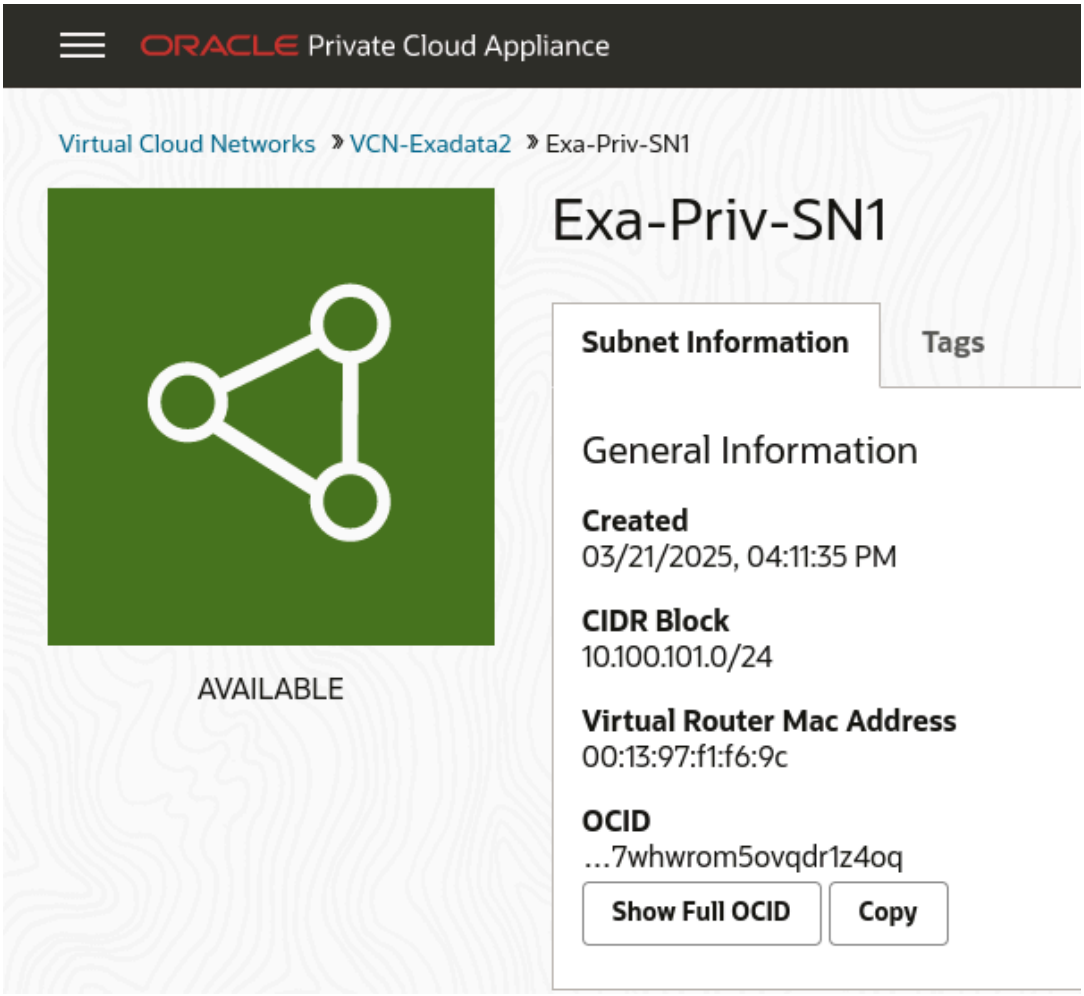
3. From the Compute Enclave, create an Internet Gateway (IGW) to provide a way for compute instances to access the data center switches. See Providing Public Access through an Internet Gateway on page 507.

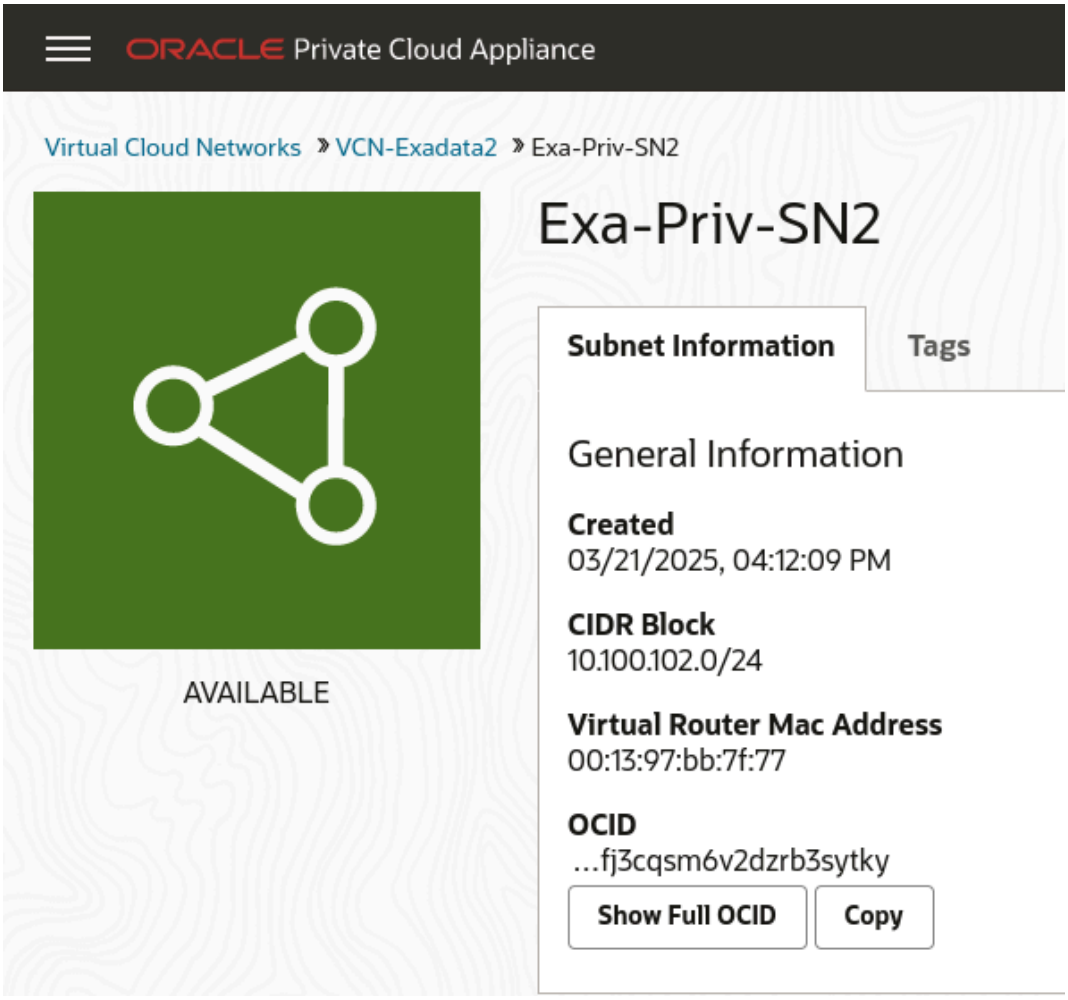


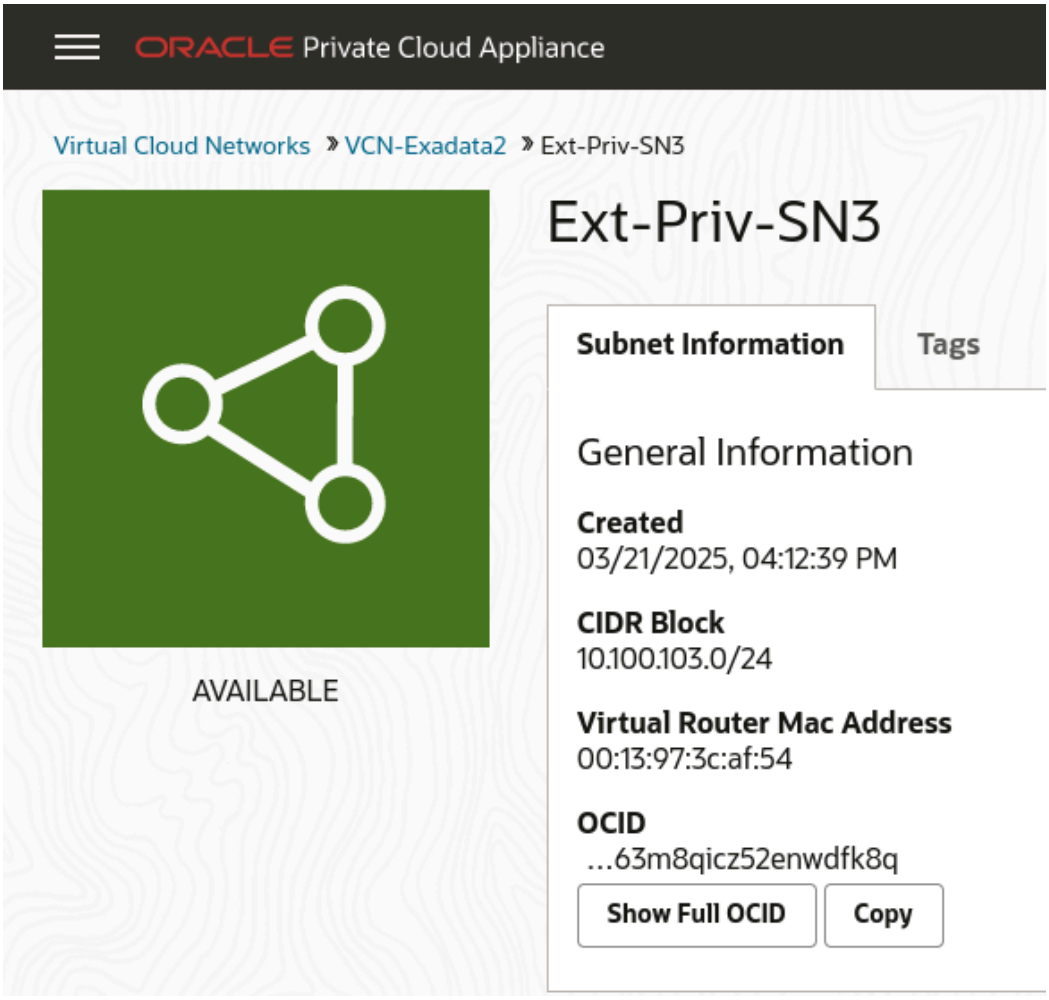
4. From the Compute Enclave, create VCNs, and Subnets. See Managing VCNs and Subnets on page 478.
   a. Create a VCN for use by the Flex network. In the Compute Web UI, create a VCN. Choose a compartment, a name for the VCN, then assign a CIDR block that meets your needs.

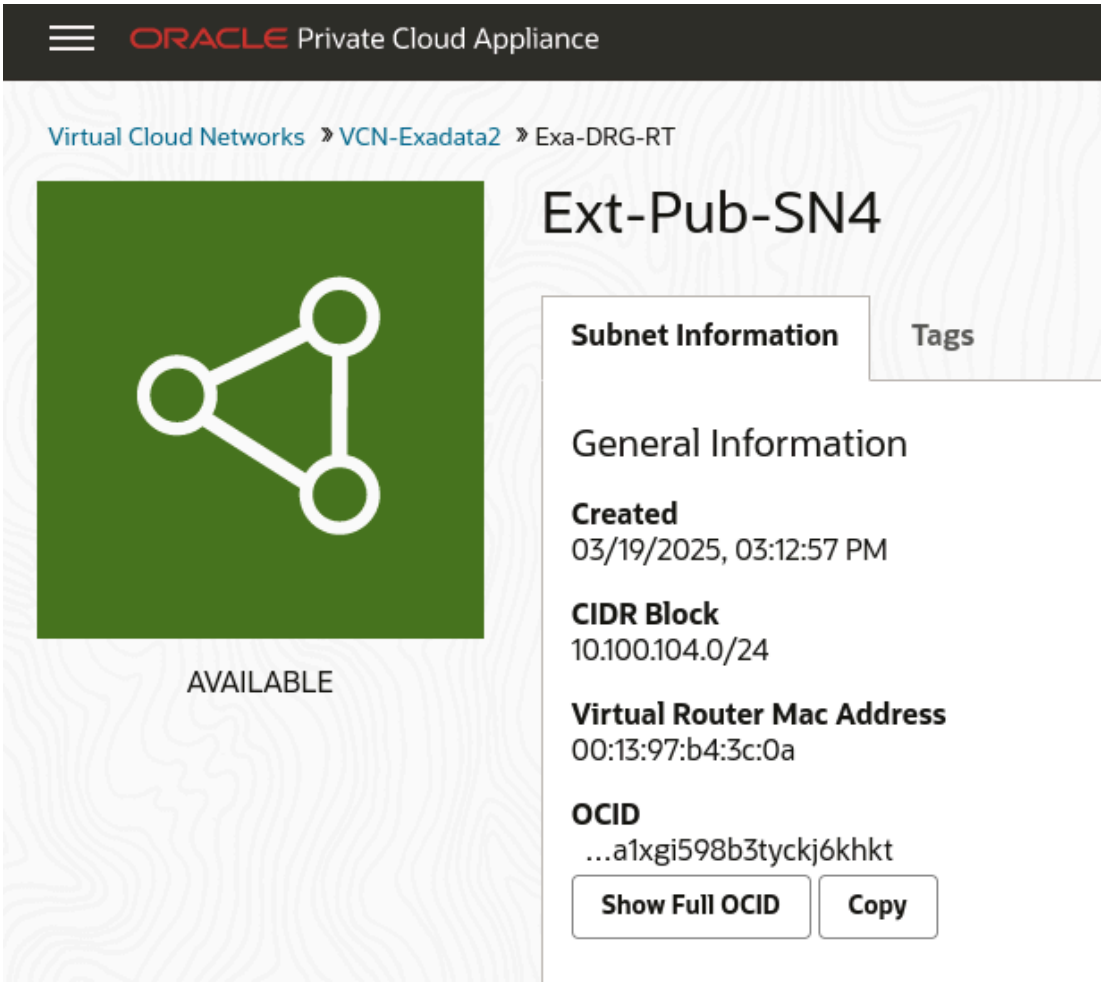


b. Create subnets within the VCN. Navigate to the VCN in the Compute Web UI, and click Create Subnet. Name the subnet, chose the compartment, and assign a CIDR block within the CIDR block range of the VCN. Next select private or public subnet, then click Create Subnet.


ORACLE Private Cloud Appliance
Virtual Cloud Networks
VCN-Exadata2
Exa-Priv-SN1
Exa-Priv-SN1
AVAILABLE
Subnet Information
Tags
General Information
Created
03/21/2025, 04:11:35 PM
CIDR Block
10.100.101.0/24
Virtual Router Mac Address
00:13:97:f1:f6:9c
OCID
...7whwrom5ovqdr1z4oq
Show Full OCID
Copy


ORACLE Private Cloud Appliance
Virtual Cloud Networks » VCN-Exadata2 » Exa-Priv-SN2
Exa-Priv-SN2
Subnet Information
Tags
AVAILABLE
General Information
Created
03/21/2025, 04:12:09 PM
CIDR Block
10.100.102.0/24
Virtual Router Mac Address
00:13:97:bb:7f:77
OCID
...fj3cqsm6v2dzrb3sytky
Show Full OCID
Copy


ORACLE Private Cloud Appliance
Virtual Cloud Networks » VCN-Exadata2 » Ext-Priv-SN3
Ext-Priv-SN3
AVAILABLE
Subnet Information
Tags
General Information
Created
03/21/2025, 04:12:39 PM
CIDR Block
10.100.103.0/24
Virtual Router Mac Address
00:13:97:3c:af:54
OCID
...63m8qicz52enwdfk8q
Show Full OCID
Copy



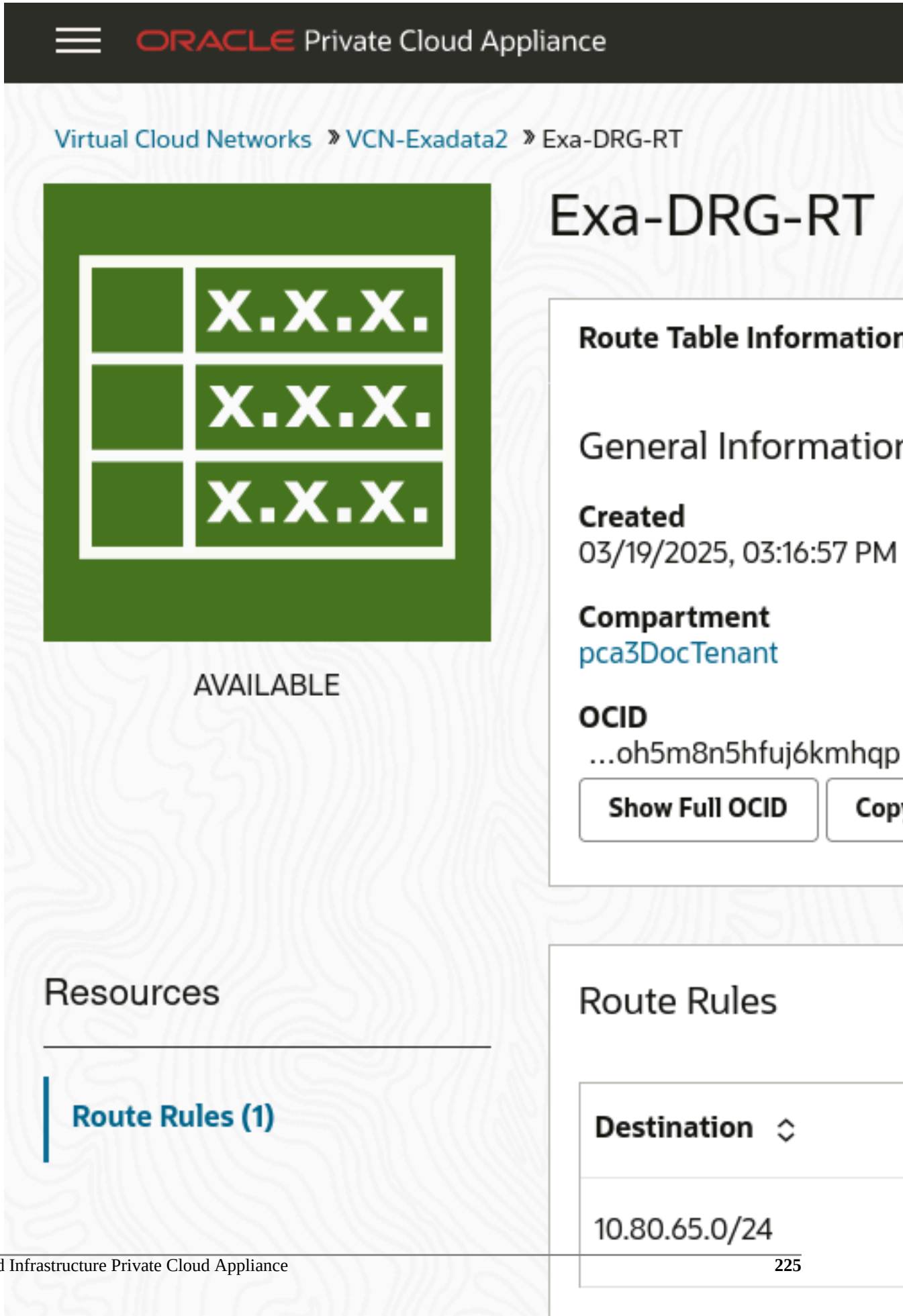
c. Note the OCID of each subnet, you need these OCIDs to enable communication between the compute instances and the Flex network in step 5 of this procedure.

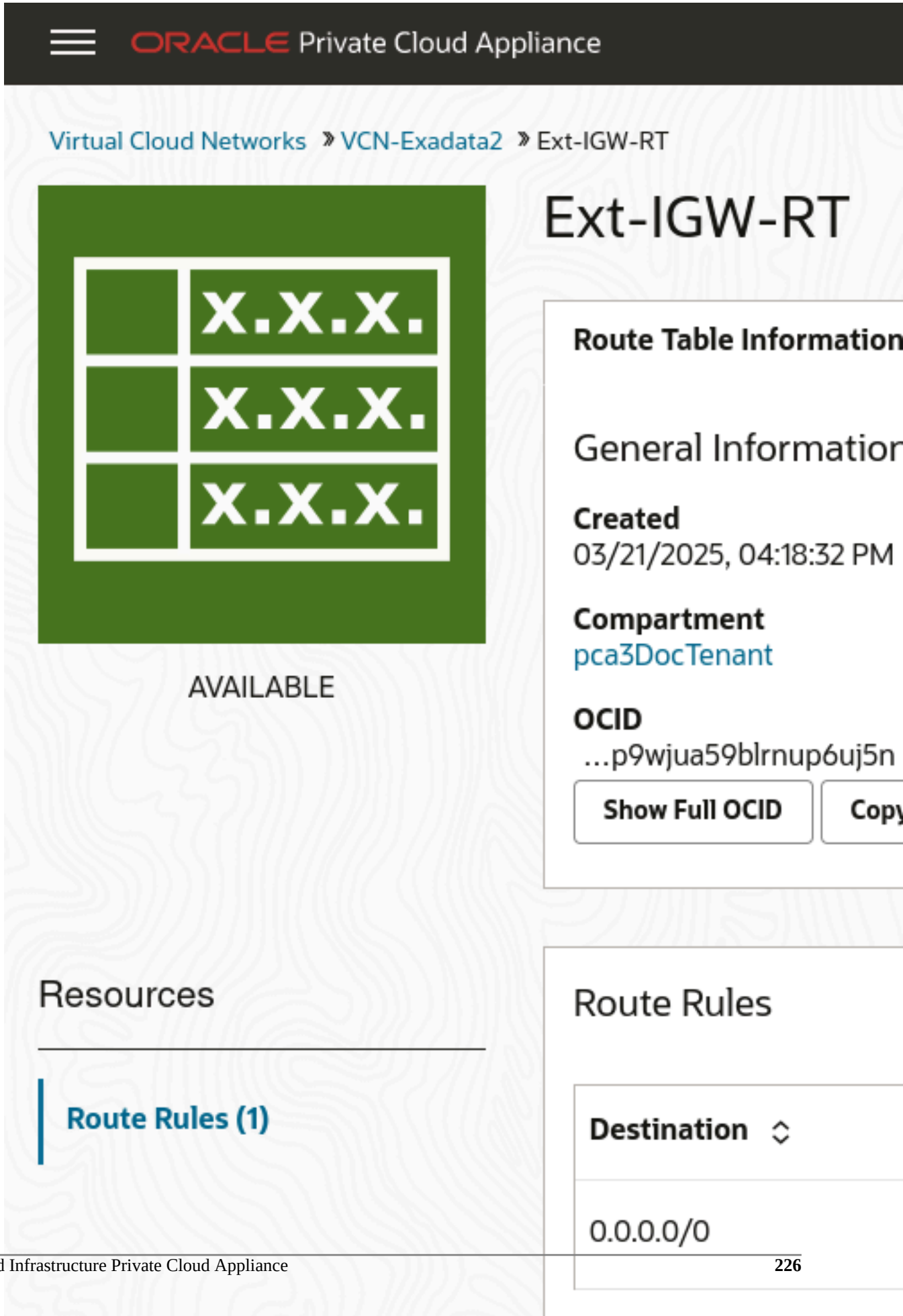
d. Create Route Tables in the VCN. Route tables are required to send traffic outside the VCN.

Navigate to the VCN in the Compute Web UI, select Route Tables in the Resources menu. Click Create Route Table, type a name for the Route Table and click Create Route Table.

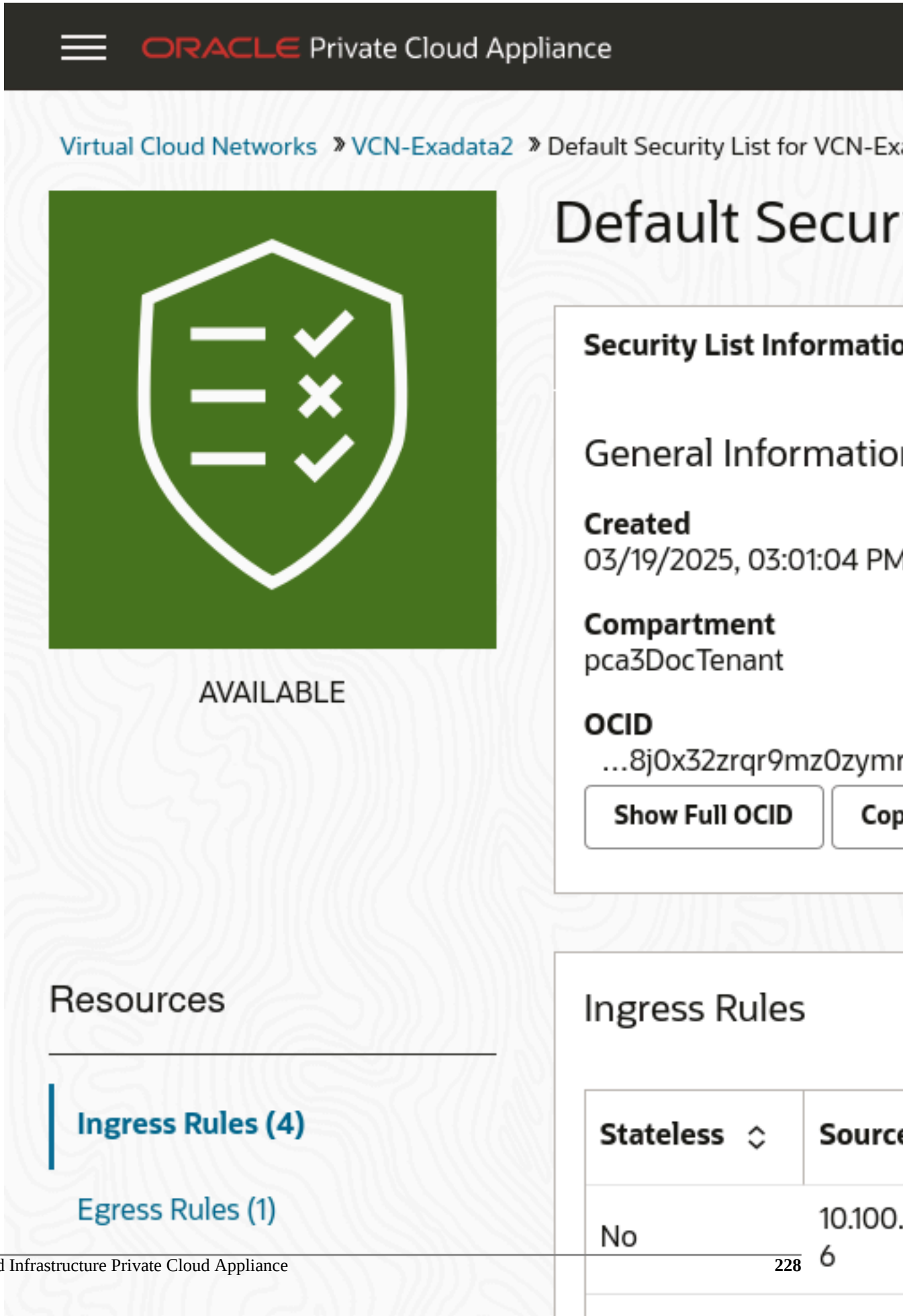
Then create any associated route rules by clicking Add Route Rules and entering a target and the destination CIDR block.

This example shows a route table that directs network traffic for the DRG, and a route table that directs network traffic for the internet gateway.


ORACLE Private Cloud Appliance
Virtual Cloud Networks » VCN-Exadata2 » Exa-DRG-RT
Exa-DRG-RT
x.x.x.
x.x.x.
x.x.x.
AVAILABLE
Route Table Information
General Information
Created
03/19/2025, 03:16:57 PM
Compartment
pca3DocTenant
OCID
...oh5m8n5hfuj6kmhqp
Show Full OCID
Resources
Route Rules (1)
Route Rules
Destination
10.80.65.0/24

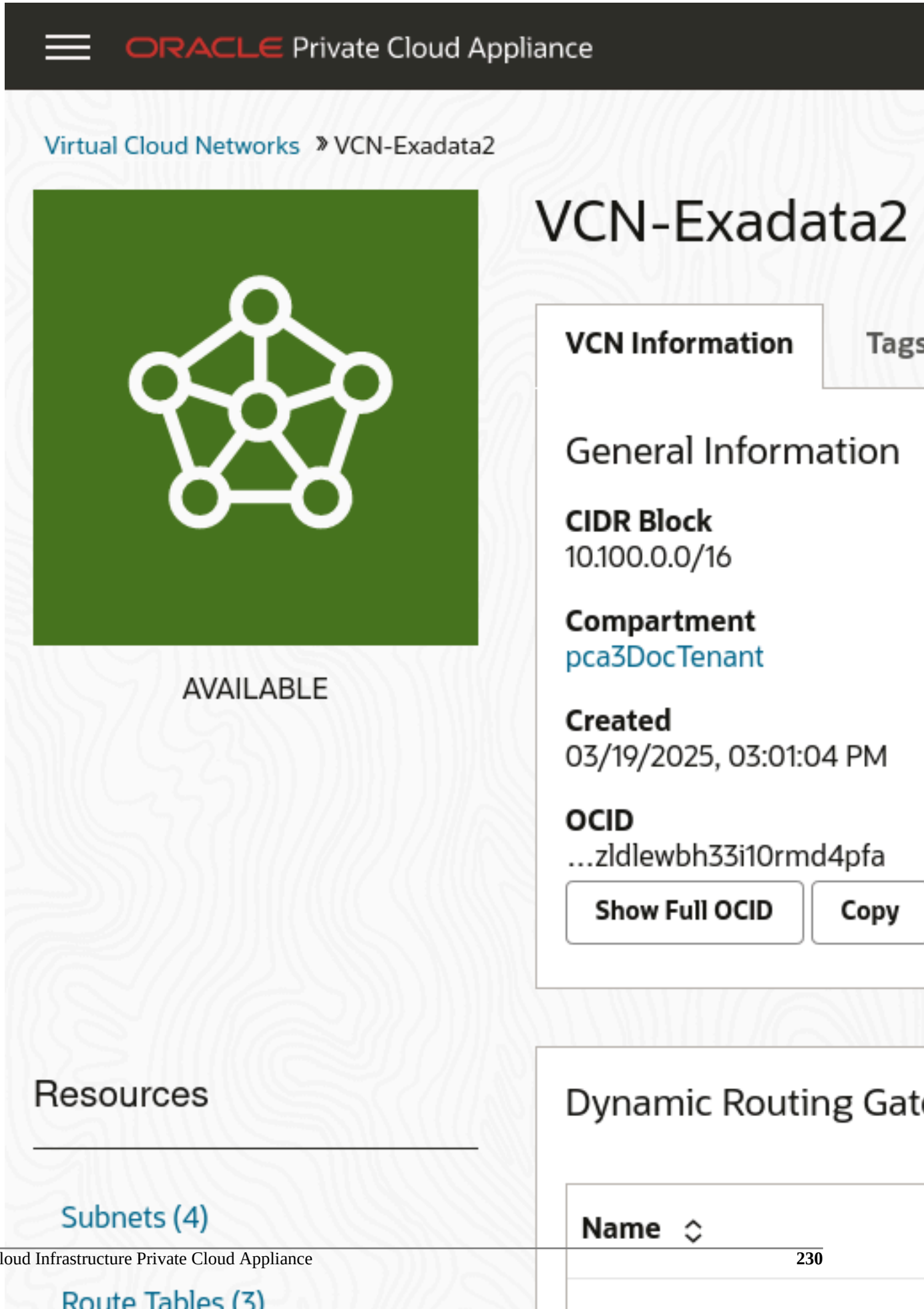

ORACLE Private Cloud Appliance
Virtual Cloud Networks » VCN-Exadata2 » Ext-IGW-RT
Ext-IGW-RT
X.X.X.
X.X.X.
X.X.X.
AVAILABLE
Route Table Information
General Information
Created
03/21/2025, 04:18:32 PM
Compartment
pca3DocTenant
OCID
...p9wjua59blrnup6uj5n
Show Full OCID
Resources
Route Rules (1)
Route Rules
Destination
0.0.0.0/0

e. Create Security Lists to enable ingress traffic. See Controlling Traffic with Security Lists on page 496.


ORACLE Private Cloud Appliance
Virtual Cloud Networks » VCN-Exadata2 » Default Security List for VCN-Ex
Default Secur
Security List Informatio
General Informatio
Created
03/19/2025, 03:01:04 PM
Compartment
pca3DocTenant
OCID
...8j0x32zrqr9mz0zymr
Show Full OCID
Cop
AVAILABLE
Resources
Ingress Rules (4)
Egress Rules (1)
Ingress Rules
Stateless
Source
No
10.100.
6

5. From the Compute Enclave, create DRG-attachments to enable the VCN to use the DRG. See Attaching VCNs to a Dynamic Routing Gateway on page 512.

   From the VCN page, select Dynamic Routing Gateway from the left menu, then click Attach Dynamic Routing Gateway. Select the appropriate tenancy, then choose the DRG you want to attach from the drop down list and click Attach Dynamic Routing Gateway.


ORACLE Private Cloud Appliance
Virtual Cloud Networks » VCN-Exadata2
VCN-Exadata2
AVAILABLE
VCN Information
Tags
General Information
CIDR Block
10.100.0.0/16
Compartment
pca3DocTenant
Created
03/19/2025, 03:01:04 PM
OCID
...zldlewbh33i10rmd4pfa
Show Full OCID
Copy
Resources
Subnets (4)
Route Tables (3)
Dynamic Routing Gat
Name

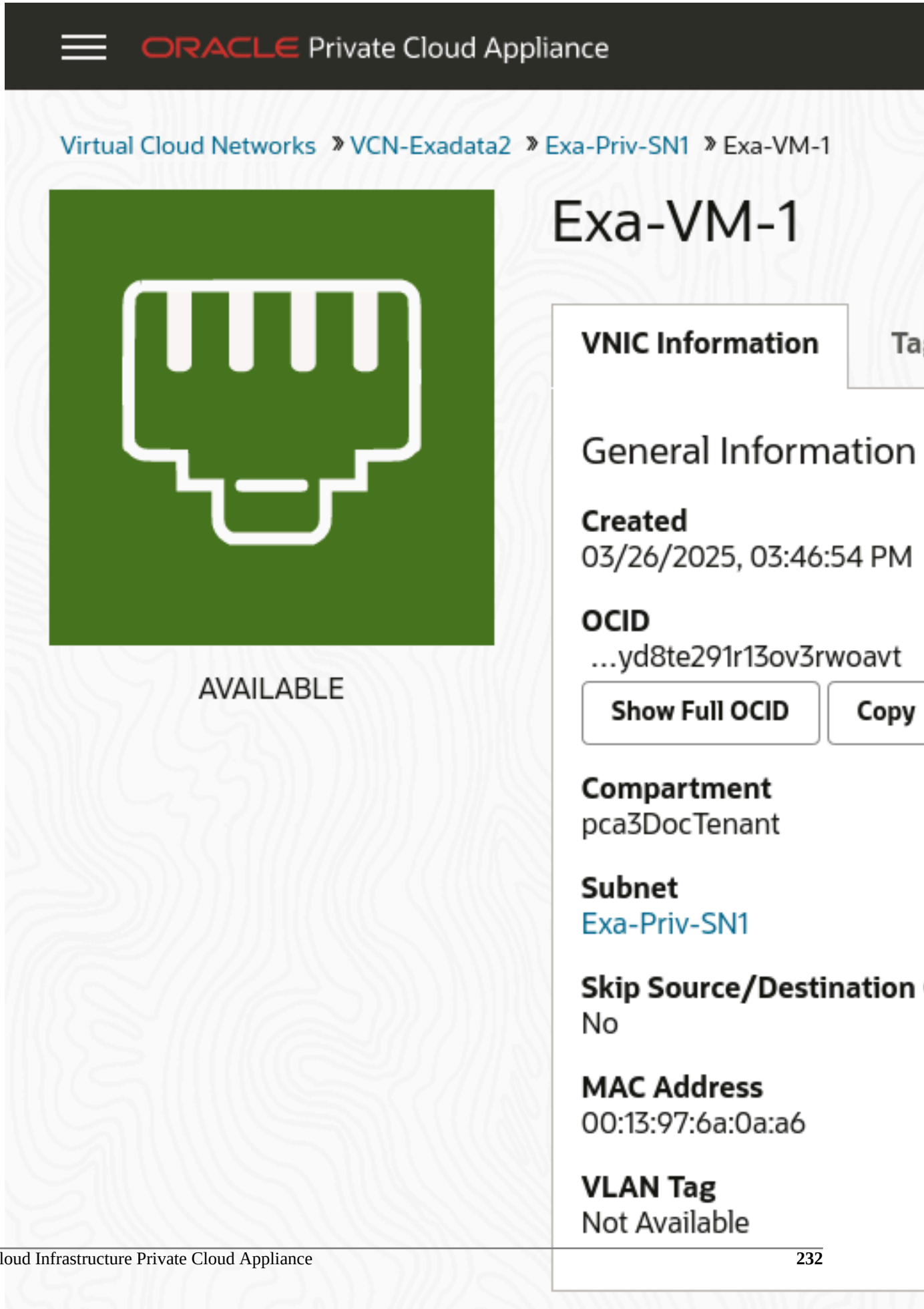
6. From the Service CLI, enable communication between the Flex network and the VM subnets.

```
PCA-ADMIN> flexNetworkEnableAccess flexNetworkId=ocid1.exadata.unique_id \
subnetId=ocid1.subnet.unique_id
Status: Success
```

7. Create compute instances in the subnets and configure their access. See Working with Instances on page 649.

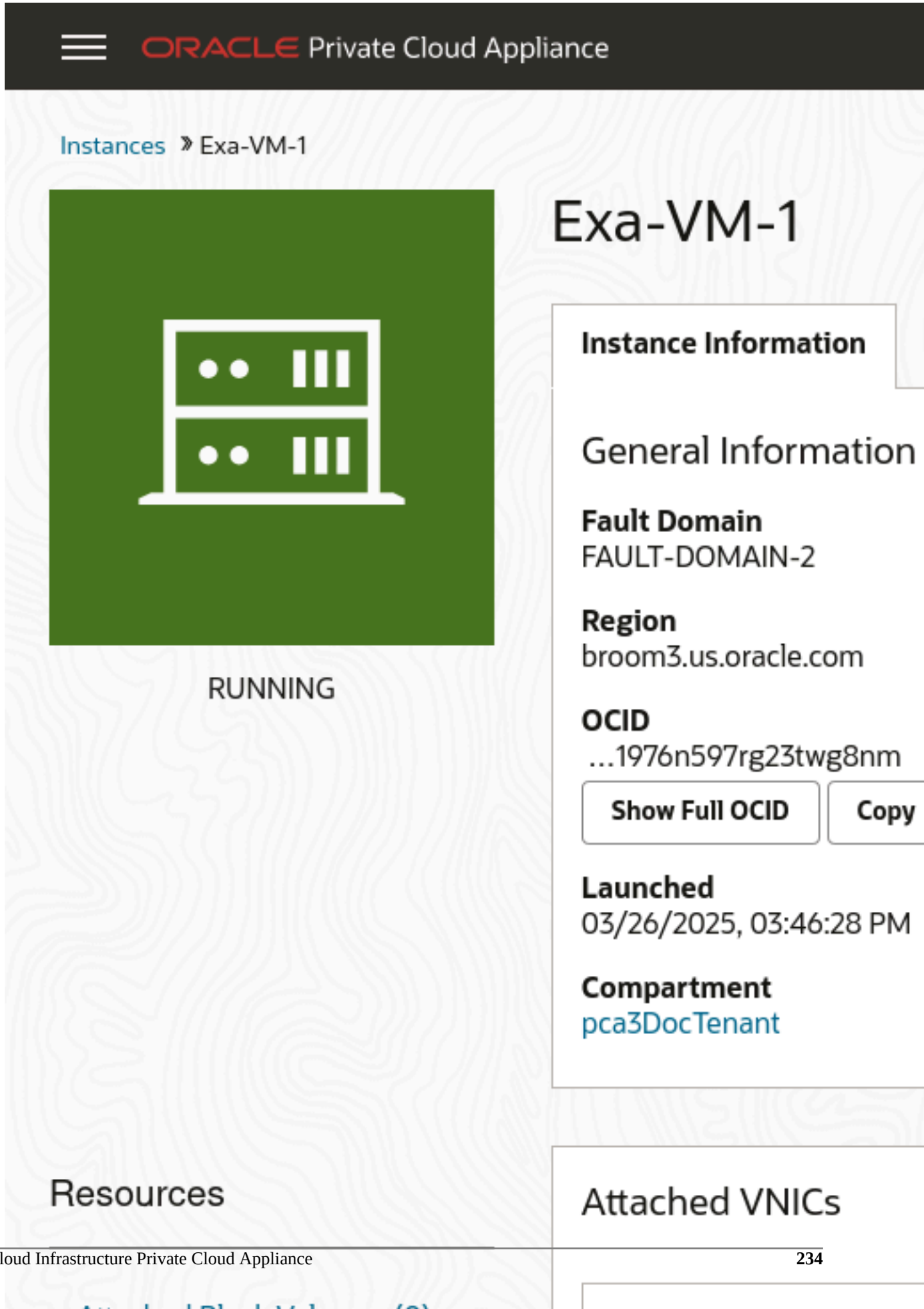
   For this example, each private subnet is configured with 2 VNICs: a primary and a secondary. Configure primary VNICs to attach to the public subnet, and secondary VNICs to attach to the DRG.

   When creating an instance, select the VCN and then the subnet. For the public compute instance, assign a public IP address.


ORACLE Private Cloud Appliance
Virtual Cloud Networks » VCN-Exadata2 » Exa-Priv-SN1 » Exa-VM-1
Exa-VM-1
AVAILABLE
VNIC Information
General Information
Created
03/26/2025, 03:46:54 PM
OCID
...yd8te291r13ov3rwoavt
Show Full OCID
Copy
Compartment
pca3DocTenant
Subnet
Exa-Priv-SN1
Skip Source/Destination
No
MAC Address
00:13:97:6a:0a:a6
VLAN Tag
Not Available

From the Compute Instance page, select the instance. From the Resources menu select Attached VNICs. You will see the primary VNIC. To create the secondary VNIC, click Create VNIC Attachment. Select the VCN and a subnet, and assign. See Creating and Attaching a Secondary VNIC on page 520.

Create primary and secondary VNICs for each instance such that the primary VNIC attaches to the public subnet, and the secondary VNIC attaches to the DRG.


ORACLE Private Cloud Appliance
Instances » Exa-VM-1
Exa-VM-1
RUNNING
Instance Information
General Information
Fault Domain
FAULT-DOMAIN-2
Region
broom3.us.oracle.com
OCID
...1976n597rg23twg8nm
Show Full OCID
Copy
Launched
03/26/2025, 03:46:28 PM
Compartment
pca3DocTenant
Resources
Attached VNICs

8. Verify connectivity between the compute instances and the external system.

Reconfiguring the Private Cloud Appliance Network Environment

The network environment is configured during the initial setup of Private Cloud Appliance. An appliance administrator can update this configuration. However, careful planning is advised, because these settings define the connections to the on-premises network and could disrupt system operations.

Note:

It is *not* supported to turn off BGP authentication by changing the BGP password to null:

```
PCA-ADMIN> edit networkConfig adminbgppassword=
```

Even if this command is completed successfully, BGP authentication remains active and the password is unchanged. This is not an error condition, but a security feature. To disable BGP authentication, you must do so explicitly. The command is:

```
PCA-ADMIN> edit networkConfig
 adminbgpauthentication=false
```

Important:

Enter the IP addresses for the uplinks exactly as they appear in the network configuration spreadsheet you filled out in preparation. The order of entry is important, especially when multiple IPs are added in the same field, because they map to specific spine and data center switches in the uplink topology.

Routing Information Cannot Be Modified

Caution:

It is not supported to change routing information. This applies to network topologies configured with either dynamic or static routing.

Updating Management Node Information

You can change the IP addresses and hostnames of the management nodes.

Using the Service Web UI

1. In the navigation menu, click Network Environment.
2. In the Network Environment Information page, click the Management Nodes tab.

   The Management Nodes details appear.
3. In the top-right corner of the page, click Edit.
4. Click Next to navigate to the page you want to edit, then update the appropriate fields.

   For field descriptions, see Initial System Installation Checklist on page 74.
5. Click Save Changes.

Using the Service CLI

1. Display the current network configuration information using the `show NetworkConfig` command.

```
PCA-ADMIN> show NetworkConfig
Data:
  Uplink Port Speed = 100
  Uplink Port Count = 2
  Uplink Vlan Mtu = 9216
```

```
  Spine1 Ip = 10.n.n.12
  Spine2 Ip = 10.n.n.13
  Uplink Netmask = 255.255.255.0
  Management VIP Hostname = ukpca01mn
  Management VIP 100g = 10.n.n.8
  NTP Server(s) = 100.n.n.254
  Uplink Port Fec = auto
  Public Ip range/list =
 10.n.n.2/32,10.n.n.3/32,10.n.n.4/32,10.n.n.5/32,10.n.n.6/32,10.n.n.7/32
  DNS Address1 = 206.n.n.1
  DNS Address2 = 206.n.n.2
  DNS Address3 = 10.n.n.197
  Management Node1 Hostname = ukpca01-mn1
  Management Node2 Hostname = ukpca01-mn2
  Management Node3 Hostname = ukpca01-mn3
  100g Management Node1 Ip = 10.n.n.9
  100g Management Node2 Ip = 10.n.n.10
  100g Management Node3 Ip = 10.n.n.11
  Object Storage Ip = 10.n.n.1
  Enable Admin Network = false
  Static Routing = true
  Spine VIP = 10.n.n.14
  Uplink Gateway = 10.n.n.1
  Uplink VLAN = 799
  Uplink Hsrp Group = 61
  BGP Authentication = false
```

2. Use the `edit NetworkConfig` command to change any of these management node parameters:

   - Management Node 1 IP
   - Management Node 1 Hostname
   - Management Node 2 IP
   - Management Node 2 Hostname
   - Management Node 3 IP
   - Management Node 3 Hostname
   - Management Node VIP
   - Management Node VIP Hostname

```
PCA-ADMIN> edit NetworkConfig mgmt01Ip100g=172.n.n.190
 mgmt02Ip100g=172.n.n.191
JobId: 52f5177d-402a-4a52-98fe-1cff9c1f26be
```

Updating the Data Center Uplink Configuration

You can change the parameters of the physical connection between Private Cloud Appliance and the on-premises network.

> **Caution:**
>
> Reconfiguring the connection to the data center causes an interruption of all network connectivity to and from the appliance. No network traffic is possible while the physical connections are reconfigured. All connections are automatically restored when the configuration update is complete.

Using the Service Web UI

1. In the navigation menu, click Network Environment.
2. In the Network Environment Information page, click the Uplink tab.

   The Uplink details appear.
3. In the top-right corner of the page, click Edit.

4. Click Next to navigate to the page you want to edit, then update the appropriate fields.

   For field descriptions, see Initial System Installation Checklist on page 74.
5. Click Save Changes.

Using the Service CLI

1. Display the current network configuration information using the `show NetworkConfig` command.

```
PCA-ADMIN> show NetworkConfig
Data:
  Uplink Port Speed = 100
  Uplink Port Count = 2
  Uplink Vlan Mtu = 9216
  Spine1 Ip = 10.n.n.12
  Spine2 Ip = 10.n.n.13
  Uplink Netmask = 255.255.255.0
  Management VIP Hostname = ukpca01mn
  Management VIP 100g = 10.n.n.8
  NTP Server(s) = 100.n.n.254
  Uplink Port Fec = auto
  Public Ip range/list =
 10.n.n.2/32,10.n.n.3/32,10.n.n.4/32,10.n.n.5/32,10.n.n.6/32,10.n.n.7/32
  DNS Address1 = 206.n.n.1
  DNS Address2 = 206.n.n.2
  DNS Address3 = 10.n.n.197
  Management Node1 Hostname = ukpca01-mn1
  Management Node2 Hostname = ukpca01-mn2
  Management Node3 Hostname = ukpca01-mn3
  100g Management Node1 Ip = 10.n.n.9
  100g Management Node2 Ip = 10.n.n.10
  100g Management Node3 Ip = 10.n.n.11
  Object Storage Ip = 10.n.n.1
  Enable Admin Network = false
  Static Routing = true
  Spine VIP = 10.n.n.14
  Uplink Gateway = 10.n.n.1
  Uplink VLAN = 799
  Uplink Hsrp Group = 61
  BGP Authentication = false
```

2. Use the `edit NetworkConfig` command to change any of these data center uplink parameters:

   - Uplink Port Speed
   - Uplink Port Count
   - Uplink VLAN MTU
   - Uplink Port FEC

```
PCA-ADMIN> edit NetworkConfig uplinkPortCount=2
JobId: 42f5137f-122a-4a52-98fe-1cfv9c1f26ve
```

Updating the Administration Network Configuration

If you use the optional Administration Network, you can update the parameters using these procedures.

> **Caution:**
>
> If you are not currently using a separate Administration Network, the Network Environment Information page in the Service Web UI will not display an Admin Network tab or any of the related configuration parameters. The command output in the Service CLI will not display any Admin Network parameters either. You must first enable the Administration Network.

> When an Administration Network is configured, it cannot be disabled again.

For general Administration Network configuration information, see Data Center Network Configuration Guidelines on page 66. For descriptions of Administration Network parameters, see Initial System Installation Checklist on page 74.

Using the Service Web UI

Scenario 1: Administration Network Disabled

If you need to enable and configure a separate Administration Network, proceed as follows:

1. In the navigation menu, click Network Environment.
2. In the top-right corner of the page, click Edit.
3. In the wizard, navigate to the Admin Network tab and set Admin Networking to *Enable*.
4. Enter all the required parameters in the respective fields on the form.
5. Click Save Changes.

Scenario 2: Administration Network Enabled

If you already configured a separate Administration Network and need to edit its settings, proceed as follows:

1. In the navigation menu, click Network Environment.
2. In the Network Environment Information page, click the Admin Network tab.

   The Admin Network details appear.
3. In the top-right corner of the page, click Edit.
4. Click Next to navigate to the page you want to edit, then update the appropriate fields.
5. Click Save Changes.

Using the Service CLI

1. If you are not currently using a separate Administration Network, the CLI output will not display any Admin Network parameters. Enable the Administration Network first.
2. Display the current network configuration information using the `show NetworkConfig` command.

```
PCA-ADMIN> show NetworkConfig
Data:
  Uplink Port Speed = 100
  Uplink Port Count = 4
  Uplink Vlan Mtu = 9216
  Spine1 Ip = 10.10.10.97,10.10.10.101
  Spine2 Ip = 10.10.10.99,10.10.10.103
  Uplink Netmask = 255.255.255.254,255.255.255.254
  Management VIP Hostname = mypca
  Management VIP = 10.10.10.107
  NTP Server(s) = 10.80.211.105,10.211.17.1,10.68.48.1
  Uplink Port Fec = auto
  Public Ip range/list =
 10.10.10.114/31,10.10.10.116/31,10.10.10.118/31,10.10.10.120/31,10.10.10.122/31
  Management Node1 Hostname = pcamn01
  Management Node2 Hostname = pcamn02
  Management Node3 Hostname = pcamn03
  Management Node1 Ip = 10.10.10.108
  Management Node2 Ip = 10.10.10.109
  Management Node3 Ip = 10.10.10.110
  Object Storage Ip = 10.10.10.113
  Enable Admin Network = true
  Admin Port Speed = 100
  Admin Port Count = 1
  Admin Vlan Mtu = 9216
  Admin Port Fec = auto
  Admin VLAN = 3915
  Admin Spine1 Ip = 10.25.0.111
```

```
  Admin Spine2 Ip = 10.25.0.112
  Admin Spine VIP = 10.25.0.110
  Admin Netmask = 255.255.255.0
  Admin Hsrp Group = 152
  Static Routing = false
  Uplink VLAN = 3911
  Peer1 Asn = 50000
  Peer1 Ip = 10.10.10.96,10.10.10.98
  Oracle Asn = 136025
  Bgp Topology = mesh
  Peer2 Asn = 50000
  Peer2 Ip = 10.10.10.100,10.10.10.102
  BGP Authentication = false
  BGP KeepAlive Timer = 60
  BGP Holddown Timer = 180
  Network Config Lifecycle State = ACTIVE
  admin DNS Address1 = 10.25.0.1
  admin Management Node1 Hostname = pcamn01admin.example.com
  admin Management Node2 Hostname = pcamn02admin.example.com
  admin Management Node3 Hostname = pcamn03admin.example.com
  admin Management Node1 Ip = 10.25.0.101
  admin Management Node2 Ip = 10.25.0.102
  admin Management Node3 Ip = 10.25.0.103
  admin Management VIP Hostname = mypcaadmin.example.com
  admin Management VIP = 10.25.0.100
```

3. Use the `edit NetworkConfig` command to change any of these administration network parameters:

> **Tip:**
>
> Enter `edit networkConfig ?` to display the parameters available for editing.

- Admin Network enable
- Management node cluster Admin VIP and host name
- Management node 1-3 Admin IP and host name
- Admin DNS IP 1-3
- Admin Port count, speed, FEC
- Admin CIDR
- Admin VLAN and MTU
- Admin Gateway IP
- Admin Netmask
- Spine 1+2 Admin IP
- Spine Admin VIP

```
PCA-ADMIN> edit NetworkConfig adminPortSpeed=25
JobId: 62f8137f-772a-4a52-98f4-1cfv9c1f24te

PCA-ADMIN> edit NetworkConfig adminCidr=10.25.0.1/24
JobId: 861381ae-cc63-44a2-a66e-8e095e4a99f9
```

Updating NTP Server Information

> **Note:**
>
> You can check the NTP connection status of a running system at any time. In the Service Web UI, it's displayed in the NTP tab of the Network

Environment Information page. In the Service CLI you can run this command:

```
PCA-ADMIN> checkNtpServers
Data:
  id              Accessible
  --              ----------
  10.64.0.252     true
  192.0.2.2       true
```

You can edit or add IP addresses for NTP servers.

Using the Service Web UI

1. In the navigation menu, click Network Environment.
2. In the Network Environment Information page, click the NTP tab.

   The NTP details appear.
3. In the top-right corner of the page, click Edit.
4. Click Next to navigate to the page you want to edit, then update the appropriate fields.

   For field descriptions, see Initial System Installation Checklist on page 74.
5. Click Save Changes.

Using the Service CLI

1. Display the current network configuration information using the `show NetworkConfig` command.

```
PCA-ADMIN> show NetworkConfig
Data:
  Uplink Port Speed = 100
  Uplink Port Count = 2
  Uplink Vlan Mtu = 9216
  Spine1 Ip = 10.n.n.12
  Spine2 Ip = 10.n.n.13
  Uplink Netmask = 255.255.255.0
  Management VIP Hostname = ukpca01mn
  Management VIP 100g = 10.n.n.8
  NTP Server(s) = 100.n.n.254
  Uplink Port Fec = auto
  Public Ip range/list =
 10.n.n.2/32,10.n.n.3/32,10.n.n.4/32,10.n.n.5/32,10.n.n.6/32,10.n.n.7/32
  DNS Address1 = 206.n.n.1
  DNS Address2 = 206.n.n.2
  DNS Address3 = 10.n.n.197
  Management Node1 Hostname = ukpca01-mn1
  Management Node2 Hostname = ukpca01-mn2
  Management Node3 Hostname = ukpca01-mn3
  100g Management Node1 Ip = 10.n.n.9
  100g Management Node2 Ip = 10.n.n.10
  100g Management Node3 Ip = 10.n.n.11
  Object Storage Ip = 10.n.n.1
  Enable Admin Network = false
  Static Routing = true
  Spine VIP = 10.n.n.14
  Uplink Gateway = 10.n.n.1
  Uplink VLAN = 799
  Uplink Hsrp Group = 61
  BGP Authentication = false
```

2. Use the `edit NetworkConfig` command to change the NTP servers. Enter multiple IP addresses in a comma-separated list:

```
PCA-ADMIN> edit NetworkConfig ntpIps=100.n.n.254,100.n.n.253
JobId: 42f5137f-122a-4a52-98fe-1cfv9c1f26ve
```

Updating DNS Server Information

You can edit or add IP addresses for DNS servers.

Using the Service Web UI

1. In the navigation menu, click Network Environment.
2. In the Network Environment Information page, click the DNS tab.

   The DNS details appear.
3. In the top-right corner of the page, click Edit.
4. Click Next to navigate to the page you want to edit, then update the appropriate fields.

   For field descriptions, see Initial System Installation Checklist on page 74.
5. Click Save Changes.

Using the Service CLI

1. Display the current network configuration information using the `show NetworkConfig` command.

```
PCA-ADMIN> show NetworkConfig
Data:
  Uplink Port Speed = 100
  Uplink Port Count = 2
  Uplink Vlan Mtu = 9216
  Spine1 Ip = 10.n.n.12
  Spine2 Ip = 10.n.n.13
  Uplink Netmask = 255.255.255.0
  Management VIP Hostname = ukpca01mn
  Management VIP 100g = 10.n.n.8
  NTP Server(s) = 100.n.n.254
  Uplink Port Fec = auto
  Public Ip range/list =
 10.n.n.2/32,10.n.n.3/32,10.n.n.4/32,10.n.n.5/32,10.n.n.6/32,10.n.n.7/32
  DNS Address1 = 206.n.n.1
  DNS Address2 = 206.n.n.2
  DNS Address3 = 10.n.n.197
  Management Node1 Hostname = ukpca01-mn1
  Management Node2 Hostname = ukpca01-mn2
  Management Node3 Hostname = ukpca01-mn3
  100g Management Node1 Ip = 10.n.n.9
  100g Management Node2 Ip = 10.n.n.10
  100g Management Node3 Ip = 10.n.n.11
  Object Storage Ip = 10.n.n.1
  Enable Admin Network = false
  Static Routing = true
  Spine VIP = 10.n.n.14
  Uplink Gateway = 10.n.n.1
  Uplink VLAN = 799
  Uplink Hsrp Group = 61
  BGP Authentication = false
```

2. Use the `edit NetworkConfig` command to change the DNS IP addresses:

   - DNS IP1
   - DNS IP2

- DNS IP3

```
PCA-ADMIN> edit NetworkConfig DnsIp2=206.n.n.2
JobId: 42f5137f-122a-4a52-98fe-1cfv9c1f26ve
```

Updating the Public IP Address Configuration

You can add public IP addresses, or change the currently configured IP addresses.

> **Caution:**
>
> Changing public IP addresses that are in use can cause system disruption.

Using the Service Web UI

1. In the navigation menu, click Network Environment.
2. In the Network Environment Information page, click the Uplink tab.

   The Uplink details appear.
3. In the top-right corner of the page, click Edit.
4. Click Next to navigate to the page you want to edit, then update the appropriate fields.

   For field descriptions, see Initial System Installation Checklist on page 74.
5. Click Save Changes.

Using the Service CLI

1. Display the current network configuration information using the `show NetworkConfig` command.

```
PCA-ADMIN> show NetworkConfig
Data:
  Uplink Port Speed = 100
  Uplink Port Count = 2
  Uplink Vlan Mtu = 9216
  Spine1 Ip = 10.n.n.12
  Spine2 Ip = 10.n.n.13
  Uplink Netmask = 255.255.255.0
  Management VIP Hostname = ukpca01mn
  Management VIP 100g = 10.n.n.8
  NTP Server(s) = 100.n.n.254
  Uplink Port Fec = auto
  Public Ip range/list =
 10.n.n.2/32,10.n.n.3/32,10.n.n.4/32,10.n.n.5/32,10.n.n.6/32,10.n.n.7/32
  DNS Address1 = 206.n.n.1
  DNS Address2 = 206.n.n.2
  DNS Address3 = 10.n.n.197
  Management Node1 Hostname = ukpca01-mn1
  Management Node2 Hostname = ukpca01-mn2
  Management Node3 Hostname = ukpca01-mn3
  100g Management Node1 Ip = 10.n.n.9
  100g Management Node2 Ip = 10.n.n.10
  100g Management Node3 Ip = 10.n.n.11
  Object Storage Ip = 10.n.n.1
  Enable Admin Network = false
  Static Routing = true
  Spine VIP = 10.n.n.14
  Uplink Gateway = 10.n.n.1
  Uplink VLAN = 799
  Uplink Hsrp Group = 61
  BGP Authentication = false
```

2. Use the `edit NetworkConfig` command to change the public IP addresses or the object storage public IP address:

- Object Storage Public IP
- Public IP Range/List

```
PCA-ADMIN> edit NetworkConfig PublicIps=
 10.n.n.17/32,10.n.n.18/32,10.n.n.19/32
JobId: 42f5137f-122a-4a52-98fe-1cfv9c1f26ve
```

Configuring Appliance Proxy Settings

The initial appliance setup procedure, as described in Completing the Initial Setup and Configuration on page 97, provides no option to add a system-wide proxy configuration. However, some of the platform and infrastructure services require connectivity to endpoints outside the appliance environment. For example, for federation with an identity provider (IDP), the IAM and Admin services must acquire metadata from that external server (for users of the Compute Enclave and Service Enclave respectively.) If network traffic passes through a proxy server in the data center, requests between the appliance and the external server cannot be completed successfully.

To enable external network communication through a data center proxy server, add the proxy configuration to the appliance network setup. Ensure that the initial appliance setup procedure has been completed first.

Using the Service Web UI

1. In the PCA Config navigation menu, click Appliance Details.

   The Appliance Details page contains system properties such as realm, region and domain.
2. To configure a proxy at the appliance level, click the Set Rack-Wide Proxy button in the top-right corner.

   A proxy configuration window appears.
3. Fill out the proxy configuration parameters:

   - **Proxy Name:** Enter the fully qualified domain name of the proxy server.
   - **Proxy Host:** Enter the proxy server IP address.
   - **Proxy Port:** Enter the port number the proxy server uses for routing requests.
   - **Proxy User Name:** If required, enter a user name for authentication with the proxy server.
   - **Proxy Password:** If required, enter the password for the proxy user name.
   - **Proxy Confirm Password:** If required, enter the proxy password again for confirmation.

   The user name and password parameters are optional, in case the proxy server requires authentication. The details are stored in the Secret Service (Vault), where services can retrieve them securely to establish their external connection.
4. Click Set Rack-Wide Proxy to save the proxy configuration.

   The proxy configuration is displayed in a separate tab on Appliance Details page.
5. To delete the proxy configuration from the appliance, go to the Appliance Details page and click Clear Rack-Wide Proxy in the top-right corner.
6. If you need to modify the stored proxy configuration, delete it and fill out the proxy configuration window again with the correct parameters.

Using the Service CLI

1. Add the proxy configuration details using the `setProxy` command.

   Syntax (entered on a single line):

```
PCA-ADMIN> setProxy
proxyName=<proxy_fqdn>
proxyHost=<proxy_ip>
proxyPort=<proxy_port>
proxyUsername=<proxy_user>
proxyPassword=<proxy_password>
proxyConfirmPassword=<proxy_password>
```

Example:

```
PCA-ADMIN> setProxy proxyName=myproxy.example.com
proxyHost=172.16.0.100
proxyPort=8080
proxyUsername=proxyuser
proxyPassword=********
proxyConfirmPassword=********
```

The user name and password parameters are optional, in case the proxy server requires authentication. The details are stored in the Secret Service (Vault), where services can retrieve them securely to establish their external connection.

2. To delete the proxy configuration from the appliance, enter the `clearProxy` command. No command parameters are required.
3. If you need to modify the stored proxy configuration, run the `setProxy` command again with the correct parameters.

Enabling External IPv6 Connectivity

The connection between Private Cloud Appliance and the on-premises network can optionally be configured to support IPv6 traffic. This is a dual stack implementation, meaning an IPv4 base configuration is applied during initial system installation and configuration, and IPv6 support is enabled in parallel.

> **Important:**
>
> IPv6 virtual network connectivity is currently only supported for compute instances with SR-IOV (single root I/O virtualization) VNICs.

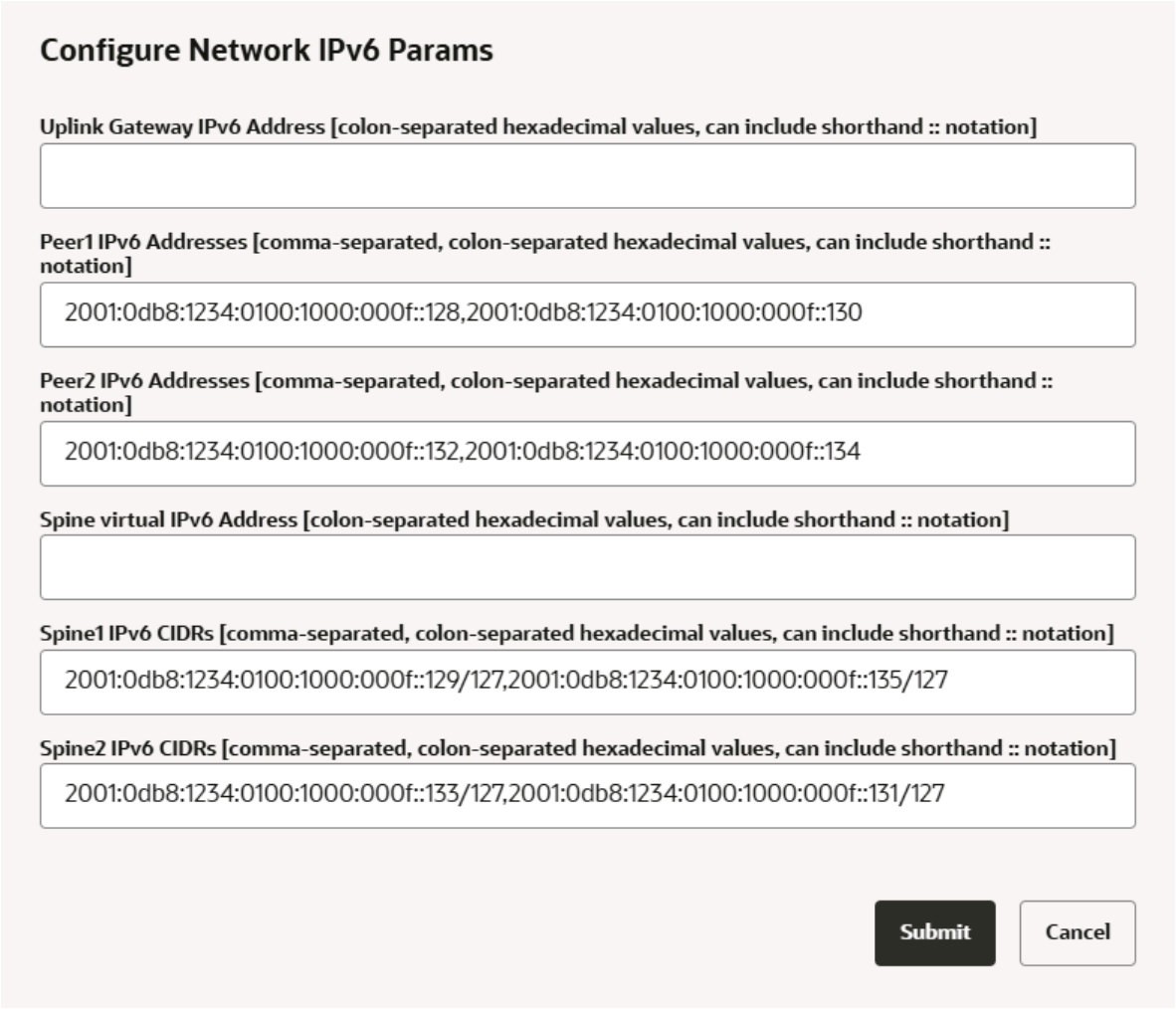
> **Tip:**
>
> Keep equivalent IPv4 and IPv6 addresses and CIDRs easy to identify by using a human readable conversion scheme.

Using the Service Web UI (preferred)

IPv6 compute instance connectivity requires IPv6 packets to be routed over the uplinks and into the subnets where the compute instances have an IPv6 address. The Service Web UI provides a network configuration page for this purpose.

1. In the navigation menu, under PCA Config, select IPv6 Network Environment.
2. In the IPv6 Network Environment Information page, in the upper right corner, click Edit.

   The IPv6 Network Configuration window is displayed.



3. Fill out the required IPv6 parameters for the system uplinks.

   The example shows a mesh topology with dynamic routing. For static routing, uplink gateway and spine virtual IPs are required parameters.
4. Click Submit to apply the IPv6 configuration to the uplinks.
5. To remove the IPv6 configuration, edit and submit with empty parameter fields.

Using the Service CLI

Alternatively, set the IPv6 configuration parameters from the CLI, using the `edit NetworkConfigV6` command.

Command syntax:

```
PCA-ADMIN> edit networkConfigV6
uplinkGatewayV6Ip=<value>
peer1V6Ip=<value>
peer2V6Ip=<value>
spineV6Vip=<value>
spine1V6Cidr=<value>
spine2V6Cidr=<value>
```

Example:

```
PCA-ADMIN> edit networkConfig \
```

```
peer1V6Ip=2001:0db8:1234:0100:1000:000f::128,2001:0db8:1234:0100:1000:000f::130
 \
peer2V6Ip=2001:0db8:1234:0100:1000:000f::132,2001:0db8:1234:0100:1000:000f::134
 \
spine1V6Cidr=2001:0db8:1234:0100:1000:000f::129/127,2001:0db8:1234:0100:1000:000f:
 \
spine2V6Cidr=2001:0db8:1234:0100:1000:000f::133/127,2001:0db8:1234:0100:1000:000f:
```

To remove the IPv6 configuration, enter the same command with the parameters set to empty values.

```
PCA-ADMIN> edit networkConfigV6 peer1V6Ip= peer2V6Ip= spine1V6Cidr=
 spine2V6Cidr=
```

Configuring the Active Directory Domain for File Storage Service

The File Storage service in Private Cloud Appliance enables users of Microsoft Windows instances to map a network drive, or mount a network share. Both the NFS and SMB protocols are supported, but SMB requires that the Microsoft Windows instances and Private Cloud Appliance belong to the same Active Directory domain.

Follow these instructions to set up the Active Directory domain in the Service Enclave.

Using the Service Web UI

1. Verify that DNS is configured on the appliance.
   a. In the navigation menu, click Network Environment.
   b. In the Network Environment Information detail page, select the DNS Servers tab and make sure that DNS servers are configured.

   DNS is required because, during domain configuration, the system searches for a matching SRV record in order to locate the controllers of the Active Directory domain.
2. In the navigation menu, click Active Directory Domain.
3. Verify that no Active Directory domain is currently configured. The configuration details should show "Status = disabled" and "Domain = Not Available".
4. Click Edit to change the Active Directory domain configuration.
5. In the Active Directory Domain Setting window, enter these parameters:
   - the name of the Active Directory domain the appliance is meant to join
   - a user name and password that enable the appliance to join the domain
   - optionally, an organizational unit
6. Click Submit to apply the new configuration.
7. Verify that the Active Directory is configured correctly. The configuration details should show "Status = online" and the newly configured domain name should appear in the Domain field.
8. To remove the ZFS Storage Appliance from the Active Directory domain again, you must use the Service CLI.

Using the Service CLI

1. Gather the information that you need to run the command:
   - the name of the Active Directory domain the appliance is meant to join
   - an account (user name and password) with authorization to join the Active Directory domain
2. Verify that DNS is configured on the appliance. During domain configuration, the system searches for a matching SRV record in order to locate the controllers of the Active Directory domain.

   ```
   PCA-ADMIN> show NetworkConfig
   Data:
     Uplink Port Speed = 100
     Uplink Port Count = 2
     Uplink Vlan Mtu = 9216
   [...]
     DNS Address1 = 192.0.2.201
   ```

```
    DNS Address2 = 192.0.2.202
    DNS Address3 = 10.25.0.101
    Management Node1 Hostname = mypca-mn1
    Management Node2 Hostname = mypca-mn2
    Management Node3 Hostname = mypca-mn3
[...]
    Network Config Lifecycle State = ACTIVE
```

3. Verify that no Active Directory domain is currently configured.

```
PCA-ADMIN> show ZFSAdDomain
Data:
  Status = disabled
  Mode = workgroup
  Service href = /api/service/v2/services/ad
  Domain href = /api/service/v2/services/ad/domain
  Workgroup href = /api/service/v2/services/ad/workgroup
  PasswordSet = false
  Preexist = false
  Workgroup = WORKGROUP
```

4. Configure the Active Directory domain by entering the name of the domain, and a user name and password that enables the appliance to join the domain.

```
PCA-ADMIN> configZFSAdDomain domain=ad.example.com user=Administrator
 password=************
JobId: 7e6abf2d-9f6a-4c32-8f18-5142f6eda3c5
```

5. Use the job ID to check the status of your command.

   When the job has completed successfully, verify the Active Directory zone configuration and status.

```
PCA-ADMIN> show ZFSAdDomain
Data:
  Status = online
  Mode = domain
  Service href = /api/service/v2/services/ad
  Domain href = /api/service/v2/services/ad/domain
  Workgroup href = /api/service/v2/services/ad/workgroup
  PasswordSet = false
  Preexist = false
```

6. To remove the ZFS Storage Appliance from the Active Directory domain again, set its configuration back to *workgroup* mode.

```
PCA-ADMIN> configZFSAdWorkgroup workgroupName=WORKGROUP
JobId: 1329e43a-3ed6-4588-b90b-a45506271df8

PCA-ADMIN> show zfsAdDomain
Data:
  Status = disabled
  Mode = workgroup
  Service href = /api/service/v2/services/ad
  Domain href = /api/service/v2/services/ad/domain
  Workgroup href = /api/service/v2/services/ad/workgroup
  PasswordSet = false
  Preexist = false
  Workgroup = WORKGROUP
```

Accessing External Interfaces of Private Cloud Appliance with Your Own CA Trust Chain

To access the external interfaces of Private Cloud Appliance, you can provide your own custom Certificate Authority (CA) certificates and include them in your own CA trust chain.

Three different CA certificates are required to access all the external interfaces of a Private Cloud Appliance.

- Admin-accessible

  - admin.<domain_name>
  - adminconsole.<domain_name>
  - alertmanager.<domain_name>
  - api.<domain_name>
  - grafana.<domain_name>
  - prometheus.<domain_name>
  - prometheus-gw.<domain_name>
  - rps.<domain_name>
- Regular uplink
  - auth.<domain_name>
  - autoscaling.<domain_name>
  - backup-provider.<domain_name>
  - compute-containers.<domain_name>
  - console.<domain_name>
  - containerengine.<domain_name>
  - dns.<domain_name>
  - filestorage.<domain_name>
  - iaas.<domain_name>
  - identity.<domain_name>
  - limits.<domain_name>
  - network-load-balancer-api.<domain_name>
  - regionregistry.<domain_name>
  - regionrepository.<domain_name>
- Object storage
  - objectstorage.<domain_name>

> **Important:**
>
> When a new service endpoint is added to Private Cloud Appliance as part of a controller software upgrade, the "Regular uplink" CA certificate must be updated. Repeat the process described in this topic to upload a new CA certificate.
>
> If you need to back out your CA certificate and revert to an Oracle-supplied certificate, contact Oracle for assistance.

The process to set up your own CA trust chain is as follows:

1. Create certificate signing requests (CSRs) from your Private Cloud Appliance.
2. With these CSRs, generate certificates signed by your Certificate Authority.
3. Upload the signed CA certificates and your CA trust chain to your Private Cloud Appliance.

Creating Certificate Signing Requests

Generate certificate signing requests (CSRs) using the Service CLI.

> **Note:**
>
> OpenSSH clients must be at least version openssh-clients-7.4p1 or later.

1. Log into the Service CLI.
2. Run the `generateCustomerCsr` command.

```
PCA-ADMIN> generateCustomerCsr
Data:
```

```
 status = success
 message = Successfully generated customer csr:
    Please download all CSR files from: /nfs/shared_storage/certs/
customer_csr/
```

3. You can add Distinguished Names to the `generateCustomerCsr` command if needed.

```
PCA-ADMIN> generatecustomerCsr country=IN state=KA locality=blr \
  organization=example organizationunit=adminexample,pca
 email=test@example.com
Data:
  status = success
  message = Successfully generated customer csr:
     Please download all CSR files from: /nfs/shared_storage/certs/
customer_csr/
```

   Allowable Distinguished Names include country, state, locality, organization unit, and email.

4. Log in to a management node and go to the shared storage directory as indicated in the command output. Download the CSRs.
   - external_tls_term.csr.pem
   - external_admin_tls_term.csr.pem
   - zfssa.csr.pem
5. Based on these CSRs, create certificates signed by your CA.

   > **Important:**
   >
   > When you generate your certificates you must add the FDQNs (and no IP addresses) from the SAN information in the CSRs.

DNS Notes

If you supply outside certificates to establish a CA trust chain, you must add PTR records to the data center DNS. A PTR (Pointer record) in DNS maps an IP address to a hostname. This behavior is the reverse of the usual IP address lookup for a supplied hostname, which is provided by an A record in DNS.

You must create `ReverseIp` lookup zones for the two `ReplicationIps` used in disaster recovery. The DNS requests are forwarded to the Private Cloud Appliance in the same way as requests for the Private Cloud Appliance Service Zone are forwarded. If only the `zfsCapacityPoolReplicationEndpoint` is defined, then only a PTR record for that IP address in is needed.

To create a `ReverseIp` lookup you need to create a DNS zone for the `ReverseIp` lookup. You create one or more reverse lookup zones depending on how the Replication IPs are configured. How to create these PTR records depends on the interface for the data center's DNS servers.

For example, if the rack domain is `myprivatecloud.example.com`, and the capacity and performance storage pool IPs are `10.170.123.98` and `10.170.123.99`, Private Cloud Appliance requires two zones with the following mappings:

```
98.123.170.10.in-addr.arpa rtype PTR rdata sn01-
dr1.myprivatecloud.example.com
99.123.170.10.in-addr.arpa rtype PTR rdata sn02-
dr1.myprivatecloud.example.com
```

For more information, see these Networking topics:

- Name Resolution on page 466
- Managing Public DNS Zones on page 537

You can proceed to the uploading process.

Uploading Custom CA Certificates

To complete the setup, you must upload the new signed CA certificates along with the CA trust chain to your Private Cloud Appliance.

Use the `uploadCustomerCerts` command to upload the CA certificates. This command uses the following parameters to provide the full paths to the certificates and the CA trust chain:

- `caTrustChain`
- `externalAdminCert`
- `externalCert`
- `zfsCert`

1. Log into the Service CLI.
2. Copy the CA certificates and your CA trust chain to the `/nfs/shared_storage` directory on the management node.
3. Run the `uploadCustomerCerts` command to upload all the CA certificates. For example:

```
PCA-ADMIN> uploadCustomerCerts externalcert=/nfs/shared_storage/
external_tls_term.cert \
zfsCert=/nfs/shared_storage/certs/zfssa.cert caTrustChain=/nfs/
shared_storage/CAPrivate.pem
Data:
 status = success
 message = Successfully uploaded customer CERTS
```

> **Important:**
>
> Upload your CA trust chain with one of the CA certificate upload commands by using the `caTrustChain` parameter.

If your Private Cloud Appliance has the Admin networking feature enabled, the `uploadCustomerCerts` command requires the additional `externalAdminCert` parameter. For example:

```
PCA-ADMIN> uploadCustomerCerts externalcert=/nfs/shared_storage/
external_tls_term.cert \
zfsCert=/nfs/shared_storage/certs/zfssa.cert caTrustChain=/nfs/
shared_storage/CAPrivate.pem \
externalAdminCert=/nfs/shared__storage/external_admin_tls_term.cert
Data:
 status = success
 message = Successfully uploaded customer CERTS
```

Status and Health Monitoring

The overall health status of Private Cloud Appliance is continually monitored, using real-time data from the hardware and platform layers. System health checks and monitoring data are the foundation of problem detection and the starting point for troubleshooting an unhealthy condition.

If necessary, administrators submit a service request with Oracle for assistance in resolving the problem. If the system supports and is registered for Oracle Auto Service Request (ASR), certain hardware failures cause a service request and diagnostic data to be automatically sent to Oracle support.

Monitoring

Independently of the built-in health checks, an administrator can consult the monitoring data at any time to verify the overall status of the system or the condition of a particular component or service. This is done through the Grafana interface, by querying the system-wide metric data stored in Prometheus.

Grafana provides a visual approach to monitoring: it allows you to create dashboards composed of a number of visualization panels. Each panel corresponds with a single metric query or a combination of metric queries, displayed

in the selected format. Options include graphs, tables, charts, diagrams, gauges, and so on. For each metric panel, thresholds can be defined. When the query result exceeds or drops below a given threshold, the display color changes, providing a quick indication of which elements are healthy, require investigation, or are malfunctioning.

A set of predefined dashboards allow administrators to start monitoring the system as soon as it is up and running. The default monitoring dashboards are grouped into the following categories:

Monitoring Dashboard Set	Description
Service Advisor	Appliance-specific collection of dashboards for monitoring the Kubernetes container orchestration environment, the containerized services it hosts, and the system health check services.
Service Level Monitoring	A read-only collection of dashboards that provide statistic data for all the microservices.
Kubernetes Monitoring	An additional collection of dashboards provided by Oracle's cloud native monitoring and visualization experts. These provide vast and detailed information about the Kubernetes cluster and its services.

The default dashboards contain metric data for the system's physical components – servers, switches, storage providers and their operating systems and firmware – as well as its logical components – controller software, platform, Kubernetes cluster and microservices, compute instances and their virtualized resources. This allows the administrator or support engineer to verify the health status of both component categories independently, and find correlations between them. For example, a particular microservice might exhibit poor performance due to lack of available memory. The monitoring data indicates whether this is a symptom of a system configuration issue, a lack of physical resources, or a hardware failure. The monitoring system has an alerting service capable of detecting and reporting hardware faults. The administrator may optionally configure a notification channel to receive alerts based on rules defined in the monitoring system.

As part of service and support operations, Oracle may ask you to report specific metric data displayed in the default dashboards. For this reason, the default dashboard configurations should always be preserved. However, if some of the monitoring functionality is inadvertently modified or broken, the defaults can be restored. In a similar way, it is possible for Oracle to create new dashboards or modify existing ones for improved monitoring, and push them to your operational environment without the need for a formal upgrade procedure.

All the open source monitoring and logging tools described here have public APIs that allow customers to integrate with their existing health monitoring and alerting systems. However, Oracle does not provide support for such custom configurations.

Fault Domain Observability

When it comes to keeping the appliance infrastructure, the compute instances and their related resources running in a healthy state, the *Fault Domain* is an extremely important concept. It groups a set of infrastructure components with the goal of isolating downtime events due to failures or maintenance, making sure that resources in other Fault Domains are not affected.

In line with Oracle Cloud Infrastructure, there are always three Fault Domains in a Private Cloud Appliance. Each of its Fault Domains corresponds with one or more physical compute nodes. Apart from using Grafana to consult monitoring data across the entire system, an administrator can also access key capacity metrics for Fault Domains directly from the Service Enclave:

- Number of compute nodes per Fault Domain
- Total and available amount of RAM per Fault Domain
- Total and available number of vCPUs per Fault Domain
- Unassigned system CPU and RAM capacity

The Fault Domain metrics reflect the actual physical resources that can be consumed by compute instances hosted on the compute nodes. The totals do not include resources reserved for the operation of the hypervisor: 40GB RAM and 4 CPU cores (8 vCPUs).

In addition to the three Fault Domains, the Service CLI displays an "Unassigned" category. It refers to installed compute nodes that have not been provisioned, and thus are not part of a Fault Domain yet. For unassigned compute nodes the memory capacity cannot be calculated, but the CPU metrics are displayed.

System Health Checks

Health checks are the most basic form of detection. They run at regular intervals as Kubernetes CronJob services, which are very similar to regular UNIX cron jobs. A status entry is created for every health check result, which is always one of two possibilities: *healthy* or *not healthy*. All status information is stored for further processing in Prometheus; the unhealthy results also generate log entries in Loki with details to help advance the troubleshooting process.

Health checks are meant to verify the status of specific system components, and to detect status changes. Each health check process follows the same basic principle: to record the current condition and compare it to the expected result. If they match, the health check passes; if they differ, the health check fails. A status change from healthy to *not healthy* indicates that troubleshooting is required.

For the purpose of troubleshooting, there are two principal data sources at your disposal: logs and metrics. Both categories of data are collected from all over the system and stored in a central location: logs are aggregated in Loki and metrics in Prometheus. Both tools have a query interface that allows you to filter and visualize the data: they both integrate with Grafana. Its browser-based interface can be accessed from the Service Web UI.

To investigate what causes a health check to fail, it helps to filter logs and metric data based on the type of failure. Loki categorizes data with a labeling system, displaying log messages that match the selected log label. Select a label from the list to view the logs for the service or application you are interested in. This list allows you to select not only the health checks but also the internal and external appliance services.

In addition, the latest status from each health check is displayed in the Platform Health Check dashboard, which is part of the Service Advisor dashboard set provided by default in Grafana.

Private Cloud Appliance runs the health checks listed below.

Health Check Service	Description
cert-checker	Verifies on each management node that no certificates have expired.
flannel-checker	Verifies that the Flannel container network service is fully operational on each Kubernetes node.
kubernetes-checker	Verifies the health status of Kubernetes nodes and pods, as well as the containerized services and their connection endpoints.
mysql-cluster-checker	Verifies the health status of the MySQL cluster database.
l0-cluster-services-checker	Verifies that low-level clustering services and key internal components (such as platform API, DHCP) in the hardware and platform layer are fully operational.
network-checker	Verifies that the system network configuration is correct.
registry-checker	Verifies that the container registry is fully operational on each management node.
vault-checker	Verifies that the secret service is fully operational on each management node.
etcd-checker	Verifies that the etcd service is fully operational on each management node.

Health Check Service	Description
zfssa-analytics-exporter	Reports ZFS Storage Appliance cluster status, active problems, and management path connection information. It also reports analytics information for a configurable list of datasets.

Centralized Logging

The platform provides unified logging across the entire system. The Fluentd data collector retrieves logs from components across the entire system and stores them in a central location, along with the appliance telemetry data. As a result, all the necessary troubleshooting and debugging information is maintained in a single data store, and does not need to be collected from different sources when an issue needs to be investigated. The overall health of the system is captured in one view, a Grafana dashboard, meaning there is no need for an administrator to check individual components.

Whenever an issue is found that requires assistance from Oracle, the administrator logs a service request. A support bundle is usually requested as part of that process. Thanks to the centralized logging, the support bundle is straightforward to generate, and remains possible even if system operation is severely compromised. Generating the support bundle is a scripted operation that produces a single compressed archive file. The administrator does need to manually upload the archive file containing the consolidated logs and other diagnostic data.

If the Private Cloud Appliance is registered for ASR, certain hardware failures cause a service request and diagnostic data to be automatically sent to Oracle support.

Serviceability

Serviceability refers to the ability to detect, diagnose and correct issues that occur in an operational system.

The primary requirement is the collection of system data: general hardware telemetry details, log files from all system components, and results from system and configuration health checks. As an engineered system, Private Cloud Appliance is designed to process the collected data in a structured manner.

To provide real-time status information, the system collects and stores metric data from all components and services in a unified way using Prometheus. Centrally aggregated data from Prometheus is visualized through metric panels in a Grafana dashboard, which enables an administrator to check overall system status at a glance. Logs are captured across the appliance using Fluentd, and collected in Loki for diagnostic purposes.

When a component status changes from healthy to not healthy, the alerting mechanism can be configured to send notifications to initiate a service workflow. If support from Oracle is required, the first step is for an administrator to open a service request and provide a problem description.

If the Private Cloud Appliance is registered for Oracle Auto Service Request (ASR), certain hardware failures cause a service request and diagnostic data to be automatically sent to Oracle support. The collection of diagnostic data is also called a support bundle. A Service Enclave administrator can also create and send a service request and supporting diagnostic data separate from ASR. For more information about ASR and support bundles, see these topics:

- Oracle Auto Service Request (ASR) on page 272
- Support Bundles on page 282

To resolve the reported issue, Oracle may need access to the appliance. For this purpose, a dedicated service account is configured during system initialization. For security reasons, this non-root account has no password. You must generate and provide a service key to allow the engineer to work on the system on your behalf. Activity related to the service account leaves an audit trail and is clearly separated from other user activity.

Most service scenarios for Oracle Engineered Systems are covered by detailed action plans, which are performed by the assigned field engineer. When the issue is resolved, or if a component has been repaired or replaced, the engineer validates that the system is healthy again before the service request is closed.

This structured approach to problem detection, diagnosis and resolution ensures that high-quality service is delivered, with minimum operational impact, delay and cost, and with maximum efficiency.

Using Grafana

With Grafana, Private Cloud Appliance offers administrators a single, visual interface to the logs and metrics collected at all levels and across all components of the system.

This section provides basic guidelines to access Grafana and navigate through the logs and monitoring dashboards. For additional information about Grafana services and how to use Grafana, see the Oracle Systems blog Oracle PCA X9-2 Monitoring and Alerting with Grafana.

The Grafana Home Page

Select one of these options to access Grafana:

- Service Enclave `admin` user.

  1. Log in to the Service Web UI.
  2. On the right side of the dashboard, click the Monitoring tile.

     The Grafana login page opens in a new browser tab. In the future, you can go directly to this Grafana login page as described in "Any Grafana user" below.
  3. Enter your user name and password at the prompts.

     You can create new users, and give the new users the direct URL to the Grafana login page. To create new Grafana users, see Managing Grafana Users on page 255. For the direct URL to the Grafana login page, see "Any Grafana user" below.

- Any Grafana user.

  You do not need to be a Service Enclave user.

  1. Go to the Grafana login page.

     The Grafana login page is `https://grafana.`***`pca_name`***`.`***`your_domain`***`/login`, where ***`pca_name`*** is the name of the Private Cloud Appliance.
  2. Enter your user name and password at the prompts.

The Welcome panel on the Grafana home page contains many links to `grafana.com` for information about how to use Grafana, such as how to create your own dashboards, queries, and alerts. You can also find Grafana tutorials in the Private Cloud Appliance 3.x playlist on the Oracle Learning YouTube Channel, or search for Grafana on Oracle Blogs.

On the left side of the home page is a vertical bar with icons that open the list of dashboards or the list of alerts, for example, or provide access to system logs as described in Viewing and Interpreting Monitoring Data in Grafana on page 258. Your user icon near the bottom of the bar enables you to change your preferences settings or log out. The Grafana logo at the top of the bar takes you back to the Grafana home page.

The Grafana Time Line

When logs and metrics are stored in Prometheus, they are given a time stamp based on the time and time zone settings of the Private Cloud Appliance. However, Grafana displays the time based on user preferences, which might result in an offset because you are in a different time zone.

Use the following instructions if you want to synchronize the time line in the Grafana visualizations with the time zone of the appliance:

1. Near the bottom of the vertical menu bar on the left side of the Grafana page, click your user account icon and click the Preferences option on the submenu that pops up.
2. In the Preferences section of the page, change the Timezone setting to the same time zone as the appliance.
3. Click the Save button at the bottom of that section to apply the change.

Monitoring Multiple Private Cloud Appliance Systems

If you need to deploy an external Grafana service with variable-driven dashboards to monitor multiple Private Cloud Appliance systems, see the following resources:

- Observability, Monitoring, and Alerting Across Multiple Oracle Private Cloud Appliance X9-2 Systems - Part 1
- Observability, Monitoring, and Alerting Across Multiple Oracle Private Cloud Appliance X9-2 Systems - Part 2

Managing Grafana Users

Grafana works with its own user accounts. A Private Cloud Appliance administrator can sign in to Grafana and add new users. User permissions can be refined through teams and folders.

Users

To add a new user, perform the following procedure as the `admin` user:

1. In the vertical menu bar on the left side of the Grafana home page, click the Server Admin (shield) icon.
2. On the Server Admin drop down menu, click Users.
3. Click the New User button.
4. Enter the requested information, and click the Create User button.

   By default, the new user has the Viewer role. You could modify the user to change the role. Another way to change a user's access is to add the user to a team that has the required access.

The following are the Grafana user roles:

- **Admin**: Has access to all organization resources, including dashboards, users, and teams.
- **Editor**: Can view and edit dashboards, folders, and playlists.
- **Viewer**: Can view dashboards and playlists.

By default, an editor can edit all of the listed resources, and a viewer can view all of the listed resources. A user with the Admin role can grant or restrict permissions to specific resources for specific roles, teams, and users. For example, click the Permissions tab on a folder to change the permissions to that folder for the Editor or Viewer roles. Click the Add Permissions button on the Permissions tab to add permissions for specific users or teams.

Teams

To create a new team, perform the following procedure as the `admin` user:

1. In the vertical menu bar on the left side of the Grafana home page, click the Configuration (gear) icon.
2. On the Configuration drop down menu, click Teams.
3. Click the New Team button.
4. Enter the requested information, and click the Create button.
5. Click the Add Member button.
6. In the Add team member box, click the drop-down arrow and select the user you want to add to the team.
7. Click the Add to team button.
8. Click the Settings tab at the top of the page to modify team settings such as home dashboard and time zone.

Folders

Use folders to grant permissions to users and teams. Perform the following procedure as the `admin` user:

1. In the vertical menu bar on the left side of the Grafana home page, click the Dashboards (grid) icon.
2. On the Dashboards drop down menu, click Manage.
3. For the folder for which you want to grant teams and users permissions, click Go to folder.
4. At the top of the folder page, click the Permissions tab.
5. Click the Add Permission button.
6. In the Add Permission For box, select the team or user, and select the role for the user or for all users on the team.
7. Click the Save button.

You can also grant permissions for specific dashboards in a folder. Perform the following procedure as the `admin` user:

1. In the vertical menu bar on the left side of the Grafana home page, click the Dashboards (grid) icon.
2. On the Dashboards drop down menu, click Manage.
3. Click the name of the folder that contains the dashboard, and then click the dashboard.
4. At the top of the dashboard page, click the gear icon.
5. In the menu on the left side of the page, click Permissions.
6. Click the Add Permission button.
7. In the Add Permission For box, select the team or user, and select the role for the user or for all users on the team.
8. Click the Save button.
9. Click the Save Dashboard button.

Managing Grafana Dashboards

Private Cloud Appliance includes a set of predefined Grafana dashboards, which are organized into folders.

Use any of the following to display the list of folders of dashboards:

- The magnifying glass icon in the vertical menu bar on the left side the page
- The Dashboards > Manage option in the menu bar
- The dashboards Home button to the right of the Grafana logo at the top of the menu bar

Click a folder name or the arrow to the right of a folder name to show the dashboards in that folder.

Use the buttons at the top of the list to toggle between showing the list of folders and showing the list of all dashboards.

In the search field at the top of the page, enter text from the name of a folder or dashboard to show only those dashboards.

Click the name of a dashboard to show the content of that dashboard. On a dashboard, you can click the star to the right of the dashboard name at the top of the page to list this dashboard on the Grafana home page for faster access.

The dashboard shows information such as the query, graphs of the data collected over time, and alerts set for that data.

You are able to modify most dashboards, but note that Oracle Support might require that information. The Grafana home page contains links to information for how to create your own dashboards and queries rather than modify dashboards that were provided by Oracle. For your custom dashboards, first create one or more folders to keep these new dashboards separate from dashboards provided by Oracle.

Managing Grafana Alerts

Private Cloud Appliance includes a set of predefined Grafana alerts. You can also add your own alerts.

You can show only alerts that are in a specified state, such as Alerting. You can display detailed information about the alert, including the values that trigger the alert and that trigger a state change. In many cases, you can change these values. If an alert is in the Alerting state, view the alert definition to determine what caused the alert to go to that state, and then use this information to evaluate the component that the alert is monitoring and determine what action might be needed.

Browsing Grafana Alerts

To view all alerts, click the bell icon in the vertical menu bar on the left side the page. An icon shows the status of each alert, and text below the alert name shows how long the alert has been in that status.

Enter text in the search field at the top of the list to show only alerts with that text in their names. Use the States list to show only alerts that are in the selected state: OK, Not OK, Alerting, No Data, Paused, Pending.

Use the "How to add an alert" button above the alerts list to create a new alert, or use information referenced on the Grafana home page to add or modify an alert, add a notification channel, and add a notification for a particular alert.

Click an alert name to see detailed information about the alert. This is the same page you see if you go to the dashboard, scroll to the metric, click the metric name, and select Edit.

Hover over the graph to list all data that is being monitored, for example each host, switch, device, or endpoint.

On the Alert tab below the graph, you can view and edit the rule. An alert rule consists of one or more queries and expressions, a condition, the frequency of evaluation, and optionally, the duration over which the condition is met. You can see how the alert state is set for various error conditions. You can send a notification message for this alert.

The state history button shows the last 50 state changes for this alert. Another button enables you to test the alert.

Adding or Configuring Notification Channels

To add or configure notification channels, click the bell icon in the menu bar on the left side the page and then select the Notification channels option, or select the Notification channels tab at the top of the list of alerts.

To change the configuration of an existing notification channel, click the name of the channel. When you are finished making changes, click the Save button. Click the Test button to send a test notification.

To add a notification channel, go to the Notification channels tab, click the New channel button, and fill out the page. Click the Save button. Click the Test button to send a test notification. Click the Back button to cancel and not create a new notification channel.

Configuring Custom External Email Notifications

To configure email notification, open a service request (SR) for Oracle Support to do the initial configuration. When the initial configuration is complete, go to the Grafana alerts page, click the Notification channels tab, click the New channel button, and fill out the page, selecting Email in the Type field.

Configuring Custom External HTTP/HTTPS Notifications

To configure external HTTP or HTTPS based custom alerts, you must first configure the proxy for Grafana as shown in the following example.

Log in to the management node that owns the management virtual IP, and run the following command:

```
$ sudo curl -u admin_user_name -XPUT \
'https://api.PCA_system_name.your_domain/v1/grafana/proxy/config?http-
proxy=proxy_fqdn:proxy_port&https-proxy=proxy_fqdn:proxy_port'
Enter host password for user 'admin_user_name':
Grafana proxy config successfully updated!
```

Run the following command so that Grafana can still contact the internal Loki and Prometheus services:

```
$ sudo curl -u admin_user_name -XPUT \
'https://api.PCA_system_name.your_domain/v1/grafana/proxy/config?no-
proxy="sauron-sauron-prometheus,sauron-sauron-alertmanager,grafana-
loki.loki.svc.cluster.local"'
```

The Grafana pod is restarted. Run the following command until you see that the Grafana pod (`sauron-sauron-grafana-`***`unique_ID`***) is running:

```
$ kubectl get pods -n sauron
```

Checking the Health and Status of Hardware and Platform Components

The predefined Grafana dashboards allow you to check the status of essential low-level components in Private Cloud Appliance, and view real-time and historic details of the relevant metrics.

The hardware and platform layers form the foundations of the system architecture. Any unhealthy condition at this level is expected to have an adverse effect on operations in the infrastructure services. The dashboards described in this section provide a good starting point for basic system health checks, and for troubleshooting if issues are found.

You might prefer to use different dashboards, metrics, and visualizations instead. The necessary data, collected across the entire system, is stored in Prometheus, and can be queried and presented through Grafana in many different ways.

Grafana Folder	Dashboard	Description
Service Monitoring	Server Stats	This comprehensive dashboard displays telemetry data for the server nodes. It includes graphs for CPU and memory utilization, disk activity, network traffic, and so on. Some panels in this dashboard display a large number of time series in a single graph. Click to display a single time series, or hover over the graph to view detailed data at a specific time.
PCA 3.0 Service Advisor	Platform Health Check	This dashboard integrates the appliance health check mechanisms into the centralized approach that Grafana provides for logging and monitoring. By default, the Platform Health Check dashboard displays all health check services. Use the buttons above the Platform Health Check list to change the content of the list. Use the Platform Service list to select a single health checker. Use the Health Check Status list to display all results or only healthy results. Use the Filters list to select a filter and a value. Typically, if you see health check failures you want to start troubleshooting. For that purpose, each health check result contains a time stamp that serves as a direct link to the related Loki logs. To view the logs related to any health check result, click the time stamp.
My Dashboards (Read Only)	Node Exporter Full	This dashboard displays a large number of detailed metric panels for a single compute or management node. Use the Host button at the top of the page to display data for a different host. This dashboard could be considered a fine-grained extension of the Server Stats dashboard. The many different panels provide detailed coverage of the server node hardware status as well as the operating system services and processes. Information that you would typically collect at the command line of each physical node is combined into a single dashboard showing live data and its evolution over time. All dashboards in the My Dashboards folder provide data that would be critical in case a system-level failure needs to be resolved. Therefore, these dashboards cannot be modified or deleted.

Viewing and Interpreting Monitoring Data in Grafana

The infrastructure services layer of Private Cloud Appliance, which is built on top of the platform and enables all the cloud user and administrator functionality, can be monitored through an extensive collection of Grafana dashboards.

These microservices are deployed across the three management nodes in Kubernetes containers, so their monitoring is largely based on Kubernetes node and pod metrics. The Kubernetes cluster also extends onto the compute nodes, where Kubernetes worker nodes collect vital additional data for system operation and monitoring.

The dashboards described in this section provide a good starting point for microservices health monitoring. You might prefer to use different dashboards, metrics and visualizations instead. The necessary data, collected across the entire system, is stored in Prometheus, and can be queried and presented through Grafana in many ways.

Grafana Folder	Dashboard	Description
Service Monitoring	ClusterLabs HA Cluster Details	This dashboard uses a bespoke Prometheus exporter to display data for HA clusters based on Pacemaker. On each HTTP request it locally inspects the cluster status, by parsing preexisting distributed data provided by the cluster components' tools. The monitoring data includes Pacemaker cluster summary, nodes and resource stats, and Corosync ring errors and quorum votes.
Service Monitoring	MySQL Cluster Exporter	This dashboard displays performance details for the MySQL database cluster. Data includes database service metrics such as uptime, connection statistics, table lock counts, as well as more general information about MySQL objects, connections, network traffic, memory and CPU usage, etc.
Service Monitoring	Service Level	This dashboard displays detailed information about RabbitMQ requests that are received by the fundamental appliance services. It allows you to monitor the number of requests, request latency, and any requests that caused an error.
Service Monitoring	VM Stats	This comprehensive dashboard displays resource consumption information across the compute instances in your environment. It includes graphs for CPU and memory utilization, disk activity, network traffic, and so on. The panels in this dashboard display a large number of time series in a single graph. You can click to display a single time series, or hover over the graph to view detailed data at a specific point on the time axis.
PCA 3.0 Service Advisor	Kube Endpoint	This dashboard focuses specifically on the Kubernetes endpoints and provides endpoint alerts. These alerts can be sent to a notification channel of your choice.
PCA 3.0 Service Advisor	Kube Ingress	This dashboard provides data about ingress traffic to the Kubernetes services and their pods. Two alerts are built-in and can be sent to a notification channel of your choice.
PCA 3.0 Service Advisor	Kube Node	This dashboard displays metric data for all the server nodes, meaning management and compute nodes, that belong to the Kubernetes cluster and host microservices pods. You can monitor pod count, CPU and memory usage, and so on. The metric panels display information for all nodes. In the graph-based panels you can click to view information for just a single node.
PCA 3.0 Service Advisor	Kube Pod	This dashboard displays metric data at the level of the microservices pods, allowing you to view the total number of pods overall and how they are distributed across the nodes. You can monitor their status per namespace and per service, and check if they have triggered any alerts.
PCA 3.0 Service Advisor	Kube Service	This dashboard displays metric data at the Kubernetes service level. The data can be filtered for specific services, but displays all by default. Two alerts are built-in and can be sent to a notification channel of your choice.

Grafana Folder	Dashboard	Description
Kubernetes Monitoring Kubernetes Monitoring Containers Kubernetes Monitoring Node	(all)	These folders contain a large and diverse collection of dashboards with a wide range of monitoring data that covers most of the operations of the Private Cloud Appliance system Kubernetes cluster. For example, these metrics provide information about deployment, ingress, and usage of CPU, disk, memory, and network resources.
OKE Monitoring	CAPOCI	This dashboard shows metrics from the Cluster API Provider for OCI, which is a component of the Private Cloud Appliance Kubernetes Engine (OKE). This dashboard monitors request status codes and response times for resources used by OKE such as compute instances and load balancers. The information about controller reconciliation is for Oracle Support.
OKE Monitoring	Cluster Time Monitoring	This dashboard shows the time taken for operations such as create or update a particular OKE cluster or node pool. Average time for these operations across all clusters and node pools also is shown.
OKE Monitoring	Metrics Meter	This dashboard shows the health of various targets used by the OKE service such as the Cluster API Provider, the Cluster API Provider for OCI, OKE, and prometheus-k8s.
OKE Monitoring	OKE Service	This dashboard shows the service level metrics for OKE. Examples of metrics on this dashboard include counts of requests such as cluster and node pool create, update, and delete, and counts of exception codes for various requests. The exception code counts help expose any patterns in request failures.

Monitoring System Capacity

It is important to track the key metrics that determine Private Cloud Appliance capacity to host your compute instances and the storage they use. Administrators have direct access to the current consumption of CPU, memory, and storage space.

The detailed data for compute node load and storage usage can be found in the Grafana dashboards. This topic explains how to access the most critical metrics directly from the Service Enclave.

Viewing CPU and Memory Usage By Fault Domain

These procedures display the number of compute nodes, the amount of total memory and free memory, and the number of total and free virtual CPUs for each fault domain.

The `UNASSIGNED` row refers to compute nodes that are not currently assigned to a fault domain. Because these compute nodes do not belong to a fault domain, their memory and CPU usage *in a fault domain* is zero.

To display this information and more for an individual compute node, select PCA Config > Rack Units from the navigation menu, or select the Rack Units tile on the Dashboard, and then click the name of a compute node in the list.

Using the Service Web UI

1. In the navigation menu, select PCA Config > Fault Domains.

2. Click the name of a fault domain to see the information for only that fault domain.

Using the Service CLI

Enter the `getFaultDomainInfo` command.

```
PCA-ADMIN> getFaultDomainInfo
Data:
  id           totalCNs   totalMemory   freeMemory   totalvCPUs
 freevCPUs
  --           --------   -----------   ----------   ----------
 ---------
  UNASSIGNED   1          0.0           0.0          0           0
  FD1          2          1072.0        976.0        176         164
  FD2          1          984.0         984.0        120         120
  FD3          1          984.0         984.0        120         120
```

The Notes column is omitted from the preceding example.

Viewing Disk Space Usage on the ZFS Storage Appliance

The Service Enclave runs a storage monitoring tool called ZFS pool manager, which polls the ZFS Storage Appliance every 60 seconds. Using the Service CLI, you can display current information about the usage of available disk space in each ZFS pool. You can also set the usage threshold that triggers a fault when the threshold is exceeded.

Checking the Storage Status of ZFS Pools

List ZFS pools.

```
PCA-ADMIN> list ZfsPool
Data:
  id                                     name
  --                                     ----
  e898b147-7cf0-4bd0-8b54-e32ec83d04cb   PCA_POOL
  c2f67943-df81-47a5-9713-06768318b623   PCA_POOL_HIGH
```

In a standard storage configuration, you only have one pool. If your system includes high-performance disk trays, then you can view usage information for each pool separately.

```
PCA-ADMIN> show ZfsPool id=e898b147-7cf0-4bd0-8b54-e32ec83d04cb
Data:
  Id = e898b147-7cf0-4bd0-8b54-e32ec83d04cb
  Type = ZfsPool
  Pool Status = Online
  Free Pool = 44879343128576
  Total Pool = 70506183131136
  Pool Usage Percent = 0.3634693989163486
  Name = PCA_POOL
  Work State = Normal
```

Configuring the Fault Threshold of the ZFS Pool Manager

By default, the fault threshold is set to 80 percent full: `usage percentage` 0.8.

```
PCA-ADMIN> show ZfsPoolManager
Data:
  Id = a6ca861b-f83a-4032-91c5-bc506394d0de
  Type = ZfsPoolManager
  LastRunTime = 2022-10-09 12:17:52,964 UTC
  Poll Interval (sec) = 60
  The minimum Zfs pool usage percentage to trigger a major fault = 0.8
  Manager's run state = Running
```

The following example sets the fault threshold to 75 percent full: `usageMajorFaultPercent=0.75`.

```
PCA-ADMIN> edit ZfsPoolManager usageMajorFaultPercent=0.75
JobId: 67cfe180-f2a2-4d59-a676-01b3d73cffae
```

Viewing System Log Data

Logs are collected from all over the Private Cloud Appliance system and aggregated in Loki.

Learn about consulting and using logs in these sections:

- Using Grafana Explore Queries on page 262
- Using Vector for Log Streaming on page 264

Using Grafana Explore Queries

System log data in Private Cloud Appliance can be queried, filtered, and displayed using Grafana Explore queries.

Loki Logs

Loki uses labels to categorize log messages. A query specifies labels, and Loki displays the service and application log messages that match the query selections.

Labels are key-value pairs. Use the following procedure to select labels for your query.

1. Open the Grafana home page.
2. Open the Explore pane.

   In the vertical menu bar on the left side of the page, click Explore (the compass icon).
3. To query Loki data, select Loki from the Explore data source menu at the top of the page to the right of the "Explore" title.

   Loki query options are displayed. For example, a Log Browser menu is shown at the top of the page.
4. Query and filter the logs.

   The following methods are similar. Both methods allow you to select labels and values from lists. The second method enables you to more easily select multiple labels and multiple values for one query.

   - Entering a Query in the Text Field on page
   - Building a Complex Query More Easily on page

   Once you have created a query, you can select the same query again from the history list.

Additional query options:

- Add query. Click the Add query button to create another query and show the result of all separate queries together in the same timeline and message list.
- Query history. Run a query that was previously run, or copy or delete the query, add a comment to the query, or star the query so that you can use the Starred button to list only starred queries. At the top of the Query history list you can enter a search string to filter the list, and you can select how to order the list.
- Recurring run. Click the arrow on the Run query button, and select an interval from the menu. To stop the recurring runs, select Off at the top of the menu.

The timeline is displayed below the Log browser section of the Explore pane. Below the timeline, the log messages that match the query are displayed.

Messages are color-coded both in the timeline and in the message list to indicate whether the message is informational, a warning, error, or other.

Use the Query type button to choose to show the results over a range of time or at just one point in time. Use the range button at the top of the page (see the clock icon) to set the range.

Select a portion of the timeline to zoom in to focus on a smaller amount of data. To zoom out, use the magnifying glass button at the top of the page next to the range button.

In the message list, click the arrow on the left side of the time stamp of a message to display all labels that match that message. You can then click the plus + magnifying glass icon to add that label to your query results or click the minus - magnifying glass icon to remove that label from your query results. Notice that the query that you entered changes.

Entering a Query in the Text Field

1. In the text field to the right of the Log browser button, enter the open brace { character.

   The closed brace is automatically added, and a list of labels pops up.
2. Select a label from the list.

   You might need to scroll the list to see all labels, or you can start typing a label name to filter the list.

   The selected label is inserted into the query in the text field, an equals sign is added, and a list of values for that label pops up.
3. Select a value from the list.

   You might need to scroll the list to see all values, or you can start typing a value name to filter the list.

   The selected value is inserted inside quotation marks.
4. If you want to further filter the query result, enter a comma.

   The list of labels pops up again, followed by the list of values after you select a label.
5. Run the query.

   Type Shift+Enter, or click the Run query button in the upper right corner of the pane.

   The timeline and log messages are displayed below the query building options.

Building a Complex Query More Easily

Click the Log browser button so that the arrow on the button points down.

A query builder is displayed with the following steps:

1. Select labels.

   Step 1 displays a row of buttons with a label name on each button. When you click one of these label buttons, a list pops up under Step 2 that shows the values for that label.

   You can click more than one label button. If you click another label button, the list of values for the new label pops up with the first list of values under Step 2.

   When you click a label button that is already selected, that label is removed from the query.
2. Choose values for the selected labels.

   Step 2 shows the list of values for each label that is selected in Step 1. You might need to scroll the list to see all possible values, or you can start typing a value name in the search field to filter all value lists.

   When you select a value from one list, some values might be removed from another list.

   You can select more than one value from a particular list. Selecting a value that is already selected removes that value from the query.

   As you select or deselect values, the query is built and displayed in Step 3.
3. Show the query result.

   Click the Show Logs button in Step 3.

   The timeline and log messages are displayed below the query building options.

   The completed query is displayed in the field to the right of the Log browser button. You can edit the query in the Log browser field and click the Run query button to show a new result.

Audit Logs

The audit logs can be consulted as separate categories. From Log browser lists, you can select the following audit labels. As described in Loki Logs on page 262, either enter the queries shown in the following list in the text field,

or select `job` or `log` from the Log labels list, and then select one of the values shown in the following list. See also the example custom query immediately following this list.

- `job="vault-audit"`

  Use this log label to filter for the audit logs of the Vault cluster. Vault, a key component of the secret service, keeps a detailed log of all requests and responses. You can view every authenticated interaction with Vault, including errors. Because these logs contain sensitive information, many strings within requests and responses are hashed so that secrets are not shown in plain text in the audit logs.
- `job="kubernetes-audit"`

  Use this log label to filter for the audit logs of the Kubernetes cluster. The Kubernetes audit policy is configured to log request metadata: requesting user, timestamp, resource, verb, etc. Request body and response body are not included in the audit logs.
- `job="audit"`

  Use this log label to filter for the Oracle Linux kernel audit daemon logs. The kernel audit daemon (`auditd`) is the userspace component of the Linux Auditing System. It captures specific events such as system logins, account modifications, and `sudo` operations.
- `log="audit"`

  Use this log label to filter for the audit logs of the ZFS Storage Appliance.

In addition to using the log labels from the list, you can also build custom queries. For example, to filter for the audit logs of the admin service and API service, enter the following query into the Log browser text field:

```
{job=~"(admin|api-server)"} | json tag="tag" | tag=~"(api-audit.log|
audit.log)"
```

To run, either press Shift+Enter, or click the Run query button in the upper right corner of the Explore pane.

Load Balancer Logs

The Load Balancer as a Service (LBaaS) logs can be consulted as separate categories. From Log browser lists, you can select the following audit labels. As described in Loki Logs on page 262, either enter the queries shown in the following list in the text field, or select `job` or `log` from the Log labels list, and then select one of the values shown in the following list.

- `job="pca-lbctl"`

  Use this log label to filter for the load balancer controller logs. You can view every client request that is being served. These logs contain API parameters and will contain error details when applicable.
- `job="pcalbmgr"`

  Use this log label to filter for the load balancer instances (manager) logs. You can view every request that is being served. These logs primarily contain the load balancer's configuration and management.

In addition to using the log labels from the list, you can also build custom queries. For example, you can view the controller and manager logs together:

```
{job=~"pca-lbctl|pca-lbmgr"}
```

To run, either press Shift+Enter, or click the Run query button in the upper right corner of the Explore pane.

Using Vector for Log Streaming

The Vector service allows you to send the Private Cloud Appliance logging information from Loki to an external location that you specify.

Beginning with Private Cloud Appliance software version 3.0.2-b1261765, Vector is installed, configured, and enabled on the appliance by default.

To specify which data you want and where you want the data sent, sign in to the currently active management node as the root user and customize the Vector configuration file It is stored in `/nfs/shared_storage/log_streaming/pca_vector.yaml`.

Edit the configuration file to customize the `sinks` section. See the Vector Sinks Reference. The following is a sample `pca_vector.yaml` file:

```
# Copyright (c) 2024, Oracle and/or its affiliates.

# DO NOT TURN API OFF
# otherwise livenessProbe will fail
api:
  enabled: true
  # Bind to 0.0.0.0. Otherwise the API will not be exposed outside the
 container.
  address: "0.0.0.0:8686"

sources:
  fluentd_source:
    type: fluent
    address: "0.0.0.0:8080"
    mode: tcp
    encoding:
      codec: json

transforms:
  log_event:
    type: remap
    inputs:
      - fluentd_source
    source: |
      log(.)

sinks:
  loki_sink:
    type: loki
    inputs:
      - fluentd_source
    endpoint: http://your_external_location:3100
    encoding:
      codec: json
    labels:
      job: "vector"
      namespace: "default"
      system: "pca_name.example.com"
      filename: "{{tag}}"
```

In the `endpoint` value, ***`your_external_location`*** can be an IP address or a domain name. At this location, you could install Grafana or use other tools to filter, manipulate, and display the data.

The value of the `filename` label that is shown in the example causes the name of the source log file to be shown in the Vector Loki Sink output. You can then use that file name as a label to search within Loki and Grafana.

The following is an example Splunk sink:

```
sinks:
  splunk_sink:
    type: splunk_hec_logs
    inputs:
      - source_id
    endpoint: https://splunk_endpoint
    token: splunk-hec-token
    encoding:
```

```
      codec: json
    tls:
      ca_file: "/path/to/ca.pem"
```

The ***splunk-hec-token*** is required to send logs to Splunk. The `ca_file` is optional if you are using HTTPS. For more information about HTTP Event Collector and how to configure and use Splunk, see the Splunk Documentation.

The following command reports the status of the log streaming pod (the Vector service):

```
# kubectl get pods -n log-streaming
NAME                          READY   STATUS    RESTARTS   AGE
log-streamer-bc4d65d78-ndrsk   1/1     Running   0          14d
```

The following command prints the logs from the `log-streamer-bc4d65d78-ndrsk` pod, which has only one container. For more information about the `kubectl logs` command, see kubectl logs on the Kubernetes site.

```
# kubectl logs log-streamer-bc4d65d78-ndrsk -n log-streaming
```

Managing Logging Levels

To help with troubleshooting, you can change the logging level of some services without restarting the service.

With the ability to raise the logging level during runtime, you don't need to log every detail by default. As a result, you can decrease the overall default logging of services, reducing the amount of storage required. The support for logging levels varies by service.

Viewing Logging Levels

- To view services that support dynamic logging, use the `getServicesSupportingLogLevelOverrides` command.

  ```
  PCA-ADMIN> getServicesSupportingLogLevelOverrides
  Command: getServicesSupportingLogLevelOverrides
  Status: Success
  Time: 2025-07-24 00:33:32,328 UTC
  Data:
    admin
    iam
    api
    limit
  PCA-ADMIN>
  ```

- To view enabled logging levels for a service, use the `getSupportedLogLevelOverrides serviceName=`*`service`* command. If logging levels aren't supported for a service, you get an error message.

  ```
  PCA-ADMIN> getSupportedLogLevelOverrides serviceName=admin
  Command: getSupportedLogLevelOverrides serviceName=admin
  Status: Success
  Time: 2025-08-04 23:28:13,889 UTC
  Data:
    Logger Name    Supported Logging Levels
    -----------    ------------------------
    AdminService   SEVERE,WARNING,INFO,FINE,FINER,FINEST
    AUDIT          SEVERE,WARNING,INFO,FINE,FINER,FINEST
  PCA-ADMIN> getSupportedLogLevelOverrides serviceName=iam
  Command: getSupportedLogLevelOverrides serviceName=iam
  Status: Success
  Time: 2025-08-04 23:27:58,208 UTC
  Data:
    Logger Name   Supported Logging Levels
    -----------   ------------------------
    root          DEBUG,INFO,WARNING,ERROR
  ```

```
  pcaiam        DEBUG,INFO,WARNING,ERROR
PCA-ADMIN> getSupportedLogLevelOverrides serviceName=api
Command: getSupportedLogLevelOverrides serviceName=api
Status: Success
Time: 2025-08-04 23:28:03,967 UTC
Data:
  Logger Name   Supported Logging Levels
  -----------   -----------------------
  root          DEBUG,INFO,WARNING,ERROR,CRITICAL
  pcaapi        DEBUG,INFO,WARNING,ERROR,CRITICAL
  audit         DEBUG,INFO,WARNING,ERROR,CRITICAL
```

- To view the current log level of the services, use the `getLogLevelOverrides` command.
-

```
PCA-ADMIN> getloglevelоverrides

Command: getloglevelоverrides
Status: Success
Time: 2025-08-22 06:17:35,817 UTC
Data:
  Service Name   Logger Name    Logger Level
  ------------   -----------    ------------
  admin          adminService   SEVERE
  admin          audit          SEVERE
  iam            root           DEBUG
  iam            pcaiam         DEBUG
  api            NOT SET        NOT SET
  limit          NOT SET        NOT SET
```

Changing Logging Levels

- To set a new log level for a service, use the `setLogLevelOverride service=`*`service`* `loggerName=`*`string`* `level=`*`log_level`* command.

> **Note:**
>
> Run the `getSupportedLogLevelOverrides serviceName=`*`service`* to find the Logger Name and the available logging levels for a service.

```
PCA-ADMIN> setloglevelоverride serviceName=iam loggername=pcaiam

Command: setloglevelоverride serviceName=iam loggername=pcaiam
 loggerlevel=ERROR
Status: Success
Time: 2025-08-22 06:37:24,095 UTC
Data:
  status = PASS
  message = LLO - successfully set log level override
```

- To remove a logging level for a service, use the `unsetLogLevelOverride service=`*`service`* `loggerName=`*`string`* `level=`*`log_level`* command.

```
PCA-ADMIN> unsetLogLevelOverride serviceName=admin loggerName=AdminService
Command: unsetLogLevelOverride serviceName=admin loggerName=AdminService
Status: Success
Time: 2025-02-13 22:29:47,886 UTC
Data:
  status = PASS
  message = LLO - successfully unset log level override
```

Consulting Audit Logs Streamed Through Vector

Private Cloud Appliance uses Vector streaming to provide easy access to a set of audit logs.

The audit log format conforms to the documented Oracle Cloud Infrastructure definition, storing auditable events with a standard envelope and a service specific payload. For more information on the OCI format, see Contents of an Audit Log Event.

With controller software version 3.0.2-b1483396 and newer, the following service logs are streamed through Vector.

- Each log entry indicates the event type and associated service.
- The *identity* section of the event message shows which user account (`principalId` and `principalName`) the request came from.
- Log rotation is applied to the Admin Service logs, which receive a numeric file name suffix. The file with the "`.0`" suffix is the most recent, and is the only one included in log streaming.

API - /var/log/api-server/api-audit.log

Each request to the API server is recorded as a single entry that summarizes how it was processed.


```
{
 "timestamp": "2025-10-28T14:19:14.548+0000",
 "podname": "api-58fd4d47df-79hzz",
 "process": 24,
 "levelname": "INFO",
 "log_type": "audit",
 "cloud_events_version": "0.1",
 "content_type": "application/json",
 "event_type_version": "2.0",
 "event_id": "/448db79d-ee36-458d-99f1-bf7be8d1527c/5421edf2-157e-4128-
bcd9-696653d9890a",
 "event_type": "compute-VolumeAttachmentCollection-GET",
 "source": "compute",
 "message": "{
  'availabilityDomain': 'AD-1',
  'compartmentId': None,
  'definedTags': None,
  'eventGroupingId': '/448db79d-ee36-458d-99f1-
bf7be8d1527c/5421edf2-157e-4128-bcd9-696653d9890a',
  'eventName': 'ListVolumeAttachments',
  'freeformTags': None,
  'identity': {
   'authType': 'native',
   'callerId':
 'ocid1.user.1749XC302V.broom6.ulmhteeqbqwi68rmtc5oj7cpgzkmjxvhzbu8exn08uihl9ng0kb
   'consoleSessionId': '',
   'credentials': None,
   'ipAddress': '253.255.130.107',
   'principalId':
 'ocid1.user.1749XC302V.broom6.ulmhteeqbqwi68rmtc5oj7cpgzkmjxvhzbu8exn08uihl9ng0kb
   'principalName': 'admin',
   'tenantId':
 'ocid1.tenancy.1749XC302V.broom6.bp4t0827a1zd5uw34s6vc9xticohkzek6fu7fd085dfe4v9q
   'userAgent': 'Jersey/3.1.5 (HttpUrlConnection 17.0.16)'
  },
  'request': {
   'action': 'GET',
   'headers': {
    'ACCEPT': 'application/json',
    'AUTHORIZATION': '*********',
    'USER-AGENT': 'Jersey/3.1.5 (HttpUrlConnection 17.0.16)',
    'HOST': 'api:9003',
    'CONNECTION': 'keep-alive'
    },
```

```
    'id': '/448db79d-ee36-458d-99f1-bf7be8d1527c/5421edf2-157e-4128-
bcd9-696653d9890a',
    'parameters': {
     'body': None,
     'query': {
      'compartmentId':
 'ocid1.compartment.1749XC302V.broom6.89n1pa1uw1iy6zigy4v5yra56zonmlbneiox8usdngtm
      'instanceId':
 'ocid1.instance.1749XC302V.broom6.cp5wa6r2xupr3gj3hlcbhn3u461umovhpucfjd25t97f8sx
      'limit': '1000',
      'lifecycleState': 'ATTACHED',
      'sortBy': 'DISPLAYNAME',
      'sortOrder': 'ASC'
     }
    },
    'path': '/20160918/volumeAttachments'
   },
   'resourceId': None,
   'resourceName': None,
   'response': {
    'headers': {
     'opc-request-id': '/448db79d-ee36-458d-99f1-bf7be8d1527c/
c5ef9bee-8863-46f9-9348-3faf1f1eb498',
     'HOST': 'api:9003',
     'principal': {
      'id':
 'ocid1.user.1749XC302V.broom6.ulmhteeqbqwi68rmtc5oj7cpgzkmjxvhzbu8exn08uihl9ng0kb
      'tenancyId':
 'ocid1.tenancy.1749XC302V.broom6.bp4t0827a1zd5uw34s6vc9xticohkzek6fu7fd085dfe4v9q
      'type': 'USER',
      'name': 'admin',
      'token': '********'
     },
     'userId':
 'ocid1.user.1749XC302V.broom6.ulmhteeqbqwi68rmtc5oj7cpgzkmjxvhzbu8exn08uihl9ng0kb
     'tenancyId':
 'ocid1.tenancy.1749XC302V.broom6.bp4t0827a1zd5uw34s6vc9xticohkzek6fu7fd085dfe4v9q
     'tagTenancyId':
 'ocid1.tenancy.1749XC302V.broom6.bp4t0827a1zd5uw34s6vc9xticohkzek6fu7fd085dfe4v9q
    },
    'message': 'ListVolumeAttachments succeeded.',
    'status': '200 OK'
   },
   'stateChange': {
    'current': None,
    'previous': None
   }
  }"
}
```

API - /var/log/api-server/api-server.log

This log file tracks the processing of each API request, including authentication, authorization, publishing to the relevant service, and the reply from the service.

```
{"timestamp": "2025-09-25T20:49:13.460+0000", "rid":
 "/0b75ee6a-9f34-4cab-90c2-09739cb1e2a1/9a414afe-4471-47c7-a7cb-
f46ed84b650f", "podname": "api-6b7f777d8d-8xd64", "process":
 30, "ocid": null, "levelname": "INFO", "module": "log",
 "lineno": 62, "pathname": "/usr/lib/python3.11/site-packages/
pcaapi/log.py", "message": "Request info is: {'request env':
 {'REQUEST_METHOD': 'GET', 'REQUEST_URI': '/metrics', 'PATH_INFO':
 '/metrics', 'QUERY_STRING': '', 'SERVER_PROTOCOL': 'HTTP/1.1',
```

```
 'SCRIPT_NAME': '', 'SERVER_NAME': 'api-6b7f777d8d-8xd64',
 'SERVER_PORT': '9003', 'UWSGI_ROUTER': 'http', 'HTTPS': 'on',
 'REMOTE_ADDR': '253.255.129.0', 'REMOTE_PORT': '48852', 'HTTP_HOST':
 '253.255.128.87:9003', 'HTTP_USER_AGENT': 'Prometheus/2.32.1',
 'HTTP_ACCEPT': 'application/openmetrics-text; version=0.0.1,text/
plain;version=0.0.4;q=0.5,*/*;q=0.1', 'HTTP_ACCEPT_ENCODING': 'gzip',
 'HTTP_X_PROMETHEUS_SCRAPE_TIMEOUT_SECONDS': '10', 'wsgi.input':
 <uwsgi._Input object at 0x7f8928e5f430>, 'wsgi.file_wrapper': <built-
in function uwsgi_sendfile>, 'wsgi.version': (1, 0), 'wsgi.errors':
 <_io.TextIOWrapper name=2 mode='w' encoding='ANSI_X3.4-1968'>,
 'wsgi.run_once': False, 'wsgi.multithread': True,
 'wsgi.multiprocess': True, 'wsgi.url_scheme': 'https',
 'uwsgi.version': b'2.0.25.1', 'uwsgi.core': 2, 'uwsgi.node':
 b'api-6b7f777d8d-8xd64'}, 'request_context': Context({'audit_event':
 <pcaapi.apiimpl.util.audit.AuditEvent object at 0x7f892946b710>, 'opc-
request-id': '/0b75ee6a-9f34-4cab-90c2-09739cb1e2a1/9a414afe-4471-47c7-
a7cb-f46ed84b650f'}), 'scheme': 'https', 'forwarded_scheme': 'https',
 'host': '253.255.128.87', 'forwarded_host': '253.255.128.87:9003',
 'port': 9003, 'root_path': '', 'URI': 'https://253.255.128.87:9003/
metrics', 'forwarded_URI': 'https://253.255.128.87:9003/metrics',
 'prefix': 'https://253.255.128.87:9003', 'forwarded_prefix':
 'https://253.255.128.87:9003', 'path': '/metrics', 'query_string':
 '', 'uri_template': None, 'remote_address': '253.255.129.0',
 'access_route': ['253.255.129.0'], 'forwarded': None, 'date': None,
 'user_agent_header': 'Prometheus/2.32.1', 'referer_header': None,
 'accept_header': 'application/openmetrics-text; version=0.0.1,text/
plain;version=0.0.4;q=0.5,*/*;q=0.1', 'client_accepts_json_header':
 True, 'if_match_header': None, 'if_none_match_header': None,
 'if_modified_since_header': None, 'if_unmodified_since_header': None,
 'if_range_header': None, 'headers': {'HOST': '253.255.128.87:9003',
 'USER-AGENT': 'Prometheus/2.32.1', 'ACCEPT': 'application/openmetrics-
text; version=0.0.1,text/plain;version=0.0.4;q=0.5,*/*;q=0.1', 'ACCEPT-
ENCODING': 'gzip', 'X-PROMETHEUS-SCRAPE-TIMEOUT-SECONDS': '10'},
 'params': {}, 'options': <falcon.request.RequestOptions object at
 0x7f893b5e0bd0>}"}
{"timestamp": "2025-09-25T20:49:13.635+0000", "rid":
 "/0b75ee6a-9f34-4cab-90c2-09739cb1e2a1/9a414afe-4471-47c7-a7cb-
f46ed84b650f", "podname": "api-6b7f777d8d-8xd64", "process": 30,
 "ocid": null, "levelname": "INFO", "module": "UWSGI", "lineno": 0,
 "pathname": null, "message": "GET /metrics => generated 1956317 bytes
 in 176 msecs (HTTP/1.1 200) 4 headers in 216 bytes (1 switches on core
 2)"}
```

IAM - /var/log/iam/iam.log

This log file collects errors and general information related to authentication and authorization requests. It tracks the lifecycle of identity resources. On systems linked to OCI, this file also tracks the synchronization of identity resources.

```
{"timestamp": "2025-09-25T20:53:17.883+0000", "rid":
 "77a7adafee88531f6f057d4ea1d95199/7368f440-c339-44df-
a59b-559406b2cd33/dfe13bdb-6746-4c82-8d64-a96176805245",
 "podname": "iam-5f5cffbbfd-fpxgw", "process": 1, "ocid":
 null, "levelname": "INFO", "module": "log", "lineno":
 125, "pathname": "/usr/lib/python3.11/site-packages/
pcacommon/log.py", "src_module": "pcaiam.authorization",
 "src_lineno": 371, "src_pathname": "/usr/lib/python3.11/site-
packages/pcaiam/authorization.py", "message": "AuthZ failed.
 Tenant=ocid1.tenancy.1742XC3020.broom7.v1fr92mhepaivxtpwvjccgl8xu8wnls6wwfdoy5o8e
 Principal=ocid1.instance.1742XC3020.broom7.f51ayedmi9r2xlmqcrdhedqxyt3f0o6w6n50u0
 Compartment=ocid1.tenancy.1742XC3020.broom7.v1fr92mhepaivxtpwvjccgl8xu8wnls6wwfdo
 PrincipalType=INSTANCE, Operation=GetInstance, Comp
```

```
 Hierarchy=['ocid1.tenancy.1742XC3020.broom7.v1fr92mhepaivxtpwvjccgl8xu8wnls6wwfdo
 Required Permissions=['INSTANCE_READ'] ", "message_data": null}
```

ADMIN - /var/log/audit.log.0

This log file keeps a record of all CLI and REST calls to the Admin Service, including all parameters.

```
{
  "asctime": "2025-09-25 20:53:08",
  "podname": "admin-558fd59789-4ms7l",
  "process": "9",
  "component": "API",
  "levelname": "INFO",
  "summary": "opc-request-id8b636b9b-72e9-4208-9fdc-56d6919d2952,
 2785 > POST https://admin.broom7.us.oracle.com/
admin/wsapi/rest/3.0.1/PeerConnection/trustExchange?
rackSerialNumber=1749XC302V&domainName=us.oracle.com&systemName=broom6&adminPublic
BEGIN+PUBLIC+KEY-----
%0AMIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAriyAQb%2BcTx2g0AK7qt6J
%0Armp%2Bw
%2BWwF6CnWsZAwzfJaGr4XEs30%2F4HrAZzSHKt62UQU3K32yAUR3ThZDA0%2BXo9%0Ajbqxi5lWR7p4Nq
%2FL82vrP1mJ2vfcD3DIKIzh3k5knoOzzT%2B%0A
%2FxX08zqakHOGX0QNQs8GoRVV5VTgQKNtM%2FAdk5TruNfnpKq98XQII5PT%2BChHeX2n
%0AdJETiIatBans4xsP8nPhdaquUbD%2B3e1rYCcFESIM5DB7oL6wyIpI2qXQeKORG
%2F49%0AxNAVg1lJ8pQfMaIkLqh1hehIGJUL2m8WpCWUMDrO56%2Fxg81wuFwbi4LnOmNS
%2FSeQ
%0ARuGBMGF2WNVzuNhKbfpydH7aqekdfop44wtEFVSOAuUpokkXljV1m2Svax1t746H
%0ArtNtJ7ioSaS2kcoLx14s2%2FhbyPFI0CXGX28QzH3ycu1cXQzyPC4Tl57KMDJe2urP
%0AEsNLdPbeqprZE4JHzVLqL7ZzcXdsVy3YeV%2FaD1kQ0qAj%2BB3T
%2BaTyCTu3J6d9FYPr%0AFgKo1bVWp8qij9G0SchllmZqXBSBBKJAA%2Bj%2BQQznaeE5VB
%2BaBHB0aYwBH8myDR1B3%0Ao86d0tbSsgtXtD3Vwzb8pLTawLbkgHV72L4BwAwLAVGEXZmQwtPj9zufnd
%0ALATjknS%2FSO1UH5DpGe2ur6ECAwEAAQ%3D%3D%0A-----END+PUBLIC+KEY-----
%0A&peerBootstrap=False",
  "message": "2785 > accept: */*2785 > accept-encoding: gzip, deflate
 2785 > authorization: [redacted] 2785 > connection: keep-alive 2785
 > content-length: 3859 2785 > content-type: application/json 2785 >
 date: Th-",
  "user": "1749XC302V",
  "success": "1",
  "log_type": "audit",
  "principalName": "1749XC302V",
  "sourceIP": "253.255.129.0"
}
```

ADMIN - /var/log/admin.log.0

This log file records detailed information about invoked commands, background tasks, system status, execution errors, and communications between the Admin Service and other services.

```
{"timestamp": "2025-09-25T20:58:18.939+0000", "podname":
 "admin-558fd59789-4ms7l", "levelname": "INFO",
 "classname": "", "methodname": "", "lineno": "", "thread":
 "WORKITEM_UPDATE_TASK(27)", "message": "UPDATE TASK SET
 RUNSTATUS='RUNNING', LASTSTARTTIME=ROUND(UNIX_TIMESTAMP(CURTIME(4))
 * 1000) WHERE NAME='WORKITEM_UPDATE_TASK' AND
 ((RUNSTATUS='STOPPED' AND COALESCE(LASTSTARTTIME,
 0)<ROUND(UNIX_TIMESTAMP(CURTIME(4))*1000-150000)) ||
 (COALESCE(LASTSTARTTIME,
 0)<ROUND(UNIX_TIMESTAMP(CURTIME(4))*1000-450000)));"}
```

Oracle Auto Service Request (ASR)

Private Cloud Appliance is qualified for Oracle Auto Service Request. ASR is integrated with My Oracle Support. When specific hardware failures occur, ASR automatically opens a service request and sends diagnostic information. The appliance administrator receives notification that a service request is open.

Using ASR is optional: the service must be registered and enabled for your appliance. To enable this feature, the Private Cloud Appliance must be configured to send hardware fault telemetry to Oracle directly at `https://transport.oracle.com/`, to a proxy host, or to a different endpoint. For example, you can use a different endpoint if you have the ASR Manager software installed in your data center as an aggregation point for multiple systems.

When a hardware problem is detected, ASR submits a service request to Oracle Support Services. In many cases, Oracle Support Services can begin work on resolving the issue before the administrator is even aware the problem exists.

ASR detects faults in the most common hardware components, such as disks, fans, and power supplies, and automatically opens a service request when a fault occurs. ASR does not detect all possible hardware faults, and it is not a replacement for other monitoring mechanisms, such as SMTP alerts, within the customer data center. ASR is a complementary mechanism that expedites and simplifies the delivery of replacement hardware. ASR should not be used for downtime events in high-priority systems. For high-priority events, contact Oracle Support Services directly.

An email message is sent to both the My Oracle Support email account and the technical contact for Private Cloud Appliance to notify them of the creation of the service request. A service request might not be filed automatically in some cases, for example if a loss of connectivity to ASR occurs. Administrators should monitor their systems for faults and call Oracle Support Services if they do not receive notice that a service request has been filed automatically.

For more information about ASR, consult the following resources:

- Oracle Auto Service Request web page
- ASR release notes on My Oracle Support: Doc ID 2152198.1.
- ASR quick start guide on My Oracle Support: Doc ID 2852505.1
- Oracle Auto Service Request user documentation

Registering Private Cloud Appliance for Oracle Auto Service Request

To register Private Cloud Appliance as an ASR client, the system must be configured to send hardware fault telemetry to Oracle.

ASR Prerequisites

Before you register for the ASR service, ensure the following prerequisites are satisfied.

1. Ensure that you have a valid My Oracle Support account.

   If necessary, create an account at https://support.oracle.com/portal/.
2. Ensure that the following are set up correctly in My Oracle Support:
   - Technical contact person at the customer site who is responsible for ASR.
   - Valid shipping address at the customer site where the system is located, so that parts are delivered to the site where they must be installed.
3. Verify connectivity to the Internet using HTTPS.

   For example, try `curl` to test whether you can access `https://support.oracle.com/portal/`.

Registering for ASR

When you register your Private Cloud Appliance for ASR, the ASR service is automatically enabled.

Hardware fault data is transferred in one of the following ways:

- Directly at `https://transport.oracle.com/`
- To a proxy host

- To a different endpoint

An example of when you would use a different endpoint is if you have the ASR Manager software installed in your data center as an aggregation point for multiple systems.

Using the Service Web UI

1. Open the navigation menu and click ASR Phone Home.
2. Click the Register button.
3. Fill in the user name and password, then complete the fields for the Phone Home configuration that you choose.
   - **Username:** Required. Enter your Oracle Single Sign On (SSO) credentials, which can be obtained from My Oracle Support.
   - **Password:** Required. Enter the password for your SSO account.
   - **Proxy Username:** To use a proxy host, enter a user name to access that host.
   - **Proxy Password:** To use a proxy host, enter the password to access that host.
   - **Proxy Host:** To use a proxy host, enter the name of that host.
   - **Proxy Port:** To use a proxy host, enter the port used to access the host.
   - **Endpoint:** If you use an aggregation point, or other endpoint for ASR data consolidation, enter that endpoint in this format: `http://`***`host`***`[:`***`port`***`]/asr`

Using the Service CLI

1. Using SSH, sign in to the management node VIP as `admin`.

   ```
   # ssh -l admin 100.96.2.32 -p 30006
   ```

2. Use the `asrClientRegister` custom command to register the appliance. Select the syntax for the selected endpoint.
   - Direct to `https://transport.oracle.com/`:

     ```
     PCA-ADMIN> asrClientRegister asrUsername=asr.user@example.com \
     password=********  confirmPassword=******** \
     endpoint=https://transport.oracle.com/
     ```

   - To a proxy host:

     ```
     PCA-ADMIN> asrClientRegister asrUsername=asr.user@example.com \
     password=********  confirmPassword=******** \
     proxyHost=zeb proxyPort=80 \
     proxyUsername=support \
     proxyPassword=******** proxyConfirmPassword=********
     ```

   - To a different endpoint:

     ```
     PCA-ADMIN> asrClientRegister asrUsername=asr.user@example.com \
     password=********  confirmPassword=******** \
     endpoint=http://asrmanager.example.com:8080/asr
     ```

3. Confirm the configuration.

   ```
   PCA-ADMIN> show asrPhonehome
   Data:
     Is Registered = true
     Overall Enable Disable = true
     Username = asr.user@example.com
     Endpoint = https://transport.oracle.com/
   ```

Testing the ASR Configuration

When configured, test your ASR configuration to ensure end-to-end communication is working properly.

Using the Service Web UI

1. Open the navigation menu and click ASR Phone Home.
2. Select Test Registration in the Controls menu.
3. Click Test Registration. A dialog confirms whether the test is successful.
4. If the test is not successful, verify your ASR configuration information and repeat the test.

Using the Service CLI

1. Using SSH, log into the management node VIP as `admin`.

```
# ssh -l admin 100.96.2.32 -p 30006
```

2. Use the `asrClientsendTestMsg` custom command to test the ASR configuration.

```
PCA-ADMIN> asrClientsendTestMsg
Status: Success
```

Updating Private Cloud Appliance Connection to Oracle Auto Service Request

A Private Cloud Appliance administrator can temporarily disable and reenable the ASR connection, or unregister the system from ASR.

Disabling the ASR Connection

You can disable ASR on an appliance to temporarily prevent fault messages from being sent and service requests created. For example, during system maintenance, components might be down but not failed or faulted. To restart the ASR service, see Enabling the ASR Connection on page 274.

Using the Service Web UI

1. Open the navigation menu and click ASR Phone Home.
2. Click the Disable button. Confirm the operation when prompted.

Using the Service CLI

1. Using SSH, sign in to the management node VIP as `admin`.

```
# ssh -l admin 100.96.2.32 -p 30006
```

2. Use the `asrClientDisable` custom command to halt the ASR service.

```
PCA-ADMIN> asrClientDisable
Status: Success
```

Enabling the ASR Connection

Follow these instructions to restart the ASR service if the ASR service is disabled.

Using the Service Web UI

1. Open the navigation menu and click ASR Phone Home.
2. Click the Enable button. Confirm the operation when prompted.

Using the Service CLI

1. Using SSH, sign in to the management node VIP as `admin`.

```
# ssh -l admin 100.96.2.32 -p 30006
```

2. Use the `asrClientEnable` custom command to start the ASR service.

```
PCA-ADMIN> asrClientEnable
Status: Success
```

Unregistering from ASR

When you unregister your system from ASR, the ASR service is automatically disabled. You do not need to perform a separate operation.

Using the Service Web UI

1. Open the navigation menu and click ASR Phone Home.
2. Click the Unregister button. Confirm the operation when prompted.

Using the Service CLI

1. Using SSH, sign in to the management node VIP as `admin`.

```
# ssh -l admin 100.96.2.32 -p 30006
```

2. Use the `asrClientUnregister` custom command to unregister the system.

```
PCA-ADMIN> asrClientUnregister
Status: Success
```

Admin Service Health Data

The Admin service in Private Cloud Appliance generates health metrics, and raises faults when certain conditions occur.

This health information is not for hardware faults but is information about resource utilization (CPU, memory, and storage), hardware run state, and health checker notifications. The hardware faults listed at the bottom of the Admin service faults summary table are reported through ASR.

Admin Service Fault Detection Configuration Summary

Fault Type	Detection Frequency	Detection Delay	Data Source	Method of Detection
CPU and memory utilization	60 sec	< 20 sec	Admin calls ComputeNode service	Faults are raised by fault task based on the compute node object attributes stored in the database.
Storage utilization	120 sec	< 20 sec	Admin calls Prometheus service	Faults are raised by fault task based on Prometheus ZFS pool usage and status data stored in the database.
Hardware run state	150 sec	< 20 sec	Admin calls Hardware `list` REST API	Faults are raised by fault task based on hardware component node/ILOM run states stored in the database.

Fault Type	Detection Frequency	Detection Delay	Data Source	Method of Detection
Health checker notification	Defined by the ZFS/Network health checker notification frequency	0 sec	Various HealthChecker services send notifications	Faults are created based on the RabbitMQ notification fault results.
Platform ILOM faults	150 sec	0 sec	Admin calls Hardware `getMgmt` and `getCompute` ILOM Health REST APIs	Faults are created based on the L1 API results for ILOM object data. For more information about ASR actionable events, see this support note: Doc ID 2833567.2.
Hardware status	On initialization, and when the `syncHardwareData` command runs	< 20 sec	Admin calls Hardware `list` REST API	Faults are raised by fault task based on the PcaSystem object attribute. For more information about ASR actionable events, see this support note: Doc ID 2833567.2.

Viewing Admin Service Faults

Using the Service Web UI

1. Click the Active Faults link at the top of the Service Web UI Home page, or click Faults on the Navigation menu.

   The Faults page is displayed.
2. At the top of the Faults page, you can toggle whether to list all faults or only active faults.
3. For more information about a fault, click the name of the fault, or click View Details on the Actions menu.

   The details page shows the description, cause, and recommended action to take.

Using the Service CLI

1. To view the list of Admin service faults, use the `list fault` command.

   Both active and cleared faults are listed.

```
PCA-ADMIN> list fault
Data:
  id                                      name
                        status    severity
  --                                      ----
                        ------    --------
  33c61b8a-dcc7-4b8f-bc0f-56915ecc62f5
 RackUnitIlomRunStateFaultStatusFault(pcacn005)     Cleared
 Critical
  f7d22180-aeae-4159-b5c8-5e55a7906a78
 RackUnitIlomRunStateFaultStatusFault(pcacn004)     Cleared
 Critical
  a4fef907-8e54-4750-9fac-6829fbade90d
 ComputeNodeCpuFaultStatusFault(pcacn006)           Cleared   Minor
```

```
 f8d93384-da30-43cd-9396-6e6671d240e2
RackUnitIlomRunStateFaultStatusFault(pcacn010)     Cleared
Critical
 8e61bb81-7a02-4c26-8ef4-c13b198f64da
ComputeNodeCpuFaultStatusFault(pcacn007)           Cleared
Warning
 3216b6f9-326b-4992-99a3-ab23cb18243b   AK-8003-F9--PCIe 3
                        Active    Minor
 ef3fb25b-0573-4524-8d1c-fb704c814446   AK-8003-HF--vnic1
                        Active    Major
 f830cd46-21ff-4d74-ba81-c82fd6f52c67
ComputeNodeCpuFaultStatusFault(pcacn005)           Cleared   Minor
 d2e71da0-ba63-4983-97da-24033d5c6447
ZfsPoolUsageFaultStatusFault(PCA_POOL)             Cleared   Major
 eecd5ef2-4a71-4137-be96-54c028212d2f
ComputeNodeMemoryFaultStatusFault(pcacn004)        Cleared   Minor
 cf68d2ee-e483-e573-b46e-c31bcbc8e968   ISTOR-8000-1S--ORACLE
SERVER E5-2L                   Cleared   Major
 0686c11d-b96b-e5aa-dfbe-a20154da4794   SPAMD-8002-FJ--ORACLE
SERVER E5-2L                   Cleared   Major
 b488a45a-80df-46e3-b0b5-a35527eb9c0e   AK-8003-F9--PCIe 10
                        Active    Minor
 ac48f88d-e181-4b03-b620-6bfbf4ad95ef
RackUnitIlomRunStateFaultStatusFault(pcacn007)     Cleared
Critical
 b4c66a7c-def3-42c2-8842-d4763afc5184
RackUnitIlomRunStateFaultStatusFault(pcacn006)     Cleared
Critical
 9fc2e45a-1cff-4f95-828d-58742c8ce12f
ComputeNodeMemoryFaultStatusFault(pcacn002)        Active    Minor
 c0124122-a91c-4110-89cc-deebe54de7ba
ComputeNodeMemoryFaultStatusFault(pcacn006)        Cleared
Critical
 ca26ed46-4d1c-4ade-9e74-af27d94cf8f4   AK-8003-HF--vnic2
                        Active    Major
 58e9ab5d-d4e7-4d94-9ca6-e85a1c88b3b8
RackUnitRunStateFaultStatusFault(sn022147XLF014)   Cleared
Critical
 474c269f-4018-45d7-97d5-da17c9c845f4
RackUnitIlomRunStateFaultStatusFault(pcacn001)     Cleared
Critical
 2b5ece1c-50fc-436a-81b3-da0c5b418fe3
RackUnitIlomRunStateFaultStatusFault(pcacn003)     Cleared
Critical
 1c164eb9-9a76-4592-8ab6-150edb8f7a75
ComputeNodeCpuFaultStatusFault(pcacn001)           Cleared
Warning
 55ed1494-6aac-4248-91cb-9ac8295d668c   AK-8003-HF--PCIe 6
                        Active    Major
 afbcc080-0b93-434b-8ead-fa673f302170   AK-8003-F9--PCIe 6
                        Active    Minor
 8b36c2db-a3b4-41c8-b416-8e733ace3aeb
PcaSystemReSyncHwStatusStatusFault(null)           Cleared
Warning
 28c5ba93-6b4e-42f3-ad61-90734b46bf30   SPENV-8000-RU--ORACLE
SERVER E5-2L                   Cleared   Critical
 3d932188-0120-489f-a512-1a244ec01e49
RackUnitIlomRunStateFaultStatusFault(pcacn009)     Cleared
Critical
 21e6faa9-68e1-47ae-a298-e2cb14d2a406
ComputeNodeMemoryFaultStatusFault(pcacn007)        Cleared   Minor
 db023304-fb7a-613b-ad9b-e277b7ce5675   SPENV-8000-A7--ORACLE
SERVER E5-2L                   Cleared   Major
```

```
  63839bf5-335b-48ff-86a0-9e981e3e9902
 RackUnitRunStateFaultStatusFault(sn012147XLF014)   Cleared
 Critical
  2e851c6e-aa29-4a25-846a-29b08967dd95
 RackUnitValidationStateStatusFault(pcacn008)       Cleared   Major
  76805c56-fcf6-48a2-b4fd-ffa77570e83c
 ComputeNodeCpuFaultStatusFault(pcacn002)           Active    Minor
  9be74faf-df4d-ea20-cfc1-92b2a6a01b06   SPENV-8000-A7--ORACLE
 SERVER E5-2L                 Cleared   Major
  1624064f-d380-4ffc-9000-d293c185d7ac
 ComputeNodeCpuFaultStatusFault(pcacn003)           Cleared
 Warning
  7ca3f7af-f0bd-45d9-bad7-15794d49e7c6
 RackUnitIlomRunStateFaultStatusFault(pcacn008)     Cleared
 Critical
  3e7a3503-7a71-4ef1-a3ad-fba2162571ab
 ComputeNodeCpuFaultStatusFault(pcacn004)           Cleared
 Warning
  0922cd8e-297e-4356-b736-b09ac382b28b   AK-8003-F9--PCIe 10
                       Active    Minor
  ab44ad2c-1105-417d-aa47-e8cb477ef0ec   AK-8003-F9--PCIe 3
                       Active    Minor
```

2. To view the details of a specific fault, including description, cause, and recommended action to take, use the `show fault` command with the specific fault ID.

```
PCA-ADMIN> show fault id=ab44ad2c-1105-417d-aa47-e8cb477ef0ec
Data:
  Id = ab44ad2c-1105-417d-aa47-e8cb477ef0ec
  Type = Fault
  Category = Internal
  Severity = Minor
  Status = Active
  Last Update Time = 2023-03-06 20:04:11,668 UTC
  Message Id = AK-8003-F9
  Time Reported = Mon Mar 06 2023 16:50:24 GMT+0000 (UTC)
  Action = Check the networking cable, switch port, and switch
 configuration. Contact your vendor for support
           if the network port remains inexplicably down. Please
 refer to the associated reference document at
           http://support.oracle.com/msg/AK-8003-F9 for the latest
 service procedures and policies regarding
           this diagnosis.
  Health Exporter = zfssa-analytics-exportersn022147XLF014
  uuid = ab44ad2c-1105-417d-aa47-e8cb477ef0ec
  Diagnosing Source = zfssa_analytics_exporter
  FaultHistoryLogIds 1 = id:fdfaa42f-de8d-4622-a9df-ea229b7bad6f
 type:FaultHistoryLog  name:
  BaseManagedObjectId = id:2147XLF015/PCIe 3/465774J-2121701684
 type:HardwareComponent  name:
  Description = Network connectivity via port mlxne4 has been lost.
  Name = AK-8003-F9--PCIe 3
  Work State = Normal
```

Viewing CPU and Memory Utilization Faults

The Admin service raises faults for the percentage of memory used and percentage of CPU used for a `ComputeNode` object. More severe faults are raised as more memory and CPU are used. When the percentage used drops below a certain percentage, any faults are cleared.

These are utilization faults (CPU and memory usage), not hardware faults. Problems with CPU and memory hardware are reported through ASR.

CPU Usage

The following table shows the default percentage of compute node CPU usage that raises different severities of faults.

CPU Percentage	Fault Severity	Fault State
< .75	Not applicable	Cleared
>= .75	Warning	Active
>= .80	Minor	Active
>= .90	Major	Active
>= .95	Critical	Active

Memory Usage

The following table shows the default percentage of compute node memory usage that raises different severities of faults.

Memory Percentage	Fault Severity	Fault State
< .75	Not applicable	Cleared
>= .75	Warning	Active
>= .80	Minor	Active
>= .90	Major	Active
>= .95	Critical	Active

Viewing Compute Node Faults in the Service CLI

To view the compute node CPU and memory usage default fault trigger settings, use the `cnUpdateManager` command:

```
PCA-ADMIN> show cnUpdateManager
Data:
  Id = caaaaaa1-a076-4e48-94b5-7bdcd4e0c42c
  Type = CnUpdateManager
  LastRunTime = 2023-03-06 23:41:33,676 UTC
  Poll Interval (sec) = 60
  The minimum CPU usage percentage to trigger a critical fault = 0.95
  The minimum CPU usage percentage to trigger a major fault = 0.9
  The minimum CPU usage percentage to trigger a minor fault = 0.8
  The minimum CPU usage percentage to trigger a warning = 0.75
  The minimum memory usage percentage to trigger a critical fault =
 0.95
  The minimum memory usage percentage to trigger a major fault = 0.9
  The minimum memory usage percentage to trigger a minor fault = 0.8
  The minimum memory usage percentage to trigger a warning = 0.75
```

To view the list of all faults and the details of a specific fault, see Viewing Admin Service Faults on page 276. The following example shows a specific compute node fault. Current usage is not shown except that

it is at least the minor fault threshold but less than the major fault threshold. To see current usage, use the Service Web UI.

```
PCA-ADMIN> show fault id=76805c56-fcf6-48a2-b4fd-ffa77570e83c
Data:
  Id = 76805c56-fcf6-48a2-b4fd-ffa77570e83c
  Type = Fault
  Category = Status
  Severity = Minor
  Status = Active
  Associated Attribute = cpuFault
  Last Update Time = 2023-03-04 01:06:25,666 UTC
  Cause = ComputeNode pcacn002 attribute cpuFault = MINOR.
  FaultHistoryLogIds 1 = id:79b44c26-cb4e-4bec-a58c-6efc7fc63fed
 type:FaultHistoryLog  name:
  FaultHistoryLogIds 2 = id:fc90a99a-031b-457f-b585-5c905e61362e
 type:FaultHistoryLog  name:
  FaultHistoryLogIds 3 = id:48068f78-1328-447d-9506-efb6f22d154d
 type:FaultHistoryLog  name:
  FaultHistoryLogIds 4 = id:d97c5819-923c-480d-8f61-2341c8403182
 type:FaultHistoryLog  name:
  FaultHistoryLogIds 5 = id:18cdd005-53c0-488c-a2df-28f2da3b1092
 type:FaultHistoryLog  name:
  FaultHistoryLogIds 6 = id:bfe1ffcd-5899-4400-914c-b467d8671e0c
 type:FaultHistoryLog  name:
  FaultHistoryLogIds 7 = id:459fa55b-8654-4c07-8ae7-6d0ef011e3b1
 type:FaultHistoryLog  name:
  FaultHistoryLogIds 8 = id:b9c8a909-f8ea-4de6-9bfe-2516e7addf73
 type:FaultHistoryLog  name:
  FaultHistoryLogIds 9 = id:6ab5d1ca-3659-49a7-8e68-946bbbeccc9f
 type:FaultHistoryLog  name:
  FaultHistoryLogIds 10 = id:d04d06a1-1e2c-404c-ac67-680e0deb34c5
 type:FaultHistoryLog  name:
  FaultHistoryLogIds 11 = id:22dd163e-528f-4346-b177-d62c7ceb9885
 type:FaultHistoryLog  name:
  FaultHistoryLogIds 12 = id:cdb2dbf5-6999-43c2-bb5f-17192bfad3e2
 type:FaultHistoryLog  name:
  FaultHistoryLogIds 13 = id:aa7b2e43-ab0b-4d78-bfe7-d4b0dd0fec4a
 type:FaultHistoryLog  name:
  BaseManagedObjectId = id:0dd96e90-de00-4fa0-82e3-16937e4601f8
 type:ComputeNode  name:
  Description = ComputeNode pcacn002 attribute cpuFault = MINOR.
  Name = ComputeNodeCpuFaultStatusFault(pcacn002)
  Work State = Normal
```

Storage Utilization Faults

The following table describes the two kinds of ZFS Storage Appliance faults raised in the Admin service. These are utilization faults (ZFS pool usage), not hardware faults. Problems with ZFS hardware are reported through ASR.

Prometheus matrix data collected for the ZFS Storage Appliance, is used to report pool usage. Total pool size per pool (`zfssa_pool_total`) and free space per pool (`zfssa_pool_free`) are used to calculate pool usage percentage. The `zfssa_pool_status` metric reports the health of a pool.

Metric Name	Metric Value Description	Fault Condition
`zfssa_pool_total` `zfssa_pool_free`	Pool usage percentage is calculated using the following formula for each pool: `(zfssa_pool_total - zfssa_pool_free) / zfssa_pool_total`	If the pool usage percentage is above a pre-configured value, a major fault is raised. The default value is 80 percent.
`zfssa_pool_status`	The `zfssa_pool_status` metric can have the following values: • `0` - exported • `1` - degraded • `2` - online • `-1` - offline • `-2` - faulted • `-3` - unavailable • `-4` - removed	A major fault is raised for any `pool`/`zfssa_node` combination that has any pool status value other than 0 or 2.

Health Checker Notification Faults

Health Checker faults are raised from notifications from the ZFS Storage Appliance and Network Health Checker components. The Admin service raises a fault for every notification it receives.

Following are example attributes of the `faultedComponents` object in the Network Health Checker component fault data:

```
"class": "cisco.fan.fail",
"severity": "Major",
"description": "Fan module has failed and needs to be replaced. This can
 lead to overheating and temperature alarms.",
[...]
"class": "cisco.power.fail",
"severity": "Major",
"description": "Power Supply has failed or has been shutdown",
```

Following are example attributes of the `faultedComponents` object in the ZFS Storage Appliance Health Checker component fault data:

```
"severity":"Major",
"type":"Fault",
"description":"An internal power supply failure has been detected.",
```

Detailed information is provided about the part that has failed.

An `action` attribute contains a brief description of what to do to fix the problem and might include a link to the appropriate support document.

Manually Clearing Faults

Faults can be manually cleared using the Service CLI. You cannot clear faults using the Service Web UI.

1. Using SSH, log into the management node VIP as `admin`.

```
# ssh -l admin 100.96.2.32 -p 30006
```

2. Use the `list fault` command to find the list of fault identifications.

```
PCA-ADMIN> list fault
Data:
id                                Name               Status   Severity
--                                ----               ------   --------
71671228...56a6a58947c6a6789      pcamn02-example    Active   Critical
524cb805...acc3458bb79t04295      RackUnit-example   Active   Major
```

3. Use the `clearFault` command with the fault identifier to clear the fault.

> **Note:**
>
> You can verify the clear fault result by running the `list fault` command again.

```
PCA-ADMIN> clearFault id=[524cb805...acc3458bb79t04295]
Status: Success

PCA-ADMIN> list fault
Data:
id                                Name               Status   Severity
--                                ----               ------   --------
71671228...56a6a58947c6a6789      pcamn02-example    Active   Critical
```

Support Bundles

Support bundles are files of diagnostic data collected from Private Cloud Appliance, which are used to evaluate and fix problems.

Support bundles can be uploaded to Oracle Support automatically or manually. They are uploaded securely and contain the minimum required data: system identity (not IP addresses), problem symptoms, and diagnostic information such as logs and status. Support bundles can be created and not uploaded. You might want to create a bundle for your own use. Creating a support bundle is a convenient way to collect related data.

Support bundles are created and uploaded in the following ways:

- Oracle Auto Service Request

  ASR automatically creates a service request and support bundle when certain hardware faults occur. The service request and support bundle are automatically sent to Oracle Support, and the Private Cloud Appliance administrator is notified. See Oracle Auto Service Request (ASR) on page 272.
- Service CLI `asrInitiateBundle` command

  The `asrInitiateBundle` command is a Service CLI command that creates a support bundle, attaches the support bundle to an existing service request, and uploads the support bundle to Oracle Support. See Collecting Support Bundles from the Service CLI on page 282.
- Oracle Linux tool `support-bundles` on management node

  The `support-bundles` command is a management node command that creates a support bundle of a specified type. Oracle Support might ask you to run this command to collect more data related to a service request, or you might want to collect this data for your own use. See #unique_169.
- Manual upload to Oracle Support

  Several methods are available for uploading support bundles or other data to Oracle Support. See #unique_170.

Collecting Support Bundles from the Service CLI

The `asrInitiateBundle` command is a Private Cloud Appliance Service CLI command that creates a support bundle, attaches the support bundle to an existing service request, and uploads it.

The `asrInitiateBundle` command takes three parameters, all required:

```
PCA-ADMIN> asrInitiateBundle mode=triage sr=SR_number bundleType=auto
```

A `triage` support bundle is collected and automatically attached to service request ***`SR_number`***. For more information about the `triage` support bundle, see #unique_171/unique_171_Connect_42_supbundles-triage-mode.

If the ASR service is enabled, `bundleType=auto` uploads the bundle to Oracle Support using the Phone Home service. For information about the Phone Home service, see Registering Private Cloud Appliance for Oracle Auto Service Request on page 272.

The bundle is saved on the management node for two days after successful upload. See #unique_169.

If you specify `mode=native` and do not specify any value for `nativeType`, then a `ZFS_BUNDLE` is uploaded.

Using the support-bundles Tool

The `support-bundles` tool collects various types of bundles, or modes, of Private Cloud Appliance diagnostic data such as health check status, command outputs, and logs.

Depending on the command options provided, these bundles might contain logs or status. All modes collect files into a bundle directory. No more than one support bundle process is allowed at one time. A support bundle lock file is created at the beginning of bundle collection and removed when bundle collection is complete.

All `support-bundles` commands return immediately, and the bundle collection runs in the background. This is because bundle collections might take hours to complete. Bundles are stored for two days, then automatically deleted.

The following types of bundles are supported:

- Triage Mode on page 284. Collects data about the current status of the Private Cloud Appliance.
- Time Slice Mode on page 285. Collects data by time slots. These results can be further narrowed by specifying pod name, job, and k8s_app label.
- Combo Mode on page 287. Collects a combination of triage and time slice data.
- Native Mode on page 287. Collects data from management, compute, and ZFS nodes and from ILOM and Cisco hosts.

A good way to start to investigate an issue is to collect a `combo` bundle. Look for NOT_HEALTHY in the `triage` mode results and compare that to what you see in the `time_slice` mode results.

The `support-bundles` command requires a mode option. All modes accept the service request number option. See the following table. Time slice and native modes have additional options.

Option	Description	Required
`-m` ***`mode`***	The type of bundle.	yes
`-sr` ***`SR_number`*** `--sr_number` ***`SR_number`***	The service request number.	no

The `support-bundles` command output is stored in the following directory on the management node, where ***`bundle-type`*** is the mode: `triage`, `time_slice`, `combo`, or `native`:

```
/nfs/shared_storage/support_bundles/SR_number_bundle-type-bundle_timestamp/
```

The ***`SR_number`*** is used if you provided the `-sr` option. If you are creating the support bundle for a service request, specify the ***`SR_number`***.

This directory contains a bundle collection progress file and an archive file, which are named as follows:

```
bundle-type_collection.log
SR_number_bundle-type-bundle_timestamp.tar.gz
```

The archive file contains a `header.json` file with the following default components:

- `current-time` - the timestamp
- `create-support-bundle` - the command line that was used

- `sr-number` - the SR number associated with the archive file

Logging in to the Management Node

To use the `support-bundles` command, log in as `root` to the management node that is running Pacemaker resources. Collect data first from the management node that is running Pacemaker resources, then from other management nodes as needed.

If you do not know which management node is running Pacemaker resources, log in to any management node and check Pacemaker cluster status. The following command shows the Pacemaker cluster resources are running on pcamn01.

```
[root@pcamn01 ~]# pcs status
Cluster name: mncluster
Stack: corosync
Current DC: pcamn01
...
Full list of resources:

scsi_fencing (stonith:fence_scsi): Stopped (disabled)
Resource Group: mgmt-rg
vip-mgmt-int (ocf::heartbeat:IPaddr2): Started pcamn01
vip-mgmt-host (ocf::heartbeat:IPaddr2): Started pcamn01
vip-mgmt-ilom (ocf::heartbeat:IPaddr2): Started pcamn01
vip-mgmt-lb (ocf::heartbeat:IPaddr2): Started pcamn01
vip-mgmt-ext (ocf::heartbeat:IPaddr2): Started pcamn01
l1api (systemd:l1api): Started pcamn01
haproxy (ocf::heartbeat:haproxy): Started pcamn01
pca-node-state (systemd:pca_node_state): Started pcamn01
dhcp (ocf::heartbeat:dhcpd): Started pcamn01
hw-monitor (systemd:hw_monitor): Started pcamn01

Daemon Status:
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
```

Triage Mode

In `triage` mode, Prometheus `platform_health_check` is queried for both HEALTHY and NOT_HEALTHY status. If NOT_HEALTHY is found, use `time_slice` mode to get more detail.

```
# support-bundles -m triage
```

The following files are in the output archive file.

File	Description
`header.json`	Timestamp and command line to generate this bundle.
`compute_node_info.json`	Pods running in the compute node.
`hardware_info.json`	Hardware component list retrieved from `hms`, all the `ipmitool fru` running at all the ready state management and compute nodes, all the `zfssa` heads information.
`management_node_info.json`	Pods running in the management node.
`rack_info.json`	Rack installation time and build version.

File	Description
`loki_search_results.log.`***`n`***	Chunk files in json.

Time Slice Mode

In time slice mode, data is collected by specifying start and end timestamps. Both of the following options are required:

- `-s start_date`
- `-e end_date`

Time slice mode has the following options in addition to the mode and service request number options. These options help narrow the data collection. If you do not specify either the `-j` or `--all` option, then data is collected from all health checker jobs.

- Only one of `--job_name`, `--all`, and `--k8s_app` an be specified.
- If none of `--job_name`, `--all`, or `--k8s_app` is specified, the pod filtering will occur on the default (`.+checker`).
- The `--all` option can collect a huge amount of data. You might want to limit your time slice to 48 hours.

Example:

```
# support-bundles -m time_slice -j flannel-checker -s
 2021-05-29T22:40:00.000Z \
-e 2021-06-29T22:40:00.000Z -l INFO
```

See more examples below.

Option	Description	Required
`-s` ***`timestamp`***	Start date in format `yyyy-mmm-ddTHH:mm:ss`	yes
`--start_date` ***`timestamp`***	The minimum argument is `yyyy-mmm-dd`	
`-e` ***`timestamp`***	End date in format `yyyy-mmm-ddTHH:mm:ss`	yes
`--end_date` ***`timestamp`***	The minimum argument is `yyyy-mmm-dd`	
`-j` ***`job_name`***	Loki job name. Default value: `.+checker`	no
`--job_name` ***`job_name`***	See Label List Query below.	
`--k8s_app` ***`label`***	The `k8s_app` label value to query within the `k8s-stdout-logs` job. See Label List Query below.	no
`--all`	Queries all job names except for jobs known for too much logging, such as `audit`, `kubernetes-audit`, and `vault-audit` and `k8s_app` label `pcacoredns`.	no
`-l` ***`level`***	Message level	no
`--levelname` ***`level`***		
`--pod_name` ***`pod_name`***	The pod name (such as `kube` or `network-checker`) to filter output based on the pod. Only the starting letters are necessary.	no

Option	Description	Required
`-t` ***`timeout`*** `--timeout` ***`timeout`***	Timeout in seconds for a single Loki query. By default it is 180 seconds.	no

Label List Query

Use the label list query to list the available job names and `k8s_app` label values.

```
# support-bundles -m label_list
2021-10-14T23:19:18.265 - support_bundles - INFO - Starting Support
 Bundles
2021-10-14T23:19:18.317 - support_bundles - INFO - Locating filter-logs
 Pod
2021-10-14T23:19:18.344 - support_bundles - INFO - Executing command -
 ['python3',
'/usr/lib/python3.6/site-packages/filter_logs/label_list.py']
2021-10-14T23:19:18.666 - support_bundles - INFO -
Label:  job
Values: ['admin', 'api-server', 'asr-client', 'asrclient-checker',
 'audit', 'cert-checker', 'ceui',
'compute', 'corosync', 'etcd', 'etcd-checker', 'filesystem', 'filter-
logs', 'flannel-checker',
'his', 'hms', 'iam', 'k8s-stdout-logs', 'kubelet', 'kubernetes-audit',
 'kubernetes-checker',
'l0-cluster-services-checker', 'messages', 'mysql-cluster-checker',
 'network-checker', 'ovm-agent',
'ovn-controller', 'ovs-vswitchd', 'ovsdb-server', 'pca-healthchecker',
 'pca-nwctl', 'pca-platform-l0',
'pca-platform-l1api', 'pca-upgrader', 'pcsd', 'registry-checker',
 'sauron-checker', 'secure',
'storagectl', 'uws', 'vault', 'vault-audit', 'vault-checker', 'zfssa-
checker', 'zfssa-log-exporter']

Label:  k8s_app
Values: ['admin', 'api', 'asr-client', 'asrclient-checker', 'brs',
 'cert-checker', 'compute',
'default-http-backend', 'dr-admin', 'etcd', 'etcd-checker',
 'filesystem', 'filter-logs',
'flannel-checker', 'fluentd', 'ha-cluster-exporter', 'has', 'his',
 'hms', 'iam', 'ilom',
'kube-apiserver', 'kube-controller-manager', 'kube-proxy', 'kubernetes-
checker', '
l0-cluster-services-checker', 'loki', 'loki-bnr', 'mysql-cluster-
checker', 'mysqld-exporter',
'network-checker', 'pcacoredns', 'pcadnsmgr', 'pcanetwork',
 'pcaswitchmgr', 'prometheus', 'rabbitmq',
'registry-checker', 'sauron-api', 'sauron-checker', 'sauron-grafana',
 'sauron-ingress-controller',
'sauron-mandos', 'sauron-operator', 'sauron-prometheus', 'sauron-
prometheus-gw',
'sauron-sauron-exporter', 'sauron.oracledx.com', 'storagectl', 'switch-
metric', 'uws', 'vault-checker',
'vmconsole', 'zfssa-analytics-exporter', 'zfssa-csi-nodeplugin',
 'zfssa-csi-provisioner', 'zfssa-log-exporter']
```

Examples:

- No job label, no k8s_app label, collect log from all health checkers.

```
# support-bundles -m time_slice -sr 3-xxxxxxxxxxx -s
 "2022-01-11T00:00:00" -e "2022-01-12T23:59:59"
```

- One job ceui.

```
# support-bundles -m time_slice -sr 3-xxxxxxxxxxx -j ceui -s
 "2022-01-11T00:00:00" -e "2022-01-12T23:59:59"
```

- One k8s_app network-checker.

```
# support-bundles -m time_slice -sr 3-xxxxxxxxxxx --k8s_app network-
checker -s "2022-01-11T00:00:00" -e "2022-01-12T23:59:59"
```

- All jobs and date.

```
# support-bundles -m time_slice -sr 3-xxxxxxxxxxx -s `date -d "2
 days ago" -u +"%Y-%m-%dT%H:%M:%S.000Z"` -e `date -d +u +"%Y-%m-%dT
%H:%M:%S.000Z"`
```

- All jobs.

```
# support-bundles -m time_slice -sr 3-xxxxxxxxxxx --all -s
 "2022-01-11T00:00:00" -e "2022-01-12T23:59:59"
```

The following files are in the output archive file.

File	Description
`header.json`	Timestamp and command line to generate this bundle.
`loki_search_results.log.`***`n`***	Chunk files in json. Time slice bundles have a limit of 500,000 logs per query, from start time.
`rack_info.json`	Rack installation time and build version.

Combo Mode

The `combo` mode is a combination of a triage bundle and a time slice bundle. The output includes an archive file and two collection log files: `triage_collection.log` and `time_slice_collection.log`.

The following files are in the output archive file.

File	Description
`triage-bundle_`***`timestamp`***`.tar.gz`	The triage bundle archive file.
`time_slice-bundle_`***`timestamp`***`.tar.gz`	The time slice bundle archive file. The time slice data collected is for `--all` jobs from one hour preceding the current time to the current time.

Native Mode

The `native_collection.log` file in the bundle directory provides collection progress information. Native bundles can take hours to collect.

The `native` mode has the following parameters in addition to mode and SR number.

Parameter	Description	Required
`-t` ***`nativetype`*** `--type` ***`nativetype`***	• `zfs_bundle` • `sosreport` • `ilom_snapshot` • `cisco_bundle` Default value: `zfs_bundle`	no
`-c` ***`component`*** `--component` ***`component`***	Component name, such as the name of a management, compute, or ZFS node, or an ILOM or Cisco host.	no

The following files are in the output archive file.

File	Description
`header.json`	Time stamp and command line to generate this bundle.
Native bundle files	These files are specific to the ***`nativetype`*** specified.
`rack_info.json`	Rack installation time and build version.

ZFS Bundle

When ***`nativetype`*** is a ZFS support bundle, collection starts on both ZFS nodes and downloads the new ZFS support bundles into the bundle directory. When ***`nativetype`*** is not specified, `zfs_bundle` is created by default.

```
# support-bundles -m native -t zfs_bundle
```

SOS Report Bundle

When ***`nativetype`*** is an SOS report bundle, the report is collected from the management node or compute node specified by the `--component` parameter. If `--component` is not specified, the report is collected from all management and compute nodes.

```
# support-bundles -m native -t sosreport -c pcamn01
```

ILOM Snapshot

When `nativeType=ilom_snapshot`, the value of the `--component` parameter is the ILOM host name of a management node or compute node. If the `--component` parameter is not specified, the report is collected from all ILOM hosts.

```
# support-bundles -m native -t ilom_snapshot -c ilom-pcacn007
```

Cisco Bundle

When ***`nativetype`*** is `cisco-bundle`, the value of the `--component` parameter is an internal Cisco management, aggregation, or access switch management host name.

```
# support-bundles -m native -t cisco-bundle -c accsn01
```

To create a `cisco-bundle` type of collection, the following conditions must be met:

- The Cisco OBFL module must be enabled on all Private Cloud Appliance Cisco switches. The Cisco OBFL module is enabled by default on all Private Cloud Appliance Cisco switches.

- The Cisco EEM module must be enabled on all Private Cloud Appliance Cisco switches. The Cisco EEM module is enabled by default on all Private Cloud Appliance Cisco switches.
- EEM (Embedded Event Manager) policy

Uploading Support Bundles to Oracle Support

After creating a support bundle on Private Cloud Appliance, you can upload it to Oracle Support by following the instructions in this topic.

Prerequisites

To upload files to Oracle Support, ensure these prerequisites are met:

- You must have a My Oracle Support user ID with Create and Update SR permissions granted by the appropriate Customer User Administrator (CUA) for each Support Identifier (SI) being used to upload files.
- For file uploads to existing service requests, the Support Identifier associated with the service request must be in your profile.
- To upload files larger than 2 GB, sending machines must have network access to connect to the My Oracle Support servers at `transport.oracle.com` to use FTPS and HTTPS.

  The Oracle FTPS service is a "passive" implementation. With an implicit configuration, the initial connection is from the client to the service on a control port of 990 and the connection is then switched to a high port to exchange data. Oracle defines a possible range of the data port of 32000 to 42000, and depending upon your network configuration you may need to enable outbound connections on both port 990 and 32000 to 42000. TLSv1.2 / ECDHE-RSA-AES128-GCM-SHA256 is the only encryption method enabled.

  The Oracle HTTPS diagnostic upload service uses the standard HTTPS port of 443 and does not require any additional ports to be opened.

  When using command line protocols, do not include your password in the command. Enter your password only when prompted.
- Oracle requires the use of TLS 1.2+ for all file transfers.
- Do not upload encrypted or password-protected files, standalone or within an archive. A Service Request update will note this as a corrupted file or reject the upload as disallowed file types were found. Files are encrypted when you use FTPS and HTTPS; additional protections are not required.
- Do not upload files with file type extensions `exe`, `bat`, `asp`, or `com`, either standalone or within an archive. A Service Request update will note that a disallowed file type was found.

Uploading Files 2 GB or Smaller

Use the SR file upload utility on the My Oracle Support Portal.

1. Log in to My Oracle Support with your My Oracle Support user name and password.
2. Do one of the following:
   - Create a new service request and in the next step, select the Upload button.
   - Select and open an existing service request.
3. Click the Add Attachment button located at the top of the page.
4. Click the Choose File button.
5. Navigate and select the file to upload.
6. Click the Attach File button.

You can also use the methods described in the next section for larger files.

Uploading Files Larger Than 2 GB

You cannot upload a file larger than 200 GB. See Splitting Files on page 290.

The `curl` commands in this section show required options and arguments. You might want to add options such as `--verbose` and `--progress-bar` to get more information about your upload. The `--progress-meter` (more information than `--progress-bar`) should be on by default, but it is disabled when `curl` is writing

other information to `stdout`. Note that some options might not be available or might behave differently on some operating systems or some versions of `curl`.

The following are the most common messages from uploading bundles to Oracle Support if you use the `--verbose` option with the `curl` command:

- `UPLOAD SUCCESSFUL`. The bundle is successfully uploaded to Oracle Support.
- `LOGIN FAILED`. The user has an authentication issue.
- `INVALID SR NUMBER`. The user does not have attach privilege to this Service Request.

FTPS

Syntax:

Be sure to include the `/` character after the service request number.

```
$ curl -T path_and_filename -u MOS_user_ID ftps://transport.oracle.com/
issue/SR_number/
```

Example:

```
$ curl -T /u02/files/bigfile.tar -u MOSuserID@example.com ftps://
transport.oracle.com/issue/3-1234567890/
```

HTTPS

Syntax:

Be sure to include the `/` character after the service request number.

```
$ curl -T path_and_filename -u MOS_user_ID https://
transport.oracle.com/upload/issue/SR_number/
```

Example:

```
$ curl -T D:\data\bigfile.tar -u MOSuserID@example.com https://
transport.oracle.com/upload/issue/3-1234567890/
```

Renaming the file during send

```
$ curl -T D:\data\bigfile.tar -u MOSuserID@example.com https://
transport.oracle.com/upload/issue/3-1234567890/NotSoBig.tar
```

Using a proxy

```
$ curl -k -T D:\data\bigfile.tar -x proxy.example.com:80 -u
 MOSuserID@example.com https://transport.oracle.com/upload/
issue/3-1234567890/
```

Splitting Files

You can split a large file into multiple parts and upload the parts. Oracle Transport will concatenate the segments when you complete uploading all the parts.

Only HTTPS protocol can be used. Only the UNIX split utility can be used. The Microsoft Windows split utility produces an incompatible format.

To reduce upload times, compress the original file before splitting.

1. Split the file.

   The following command splits the file `file1.tar` into 2 GB parts named `file1.tar.partaa` and `file1.tar.partab`.

> **Important:**
>
> Specify the `.part` extension exactly as shown below.

```
$ split –b 2048m file1.tar file1.tar.part
```

2. Upload the resulting `file1.tar.partaa` and `file1.tar.partab` files.

> **Important:**
>
> Do not rename these output part files.

```
$ curl -T file1.tar.partaa -u MOSuserID@example.com https://
transport.oracle.com/upload/issue/SR_number/
$ curl -T file1.tar.partab -u MOSuserID@example.com https://
transport.oracle.com/upload/issue/SR_number/
```

3. Send the command to put the parts back together.

   The spit files will not be attached to the service request. Only the final concatenated file will be attached to the service request.

```
$ curl -X PUT -H X-multipart-total-size:original_size -u
 MOSuserID@example.com https://transport.oracle.com/upload/
issue/SR_number/file1.tar?multiPartComplete=true
```

   In the preceding command, ***original_size*** is the size of the original unsplit file as shown by a file listing.

4. Verify the size of the newly attached file.

> **Note:**
>
> This verification command must be executed immediately after the concatenation command in Step 3. Otherwise, the file will have begun processing and will no longer be available for this command.

```
$ curl -I -u MOSuserID@example.com https://transport.oracle.com/upload/
issue/SR_number/file1.tar
    X-existing-file-size: original_size
```

Resuming an Interrupted HTTPS Upload

You can resume a file upload that terminated abnormally. Resuming can only be done by using HTTPS. Resuming does not work with FTPS. When an upload is interrupted by some event, the start with retrieving the file size of the interrupted file

1. Determine how much of the file has already been uploaded.

```
$ curl -I -u MOSuserID@example.com https://transport.oracle.com/upload/
issue/SR_number/myinfo.tar
HTTP/1.1 204 No Content
Date: Tue, 15 Nov 2022 22:53:54 GMT
Content-Type: text/plain
X-existing-file-size: already_uploaded_size
X-Powered-By: Servlet/3.0 JSP/2.2
```

2. Resume the file upload.

Note the file size returned in “X-existing-file-size” in Step 1. Use that file size after the `-C` switch and in the `-H “X-resume-offset:”` switch.

```
$ curl -Calready_uploaded_size -H "X-resume-offset: already_uploaded_size"
 -T myinfo.tar -u MOSuserID@example.com https://transport.oracle.com/
upload/issue/SR_number/myinfo.tar
```

3. Verify the final file size.

```
$ curl -I -u MOSuserID@example.com https://transport.oracle.com/upload/
issue/SR_number/myinfo.tar
-H X-existing-file-size: original_size
```

In the preceding command, ***`original_size`*** is the size of the original file as shown by a file listing.

Intrusion Monitoring

Private Cloud Appliance provides utilities designed to run virus scans and ensure file integrity on management nodes and compute nodes.

The intrusion monitor is enabled by default. It detects and reports software level intrusions on all management and compute nodes installed in the system.

- ClamAV is used to scan for viruses, malware, and other types of threats to the system software.
- File integrity monitoring is based on Advanced Intrusion Detection Environment. AIDE is a tool that creates a baseline database of critical files, and periodically compares the current state to that baseline. Inconsistencies in file attributes or contents might indicate unauthorized changes, and are reported for further investigation.

Integrity checks and virus scans run once every week. The result of each run is saved to Loki logs and is viewable in Grafana. See also Loki Logs on page 262 for information about finding and managing log data.

The latest versions of the monitoring service databases are installed with the appliance controller software upgrade. Configure intrusion monitoring from the Service CLI.

- To disable the service:

```
PCA-ADMIN> intrusionMonitorDisable
Data:
  status = success
  message = Intrusion Monitor Timer disabled successfully on all nodes.
```

- To enable the service and set the weekly schedule:

```
PCA-ADMIN> intrusionMonitorEnable day=Friday time=22:30:00
Data:
  status = success
  message = Intrusion Monitor Timer configured successfully on all nodes.
```

The day and time are optional parameters. If you leave them out, the defaults will be used: Sunday at 00:00:00.

System Backup and Restore

The integrated backup service protects the Private Cloud Appliance system configuration against data loss and corruption. It stores the data required for system and service operation so that any crucial service or component can be restored to its last known healthy state. Backups of user environments in the Compute Enclave are not included.

The following file systems are backed up:

- PCA project: `obj_share` and `MGMT_ROOT`
- `public_ostore_project`: File systems related to each tenancy created on the rack (for COS operations)
- `private_ostore` project: File system in `private_ostore_project`

In line with the microservice-based deployment model, the backup service orchestrates the various backup operations across the entire system and ensures data consistency and integrity, but it does not define the individual component backup requirements. That logic is part of the component backup plugins.

The backup plugin is the key element that determines which files are backed up for a given system component or service, and how the data is collected. For example, a simple file copy may work for certain files while a snapshot is required for other data, or in some cases a service may need to be stopped to allow the backup to be created. The plugin also determines the backup frequency and retention time. Each plugin registers with the backup service so that the backup service is aware of the active plugins and can schedule the required backup operations in a consistent manner as Kubernetes CronJobs. Plugins are aggregated into a backup profile; the backup profile is the list of tasks that the backup service performs when a backup job is run.

The backup data collected through the plugins is then stored by the backup service in a dedicated NFS share on the internal ZFS Storage Appliance, using ZFS encryption to ensure that the data at rest is secure. If required, the backup files can optionally be replicated to an external storage location.

When restoring a service or component from a backup, the service again relies on the logic provided by the plugin. A component restore process has two major phases: verification and data management. In the verification phase, the backup is evaluated for completeness and appropriateness in relation to the current condition of the component. In the data management phase, the required actions are taken to stop or suspend a component, replace the data, and restart or resume normal operation. As with backup, the operations to restore the data are specific to the component in question.

The default backup and restore implementation is to perform all tasks in a global backup profile that covers the MySQL cluster database, a snapshot of the ZFS projects on the storage appliance, and all registered component backup plugins. The default profile is processed daily at midnight UTC and has a 14-day retention policy; older backups are deleted. Backups are stored in `/nfs/shared_storage/backups/backup_*`. All restore operations must be performed manually on a per-component basis.

Important:

- Automated restore operations based on the backup plugins are not possible. If a manual restore from a backup is required, contact Oracle for assistance.
- Monitoring data from Prometheus is not included in automated backups. To preserve your Prometheus data, create a backup and restore it manually. For more information, refer to the support note with Doc ID 3021643.1.
- Automatic purging of backups – regardless of whether it is a standard daily backup or a manually triggered operation – is critical for backups of the **MySQL database**. If a MySQL backup must be stored longer than the retention period, for example because it represents an important restore point, ensure that the data is copied to another location before the retention period expires. Contact your Oracle representative for assistance.
- Certain snapshots are not governed by any retention policy. If these are no longer needed, you must delete them manually. See Converted Snapshots on page 296.

Activating the Standard Daily Backup

System backups are not available by default. It is critical that you activate standard daily backups on your Private Cloud Appliance.

Caution:

Ensure that daily backups are activated after system initialization. If this procedure is omitted, you will not be able to restore a component or service from a last known good state.

To activate system backups, set up a Kubernetes CronJob by running the applicable script from the management node that owns the virtual IP of the cluster.

When the system initialization process is complete, perform the following procedure to activate system backups:

1. Log on to one of the management nodes.
2. Retrieve the name of the Kubernetes pod that runs the backup and restore service. Use the following command:

```
# kubectl get pods -A | grep brs
default    brs-5bdc556546-gxtx9        3/3    Running    0    17d
```

3. Run the `default-backup` script as shown in the following example to set up the Kubernetes CronJob to make a daily backup.

```
kubectl exec brs-5bdc556546-gxtx9 -c brs -- /usr/sbin/default-backup
```

   This backup runs every day at 00:00 local system time and is retained for 14 days.
4. Verify that the CronJob has been added in the default namespace.

```
# kubectl get cronjobs -A
NAMESPACE      NAME                              SCHEDULE       SUSPEND
 ACTIVE   LAST SCHEDULE   AGE
default        brs-cronjob-1629969790-backup     0 0 * * *      False     0
        <none>          32s
health-check   cert-checker                      */10 * * * *   False     0
        4m6s            17d
health-check   etcd-checker                      */10 * * * *   False     0
        4m6s            17d
health-check   flannel-checker                   */10 * * * *   False     0
        4m6s            17d
health-check   kubernetes-checker                */10 * * * *   False     0
        4m6s            17d
health-check   l0-cluster-services-checker       */10 * * * *   False     0
        4m6s            17d
health-check   mysql-cluster-checker             */10 * * * *   False     0
        4m6s            17d
health-check   network-checker                   */10 * * * *   False     0
        4m6s            17d
health-check   registry-checker                  */10 * * * *   False     0
        4m6s            17d
health-check   sauron-checker                    */10 * * * *   False     0
        4m6s            17d
health-check   vault-checker                     */10 * * * *   False     0
        4m6s            17d
sauron         sauron-sauron-prometheus-gw-cj    30 19 * * *    False     0
        18h             17d
```

   When this `brs-cronjob-`***`unique_ID`***`-backup` CronJob runs, any backups that were previously created by a `brs` manual system backup job that are more than 14 days old are deleted.

   ZFS Storage Appliance manual snapshots are not deleted if they were not created by using any `brs` job, and their snapshot name is not in the following form (the `brs` snapshot naming convention):

```
projectname/filesystemname_timestamp
```

   Backups are created on the ZFS Storage Appliance at the following location, as seen from the management node mount point:`/nfs/shared_storage/backups/`.

   Each backup is identified by its unique path containing the job OCID and timestamp:

```
/nfs/shared_storage/backups/ocid1.backup_cronjob.unique_ID/
backup_timestamp/
```

Performing a Manual Backup Operation

Ensure that Private Cloud Appliance system backups are active. In addition, you can manually perform system backups to create critical restore points.

Manual backups are initiated from the Service CLI.

1. Select a strategy: create a full system backup, or back up individual components.
2. Run the backup command with the target parameter of your choice.

   Target options are: `layer0`, `zfs`, `vault`, `mysql`, `loki`, `sauron`, `all`.

   - To create a full system backup, select `target=all`.

     ```
     PCA-ADMIN> backup target=all
     Data:
       Type = BackupJob
       Job Id = ocid1.brs-job.2147XLD01D....<unique_ID>
       Display Name = brs-job-1698401412-backup
       Profile Id = ocid1.backup_profile.2147XLD01D....<unique_ID>
       Time Created = 2023-10-27T10:10:12Z
       Lifecycle State = CREATING
       Retention = 14
     ```

   - To create an individual component backup, select the appropriate target from the list: `layer0 | zfs | vault | mysql | loki | sauron`. For example:

     ```
     PCA-ADMIN> backup target=layer0
     Data:
       Type = BackupJob
       Job Id = ocid1.brs-job.2147XLD01D....<unique_ID>
       Display Name = brs-job-1698401607-backup
       Profile Id = ocid1.backup_profile.2147XLD01D....<unique_ID>
       Time Created = 2023-10-27T10:13:27Z
       Lifecycle State = CREATING
       Retention = 14
     ```

3. Use the backup job ID to check the status of the backups.

   ```
   PCA-ADMIN> getBackupJobs
   Data:
     id
                            Display Name                Components
     --
                            ------------                ----------
     ocid1.brs-
   job.PCA3X62D9C1.mypca.wtczj8r3zl7wjftxbkskaj6j60x6xqai3lpjoxp7ywi7nmrcuyo4vathc8rj
      brs-job-1698401412-backup   layer0,zfs,vault,mysql,loki,sauron
     ocid1.brs-
   job.PCA3X62D9C1.mypca.089a7b8cuqz0cam2r7xexo4i4p7j7ia7sqhl9f8w89dyp9q3y10dnbaac6mu
      brs-job-1698401607-backup   layer0

   PCA-ADMIN> getBackupJob backupJobId=ocid1.brs-
   job.PCA3X62D9C1.mypca.wtczj8r3zl7wjftxbkskaj6j60x6xqai3lpjoxp7ywi7nmrcuyo4vathc8rj
   Data:
     Type = BackupJob
     Job Id = ocid1.brs-
   job.PCA3X62D9C1.mypca.wtczj8r3zl7wjftxbkskaj6j60x6xqai3lpjoxp7ywi7nmrcuyo4vathc8rj
     Display Name = brs-job-1698401412-backup
     Time Created = 2023-10-27T10:10:12Z
     Status = success
     Components = layer0,zfs,vault,mysql,loki,sauron
   ```

4. Confirm that the backup operations have completed successfully.

Converted Snapshots

Converted snapshots and any manually created snapshots are not governed by retention policies of the Private Cloud Appliance Backup and Restore service. Snapshots that are no longer needed, must be deleted manually.

What Is a Converted Snapshot?

If you change any settings in a backup policy other than the display name of the policy, snapshots that were created under that policy are detached and marked as converted. For example, if you update a backup policy to add a new schedule, change a schedule, or delete a schedule, all existing snapshots are detached and marked as converted on the ZFS Storage Appliance.

A converted snapshot has been converted from being managed by a backup policy to requiring manual management. Specifically, snapshots that are marked as converted will not be automatically deleted because they are no longer governed by any retention policy. When these snapshots are no longer needed, you must delete them manually.

How to Identify Converted Snapshots

The following procedure describes how to identify converted snapshots. In addition to converted snapshots, any snapshots that you created manually must be deleted manually.

1. Log on to one of the management nodes.
2. Show `storagectl` content.

```
# helm history storagectl
REVISION UPDATED STATUS CHART APP VERSION DESCRIPTION
3 Mon Sep 23 14:01:17 2024 superseded
 storagectl-3.0.202-3.10.0.0.1.75.g94b57a2 3.0.202-3.10.0.0.1.75.g94b57a2
 Upgrade complete
4 Tue Sep 24 09:39:19 2024 superseded
 storagectl-3.0.202-3.10.0.0.0.79.g1f63430 3.0.202-3.10.0.0.0.79.g1f63430
 Upgrade complete
5 Wed Sep 25 03:20:06 2024 superseded
 storagectl-3.0.202-3.10.0.0.0.80.gbb09c40 3.0.202-3.10.0.0.0.80.gbb09c40
 Upgrade complete
6 Thu Sep 26 09:33:19 2024 superseded
 storagectl-3.0.202-3.10.0.0.1.66.g435af30 3.0.202-3.10.0.0.1.66.g435af30
 Upgrade complete
7 Tue Oct 1 19:37:24 2024 superseded
 storagectl-3.0.202-3.10.0.0.1.75.gf712a7e 3.0.202-3.10.0.0.1.75.gf712a7e
 Upgrade complete
8 Tue Oct 1 19:46:22 2024 superseded
 storagectl-3.0.202-3.10.0.0.1.75.gf712a7e 3.0.202-3.10.0.0.1.75.gf712a7e
 Upgrade complete
9 Tue Oct 1 19:51:08 2024 superseded
 storagectl-3.0.202-3.10.0.0.1.79.ged909dd 3.0.202-3.10.0.0.1.79.ged909dd
 Upgrade complete
10 Tue Oct 1 19:53:25 2024 superseded
 storagectl-3.0.202-3.10.0.0.1.75.gf712a7e 3.0.202-3.10.0.0.1.75.gf712a7e
 Upgrade complete
11 Tue Oct 1 20:00:18 2024 superseded
 storagectl-3.0.202-3.10.0.0.1.79.ged909dd 3.0.202-3.10.0.0.1.79.ged909dd
 Upgrade complete
12 Tue Oct 1 21:45:03 2024 deployed
 storagectl-3.0.202-3.11.0.0.1.1.g1e7dc83 3.0.202-3.11.0.0.1.1.g1e7dc83
 Upgrade complete
```

3. Extract the `storagectl` chart from `/nfs/shared_storage/`.

   In this example, we use the chart from the latest revision (the last line) shown in Step 2.

```
# tar -xvf /nfs/shared_storage/charts/
storagectl-3.0.202-3.11.0.0.1.1.g1e7dc83.tgz -C /target_directory
```

```
# cd /target_directory
```

4. Delete the file `job.batch/storagectl-list-converted-snapshots` if it exists.

```
# kubectl delete job.batch/storagectl-list-converted-snapshots
```

5. Rename the job definition file to remove the underscore prefix.

```
# cp storagectl/templates/_storagectl-list-converted-snapshots.yaml
 storagectl/templates/storagectl-list-converted-snapshots.yaml
```

6. (Optional) Set input parameters such as `volumeId` or `compartmentId` (the volume or compartment for which you want to list converted snapshots) or output file name.

   If you do not set either a volume OCID or a compartment OCID, then the list will include all converted snapshots, which could be a large list.

```
# kubectl edit configmap storagectl-list-converted-snapshots-config
```

7. Deploy the Kubernetes job.

```
# helm template -s templates/storagectl-list-converted-snapshots.yaml
 storagectl/ | kubectl apply -f -
```

8. Check the status of the job.

```
# kubectl get all -A | grep storagectl
```

   When the Kubernetes job is finished, the corresponding `pod/storagectl-list-converted-snapshots-*` is complete and the output file contains the list of converted snapshots.

9. View the list of converted snapshots in the output file.

```
# cat /nfs/shared_storage/pca-platform/kubernetes/storagectl-data/
converted_snapshots_data.json
```

   The snapshots in this list will not be automatically deleted. If these snapshots are no longer needed, delete them manually.

Restoring the System from a Backup

Restoring Private Cloud Appliance system data from a backup is an Oracle Support procedure.

If a system component or service is in a bad state, or if lost system data needs to be recovered from a critical restore point, a backup can be restored by trained personnel. Contact your Oracle representative for assistance.

Disaster Recovery

The goal of disaster recovery (DR) is to provide high availability at the level of an installation site, and to protect critical workloads hosted on Private Cloud Appliance against outages and data loss.

The current Private Cloud Appliance controller software provides a Disaster Recovery service with orchestration of DR operations from within the Service Enclave. The service is also called *native DR* because it is built directly into the infrastructure services layer.

> **Note:**
>
> Existing installations might use the previous generation disaster recovery, which relies on a third system running Oracle Enterprise Manager with Oracle Site Guard. A path is provided to migrate to the current DR service.
>
> First-generation DR documentation is available on Oracle Help Center in the Private Cloud Appliance Administrator Guide. See the Disaster Recovery chapter.

Setting up disaster recovery is the responsibility of an appliance administrator or Oracle engineer. The systems participating in the DR setup are fully operational environments on their own, running in different physical locations. A mutual peer connection must be established first, so they can operate as each other's standby or replica in case an outage occurs at one of the sites. The DR services on the primary and standby racks communicate using REST API calls. Local commands are sent to the appropriate cloud infrastructure service through the platform layer's internal messaging and administration services.

The DR service makes a clear distinction between resources and operations. The administrator determines which workloads and resources are under disaster recovery protection by creating and managing *DR configurations*. The DR configurations define which compute instances are protected against site-level incidents, and map the relevant source and target compartments and networking resources between the peered systems.

DR operations are defined in a DR plan, which outlines the steps to perform during a switchover, failover, or postfailover operation. DR configurations and DR plans are shared between the peered systems, and can be created and maintained from the primary or standby system. DR operations could also be executed from either system, except for failover, which is always triggered from the standby.

It is important to understand what is covered under disaster recovery and what is not.

Disaster recovery supports:

- Compute instances
- The block volumes associated with these compute instances

The following limitations apply to the disaster recovery feature:

- File systems are not supported
- Object storage is not supported
- OKE clusters are not supported
- Application and network load balancers are not supported
- SR-IOV instances are not supported
- Compute instances with shared block volumes are not supported.

Migrating to the Native Disaster Recovery Service

Existing Private Cloud Appliance environments set up for first generation disaster recovery can be migrated to the native Disaster Recovery Service as soon as the systems have been upgraded or patched to appliance software version 3.0.2-b1261765 or later.

First, ensure that both environments are running the minimum required appliance software version. A first-generation disaster recovery setup cannot operate as the standby of a system running the newer native DR service, or the other way around. As soon as a peer connection has been established, first-generation DR configurations can no longer be used, but you can migrate them to the native DR service.

Collecting Oracle Site Guard DR Plans

To extract the DR plan data from Site Guard, you need the `get_em_plan.sh` script, and a valid user name and password for the Oracle Enterprise Manager database schema. During appliance upgrade or patching, the script is saved to the directory `/var/lib/pca-foundation/scripts`.

1. Log in to the system that runs the Enterprise Manager database, where the data from the Site Guard plugin is stored.
2. Copy the `get_em_plan.sh` script to this system. Run the script from the command line and specify an output file. The data is stored in CSV format.

   > **Note:**
   >
   > To retrieve the plan data for a single DR configuration, add its name to the end of the command, separated by a space.

   ```
   $ ./get_em_plan.sh em_plan_data.csv
   ```

3. When prompted by the script, enter the database user name and password.

   The DR plan data is retrieved from the database and saved to the `*.csv` output file you specified.
4. Copy the `*.csv` file before logging out.

Switching to the Native DR Service

Before converting the first-generation DR plans, you must set up peering between both systems, and activate the native DR service.

1. Ensure that an active peer connection exists between the primary and standby appliance. See Establishing a Peer Connection on page 300.
2. Activate the native DR service by updating the existing service configuration. The serial number of the peer system is a required parameter.

```
PCA-ADMIN> drNativeUpdateService peerSerialNumber=<peer_serial>
```

Converting to Native DR Plans

You can import the DR plan data from Site Guard into the native DR service using the Service CLI.

1. Ensure that the `*.csv` output file with the DR plan data from Site Guard is in a location that the DR service pod can access.
2. Create DR plans from the `*.csv` file with the `create DrPlan` command.

   For each DR configuration that matches a record in the `*.csv` file, the DR plans are converted and saved in the native DR format.

   Syntax (entered on a single line):

```
create DrPlan
drPlanDataFile=<csv_file_name>
configsToMigrate=["all"|"config_id_1,config_id_n"]
```

   - To convert DR plan data for a subset of DR configurations from the CSV file, specify the IDs of the required configurations:

```
PCA-ADMIN> create DrPlan drPlanDataFile=/home/admin/em_plan_data.csv \
configsToMigrate="config1_id, config2_id, config3_id"
  JobId: 146c6422-cd82-4ecf-9e5f-a99acfc9baab
  Data: DrPlan id: null::null. Successfully started job for plan
 creation using /home/admin/em_plan_data.csv
```

   - To convert all DR plans for all matching DR configurations from the CSV file:

```
PCA-ADMIN> create DrPlan drPlanDataFile=/home/admin/em_plan_data.csv
 configsToMigrate="all"
  JobId: 862b045a-c4f6-4f8c-8130-9d9b487bd363
  Data: DrPlan id: null::null. Successfully started job for plan
 creation using /home/admin/em_plan_data.csv
```

3. Use the job ID to check the status of the operation you started.

```
PCA-ADMIN> show Job id=862b045a-c4f6-4f8c-8130-9d9b487bd363
Data:
  Id = 862b045a-c4f6-4f8c-8130-9d9b487bd363
  Type = Job
  Associated Work Request Id = 8671211c-68b3-426e-aa7c-d93d9ffb32e4
  Done = true
  Name = CREATE_TYPE
  Progress Message = Plan migration succeeded.
  Run State = Succeeded
```

```
  Transcript = Created job CREATE_TYPE
  Username = admin
  WorkItemIds 1 = id:08cb4f9a-575b-4536-b832-b85948c90212  type:WorkItem
 name:
```

Use the work request ID to display more details.

```
PCA-ADMIN> drGetJob jobId=8671211c-68b3-426e-aa7c-d93d9ffb32e4
Data:
  Type = drplan_migration
  Job Id = 8671211c-68b3-426e-aa7c-d93d9ffb32e4
  Status = finished
  Result = succeeded
  Message = job successfully retrieved
  Response =
    config_id: b51a51ac-043e-4b6d-ab29-20f86d905e81: plan_name: fo1
 status: success
    config_id: b51a51ac-043e-4b6d-ab29-20f86d905e81: plan_name: sw1
 status: success
    config_id: b51a51ac-043e-4b6d-ab29-20f86d905e81: plan_name: pfo1
 status: success
```

4. To confirm that the DR plans have been added, use the `drListPlan` command.

```
PCA-ADMIN> drListPlan drConfigId=b51a51ac-043e-4b6d-ab29-20f86d905e81
Data:
  id                                          Plan Name   Operation
  --                                          ---------   ---------
  b51a51ac-043e-4b6d-ab29-20f86d905e81::fo1   fo1         failover
  b51a51ac-043e-4b6d-ab29-20f86d905e81::sw1   sw1         switchover
  b51a51ac-043e-4b6d-ab29-20f86d905e81::pfo1  pfo1        postfailover
```

5. Modify the DR plans as required. For details, see Working with Disaster Recovery Plans on page 318.

Establishing a Peer Connection

The disaster recovery service requires a mutual, symmetrical peer connection between two Private Cloud Appliance system installations. Dedicated cabling must be installed at each site, and the disaster recovery service on each rack must be configured to accept the other rack as its standby system.

To establish a peer connection, follow these steps:

1. Adding Cable Connections for Peering on page 300
2. Creating a Local Endpoint on page 301
3. Completing the Peering Configuration on page 303

Adding Cable Connections for Peering

A peer connection between Private Cloud Appliance racks requires additional physical connections. Dedicated cabling must be installed between the spine switches and the data center network.

Racks with factory-installed software version 3.0.2-b1261765 or later already have all the necessary internal network interfaces and connections. Only external cabling for the peer connection is required. The ZFS Storage Appliances use the same physical connections for their data replication.

In existing installations where first-generation disaster recovery is configured, an active replication network between the ZFS Storage Appliances is in place. When upgrading or patching to the latest appliance software, the existing replication network remains active. The new physical connections from the spine switches are used for peering traffic only.

Data Center Cabling for Rack Peering

Direct peering between racks requires dedicated cabling for each participating system. The additional connections between the spine switches and the data center network are the physical basis on which the network tunnels of the peer connection are configured.

For the purpose of peering, port 6 on each spine switch must be connected to the data center network. To provide the required connection speed of 10 or 25 Gbps, a 4-way breakout cable is attached to spine port 6. From the breakout cable, 1 transceiver is connected to the data center network. Cabling must be identical for both spine switches.

Internal Cabling for Rack Peering

Appliance rack configurations shipped from the factory before the release of the native DR service do not have the required internal cabling to enable replication through the peer connection tunnels. They lack these crucial components:

- PCIe 25GbE network interface card in some models of the ZFS Storage Appliance Controllers
- Ethernet cabling between ZFS Storage Appliance Controllers and spine switches (port 27)

These components can be added to existing installations, so their hardware configuration is equivalent to racks with factory-installed software version 3.0.2-b1261765 or later. Contact Oracle for assistance and additional information.

Backward Compatibility

The native DR service supports both cabling layouts:

- **Peering Topology:** combined peer connection and storage replication network through the spine switches
- **Compatibility Topology:** peer connection and physically separated direct replication link between the ZFS Storage Appliances

The compatibility topology provides different options for existing installations after upgrading or patching to software version 3.0.2-b1261765 or later. If you have a first-generation DR setup, you can choose to continue with this configuration, on condition that you do **not** establish a peer connection at the appliance level. However, Oracle recommends migrating the existing configuration to the native DR service, in accordance with your infrastructure design and maintenance schedule. Data center cabling for the peer connection must be added, but you can continue to use the existing storage replication connection. For more information, see Migrating to the Native Disaster Recovery Service on page 298.

Creating a Local Endpoint

Traffic between peered Private Cloud Appliance systems flows through tunnels between endpoints. A rack must have a local endpoint configured before it can participate in a peer connection.

Assuming the network configuration remains the same, a local endpoint is set up once. It remains configured if a peer connection is deleted, so it can be reused for any new connection. These parameters are required to create the local endpoint:

- an IP address for each spine switch
- the IP addresses of the data center gateways to which the spines are connected
- IP address of the capacity ZFS pool and, if present, the high-performance ZFS pool
- the ASN ID of your network environment, if applicable

Network Configuration Guidelines

A local endpoint requires a /29 address block, which has 6 usable IP addresses. Within that /29 range, the spine switch pair is assigned 3 IPs (one is shared). Each ZFS pool is assigned 1 IP *outside* the spine switch subnet. To allow for additional future peer connections, it is recommended to reserve a data center IP range of at least /25 in size, which corresponds with 16 address blocks of /29 size.

When setting up the local endpoint, you must provide the netmask with the peering network IPs, but not with IPs that have already been configured, such as the data center addresses. The gateway IPs must be provided by the network administrator, and assigned to the data center switches to which the spine switches are connected. Note that spine 1 corresponds with gateway 1, and spine 2 with gateway 2.

The native DR service sets a host name for each ZFS pool: `sn01-dr1.<example.com>` for the capacity pool, and `sn02-dr1.<example.com>` for the performance pool (if present). Replication connectivity between the ZFS Storage Appliances of the peered systems relies on DNS, not the IP addresses. Ensure that the fully qualified domain names resolve on both sides of the peer connection. Add PTR (pointer record) entries to the data center DNS so that requests for the remote environment are forwarded to the remote DNS. For example:

- System A:

```
129.128.212.10.in-addr.arpa rtype PTR rdata sn01-dr1.sys-
b.example.com
130.128.212.10.in-addr.arpa rtype PTR rdata sn02-dr1.sys-
b.example.com
```

- System B:

```
34.224.86.10.in-addr.arpa rtype PTR rdata sn01-dr1.sys-a.example.com
35.224.86.10.in-addr.arpa rtype PTR rdata sn02-dr1.sys-a.example.com
```

First Generation DR Migration

> **Caution:**
>
> If your systems are set up with a first-generation disaster recovery configuration, and you are migrating to the native disaster recovery service, perform these steps:

1. Gather the existing disaster recovery configuration details on both appliances, using the `drShowService` command.

```
PCA-ADMIN> drShowService
Data:
  Local Ip = 10.100.3.83/28
  Local Ip Perf = 10.100.3.84/28
  Remote Host = sn01-dr1.exmaple.com
  Remote Host Perf = sn02-dr1.exmaple.com
  Replication = ENABLED
  Replication High = ENABLED
  Message = Successfully retrieved site configuration
  maxConfig = 12
  gateway IP = 10.100.3.81
  gateway IP Perf = 10.100.3.81
  Job Retention Hours = 48
```

2. Remove the ZFS pool replication IPs from the existing network configuration.

```
PCA-ADMIN> edit networkConfig ZFSCapacityPoolReplicationEndpoint=""
PCA-ADMIN> edit networkConfig ZFSPerfPoolReplicationEndpoint=""
```

3. Use the storage IP addresses (and subnet mask) already assigned to the ZFS Storage Appliance Controllers for replication between the storage pools.

Creating the Local Endpoint

When you have obtained all required IP addresses, create the appliance local endpoint using either the Service CLI or Service Web UI.

- **Service CLI:**

  Enter the following command on a single line, replacing the sample IPs with the ones you obtained:

```
PCA-ADMIN> create LocalEndpoint \
spine1Ip=<10.212.128.3/29> datacenterGateway1Ip=<10.212.128.1> \
spine2Ip=<10.212.128.4/29> datacenterGateway2Ip=<10.212.128.2> \
```

```
zfsCapacityPoolEndpointIp=<10.212.128.129/29>
 zfsPerformancePoolEndpointIp=<10.212.128.130/29> \
localAsn=<136025>
```

Check the local endpoint configuration with the command: `getLocalEndpoint`.

- **Service Web UI:**

  Under Disaster Recovery Service, open the Local Endpoint page. In the top-right corner, select Create.

  In the pop-up window, enter the IP addresses in the respective fields. Select Create Local Endpoint to apply the settings.

  In the Local Endpoint page, the Information tab indicates the endpoint is configured. Select the Configuration tab to display the details.

Updating the Local Endpoint

You can change the IP addresses of the ZFS pool replication interfaces without deleting and recreating the entire local endpoint configuration or peer connection. Remember to update the corresponding DNS records in the data center.

To update the local endpoint:

- **Service CLI:**

  Enter the `updateLocalEndpoint` command and include the parameters you want to change.

  ```
  PCA-ADMIN> updateLocalEndpoint
   zfsCapacityPoolEndpointIp='10.212.128.241/26'
  ```

- **Service Web UI:**

  Under Disaster Recovery Service, open the Local Endpoint page. In the top-right corner, select Edit.

  In the pop-up window, enter the new ZFS pool IP addresses in the respective fields. Select Submit to apply the settings.

  In the Local Endpoint page, select the Configuration tab to confirm the updated parameters.

Deleting the Local Endpoint

The local endpoint cannot be deleted if it is part of an existing peer configuration. Delete the peer connection first.

To delete the local endpoint:

- **Service CLI:**

  Enter the command: `deleteLocalEndpoint`.

- **Service Web UI:**

  Under Disaster Recovery Service, open the Local Endpoint page. In the top-right corner, select Delete.

Completing the Peering Configuration

When two Private Cloud Appliance systems have been cabled correctly, and their local endpoints have been configured, the peer connection can be created.

The peer connection is a symmetrical configuration, meaning the setup must be performed on each connected appliance. The administrators exchange the relevant configuration details of their system, so they can each include the peer details required for creating the connection. A trust relationship between the appliances is established through a CA chain stored in the Secret Service (Vault).

When the first appliance completes its side of the connection setup, it goes into a waiting state. By design, the appliance with the IP address ending with the lowest value initiates the connection. As soon as the entry for the peer appliance is detected, the CA certificates are verified and the mutual trust relationship is confirmed. After successful peering, a pair of secure tunnels is established between the spine switches. These allow the administration services on the appliances to exchange information with each other.

These parameters are required to create a peer connection:

- the IP addresses (4 in total) of the local and the remote endpoint for each tunnel
- the IP addresses of the remote spine switches in the peer appliance
- properties of the peer appliance: domain name, system name, serial number, ASN ID if applicable
- properties of the peer Admin Service: host name, admin user name, admin password, CA chain

The network configuration must allow peer-to-peer connectivity between the replication endpoints, or use routable IPs when both systems are in separate address spaces. Ensure that the new network setup does not overlap with existing connections between the appliance and the data center.

A peer connection requires a /30 subnet, with 2 IPs assigned to each local endpoint. When setting up the connection, you include the netmask for the local endpoint IPs, but not for the remote endpoint IPs and remote spine switch IPs.

Creating the Peer Connection

When you have obtained all required parameters, create the peer connection.

- **Service CLI:**

  Enter the following command on a single line, replacing the sample parameters with the ones you obtained:

```
PCA-ADMIN> create PeerConnection name=<peerconnection1>
 description=<"my peer connection"> \
peerSerialNumber=<1654BF2465> peerSystemName=<mypca1>
 peerDomainName=<mydomain.com> \
localEndpoint1Ip=<172.16.21.1/30> remoteEndpoint1Ip=<172.16.21.2> \
localEndpoint2Ip=<172.16.21.5/30> remoteEndpoint2Ip=<172.16.21.6> \
remoteSpine1Ip=<10.212.128.10> remoteSpine2Ip=<10.212.128.11> \
peerAdminHostname=<mypca1.mydomain.com> peerAdminUserName=<admin>
 peerAdminPassword=<password> \
peerAdminCaChain=<ca_string>
remoteAsn=<136025>
```

  Check the peer connection configuration using the following commands:

```
PCA-ADMIN> list PeerConnection
Data:
  id                                      Name              Peer Admin
 Hostname    Peer Rack Serial Number   Lifecycle State
  --                                      ----
 -------------------    ----------------------   ---------------
  ocid1.drpeerconnection....unique_ID   peerconnection1
 mypca1.mydomain.com    1654BF2465                ACTIVE

PCA-ADMIN> show peerConnection
 id=ocid1.drpeerconnection....unique_ID
Data:
  Id = ocid1.drpeerconnection....unique_ID
  Type = PeerConnection
  Lifecycle Sub State = ACTIVE
  Lifecycle State = ACTIVE
  Peer Rack Serial Number = 1654BF2465
  Local Endpoint 1 Ip = 172.16.21.1/30
  Local Endpoint 2 Ip = 172.16.21.5/30
  Remote Endpoint 1 Ip = 172.16.21.2
  Remote Endpoint 2 Ip = 172.16.21.6
  Remote Spine 1 Ip = 10.212.128.10
  Remote Spine 2 Ip = 10.212.128.11
  Peer Admin CaChain = -----BEGIN CERTIFICATE-----
\nMIIFbjCCA1agAwIBAgIQfMPkn17+ZTNl/jZjYzbpn[...]
  Peer Admin Hostname = mypca1.mydomain.com
  Peer Rack Domain Name = mydomain.com
```

```
  Peer Rack System Name = mypca1
  Peer Rack Admin User Name = admin
  Peer Rack Admin User Password = *******
  Remote Asn = 136025
  ProgressRecordIds 1 = id:d39144d6-feef-4988-ba71-fac4b046fff8
 type:ProgressRecord  name:
  ProgressRecordIds 2 = id:940b397f-993c-4ab9-9708-909dabb65c47
 type:ProgressRecord  name:
  ProgressRecordIds 3 = id:64b31360-3d0d-4dc2-a925-35164143eb25
 type:ProgressRecord  name:
  ProgressRecordIds 4 = id:7e8d9e2e-74b1-4d31-9098-7a09d719ec6a
 type:ProgressRecord  name:
  ProgressRecordIds 5 = id:2309bcdc-1689-410b-a93e-528444ada2a5
 type:ProgressRecord  name:
  ProgressRecordIds 6 = id:8a4d5747-d8fe-48e2-96e2-f4c797964cbe
 type:ProgressRecord  name:
  Name = peerconnection1
  Work State = Normal
```

- **Service Web UI:**

  Under Disaster Recovery Service, open the Peer Connections page. In the top-right corner, click Create Peer Connection.

  In the pop-up window, enter all parameters in the respective fields. Click Create Peer Connection to apply the settings.

  In the Peer Connections page, the table displays a new entry for the connection you created. Click the name in the table to display the detail page of the peer connection, and review its configuration parameters.

Updating the Peer Connection

There is no CLI command or UI function to modify the peer connection once it's configured. Changing the peer connection requires that you delete it and create a new connection with the updated parameters.

Deleting the Peer Connection

If a peer connection is no longer used, you can delete it. Ensure that the peer configuration is removed from each connected appliance.

- **Service CLI:**

  Look up the ID of the peer connection you want to delete, then enter the delete command as shown.

```
PCA-ADMIN> list PeerConnection
Data:
  id                                      Name              Peer Admin
 Hostname    Peer Rack Serial Number   Lifecycle State
  --                                      ----
 ------------------    ----------------------   ---------------
  ocid1.drpeerconnection....unique_ID   peerconnection1
 mypca1.mydomain.com    1654BF2465                ACTIVE

PCA-ADMIN> delete peerConnection
 id=ocid1.drpeerconnection....unique_ID
```

- **Service Web UI:**

  Under Disaster Recovery Service, open the Peer Connections page. In the table, click the name of the connection you want to delete. The peer connection detail page is displayed. In the top-right corner, click Delete.

Configuring the Disaster Recovery Service

The Private Cloud Appliance native Disaster Recovery Service is configured on top of an active peer connection between two systems located at different sites. When the appliances have been peered, the administrator can set up the DR service.

The DR service is set up with a single command and the peer serial number as a parameter. The system verifies that replication targets are configured correctly, collects the required settings from the peer connection, and stores the information in a metadata file used for enabling the DR service. The setup must be performed on both peered appliances for the DR service to become active.

Using the Service CLI

Obtain the serial number of the peered appliance (`list PeerConnection`). To check if a DR service setup already exists, enter the `drNativeListService` command.

To enable the DR service over an existing peer connection, enter the setup command as follows:

```
PCA-ADMIN> drNativeSetupService peerSerialNumber=<peer_serial>
```

To verify the DR service details, enter the list and show commands as follows:

```
PCA-ADMIN> drNativeListService
Data:
  id           Peer Serial Number   Remote Host
  --           ------------------   -----------
  1654BF2465   1654BF2465           mydrhost1.mypca2.mydomain.com

PCA-ADMIN> drNativeShowService peerSerialNumber=1654BF2465
Data:
  Type = DrSiteConfig
  Local Ip = 192.168.8.17/28
  Remote Host = mydrhost1.mypca2.mydomain.com
  Replication = ENABLED
  Replication High = DISABLED
  Message = Successfully retrieved site configuration
  maxConfig = 20
  Job Retention Hours = 24
  Peer Serial Number = 1654BF2465
```

Using the Service Web UI

Under Disaster Recovery Service, open the Native Services page. In the top-right corner, click Setup Native Service.

In the pop-up window, enter the serial number of the peered appliance in the designated field. (Max Config and Job Retention Hours are optional parameters.) Click Setup Native Service to apply the settings.

Updating the Disaster Recovery Service

You can change certain properties of the DR service: the retention period for jobs, and the maximum number of DR configurations (up to 20). Ensure that the service configuration is identical on both peered appliances.

If you are migrating from first-generation DR to the native DR service, the same command is used to update the original metadata file so it can be used by the native DR service.

Using the Service CLI

Obtain the serial number of the peered appliance (`drNativeListService`). Enter the update command and new parameters on a single line as follows:

```
PCA-ADMIN> drNativeUpdateService peerSerialNumber=<peer_serial> \
maxConfig=<12> jobRetentionHours=<48>
```

Using the Service Web UI

Under Disaster Recovery Service, open the Native Services page. Click the DR service. In the top-right corner of the DR service detail page, click Edit. In the popup window, make the necessary changes and save them.

Disabling the Disaster Recovery Service

To deactivate disaster recovery, delete the DR service setup from both peered appliances.

Using the Service CLI

Obtain the serial number of the peered appliance (`drNativeListService`). Enter the delete command as follows:

```
PCA-ADMIN> drNativeDeleteService peerSerialNumber=<peer_serial>
```

Using the Service Web UI

Under Disaster Recovery Service, open the Native Services page. Click the DR service. In the top-right corner of the DR service detail page, click Delete.

Working with Disaster Recovery Configurations

Disaster recovery (DR) configurations specify the resources that play a critical role in protecting Private Cloud Appliance workloads against site-level incidents.

The resources in a DR configuration include compute instances with their associated block volumes, as well as the compartments they belong to and the network resources that provide their connectivity. Relevant network resources and compartment hierarchies must be set up on both the primary and standby rack, and associated with each other through site mappings.

Rules and Conditions

When populating DR configurations, respect the following rules regarding compute and storage resources.

- A compute instance must be stopped before it can be added to a DR configuration. There is one exception: when all volumes attached to the instance are also attached to one or more instances already included in the same DR configuration.
- A compute instance must be stopped before it can be removed from a DR configuration. There is one exception: when all volumes attached to the instance are also attached to one or more instances still included in the same DR configuration.
- All compute instances in a DR configuration must be stopped before the DR configuration can be deleted.
- A volume attached to a compute instance might be created from another source volume or volume backup. Such an instance (instance T) can be added to a DR configuration on condition that the source volume is not attached to any instance in any DR configuration. Note that *source volume* also refers to the volume used for the volume backup, and its direct or indirect source.

  Alternatively, the instance with the source volume attached can be added to a DR configuration on condition that *instance T* is not added to any DR configuration. Due to the volume source/target relationship, only one of the instances involved can be part of a DR configuration, not both.
- A DR configuration must be refreshed when configuration for any of its instances has changed, including changes for the instance and the attached storage or network resources. This is to ensure the instances on the standby rack after switchover or failover are started with needed attributes preserved.
- Refreshing a DR configuration results in a failure in case a volume ***and*** the source from which it was created, are both attached to one or more compute instances in any DR configuration.
- Instances in a DR configuration preserve the primary and secondary private IPs upon switchover or failover on condition that the subnet for the IP address is tagged appropriately. The subnet must be assigned a *freeform tag*, with the key `preserve_private_ips` and a value set to anything.

When this freeform tag is applied, any instance with a VNIC attached to this subnet preserves the associated primary and secondary IP addresses after switchover or failover. The precheck on the standby rack verifies whether the IP address is already in use, and fails if this is true.

Instructions

These sections explain how to build DR configurations:

- Creating and Maintaining Disaster Recovery Configurations on page 308
- Creating and Maintaining Site Mappings on page 311
- Adding and Removing Compute Instances on page 314

Creating and Maintaining Disaster Recovery Configurations

A DR configuration is the parent object to which you add Private Cloud Appliance cloud resources that you want to protect against system outages. You start by creating an empty DR configuration with an associated ZFS project on the ZFS Storage Appliance.

If your system contains optional high-performance disk shelves, two ZFS projects are created for each DR configuration: one in the standard pool and one in the high-performance pool. When you add instances to the DR configuration that have disks running on standard as well as high-performance storage, those storage resources are automatically added to the ZFS project in the appropriate pool.

Creating a DR Configuration

Using the Service CLI

1. Gather the information that you need to run the command:
   - a unique name for the DR configuration
   - a unique name for the associated ZFS storage project
2. Create an empty DR configuration with the `create DrConfig` command.

   Syntax (entered on a single line):

   ```
   create DrConfig
   configName=<DR_configuration_name>
   zfsProjectName=<ZFS_storage_project_name>
   ```

   Example:

   ```
   PCA-ADMIN> create DrConfig configName=mydrconf1
    zfsProjectName=mydrconf1-project
   JobId: 16047c38-1a7e-48ac-9588-297b915d49fc
   Data: DrConfig id: d3cd87de-afd4-4718-a3e6-1105b56b42d8.
    Successfully started job to create config mydrconf1
   ```

3. Use the job ID to check the status of the operation you started.

   ```
   PCA-ADMIN> show Job id=16047c38-1a7e-48ac-9588-297b915d49fc
   Data:
     Id = 16047c38-1a7e-48ac-9588-297b915d49fc
     Type = Job
     Associated Work Request Id = 20d20925-39d1-4f9a-beef-5e9d3ca465d5
     Done = true
     Name = CREATE_TYPE
     Progress Message = DrConfig id: d3cd87de-afd4-4718-
   a3e6-1105b56b42d8. Successfully created DR config mydrconf1:
    d3cd87de-afd4-4718-a3e6-1105b56b42d8
     Run State = Succeeded
     Transcript = Created job CREATE_TYPE
     Username = admin
   ```

```
  WorkItemIds 1 = id:e82eeec2-7881-45c5-ae84-6a3894ef66c6
 type:WorkItem  name:
```

4. When the DR configuration is created, the storage project for data replication is set up on the ZFS Storage Appliances. Note the DR configuration ID. You need it for all subsequent commands to modify the configuration.
5. To display a list of existing DR configurations, use the `drGetConfigs` command.

```
PCA-ADMIN> drGetConfigs
Data:
  id                                     Config Name
  --                                     -----------
  d3cd87de-afd4-4718-a3e6-1105b56b42d8   mydrconf1
  e8291afa-a413-4932-880a-abb8ac22c85d   mydrconf2
  7ad05d9f-731c-41b8-b477-35da4b999071   mydrconf3
```

6. To display the status and details of a DR configuration, use the `show DrConfig` command.

```
PCA-ADMIN> show DrConfig id=d3cd87de-afd4-4718-a3e6-1105b56b42d8
Data:
  Id = d3cd87de-afd4-4718-a3e6-1105b56b42d8
  Type = DrConfig
  Config State = Enabled
  Config Name = mydrconf1
  Config Id = d3cd87de-afd4-4718-a3e6-1105b56b42d8
  Zfs Project Name = mydrconf1-project
  Message = Successfully retrieved config data
  Message = Successfully retrieved config data
  Replica State = OK
  Replica Lag In Seconds = 3
```

Using the Service Web UI

1. Under Disaster Recovery Service, open the DR Configurations page. In the top-right corner, click Create Configuration.
2. In the Create Configuration window, enter the following information:
   - a unique name for the DR configuration
   - a unique name for the associated ZFS storage project
3. Click Submit. A new, empty DR configuration appears in the table.

   Next, you add site mappings and instances to the DR configuration.

About the DR Configuration State

The normal working state of a DR configuration is "Enabled". However, as part of the process of DR plan execution, the state of the DR configurations changes. In general, an administrator is not expected to manually change the state, but it might be useful, and it is possible to temporarily disable a DR configuration.

The following configuration states occur:

- **Enabled:** The DR configuration is in active working state. The mappings and instances can be modified, and DR plans can be modified and executed.
- **Disabled:** The DR configuration is locked. The mappings and instances cannot be modified, and DR plans cannot be created, edited, or executed.
- **Frozen:** The DR configuration is locked, like in the disabled state. It is no longer usable because it was migrated to the standby appliance.

Refreshing a DR Configuration

To ensure that the replication information stored in a DR configuration is updated with all the latest changes in your environment, you must refresh the DR configuration after making changes to the protected resources. Refreshing

the DR configuration is important for the continuation of service after a DR plan is executed, because it keeps the relevant metadata up-to-date with the current status of the resources under DR protection.

Using the Service CLI

1. Look up the ID of the DR configuration you want to refresh (`drGetConfigs`).
2. Refresh the data stored in the selected DR configuration with the `drRefreshConfig` command.

```
PCA-ADMIN> drRefreshConfig drConfigId=b51a51ac-043e-4b6d-
ab29-20f86d905e81
Data:
  Message = Successfully started job to refresh DR config
 b51a51ac-043e-4b6d-ab29-20f86d905e81
  Job Id = ed3a509e-4ae9-4314-9ec5-fb12ff78615a
```

3. Use the job ID to check the status of the operation you started.

```
PCA-ADMIN> drGetJob jobId=ed3a509e-4ae9-4314-9ec5-fb12ff78615a
Data:
  Type = refresh_config
  Job Id = ed3a509e-4ae9-4314-9ec5-fb12ff78615a
  Status = finished
  Result = success
  Message = job successfully retrieved
  Response = Successfully refreshed DR config [mydrconf1]
```

Using the Service Web UI

1. Under Disaster Recovery Service, open the DR Configurations page.
2. In the table, click the DR configuration to open its detail page.
3. In the top-right corner, select Controls, then click Refresh Configuration.
4. A DR job is started. When it completes successfully, the DR configuration has been refreshed.

   To track progress, under Disaster Recovery Service, select Jobs. The Jobs table reports the status of each job. Click a record in the table to display the job details.

Deleting a DR Configuration

When you no longer need a DR configuration, you can remove it with a single command. However, you must stop all compute instances that are part of the DR configuration before you can delete it. Deleting a DR configuration also removes all site mappings and cleans up the associated storage projects on the local ZFS Storage Appliance.

If required, the storage projects replicated on the standby ZFS Storage Appliance can be cleaned up manually. If the native DR service is disabled/deleted on both systems, all related storage project on both systems are cleaned up automatically as part of the process.

Using the Service CLI

1. Stop all the compute instances that are part of the DR configuration you want to delete.
2. Look up the ID of the DR configuration you want to delete (`drGetConfigs`).
3. Delete the selected DR configuration with the `drDeleteConfig` command.

```
PCA-ADMIN> delete DrConfig id=d3cd87de-afd4-4718-a3e6-1105b56b42d8
JobId: a687aafa-7c5c-4a68-99da-62e9384140be
Data: Successfully started job to delete config d3cd87de-afd4-4718-
a3e6-1105b56b42d8
```

4. Use the job ID to check the status of the operation you started.

```
PCA-ADMIN> show Job id=a687aafa-7c5c-4a68-99da-62e9384140be
Data:
  Id = a687aafa-7c5c-4a68-99da-62e9384140be
```

```
  Type = Job
  Associated Work Request Id = e5f77289-b263-4a5c-9163-7ce4629cc550
  Done = true
  Name = DELETE_TYPE
  Progress Message = Successfully deleted DR config [mydrconf1]
  Run State = Succeeded
  Transcript = Created job DELETE_TYPE
  Username = admin
  WorkItemIds 1 = id:622a4f37-9222-4ea4-af2c-351cf28e739b
 type:WorkItem  name:
```

Using the Service Web UI

1. Under Disaster Recovery Service, open the DR Configurations page. All configurations are displayed in the table.
2. In the Actions column, open the quick menu (3 dots) and click Delete.

   Alternatively, click the DR configuration to display its detail page. In the top-right corner, select Controls, then click Delete.
3. When prompted, click Confirm. A DR job is started. When it completes successfully, the DR configuration is removed from the table.

   To track progress, under Disaster Recovery Service, select Jobs. The Jobs table reports the status of each job. Click a record in the table to display the job details.

Creating and Maintaining Site Mappings

Site mappings determine how and where on the standby Private Cloud Appliance system the instances should be brought back up in case the primary system experiences an outage.

Each site mapping consists of a source object on the primary system and a corresponding target object on the standby system. Ensure that these resources exist on both systems before you add the site mappings to the DR configuration.

Adding Site Mappings

These are the site mapping types you can include in a DR configuration:

- Compartment: specifies that, if a failover or switchover occurs, instances from the source compartment must be brought up in the target compartment on the standby system
- Subnet: specifies that, if a failover or switchover occurs, instances connected to the source subnet must be connected to the target subnet on the standby system

Using the Service CLI

1. Gather the information that you need to run the command:
   - DR configuration ID (`drGetConfigs`)
   - Mapping source and target object OCIDs

     Use the Compute Enclave UI or CLI on the primary and standby system respectively.

     CLI commands:

     - `oci iam compartment list`
     - `oci network subnet list --compartment-id "ocid1.compartment.....uniqueID"`
2. Add a site mapping to the DR configuration with the `drAddSiteMapping` command.

   Syntax (entered on a single line):

```
drAddSiteMapping
drConfigId=<DR_configuration_id>
objType=[compartment | subnet]
sourceId=<source_object_OCID>
```

```
targetId=<target_object_OCID>
```

Examples:

```
PCA-ADMIN> drAddSiteMapping drConfigId=d3cd87de-afd4-4718-
a3e6-1105b56b42d8 objType=compartment \
sourceId=ocid1.compartment....<region1>....uniqueID
 targetId=ocid1.compartment....<region1>....uniqueID
Data:
  Message = Successfully added site mapping to DR config [mydrconf1]
  Returned Object Id = 77233b2c-aa4a-4314-8196-4d8dd1d20721

PCA-ADMIN> drAddSiteMapping drConfigId=d3cd87de-afd4-4718-
a3e6-1105b56b42d8 objType=subnet \
sourceId=ocid1.subnet....<region1>....uniqueID
 targetId=ocid1.subnet....<region2>....uniqueID
Data:
  Message = Successfully added site mapping to DR config [mydrconf1]
  Returned Object Id = 4467bb39-7256-4715-9a01-bedd25f73a82
```

3. Repeat the command with the OCIDs of all the source and target objects that you want to include in the site mappings of the DR configuration.
4. To display the list of site mappings included in the DR configuration, use the `drGetSiteMappings` command. The DR configuration ID is a required parameter.

   Syntax:

```
PCA-ADMIN> drGetSiteMappings drConfigId=d3cd87de-afd4-4718-
a3e6-1105b56b42d8
Data:
  id                                     Object Type
  --                                     -----------
  77233b2c-aa4a-4314-8196-4d8dd1d20721   compartment
  6ef9e3b6-6704-4260-b345-f55e364ba9e9   compartment
  4467bb39-7256-4715-9a01-bedd25f73a82   subnet
  096a5499-9720-4a4f-a8be-ce055f0efd5d   subnet
```

5. To display the status and details of a site mapping included in the DR configuration, use the `drGetSiteMapping` command. The DR configuration ID and site mapping ID are required parameters.

```
PCA-ADMIN> drGetSiteMapping drConfigId=d3cd87de-afd4-4718-
a3e6-1105b56b42d8 mappingId=4467bb39-7256-4715-9a01-bedd25f73a82
Data:
  Type = DrSiteMapping
  Mapping Id = 4467bb39-7256-4715-9a01-bedd25f73a82
  Object Type = subnet
  Source Id = ocid1.subnet1....<region1>....uniqueID
  Target Id = ocid1.subnet1....<region2>....uniqueID
  Message = Successfully retrieved mapping data
```

Using the Service Web UI

1. Collect the OCIDs of the source and target objects you are going to map. Use the Compute Enclave UI or CLI on the primary and standby system respectively. CLI commands:
   - `oci iam compartment list`
   - `oci network subnet list --compartment-id "ocid1.compartment.....uniqueID"`
2. Under Disaster Recovery Service, open the DR Configurations page. In the table, click the configuration to which you want to add site mappings. The DR Configuration detail page appears.
3. In the Resources section, click Site Mappings.

On the right hand side of the Site Mappings box, click Add Site Mapping.

4. In the Add Site Mapping window, enter the following information in the respective fields:

   - **Object Type:** Specify which resource type you are mapping. Enter either compartment or subnet.
   - **Target ID:** Enter the OCID of the target object on the remote system.
   - **Source ID:** Enter the OCID of the source object on the local system.

   Click Add Site Mapping. The new site mapping appears in the resources table.

   Optionally, click the site mapping to display its detail page.
5. Repeat these steps to add all the required site mappings.

   As cloud resources across the environment are added and removed over time, verify on a regular basis that the site mappings in the DR configuration are up-to-date.

Removing Site Mappings

You can remove a site mapping from the DR configuration if it is no longer required.

Using the Service CLI

1. Gather the information that you need to run the command:

   - DR configuration ID (`drGetConfigs`)
   - Site mapping ID (`drGetSiteMappings`)
2. Remove the selected site mapping from the DR configuration with the `drRemoveSiteMapping` command.

   Syntax (entered on a single line):

   ```
   drRemoveSiteMapping
   drConfigId=<DR_configuration_id>
   mappingId=<site_mapping_id>
   ```

   Example:

   ```
   PCA-ADMIN> drRemoveSiteMapping drConfigId=d3cd87de-afd4-4718-
   a3e6-1105b56b42d8 mappingId=1ce736ef-13f9-40a9-aa6b-3e8960261224
   Data:
     Message = Successfully removed mapping [1ce736ef-13f9-40a9-
   aa6b-3e8960261224] from DR config [mydrconf1]
   ```
3. Repeat the command with the IDs of all the site mappings that you want to remove from the DR configuration.

Using the Service Web UI

1. Under Disaster Recovery Service, open the DR Configurations page. In the table, click the configuration for which you want to edit site mappings. The DR Configuration detail page appears.
2. In the Resources section, click Site Mappings. All existing site mappings are displayed in the table.
3. In the Actions column, open the quick menu (3 dots) and click Remove Site Mapping.

   Alternatively, click the site mapping to display its detail page. In the top-right corner, click Remove Site Mapping.
4. Repeat until all obsolete site mappings have been removed.

   As cloud resources across the environment are added and removed over time, verify on a regular basis that the site mappings in the DR configuration are up-to-date.

Adding and Removing Compute Instances

When the relevant site mappings have been set up in the DR configuration, you add the required compute instances. Their data and disks are stored in the ZFS storage project associated with the DR configuration, and replicated over the peer connection between the Private Cloud Appliance systems.

Adding Instances

Compute instances must be stopped before you add them to a DR configuration. Instances can be added one at a time, per compartment, or all at once across all compartments.

> **Note:**
>
> In the Service Web UI, compute instances must be added one by one.

Using the Service CLI

1. Gather the information that you need to run the command:
   - DR configuration ID (`drGetConfigs`)
   - Instance OCIDs from the Service Enclave UI or CLI (`oci compute instance list --compartment-id <compartment_OCID>`)
2. Add compute instances to the DR configuration with the `drAddComputeInstance` command.

   Syntax (entered on a single line):

   ```
   drAddComputeInstance
   drConfigId=<DR_configuration_id>
   instanceId="<instance-1_OCID>","<instance-n_OCID>"
   compartmentId="<compartment-1_OCID>","<compartment-1_OCID>"
   all=True
   ```

   > **Note:**
   >
   > When selecting "all", the instance and compartment parameters cannot be included in the command.

   Examples:

   - individual instances: single instance or comma-separated list

     ```
     PCA-ADMIN> drAddComputeInstance drConfigId=d3cd87de-afd4-4718-
     a3e6-1105b56b42d8
      instanceId="ocid1.instance1....uniqueID","ocid1.instance2....uniqueID"
     Data:
       Message = Successfully started job to add instance to config
      d3cd87de-afd4-4718-a3e6-1105b56b42d8
       Job Id = dc0787ef-c254-4d91-b7dc-ca4d2412f461
     ```

   - all instances in a compartment: single compartment or comma-separated list

     ```
     PCA-ADMIN> drAddComputeInstance drConfigId=d3cd87de-afd4-4718-
     a3e6-1105b56b42d8
      compartmentId="ocid1.compartment1....uniqueID","ocid1.compartment2....uniquel
     Data:
       Message = Successfully started job to add instance to config
      d3cd87de-afd4-4718-a3e6-1105b56b42d8
       Job Id = 3801d0be-f2ab-48d5-ad12-cd8afd444bc2
     ```

   - all instances across all mapped compartments

     ```
     PCA-ADMIN> drAddComputeInstance drConfigId=d3cd87de-afd4-4718-
     a3e6-1105b56b42d8 all=True
     Data:
     ```

```
  Message = Successfully started job to add instance to config
 d3cd87de-afd4-4718-a3e6-1105b56b42d8
  Job Id = d0a11091-f202-4350-b59a-f0cddc9fb035
```

3. Use the job ID to check the status of the operation you started.

```
PCA-ADMIN> drGetJob jobId=3801d0be-f2ab-48d5-ad12-cd8afd444bc2
Data:
  Type = add_computeinstance
  Job Id = 3801d0be-f2ab-48d5-ad12-cd8afd444bc2
  Status = finished
  Result = success
  Message = job successfully retrieved
  Response = Successfully added the following instances to mydrconf1
 ['ocid1.instance1....uniqueID','ocid1.instance2....uniqueID']
```

4. Repeat the `drAddComputeInstance` command to add all the required compute instances to the DR configuration.
5. To display the list of instances included in the DR configuration, use the `drGetComputeInstances` command. The DR configuration ID is a required parameter.

```
PCA-ADMIN> drGetComputeInstances drConfigId=d3cd87de-afd4-4718-
a3e6-1105b56b42d8
Data:
  id
  --
  ocid1.instance1....uniqueID
  ocid1.instance2....uniqueID
  ocid1.instance3....uniqueID
  ocid1.instance4....uniqueID
```

6. To display the status and details of an instance included in the DR configuration, use the `drGetComputeInstance` command. The DR configuration ID and instance ID are required parameters.

```
PCA-ADMIN> drGetComputeInstance drConfigId=d3cd87de-afd4-4718-
a3e6-1105b56b42d8 instanceId=ocid1.instance1....uniqueID
Data:
  Compartment Id = ocid1.compartment....uniqueID
  Boot Volume Id = ocid1.bootvolume....uniqueID
  Compute Instance Shape = VM.PCAStandard1.Flex
  Message = Successfully retrieved instance data
```

Using the Service Web UI

1. Collect the OCIDs of the compute instances that need to be added to the DR configuration. Use the Service Enclave UI, or these CLI commands:
   - `oci iam compartment list`
   - `oci compute instance list --compartment-id <compartment_OCID>`
2. Under Disaster Recovery Service, open the DR Configurations page. In the table, click the configuration to which you want to add compute instances. The DR Configuration detail page appears.
3. In the Resources section, click Compute Instances.

   On the right hand side of the Compute Instances box, click Add Instance.
4. In the Add Compute Instances window, select compute instances using one of these methods:
   - By instance OCID

     Add a single instance by entering its OCID. Add multiple instances by entering their OCIDs as a comma-separated list.
   - By compartment OCID

Add all instances in a compartment by entering the compartment OCID. Add instances from multiple compartments by entering the compartment OCIDs as a comma-separated list.

- All

  Add all instances from all mapped compartments.

Click Submit. A DR job is started. When it completes successfully, the instances appear in the resources table.

To track progress, under Disaster Recovery Service, select Jobs. The Jobs table reports the status of each job. Click a record in the table to display the job details.

5. Check regularly for instances requiring DR protection, and repeat these steps whenever necessary.

   As cloud resources across the environment are added and removed over time, verify on a regular basis that the list of compute instances in the DR configuration is up-to-date.

Removing Instances

Instances can only be part of a single DR configuration. You can remove a compute instance from the DR configuration to which it was added.

Using the Service CLI

1. Gather the information that you need to run the command:
   - DR configuration ID (`drGetConfigs`)
   - Instance OCID (`drGetComputeInstances`)
2. Remove the selected compute instances from the DR configuration with the `drRemoveComputeInstance` command.

   Syntax (entered on a single line):

```
drRemoveComputeInstance
drConfigId=<DR_configuration_id>
instanceId=<instance_OCID>
compartmentId=<compartment_OCID>
all=True
```

> **Note:**
>
> When selecting "all", the instance and compartment parameters cannot be included in the command.

   Examples:

   - single instance

```
PCA-ADMIN> drRemoveComputeInstance drConfigId=d3cd87de-afd4-4718-
a3e6-1105b56b42d8 instanceId=ocid1.instance....uniqueID
Data:
  Message = Successfully started job to remove instance(s) from DR
 config d3cd87de-afd4-4718-a3e6-1105b56b42d8
  Job Id = 20f6735e-7702-4d27-ba19-41cf08a86d90
```

   - all instances in a compartment

```
PCA-ADMIN> drRemoveComputeInstance drConfigId=d3cd87de-afd4-4718-
a3e6-1105b56b42d8 compartmentId=ocid1.compartment....uniqueID
Data:
  Message = Successfully started job to remove instance(s) from DR
 config d3cd87de-afd4-4718-a3e6-1105b56b42d8
  Job Id = c4b0ff83-45e3-4498-98bb-e8a2f6b904e8
```

- all instances in the DR configuration

```
PCA-ADMIN> drRemoveComputeInstance drConfigId=d3cd87de-afd4-4718-
a3e6-1105b56b42d8 all=True
Data:
  Message = Successfully started job to remove instance(s) from DR
 config d3cd87de-afd4-4718-a3e6-1105b56b42d8
  Job Id = 0a2cdc62-1d13-4774-b3c6-d2a30b75db5e
```

3. Use the job ID to check the status of the operation you started.

```
PCA-ADMIN> drgetjob jobId=20f6735e-7702-4d27-ba19-41cf08a86d90
Data:
  Type = remove_computeinstance
  Job Id = 20f6735e-7702-4d27-ba19-41cf08a86d90
  Status = finished
  Result = success
  Message = job successfully retrieved
  Response = Successfully removed the following instances from
 mydrconf1 ['ocid1.instance....uniqueID']
```

4. Repeat the `drRemoveComputeInstance` command to remove all the required compute instances from the DR configuration.

Using the Service Web UI

1. Under Disaster Recovery Service, open the DR Configurations page. In the table, click the configuration from which you want to remove compute instances. The DR Configuration detail page appears.
2. In the Resources section, click Compute Instances. All instances included in the DR configuration are displayed in the table.
3. To quickly remove a single instance, in the Actions column, open the quick menu (3 dots) and click Delete. To remove multiple instances, proceed to the next step.
4. Open the Controls menu in the top-right corner of the DR Configuration detail page and click Remove Instances.
5. In the Remove Compute Instances window, select compute instances for removal using one of these methods:

   - By instance OCID

     Remove a single instance by entering its OCID. Remove multiple instances by entering their OCIDs as a comma-separated list.
   - By compartment OCID

     Remove all instances in a compartment by entering the compartment OCID. Remove instances included from multiple compartments by entering the compartment OCIDs as a comma-separated list.
   - All

     Remove all instances included from all mapped compartments.

   Click Submit. A DR job is started. When it completes successfully, the instances are removed from the resources table.

   To track progress, under Disaster Recovery Service, select Jobs. The Jobs table reports the status of each job. Click a record in the table to display the job details.
6. Check regularly for instances that no longer require DR protection, and repeat these steps whenever necessary.

   As cloud resources across the environment are added and removed over time, verify on a regular basis that the list of compute instances in the DR configuration is up-to-date.

Working with Disaster Recovery Plans

A disaster recovery (DR) plan describes the operations that must be performed on the Private Cloud Appliance resources that are under the protection of the disaster recovery service.

A DR plan is associated with a DR configuration, and is executed by an administrator either when a site-level incident is detected (failover), or when one of the sites must be taken offline (switchover). After a failover, when the affected system is back online, postfailover operations are performed to ensure that both systems are ready to run new DR operations.

These sections explain how to build and execute DR plans:

- Creating and Maintaining Disaster Recovery Plans on page 319
- Customizing the Steps in a Disaster Recovery Plan on page 323
- Executing a Disaster Recovery Plan on page 328

About DR Operations and Default Plans

The native DR service provides plans with default steps for each type of operation. DR plan steps can be customized. The built-in plans are configured as follows:

Switchover Plan

When a switchover is performed, there is no outage, so both peered systems are online. The goal is to move all resources covered in the DR configuration from the primary system (A) to the standby system (B). When completed, system B becomes the primary and system A the standby for the resources in question.

The plan starts with prechecks to ensure that both systems meet the requirements to allow compute instances to be stopped on the primary system and started again on the standby system. The prechecks include site mappings as well as other critical elements, such as tags, security lists, or network security groups. The role reversal precheck specifically ensures that the ZFS Storage Appliance in each rack is in the correct state.

When the prechecks are completed without errors, the DR configuration on the primary system (A) is frozen and its compute instances are stopped, so the role reversal can begin. Based on resource metadata exchanged between the peered systems, and replicated data on the standby ZFS Storage Appliance, the target system (B) is prepared to assume the primary role for the instances in the DR configuration. The replication process is reversed and ready to use the source system (A) as its standby as soon as the switchover is complete.

Using the replicated volumes, the compute instances in the DR configuration are launched on the standby system (B). An identical DR configuration is created on the standby system, with all source and target resources in the site mappings inverted. The metadata of the newly launched instances is stored in the DR configuration. On the primary system (A) a cleanup is performed: the DR configuration is disabled and its compute instances are terminated.

To complete the switchover, data replication from the new primary system (B) to the standby system (A) is started, the DR plans are moved to the new standby system (A), and the storage project and metadata associated with the original DR configuration are deleted from system A.

Failover Plan

A failover is performed on the standby system, when one of the peered systems goes down. The goal is to recover all resources covered in the DR configuration on the standby system (B), allowing continuation of service. The failover steps are similar to the switchover plan, but none of the operations on the primary system (A) can be performed. The primary system cannot be cleaned up until it comes back online.

The plan starts with prechecks to ensure that the standby system and its ZFS Storage Appliance are in the correct state to bring up the resources covered in the DR configuration. When the prechecks are completed without errors, the role reversal begins.

Using the replicated metadata and resources, the compute instances in the DR configuration are launched on the standby system (B), which assumes the primary role. An identical DR configuration is created on system B, which has become the primary, with inverted site mappings and metadata collected from the newly

launched instances. In preparation of the original primary system (A) coming back online, the replication process is reversed and ready to use system A as the standby.

When the original primary system (A) eventually comes online, the remaining steps to return the DR configuration to a correct working state are performed by executing the postfailover plan.

Postfailover Plan

A postfailover plan is performed after a failover, when the system that experienced an outage comes back online, and the peer connection is restored. The goal is to clean up the DR configuration on the primary system that went down (A), and set it up as the standby for the new primary system (B).

There are no prechecks in a postfailover plan. System A is back online after an outage and needs to be cleaned up: the DR configuration is disabled and its compute instances are terminated. Data replication from the new primary system (B) to the standby system (A) is started, the DR plans are moved to the new standby system (A), and the storage project and metadata associated with the original DR configuration are deleted from system A.

To move resources that were originally hosted on system A back from system B, the administrator must perform a switchover from B to A for the relevant DR configuration(s).

Creating and Maintaining Disaster Recovery Plans

Each DR plan defines the steps to perform on Private Cloud Appliance cloud resources when a DR operation is executed: failover, switchover, or postfailover.

Each DR configuration has an associated DR plan file. This file can contain multiple DR plans, each identified by a unique name assigned by the administrator during creation. The DR plan file is stored in JSON format in the directory `/mnt/dr_metadata` on the standby rack.

Creating a DR Plan

The DR service provides default plans for each standard operation. You create them with a simple command to get started. If necessary, you can customize the plan steps afterward.

Using the Service CLI

1. Look up the ID of the DR configuration for which you want to create a DR plan (`drGetConfigs`).
2. Create a default DR plan with the `create DrPlan` command.

   Typically, a DR configuration has associated DR plans for each operation type. In this example, plans are created for failover, switchover, and postfailover.

   ```
   PCA-ADMIN> create DrPlan drConfigId=6e797d8b-7245-4d49-8e68-
   bf67f2d53041 operation=SWITCHOVER planName=sw1
   JobId: eae66f69-7b99-420e-b324-7d8964b2202b
   Data: DrPlan id: 6e797d8b-7245-4d49-8e68-bf67f2d53041::sw1.
    Successfully started job for DR Plan Create for config_id
    6e797d8b-7245-4d49-8e68-bf67f2d53041

   PCA-ADMIN> create DrPlan drConfigId=6e797d8b-7245-4d49-8e68-
   bf67f2d53041 operation=FAILOVER planName=fo1

   PCA-ADMIN> create DrPlan drConfigId=6e797d8b-7245-4d49-8e68-
   bf67f2d53041 operation=POSTFAILOVER planName=pfo1
   ```

3. Use the job ID to check the status of the operation you started.

   ```
   PCA-ADMIN> show Job id=eae66f69-7b99-420e-b324-7d8964b2202b
   Data:
     Id = eae66f69-7b99-420e-b324-7d8964b2202b
     Type = Job
     Associated Work Request Id = ec0f39df-6256-4c4c-a839-0d00a8f326dc
     Done = true
     Name = CREATE_TYPE
   ```

```
  Progress Message = DrPlan id: 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1. Successfully created [sw1] plan on STANDBY for
 DR operation [switchover] for config [6e797d8b-7245-4d49-8e68-
bf67f2d53041]
  Run State = Succeeded
  Transcript = Created job CREATE_TYPE
  Username = admin
  WorkItemIds 1 = id:5ca6d187-e01a-40e2-bc97-3193a9a88742
 type:WorkItem  name:
```

4. To display a list of existing DR plans for a DR configuration, use the `drListPlan` command.

```
PCA-ADMIN> drListPlan drConfigId=6e797d8b-7245-4d49-8e68-
bf67f2d53041
Data:
  id                                          Plan Name   Operation
  --                                          ---------   ---------
  6e797d8b-7245-4d49-8e68-bf67f2d53041::sw1    sw1
 switchover
  6e797d8b-7245-4d49-8e68-bf67f2d53041::fo1    fo1         failover
  6e797d8b-7245-4d49-8e68-bf67f2d53041::pfo1   pfo1
 postfailover
```

5. To display the status and details of a DR plan, use the `show DrPlan` command.

```
PCA-ADMIN> show DrPlan id=6e797d8b-7245-4d49-8e68-bf67f2d53041::sw1
Data:
  Id = 6e797d8b-7245-4d49-8e68-bf67f2d53041::sw1
  Type = DrPlan
  Plan Id = 6e797d8b-7245-4d49-8e68-bf67f2d53041::sw1
  Plan Name = sw1
  Config Id = 6e797d8b-7245-4d49-8e68-bf67f2d53041
  Operation = switchover
  Steps 1 - Step Id = 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::precheck
  Steps 1 - Step Name = PRECHECK
  Steps 1 - Enabled = true
  Steps 1 - Last Status = norun
  Steps 1 - Command = None
  Steps 1 - Check Only = true
  Steps 2 - Step Id = 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::role_reversal_precheck
  Steps 2 - Step Name = ROLE_REVERSAL_PRECHECK
  Steps 2 - Enabled = true
  Steps 2 - Last Status = norun
  Steps 2 - Command = None
  Steps 2 - Check Only = true
  Steps 3 - Step Id = 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::stop_primary
  Steps 3 - Step Name = STOP_PRIMARY
  Steps 3 - Enabled = true
  Steps 3 - Last Status = norun
  Steps 3 - Command = None
  Steps 3 - Check Only = false
  Steps 4 - Step Id = 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::role_reversal
  Steps 4 - Step Name = ROLE_REVERSAL
  Steps 4 - Enabled = true
  Steps 4 - Last Status = norun
  Steps 4 - Command = None
  Steps 4 - Check Only = false
  Steps 5 - Step Id = 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::start_standby
  Steps 5 - Step Name = START_STANDBY
```

```
  Steps 5 - Enabled = true
  Steps 5 - Last Status = norun
  Steps 5 - Command = None
  Steps 5 - Check Only = false
  Steps 6 - Step Id = 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::cleanup_primary
  Steps 6 - Step Name = CLEANUP_PRIMARY
  Steps 6 - Enabled = true
  Steps 6 - Last Status = norun
  Steps 6 - Command = None
  Steps 6 - Check Only = false
  Steps 7 - Step Id = 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::post_config
  Steps 7 - Step Name = POST_CONFIG
  Steps 7 - Enabled = true
  Steps 7 - Last Status = norun
  Steps 7 - Command = None
  Steps 7 - Check Only = false
```

Using the Service Web UI

1. Under Disaster Recovery Service, open the DR Configurations page. In the table, click the configuration to which you want to add a DR plan. The DR Configuration detail page appears.
2. In the Resources section, click Plans.

   On the right hand side of the Plans box, click Add Plan.
3. In the Create DR Plan window, enter the following information in the respective fields:
   - **Plan Name:** Enter a name for this DR plan.
   - **Operation:** Select a standard operation: switchover, failover, postfailover.
   - **Steps:** Leave blank if you want the default steps for the selected operation. Otherwise, specify the steps to add to the DR plan. The options are: 'precheck', 'role_reversal_precheck', 'stop_primary', 'role_reversal', 'start_standby', 'cleanup_primary', 'post_config'.
   - **Dr Plan Data File:** Used for migrating first-generation DR plans. Leave blank.
   - **Configurations To Migrate:** Used for migrating first-generation DR plans. Leave blank.
4. Click Create DR Plan. A DR job is started. When it completes successfully, the DR plan appears in the resources table.

   To track progress, under Disaster Recovery Service, select Jobs. The Jobs table reports the status of each job. Click a record in the table to display the job details.
5. Repeat these steps to add all the required DR plans.
6. Optionally, click a DR plan name to display its detail page.

Changing a DR Plan

A DR plan has a complex structure with many parameters, which makes it difficult to update from the command line in particular. Instead, you can change the individual steps that make up the DR plan. See Customizing the Steps in a Disaster Recovery Plan on page 323.

Deleting a DR Plan

You can remove a DR plan if it is no longer required. To delete all plans associated with a DR configuration at once, use the command `drDeleteAllPlans`.

Using the Service CLI

1. Look up the ID of the DR plan you want to delete (`drListPlan`).
2. Remove the selected DR plan with the `delete DrPlan` command.

```
PCA-ADMIN> delete DrPlan id=6e797d8b-7245-4d49-8e68-
bf67f2d53041::fo2
```

```
JobId: 603d480f-1e0f-4229-b596-aaaf8588e682
Data: DrPlan id: 6e797d8b-7245-4d49-8e68-bf67f2d53041::fo2.
 Successfully started job for DR Plan delete for config_id
 6e797d8b-7245-4d49-8e68-bf67f2d53041
```

3. Use the job ID to check the status of the operation you started.

```
PCA-ADMIN> show Job id=603d480f-1e0f-4229-b596-aaaf8588e682
Data:
  Id = 603d480f-1e0f-4229-b596-aaaf8588e682
  Type = Job
  Associated Work Request Id = 391a0799-235a-4b26-aa99-4b5dd14ba19a
  Done = true
  Name = DELETE_TYPE
  Progress Message = DrPlan id: 6e797d8b-7245-4d49-8e68-
bf67f2d53041::fo2. Successfully deleted [fo2] plan on STANDBY for
 config [6e797d8b-7245-4d49-8e68-bf67f2d53041]
  Run State = Succeeded
  Transcript = Created job DELETE_TYPE
  Username = admin
  WorkItemIds 1 = id:86d0bd13-5f9c-4513-9404-60d8980b2243
 type:WorkItem  name:
```

Using the Service Web UI

1. Under Disaster Recovery Service, open the DR Configurations page. In the table, click the configuration for which you want to delete a DR plan. The DR Configuration detail page appears.
2. In the Resources section, click Plans. All existing DR plans are displayed in the table.
3. In the Actions column, open the quick menu (3 dots) and click Delete.

   Alternatively, click the DR plan name to display its detail page. In the top-right corner, click Delete.
4. When prompted, click Confirm. A DR job is started. When it completes successfully, the DR plan is removed from the table.

   To track progress, under Disaster Recovery Service, select Jobs. The Jobs table reports the status of each job. Click a record in the table to display the job details.
5. Repeat until all obsolete DR plans have been removed.

Deleting All DR Plans

Instead of deleting DR plans one by one, you can delete all plans associated with a DR configuration at once.

Using the Service CLI

1. Look up the ID of the DR configuration for which you want to delete all plans (`drGetConfigs`).
2. Remove the selected DR plan with the `drdeleteallPlans` command.

```
PCA-ADMIN> drdeleteallPlans drConfigId=6e797d8b-7245-4d49-8e68-
bf67f2d53041
JobId: b595dd62-8046-4ca7-90a0-dcbbf084e663
Data: DrPlan id: 6e797d8b-7245-4d49-8e68-bf67f2d53041::all_plans.
 Successfully started job for DR Plan delete for config_id
 6e797d8b-7245-4d49-8e68-bf67f2d53041
```

3. Use the job ID to check the status of the operation you started.

```
PCA-ADMIN> show Job id=b595dd62-8046-4ca7-90a0-dcbbf084e663
Data:
  Id = b595dd62-8046-4ca7-90a0-dcbbf084e663
  Type = Job
  Associated Work Request Id = d7cfb184-4e38-469d-b189-bb809386f5d4
  Done = true
  Name = DELETE_TYPE
```

```
  Progress Message = DrPlan id: 6e797d8b-7245-4d49-8e68-
bf67f2d53041::all_plans. Successfully deleted [all_plans] plan on
 STANDBY for config [6e797d8b-7245-4d49-8e68-bf67f2d53041]
  Run State = Succeeded
  Transcript = Created job DELETE_TYPE
  Username = admin
  WorkItemIds 1 = id:6cfec1c9-4a72-492d-a414-38bf4df6cf3a
 type:WorkItem  name:
```

Using the Service Web UI

1. Under Disaster Recovery Service, open the DR Configurations page. In the table, click the configuration for which you want to delete a DR plan. The DR Configuration detail page appears.
2. In the Resources section, click Plans. All existing DR plans are displayed in the table.

   Verify that all plans in the table should be deleted.
3. In the top-right corner, select Controls, then click Delete All Plans.
4. When prompted, click Confirm. A DR job is started. When it completes successfully, all DR plan are removed from the table.

   To track progress, under Disaster Recovery Service, select Jobs. The Jobs table reports the status of each job. Click a record in the table to display the job details.

Customizing the Steps in a Disaster Recovery Plan

Customizing the plan steps is the most convenient way to change a Private Cloud Appliance disaster recovery (DR) plan with default configuration. The administrator can enable or disable steps, configure steps to run in check-only mode, insert steps, and remove steps.

> **Note:**
>
> Custom steps and custom commands are not available in appliance software version 3.0.2-b1261765.

Changing the Properties of a DR Plan Step

You can enable or disable an existing DR plan step, and decide whether it should be performed in check-only mode.

Using the Service CLI

1. Look up the ID of the DR plan step you want to change (`show DrPlan`). If you don't have the DR plan ID, look it up using the commands `drGetConfigs` and `drListPlan`.
2. Change the DR plan step with the `edit DrPlanStep` command. These are the properties you can change in order to affect how a DR plan is executed:

   - `enabled=[True|False]` – determines whether this step is performed or not
   - `checkOnly=[True|False]` – determines whether this step is performed in check-only mode
   - `command=[string]` – (NOT available) specifies a custom command to be run as part of this step

     For standard DR plan operations, the command parameter must be set to `None`.

```
PCA-ADMIN> edit DrPlanStep id=6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::precheck checkOnly=False
JobId: d281141c-c388-490e-b038-239598488bc6
Data: DrPlanStep id: 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::precheck. Successfully started job for DR Plan
 Step update for config_id 6e797d8b-7245-4d49-8e68-bf67f2d53041
```

3. Use the job ID to check the status of the operation you started.

```
PCA-ADMIN> show Job id=d281141c-c388-490e-b038-239598488bc6
Data:
  Id = d281141c-c388-490e-b038-239598488bc6
  Type = Job
```

```
  Associated Work Request Id = 8f40d0a3-b2ac-4742-bed8-70beb26d669d
  Done = true
  Name = MODIFY_TYPE
  Progress Message = DrPlanStep id: 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::precheck. Successfully updated plan step
 [precheck] in DR Plan [sw1] for config [6e797d8b-7245-4d49-8e68-
bf67f2d53041]
  Transcript = Created job MODIFY_TYPE
  Username = admin
  WorkItemIds 1 = id:92a94a5e-4773-4ba8-ac4b-990496c5c2f9
 type:WorkItem  name:
```

4. To display the status and updated details of a DR plan step, use the `show DrPlanStep` command.

```
PCA-ADMIN> show DrPlanStep id=6e797d8b-7245-4d49-8e68-
bf67f2d53041::fo1::precheck
Data:
  Id = 6e797d8b-7245-4d49-8e68-bf67f2d53041::fo1::precheck
  Type = DrPlanStep
  Step Id = 6e797d8b-7245-4d49-8e68-bf67f2d53041::fo1::precheck
  Step Name = PRECHECK
  Config Id = 6e797d8b-7245-4d49-8e68-bf67f2d53041
  Plan Name = fo1
  Enabled = true
  Last Status = norun
  Command = None
  Check Only = false
```

Using the Service Web UI

1. Under Disaster Recovery Service, open the DR Configurations page. In the table, click the configuration for which you want to modify a DR plan. The DR Configuration detail page appears.
2. In the Resources section, click Plans. Click the name of the plan for which you want to edit the steps. The DR Plan detail page appears. The Resources section displays all steps in the plan.
3. In the Actions column, open the quick menu (3 dots) for the step you want to change, and click Edit.
4. In the Update DR Plan Step window, edit the properties as needed:
   - **Step Name:** Do not change the selected step.
   - **Enabled:** Select Yes or No to determine whether this step is performed or not during plan execution.
   - **Insert Type:** Do not change. This field is used only for adding a new step.
   - **Insert Location:** Do not change. This field is used only for adding a new step.
   - **Command:** (NOT available.) Specify a custom command to be run as part of this step.

     For standard DR plan operations, this field must remain empty. Custom commands only apply to custom DR plan steps.
   - **Check Only:** Select Yes or No to determine whether this step is performed or not in check-only mode.
5. Click Update DR Plan Step to apply your changes.

Inserting a DR Plan Step

Extra steps can be added to a DR plan.

Using the Service CLI

1. Gather the information that you need to run the command:
   - DR configuration ID (`drGetConfigs`)
   - DR plan name (`drListPlan`)
   - name of the step before or after which the new step must be inserted (`show DrPlan`)
2. Add a step to the DR plan with the `create DrPlanStep` command.

Note the mandatory and optional parameters. If the `insertType` and `insertLocation` parameters are not provided, the new step is added as the final step of the DR plan.

Syntax (entered on a single line):

```
create DrPlanStep
drConfigId=<DR_configuration_id>
planName=<DR_plan_name>
stepName=<DR_plan_new_step_name>
[optional:]
insertType=[BEFORE|AFTER]
insertLocation=<DR_plan_existing_step_name>
enabled=[True|False]
checkOnly=[True|False]
command=<execution_path>
```

> **Note:**
>
> Custom steps and custom commands are not available in appliance software version 3.0.2-b1261765.
>
> - Do not use the `command` parameter.
> - Select a step name from this list: `precheck, role_reversal_precheck, stop_primary, role_reversal, start_standby, cleanup_primary, post_config`.

Example:

```
PCA-ADMIN> create DrPlanStep drConfigId=6e797d8b-7245-4d49-8e68-
bf67f2d53041 planName=sw1 stepName=ROLE_REVERSAL insertType=BEFORE
 insertLocation=START_STANDBY
JobId: 7a162b6c-1ddc-410c-b27a-0996fb2d26df
Data: DrPlanStep id: 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::role_reversal. Successfully started job for DR
 Plan Step Create for config_id 6e797d8b-7245-4d49-8e68-bf67f2d53041
```

3. Use the job ID to check the status of the operation you started.

```
PCA-ADMIN> show Job id=7a162b6c-1ddc-410c-b27a-0996fb2d26df
Data:
  Id = 7a162b6c-1ddc-410c-b27a-0996fb2d26df
  Type = Job
  Associated Work Request Id = a4c76ef9-f91d-402e-986f-9795738fb429
  Done = true
  Name = CREATE_TYPE
  Progress Message = Successfully added step [role_reversal] in plan
 [sw1] for config [6e797d8b-7245-4d49-8e68-bf67f2d53041]
  Run State = Succeeded
  Transcript = Created job CREATE_TYPE
  Username = admin
  WorkItemIds 1 = id:5c376fae-f281-4f7a-984f-b996b19d367d
 type:WorkItem  name:
```

4. Repeat the `create DrPlanStep` command to add all the required steps to the DR plan.
5. To display the status and updated details of a DR plan, use the `show DrPlan` command.

```
PCA-ADMIN> show DrPlan id=6e797d8b-7245-4d49-8e68-bf67f2d53041::sw1
Data:
  Id = 6e797d8b-7245-4d49-8e68-bf67f2d53041::sw1
  Type = DrPlan
  Plan Id = 6e797d8b-7245-4d49-8e68-bf67f2d53041::sw1
```

```
  Plan Name = sw1
  Config Id = 6e797d8b-7245-4d49-8e68-bf67f2d53041
  Operation = switchover
  Steps 1 - Step Id = 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::precheck
  Steps 1 - Step Name = PRECHECK
  Steps 1 - Enabled = true
  Steps 1 - Last Status = norun
  Steps 1 - Command = None
  Steps 1 - Check Only = true
  Steps 2 - Step Id = 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::role_reversal_precheck
  Steps 2 - Step Name = ROLE_REVERSAL_PRECHECK
  Steps 2 - Enabled = true
  Steps 2 - Last Status = norun
  Steps 2 - Command = None
  Steps 2 - Check Only = true
  Steps 3 - Step Id = 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::stop_primary
  Steps 3 - Step Name = STOP_PRIMARY
  Steps 3 - Enabled = true
  Steps 3 - Last Status = norun
  Steps 3 - Command = None
  Steps 3 - Check Only = false
  Steps 4 - Step Id = 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::role_reversal
  Steps 4 - Step Name = ROLE_REVERSAL
  Steps 4 - Enabled = true
  Steps 4 - Last Status = norun
  Steps 4 - Command = None
  Steps 4 - Check Only = false
  Steps 5 - Step Id = 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::start_standby
  Steps 5 - Step Name = START_STANDBY
  Steps 5 - Enabled = true
  Steps 5 - Last Status = norun
  Steps 5 - Command = None
  Steps 5 - Check Only = false
  Steps 6 - Step Id = 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::cleanup_primary
  Steps 6 - Step Name = CLEANUP_PRIMARY
  Steps 6 - Enabled = true
  Steps 6 - Last Status = norun
  Steps 6 - Command = None
  Steps 6 - Check Only = false
  Steps 7 - Step Id = 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::post_config
  Steps 7 - Step Name = POST_CONFIG
  Steps 7 - Enabled = true
  Steps 7 - Last Status = norun
  Steps 7 - Command = None
  Steps 7 - Check Only = false
```

Using the Service Web UI

1. Under Disaster Recovery Service, open the DR Configurations page. In the table, click the configuration for which you want to modify a DR plan. The DR Configuration detail page appears.
2. In the Resources section, click Plans. Click the name of the plan to which you want to add one or more steps. The DR Plan detail page appears. The Resources section displays all steps in the plan.
3. In the Resources section of the DR Plan detail page, on the right hand side of the Steps box, click Add Step.
4. In the Add DR Plan Step window, enter the following information in the respective fields:

- **Step Name:** Select a standard step from the list.
- **Enabled:** Select Yes or No to determine whether this step is performed or not during plan execution.
- **Insert Type:** Select Before or After the insert location.
- **Insert Location:** Select the existing step before or after which you want the new step to be inserted.
- **Command:** (NOT available.) Specify a custom command to be run as part of this step.

  For standard DR plan operations, this field must remain empty.
- **Check Only:** Select Yes or No to determine whether this step is performed or not in check-only mode.

5. Click Add DR Plan Step to insert this new step in the selected location.
6. Repeat until all the required DR plan steps have been added.

Deleting a DR Plan Step

Steps can be removed from a DR plan.

Using the Service CLI

1. Look up the ID of the DR plan step you want to delete (`show DrPlan`). If you don't have the DR plan ID, look it up using the commands `drGetConfigs` and `drListPlan`.
2. Delete the DR plan step with the `delete DrPlanStep` command.

```
PCA-ADMIN> delete DrPlanStep id=6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::precheck
JobId: c75a4c71-0525-40b7-9618-c2a4e8fcb051
Data: DrPlanStep id: 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::precheck. Successfully started job for DR Plan
 Step delete for config_id 6e797d8b-7245-4d49-8e68-bf67f2d53041
```

3. Use the job ID to check the status of the operation you started.

```
PCA-ADMIN> show Job id=c75a4c71-0525-40b7-9618-c2a4e8fcb051
Data:
  Id = c75a4c71-0525-40b7-9618-c2a4e8fcb051
  Type = Job
  Associated Work Request Id = d7bd3873-e5a0-4437-b6c2-8ca0cce83953
  Done = true
  Name = DELETE_TYPE
  Progress Message = DrPlanStep id: 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::precheck. Successfully deleted step [precheck] in
 plan [sw1] for config [6e797d8b-7245-4d49-8e68-bf67f2d53041]
  Run State = Succeeded
  Transcript = Created job DELETE_TYPE
  Username = admin
  WorkItemIds 1 = id:b462883b-6e6e-49b6-b455-1ceb54a5e2e3
 type:WorkItem  name:
```

4. Repeat the `delete DrPlanStep` command to remove all the required steps from the DR plan.
5. To display the status and updated details of a DR plan, use the `show DrPlan` command.

```
PCA-ADMIN> show DrPlan id=6e797d8b-7245-4d49-8e68-bf67f2d53041::sw1
Data:
  Id = 6e797d8b-7245-4d49-8e68-bf67f2d53041::sw1
  Type = DrPlan
  Plan Id = 6e797d8b-7245-4d49-8e68-bf67f2d53041::sw1
  Plan Name = sw1
  Config Id = 6e797d8b-7245-4d49-8e68-bf67f2d53041
  Operation = switchover
  Steps 1 - Step Id = 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::stop_primary
  Steps 1 - Step Name = STOP_PRIMARY
```

```
  Steps 1 - Enabled = true
  Steps 1 - Last Status = norun
  Steps 1 - Command = None
  Steps 1 - Check Only = false
  Steps 2 - Step Id = 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::role_reversal
  Steps 2 - Step Name = ROLE_REVERSAL
  Steps 2 - Enabled = true
  Steps 2 - Last Status = norun
  Steps 2 - Command = None
  Steps 2 - Check Only = false
  Steps 3 - Step Id = 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::start_standby
  Steps 3 - Step Name = START_STANDBY
  Steps 3 - Enabled = true
  Steps 3 - Last Status = norun
  Steps 3 - Command = None
  Steps 3 - Check Only = false
  Steps 4 - Step Id = 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::cleanup_primary
  Steps 4 - Step Name = CLEANUP_PRIMARY
  Steps 4 - Enabled = true
  Steps 4 - Last Status = norun
  Steps 4 - Command = None
  Steps 4 - Check Only = false
  Steps 5 - Step Id = 6e797d8b-7245-4d49-8e68-
bf67f2d53041::sw1::post_config
  Steps 5 - Step Name = POST_CONFIG
  Steps 5 - Enabled = true
  Steps 5 - Last Status = norun
  Steps 5 - Command = None
  Steps 5 - Check Only = false
```

Using the Service Web UI

1. Under Disaster Recovery Service, open the DR Configurations page. In the table, click the configuration for which you want to modify a DR plan. The DR Configuration detail page appears.
2. In the Resources section, click Plans. Click the name of the plan for which you want to edit the steps. The DR Plan detail page appears. The Resources section displays all steps in the plan.
3. In the Actions column, open the quick menu (3 dots) for the step you want to remove, and click Delete.

   When prompted, click Confirm.
4. Repeat until all obsolete DR plan steps have been removed.

Executing a Disaster Recovery Plan

A disaster recovery (DR) plan can be executed from either the standby or the primary Private Cloud Appliance. However, a failover plan is always executed from the standby system., because the primary rack is down in this scenario.

A switchover can be performed for the purpose of testing the disaster recovery setup, or when extensive maintenance is required on the primary system. To return both appliances to their normal working state after a failover, a postfailover plan is executed on each system when the primary is back online. The switchover plan has postfailover steps built in, so it does not require an additional run of the postfailover plan.

As a result of executing a DR plan, resources are moved between peered systems and the primary system changes. Those resources are not automatically moved back to their original host system. To move resources back to their original environment, you must perform another switchover for the relevant DR configuration(s).

Performing a Switchover

A switchover allows the administrator to move resources away from a system so it can be taken offline, for example in case of planned maintenance. A (second) switchover is also performed to move resources back to their original host system, after they were impacted by a failover or switchover.

Using the Service CLI

1. Look up the ID of the switchover DR plan you want to execute. Use `drGetConfigs` to find the DR configuration, and display its associated DR plans using `drListPlan`.
2. From the primary or standby appliance, execute the switchover DR plan with the `drExecutePlan` command.

   > **Note:**
   >
   > To run the command in check-only mode, add the parameter `checkOnly=True`. Only the DR plan steps enabled for check-only mode will be performed.

   ```
   PCA-ADMIN> drExecutePlan planId=6e797d8b-7245-4d49-8e68-
   bf67f2d53041::sw1
   JobId: 92b4acc2-2dff-492c-9ba2-0a2ac058baa5
   Data: DrPlan id: 6e797d8b-7245-4d49-8e68-bf67f2d53041::sw1.
    Successfully started job for DR Plan Execute for config_id
    6e797d8b-7245-4d49-8e68-bf67f2d53041, plan_name sw1
   ```

3. Use the job ID to check the status of the operation you started.

   ```
   PCA-ADMIN> show Job id=92b4acc2-2dff-492c-9ba2-0a2ac058baa5
   Data:
     Id = 92b4acc2-2dff-492c-9ba2-0a2ac058baa5
     Type = Job
     Associated Work Request Id = c6cca56c-a1cc-421c-9ded-acf0e7cd9da2
     Done = false
     Name = OPERATION-EXECUTE_DR_PLAN
     Progress Message = DrPlan id: 6e797d8b-7245-4d49-8e68-
   bf67f2d53041::sw1. Successfully started job for DR Plan Execute for
    config_id 6e797d8b-7245-4d49-8e68-bf67f2d53041, plan_name sw1
     Run State = Active
     Transcript = Created job OPERATION
     Username = admin
     WorkItemIds 1 = id:e06881fc-ea57-4835-bb86-e1244d3787c3
    type:WorkItem  name:
   ```

4. Ensure that the job completes successfully.

   ```
   PCA-ADMIN> show Job id=92b4acc2-2dff-492c-9ba2-0a2ac058baa5
   Data:
     Id = 92b4acc2-2dff-492c-9ba2-0a2ac058baa5
     Type = Job
     Associated Work Request Id = c6cca56c-a1cc-421c-9ded-acf0e7cd9da2
     Done = true
     Name = OPERATION-EXECUTE_DR_PLAN
     Progress Message = DrPlan id: 6e797d8b-7245-4d49-8e68-
   bf67f2d53041::sw1. DrPlan id: 6e797d8b-7245-4d49-8e68-
   bf67f2d53041::sw1. drexecuteplan succeeded for config
    [6e797d8b-7245-4d49-8e68-bf67f2d53041] Operation: [switchover]
    plan_name: [sw1]. Response: [Successfully completed checks for
    switchover for DR config_id 6e797d8b-7245-4d49-8e68-bf67f2d53041.
    Plan Execution Status: [precheck : pass , role_reversal_precheck :
    pass , stop_primary : norun , role_reversal : norun ,
    start_standby : norun , cleanup_primary : norun , post_config :
    norun , ]]
   ```

```
  Run State = Succeeded
  Transcript = Created job OPERATION
  Username = admin
  WorkItemIds 1 = id:e06881fc-ea57-4835-bb86-e1244d3787c3
type:WorkItem  name:
```

After successful completion, all instances included in the DR configuration have been recovered and are running on the standby appliance.

Using the Service Web UI

1. Under Disaster Recovery Service, open the DR Configurations page. In the table, click the configuration for which you want to perform a switchover. The DR Configuration detail page appears.
2. In the Resources section, click Plans.
3. In the Actions column, open the quick menu (3 dots) for the switchover plan of your choice, and click Execute Plan.

   Alternatively, click the DR plan name to display its detail page. In the top-right corner, click Execute Plan.
4. When prompted, choose whether to execute the full plan or a subset of the steps in *check-only* mode.

   Click Confirm. A DR job is started. When it completes successfully, all steps in the switchover DR plan have been performed as expected.

   To track progress, under Disaster Recovery Service, select Jobs. The Jobs table reports the status of each job. Click a record in the table to display the job details.

   After successful completion, all instances included in the DR configuration have been recovered and are running on the standby appliance.

Performing a Failover

The native DR service does not provide automated failover. An administrator must confirm that the primary appliance is down, and execute the failover plan from the standby appliance. A failover is meant to allow continuation of service when the primary system experiences an outage.

When one appliance is down, the peer rack reports a fault with a name containing "peer connect" and the rack serial number. Use the Service CLI to check the fault list (`list fault <parameters>`) and display the details of the peer connection problem. For example:

```
PCA-ADMIN> show fault id=57701191-5764-480b-826c-38c4b1970dde
Data:
  Cause =  1742XC3024 : network is not in a CONNECTED state: CONNECTING
  Action = Please contact customer support for solution
  Health Exporter = peerconnect-checker
  Diagnosing Source = peer connect health checker
  Faulted Component Type = SOFTWARE
  Description = 1749XC302P-- 1742XC3024 : network is not in a CONNECTED
 state: CONNECTING
  Name = 1749XC302P--PCA-8000-UY--peerconnect
```

Using the Service CLI

1. Look up the ID of the failover DR plan you need to execute. Use `drGetConfigs` to find the DR configuration, and display its associated DR plans using `drListPlan`.
2. From the standby appliance, execute the failover DR plan with the `drExecutePlan` command.

> **Note:**
>
> To run the command in check-only mode, add the parameter `checkOnly=True`. Only the DR plan steps enabled for check-only mode will be performed.

```
PCA-ADMIN> drExecutePlan planId=6e797d8b-7245-4d49-8e68-
bf67f2d53041::fo1
JobId: 49521287-c148-4791-9626-13190fce3d1d
Data: DrPlan id: 6e797d8b-7245-4d49-8e68-bf67f2d53041::fo1.
 Successfully started job for DR Plan Execute for config_id
 6e797d8b-7245-4d49-8e68-bf67f2d53041, plan_name fo1
```

3. Use the job ID to check the status of the operation you started.

```
PCA-ADMIN> show Job id=49521287-c148-4791-9626-13190fce3d1d
Data:
  Id = 49521287-c148-4791-9626-13190fce3d1d
  Type = Job
  Associated Work Request Id = c8e3b554-a3ef-4e9b-a52c-c9a518f70974
  Done = false
  Name = OPERATION-EXECUTE_DR_PLAN
  Progress Message = DrPlan id: 6e797d8b-7245-4d49-8e68-
bf67f2d53041::fo1. Successfully started job for DR Plan Execute for
 config_id 6e797d8b-7245-4d49-8e68-bf67f2d53041, plan_name fo1
  Run State = Active
  Transcript = Created job OPERATION
  Username = admin
  WorkItemIds 1 = id:d7a09483-ef2e-4e03-81bb-fed5ee661428
 type:WorkItem  name:
```

4. Ensure that the job completes successfully.

```
PCA-ADMIN> show Job id=49521287-c148-4791-9626-13190fce3d1d
Data:
  Id = 49521287-c148-4791-9626-13190fce3d1d
  Type = Job
  Associated Work Request Id = c8e3b554-a3ef-4e9b-a52c-c9a518f70974
  Done = true
  Name = OPERATION-EXECUTE_DR_PLAN
  Progress Message = DrPlan id: 6e797d8b-7245-4d49-8e68-
bf67f2d53041::fo1. DrPlan id: 6e797d8b-7245-4d49-8e68-
bf67f2d53041::fo1. drexecuteplan succeeded for config
 [6e797d8b-7245-4d49-8e68-bf67f2d53041] Operation: [failover]
 plan_name: [fo1]. Response: [Successfully completed checks for
 failover for DR config_id 6e797d8b-7245-4d49-8e68-bf67f2d53041.
 Plan Execution Status: [precheck : pass , role_reversal_precheck :
 pass , role_reversal : pass , start_standby : pass , ]]
  Run State = Succeeded
  Transcript = Created job OPERATION
  Username = admin
  WorkItemIds 1 = id:d7a09483-ef2e-4e03-81bb-fed5ee661428
 type:WorkItem  name:
```

After successful completion, all instances included in the DR configuration have been recovered and are running on the standby appliance.

Using the Service Web UI

1. Under Disaster Recovery Service, open the DR Configurations page. In the table, click the configuration for which you want to perform a switchover. The DR Configuration detail page appears.
2. In the Resources section, click Plans.

3. In the Actions column, open the quick menu (3 dots) for the failover plan of your choice, and click Execute Plan.

   Alternatively, click the DR plan name to display its detail page. In the top-right corner, click Execute Plan.
4. When prompted, choose whether to execute the full plan or a subset of the steps in *check-only* mode.

   Click Confirm. A DR job is started. When it completes successfully, all steps in the switchover DR plan have been performed as expected.

   To track progress, under Disaster Recovery Service, select Jobs. The Jobs table reports the status of each job. Click a record in the table to display the job details.

   After successful completion, all instances included in the DR configuration have been recovered and are running on the standby appliance.

Performing Postfailover Operations

A postfailover is performed after a failover, when the system that experienced an outage comes back online. The plan can be executed from either of the peered systems. During postfailover, the DR configuration is cleaned up on the primary system that went down. The original standby system becomes the primary for the resources covered by the DR configuration, using the original primary as the new target for DR data replication.

Using the Service CLI

1. After a failover, confirm that the primary appliance is back online and in healthy condition.

   Ensure that the peering status is *active* and replication is *enabled*. Neither rack should report an active fault with a name containing "peer connect". (Check with Service CLI command `list fault`.)
2. Look up the ID of the postfailover DR plan you want to execute. Use `drGetConfigs` to find the DR configuration, and display its associated DR plans using `drListPlan`.
3. From the primary or standby appliance, execute the postfailover DR plan with the `drExecutePlan` command.

   > **Note:**
   >
   > For postfailover operations, the check-only mode does not apply.

```
PCA-ADMIN> drExecutePlan planId=6e797d8b-7245-4d49-8e68-
bf67f2d53041::pfo1
JobId: 56d040ba-30a6-4bea-b924-78ebabed2626
Data: DrPlan id: 6e797d8b-7245-4d49-8e68-bf67f2d53041::pfo1.
 Successfully started job for DR Plan Execute for config_id
 6e797d8b-7245-4d49-8e68-bf67f2d53041, plan_name pfo1
```

4. Use the job ID to check the status of the operation you started.

```
PCA-ADMIN> show Job id=56d040ba-30a6-4bea-b924-78ebabed2626
Data:
  Id = 56d040ba-30a6-4bea-b924-78ebabed2626
  Type = Job
  Associated Work Request Id = b4ad564b-e385-4688-94ff-11bf5267d72e
  Done = false
  Name = OPERATION-EXECUTE_DR_PLAN
  Progress Message = DrPlan id: 6e797d8b-7245-4d49-8e68-
bf67f2d53041::pfo1. Successfully started job for DR Plan Execute for
 config_id 6e797d8b-7245-4d49-8e68-bf67f2d53041, plan_name pfo1
  Run State = Active
  Transcript = Created job OPERATION
  Username = admin
  WorkItemIds 1 = id:2e4db010-239e-41a1-aa0d-cb97167c64fc
 type:WorkItem  name:
```

5. Ensure that the job completes successfully.

```
PCA-ADMIN> show Job id=56d040ba-30a6-4bea-b924-78ebabed2626
Data:
  Id = 56d040ba-30a6-4bea-b924-78ebabed2626
  Type = Job
  Associated Work Request Id = b4ad564b-e385-4688-94ff-11bf5267d72e
  Done = true
  Name = OPERATION-EXECUTE_DR_PLAN
  Progress Message = DrPlan id: 6e797d8b-7245-4d49-8e68-
bf67f2d53041::pfo1. DrPlan id: 6e797d8b-7245-4d49-8e68-
bf67f2d53041::pfo1. drexecuteplan succeeded for config
 [6e797d8b-7245-4d49-8e68-bf67f2d53041] Operation: [postfailover]
 plan_name: [pfo1]. Response: [Successfully completed checks for
 postfailover for DR config_id 6e797d8b-7245-4d49-8e68-bf67f2d53041.
 Plan Execution Status: [stop_primary : pass , cleanup_primary :
 pass , post_config : pass , ]]
  Run State = Succeeded
  Transcript = Created job OPERATION
  Username = admin
  WorkItemIds 1 = id:2e4db010-239e-41a1-aa0d-cb97167c64fc
 type:WorkItem  name:
```

After successful completion, all instances impacted by the switchover or failover have been restored and are running on the appliance where they were hosted before.

Using the Service Web UI

1. After a failover, confirm that the primary appliance is back online and in healthy condition.

   Ensure that the peering status is *active* and replication is *enabled*. Neither rack should report an active fault with a name containing "peer connect". (Display active faults in the Service CLI.)
2. Under Disaster Recovery Service, open the DR Configurations page. In the table, click the configuration for which you want to perform postfailover operations. The DR Configuration detail page appears.
3. In the Resources section, click Plans.
4. In the Actions column, open the quick menu (3 dots) for the postfailover plan of your choice, and click Execute Plan.

   Alternatively, click the DR plan name to display its detail page. In the top-right corner, click Execute Plan.
5. When prompted, click Confirm.

   > **Note:**
   >
   > For postfailover operations, the check-only mode does not apply.

   A DR job is started. When it completes successfully, all steps in the postfailover DR plan have been performed as expected.

   To track progress, under Disaster Recovery Service, select Jobs. The Jobs table reports the status of each job. Click a record in the table to display the job details.

   When the job has completed successfully, all instances impacted by the switchover or failover have been restored and are running on the appliance where they were hosted before.

Disaster Recovery Service and Peering Health Check

When the Disaster Recovery (DR) service is active, Private Cloud Appliance monitors the status of its local service and the health of the system it has a peer connection with.

Internally, the controller software ensures that the container hosting the DR service is running and functioning correctly. Using a Kubernetes liveness probe, messages are exchanged over the RabbitMQ bus at regular intervals to

confirm status. If the communication times out or the response indicates the container is unhealthy, Kubernetes tries to restart it.

To verify the operational status of the DR service, health checks are performed for the local service and the remote target. They report the status of the DR service, the replication status, and the presence of a peer connection. For an active peer connection, the health status of the remote DR service is also reported.

Data from DR service health checks is stored in Prometheus.

DR Metric	Description
`dr_health_status`	Status of the local DR service. Possible values are: • 0 = healthy • 1 = unhealthy • 2 = service not set up • 3 = peer connection not enabled
`dr_peer_health_status`	Status of the DR service on the peer system. Possible values are: • 0 = healthy • 1 = unhealthy • 2 = service not set up • 3 = peer connection not enabled

Disaster Recovery Service Metrics

The native Disaster Recovery (DR) service provides metric data in a different way compared to other Private Cloud Appliance microservices. No DR metrics are exposed through Grafana by default, but they appear after the service setup is complete.

A prebuilt Grafana dashboard is not provided. If you prefer to display DR metrics in a dashboard you can build one that suits your requirements. For more information, see #unique_118. The DR metrics are available from the Grafana Explore module. Using the Prometheus data source, you open the Metrics drop-down and select "*dr*" to display the list.

DR Metric	Available after...	Description
`dr_replication`	Native DR service setup	indicates that the DR service is enabled (value=1) and displays target details
`dr_job_retention_hours`	Native DR service setup	indicates how long DR jobs are stored
`dr_max_config`	Native DR service setup	indicates the maximum number of DR configurations
`dr_peering_status`	Native DR service setup	indicates that a peer connection with another appliance is enabled (value=1)
`dr_replication_sync_status`	Native DR service setup	indicates whether data sync between peered systems is successful (value=0) or not (value=1)
`dr_health_status`	Native DR service setup	indicates whether the local DR service is healthy (value=0)

DR Metric	Available after...	Description
`dr_configcount`	creating DR configurations	indicates the total number of DR configurations present
`dr_instances`	creating DR configurations	indicates how many instances are included in each DR configuration
`dr_operation_success_count`	running DR plan operations	indicates the total number of successful DR operations per DR plan
`dr_operation_fail_count`	running DR plan operations	indicates the total number of failed DR operations per DR plan

Note:

For existing DR configurations, metrics are not generated immediately after the appliance software upgrade that enables the functionality. They appear when a DR operation has been run, or when a new DR configuration is added. This also applies to a DR setup that was migrated to the native DR service.

Upgrading the Controller Software on Private Cloud Appliance

Upgrading Private Cloud Appliance is the responsibility of the appliance administrator. The system provides a framework to verify upgrade readiness in advance, and to perform an upgrade as one workflow orchestrating all individual tasks in the correct order with progress tracking.

The source content for upgrades – packages, archives, deployment charts and so on – is delivered through ISO images. During the preparation of the upgrade environment, the ISO images are downloaded to shared storage, and the upgrader itself is upgraded to the latest version included in the ISO content. At this stage the appliance is ready for upgrades to be applied.

The administrator can perform an upgrade either through the Service Web UI or the Service CLI. In general, a full rack upgrade is performed because it conveniently integrates all steps into a single workflow. In certain situations it might be advisable to upgrade in phases, work through components in groups, or even target an individual component. Those options are also available.

Upgrade Plan and History

During the preparation stages of the upgrade environment, the metadata file with the target component versions is compared to the current installation, and the result of this metadata comparison is recorded in a detailed upgrade plan. For each item, the plan contains the current and target versions and builds, and whether upgrade is required or not. If an item needs upgrading, the plan indicates which infrastructure components are impacted, whether a reboot is required, and how much time the upgrade takes.

Every reboot can affect the performance and availability of the appliance. For minimum risk and optimal use of time, the logic to populate the upgrade plan includes detailed evaluation of new packages. Many packages can be upgraded without a reboot or only require a service restart. The reboot flag in the upgrade plan is set only if a component absolutely must be rebooted.

The upgrade plan serves as a checklist that drives all upgrade procedures in a prescribed order. An upgrade job is started for every upgrade command, but if the plan indicates that a component is already at the required version, no further action is taken and the upgrade is skipped. However, a same-version upgrade can be forced if necessary.

As the upgrade operations progress, the upgrade plan is updated to reflect the current status, so that an administrator can check at any time which components have already been upgraded or still need to be upgraded. When current and target versions are identical for each item in the upgrade plan, the entire system is up-to-date.

At any time during or after an upgrade, the administrator can use the job framework to verify the status of requests and jobs. An upgrade command corresponds with an upgrade request, which triggers a workflow that might contain several upgrade jobs. Jobs, in turn, consist of a series of tasks. Details about all these elements can be retrieved using either the Service Web UI or the Service CLI. While the upgrade plan shows the status at a point in time, the job framework lets an administrator view status and history by drilling down into the details of Upgrader activity, which also makes troubleshooting easier.

For the purpose of long-term tracking, metadata files and upgrade plans are saved, while detailed information about all component upgrades and patches is captured in upgrade jobs. The upgrade history feature allows an administrator to drill down into the upgrade and patching activity on the appliance. The upgrade history presents all upgrade and patch information in a categorized way so you can see which version upgrades have been performed, which jobs have been run for each of those upgrades, and from which source (ISO upgrade or ULN patch). Details include build versions, component versions before and after, job completion, success or failure, time stamps, and duration.

> **Note:**
>
> Only ISO upgrades are possible from software version 3.0.2-b1261765. ULN based patching has been deprecated. However, if the system was patched in the past, those records appear in the upgrade history.

Pre-Checks

All upgrade operations are preceded by a verification process to ensure that system components are in the correct state to be upgraded. For these pre-checks, the upgrade mechanism relies on the platform-level health checks. Even though health checks are executed continually for monitoring purposes, they must be run specifically before an upgrade operation. The administrator is not required to run the pre-checks manually; they are executed by the upgrade code when an upgrade command is entered. All checks must pass for an upgrade to be allowed to start, even if a single-component upgrade is selected.

Prerequisite software versions are enforced in releases from January 2024 onward. During the upgrade or patch preparations, the Upgrader service validates the currently installed appliance software version against the new target version. If the appliance is not running at least the minimum required version, the workflow rolls back the environment to its previous state. You must first install the minimum required version, before proceeding with the intended target version.

Certain upgrade procedures require that the administrator first sets a provisioning lock and maintenance lock. While the locks are active, no provisioning operations or other conflicting activity can occur, meaning the upgrade process is protected against potential disruptions. After the upgrade has completed, the maintenance and provisioning locks must be released so the system returns to full operational mode.

In large composite workflows, like the full rack upgrade and the upgrade of all compute nodes, locks are applied and released programmatically. Administrators do not need to lock and unlock components manually, but they must monitor the upgrade workflow for potential interruptions, and take corrective action if issues occur with node locking, compute instance migrations, and so on.

Full Rack Upgrade

The Upgrader Workflow Service (UWS) adds an orchestration layer to the upgrade process. It brings the modular component upgrades together in an integrated workflow, allowing the administrator to start an upgrade of the entire appliance with a single command. The upgrade logic of component selection, order, and timing, is determined by the upgrade plan. The workflow manages the operations required to complete the upgrade plan by generating requests and monitoring the results of the associated jobs.

The workflow-based approach is introduced in the controller software version that migrates the full management cluster to Oracle Linux 8. The full rack upgrade is the preferred way to upgrade Private Cloud Appliance with all

releases from April 2025 onward. In case of unforeseen orchestration problems or specific component requirements, Oracle will provide guidance on how to upgrade components in groups or individually.

Component order is determined by the upgrade plan for the specific release. The rack upgrade workflow includes the these components:

1. ZFS Storage Appliance firmware (including ILOMs of storage controllers)
2. Compute nodes
3. Management nodes
   - Host operating system of the 3 nodes
   - MySQL cluster database
   - Secret service (including Etcd and Vault)
   - Kubernetes container orchestration packages (platform layer)
   - Containerized microservices
4. Oracle Cloud Infrastructure images
5. ILOM firmware (all nodes)
6. Switch firmware (all switches)

Component Upgrades

Private Cloud Appliance upgrades are designed to be modular, enabling individual components to be upgraded without service interruptions in the operational environment. Oracle strongly recommends using the full rack upgrade workflow, but procedures for subsets and individual components are available for particular scenarios.

- **ZFS Storage Appliance firmware**

  Use this option to upgrade the operating software on the ZFS Storage Appliance. Both controllers, which operate in an active-active cluster configuration, are upgraded as part of the same process. If a new ILOM firmware is available, it is also upgraded on both controllers in the process.
- **Compute node**

  Use this option to install the latest Oracle Linux kernel and user space packages, as well as appliance-specific tools and optimizations. You can upgrade a single node by providing its host IP address, or run a workflow that upgrades all compute nodes installed in the appliance. There can be no concurrent upgrades; the nodes are locked and upgraded one by one.
- **Management cluster**

  Use this option to upgrade the host OS and the services running on all three management nodes, including the clustered MySQL database and Secret service (etcd/Vault). The full management node cluster upgrade integrates multiple component upgrades into a global workflow, so all jobs are performed in a predefined order. With a single command, all three management nodes in the cluster are upgraded sequentially and component by component. This means an upgrade of a given component is performed on each of the three management nodes before the global workflow moves to the next component upgrade.
- **Host operating system**

  Use this option to upgrade the Oracle Linux operating system on a management node. You can upgrade the host OS of a single node by providing its host IP address, or run a workflow that upgrades the host OS sequentially on all three management nodes.
- **Kubernetes cluster**

  Use this option to upgrade the Kubernetes cluster, which is the container orchestration environment where services are deployed. The Kubernetes cluster runs on all the management nodes and compute nodes; its upgrade involves three major operations:

  - Upgrading the Kubernetes packages and all dependencies: kubeadm, kubelet, kubectl and so on.
  - Upgrading the Kubernetes container images: kube-apiserver, kube-controller-manager, kube-proxy and so on.
  - Updating any deprecated Kubernetes APIs and services YAML manifest files.
- **Platform services**

Use this option to upgrade the containerized services running within the Kubernetes cluster on the management nodes. The service upgrade mechanism is based on Helm, the Kubernetes equivalent of a package manager. For services that need to be upgraded, new container images and Helm deployment charts are delivered through an ISO image and uploaded to the internal registry. None of the operations up to this point have an effect on the operational environment.

At the platform level, an upgrade is triggered by restarting the pods that run the services. The new deployment charts are detected, causing the pods to retrieve the new container image when they restart. If a problem is found, a service can be rolled back to the previous working version of the image.

> **Note:**
>
> In specific circumstances it is possible to upgrade certain platform services individually, by adding an optional JSON string to the command. This option should not be used unless Oracle provides explicit instructions to do so.

- **ILOM firmware**

  Use this option to upgrade the Oracle Integrated Lights Out Manager (ILOM) firmware of a component installed in the appliance. You can upgrade a single ILOM by providing its IP address, or run a workflow that upgrades all ILOMs of all appliance components. After the firmware is upgraded successfully, the ILOM is automatically rebooted. However, the administrator must manually restart the server for all changes to take effect.

- **Switch firmware**

  Use this option to upgrade the operating software of the switches. You can specify a switch category to upgrade: the leaf switches, the spine switches, or the management switch. You can also run a workflow that upgrades all switches installed in the appliance.

- **Oracle-provided images**

  Use this option to add the latest available Oracle Cloud Infrastructure Images for Private Cloud Appliance. Existing images that are no longer needed, must be removed manually.

Important Notes for Private Cloud Appliance Software Upgrade

Read and understand this important information before preparing the upgrade of Private Cloud Appliance.

Do Not Modify Software Installation

Do not install or upgrade individual packages on the appliance components. Only upgrades as described in this chapter are supported. If security and other updates must be applied outside of the upgrade workflow, explicit instructions will be provided by Oracle

Patching Is Deprecated

Patching is separate from the upgrade functionality and uses a ULN mirror to download supported packages to the shared storage on the management nodes. Legacy documentation is still available for systems running older software. See Oracle Private Cloud Appliance Patching Guide.

As of software version 3.0.2-b1261765, systems must be upgraded using ISO images. The upgrade to version 3.0.2-b1443573 migrates the system from Oracle Linux 7 to Oracle Linux 8. New packages will no longer be made available through Private Cloud Appliance ULN channels.

Minimum Recommended Version

In these upgrade instructions it is assumed that your system is running at least the minimum recommended appliance software version: 3.0.2-b1261765. The main topics provide detailed instructions to move your system on to the latest available version.

If your active Private Cloud Appliance software is older than the minimum recommended version, please follow the guidelines and instructions in the legacy Private Cloud Appliance Upgrade Guide: Upgrading from Earlier Software Versions. When the appliance is running at least the minimum recommended version, follow the instructions here.

Upgrade Strategy

We recommend to run the latest available software on your Private Cloud Appliance. It improves protection against vulnerabilities and allows you to take advantage of all new features, bug fixes, and functional improvements.

The latest Upgrader code automatically enforces prerequisite software versions. During the upgrade or patch preparations, the Upgrader service validates the currently installed appliance software version against the new target version. If the appliance is not running at least the minimum required version, the Upgrader exits the process and rolls back the environment to its previous state. You must first install the prerequisite version as indicated in the log.

Upgrade Order

Components must be upgraded in a prescribed order. The upgrade plan helps manage the order of upgrade operations. For more information, see Checking Status and Progress of Private Cloud Appliance Upgrade Plan on page 357.

Oracle-Provided Images

At the end of the full management node cluster upgrade procedure, the Oracle Cloud Infrastructure images are automatically updated. The management node upgrade procedure includes the steps described in Upgrading Oracle Cloud Infrastructure Images on page 384 so they do not need to be performed separately.

Dedicated GPG Key

As of appliance software version 3.0.2-b1483396 the Private Cloud Appliance RPM packages are signed with a new GPG key. Previously the system relied on the Oracle Linux signing servers and keys for the appliance-specific packages. The new signing key is added to the RPM database during software upgrade and is expected to work seamlessly.

Update Custom CA Certificates

If your Private Cloud Appliance has been configured to use your own CA certificates for secure connection to the external appliance interfaces, you might need to update them after upgrading to a new software version. If the new software version changes the endpoints, for example because a new cloud service was added, the current certificate cannot validate connections to new endpoints. To resolve this, you generate new certificate signing requests (CSR) when the new appliance software is active, ask your Certificate Authority to provide new signed certificates, and upload those to the appliance. All current endpoints are now included in the updated certificates.

For more information about using your own CA certificates, see Accessing External Interfaces of Private Cloud Appliance with Your Own CA Trust Chain on page 247.

Preparing for Private Cloud Appliance Upgrade

During the preparation phase you ensure that a backup exists for vital Private Cloud Appliance system components, and that new software sources are ready to be deployed.

> **Caution:**
>
> When upgrading or patching Private Cloud Appliance, ensure that no other Service Enclave operations are ongoing or started. The upgrade or patching process might be adversely affected by appliance administration activities, including but not limited to: compute node operations (such as provisioning or reboot), operations on the management node cluster, configuration changes in any hardware or software component, rack network connectivity changes, password changes, and so on.
>
> Appliance administrators must always use the supported interfaces: the Service CLI and Service Web UI. Performing operations not provided through the API, from the Linux command line and/or with elevated access

privileges, is not supported, unless such explicit instructions are provided by Oracle.

Caution:

The granular appliance architecture with built-in redundancy allows administrators to upgrade or patch components without downtime. However, resource capacity and performance might be reduced while an upgrade or patch workflow is in progress.

We recommend that administrators responsible for upgrade or patching notify all Compute Enclave users in advance about such planned maintenance operations.

This is particularly important for users of theKubernetes Engine (OKE), because new cluster deployments are not allowed during the maintenance window, and some types of application clusters might experience service interruptions.

My Oracle Support

Software versions and upgrades for Private Cloud Appliance are made available for download through My Oracle Support. All files required to upgrade the appliance hardware and software components to a given release are packaged into one or more ISO image files. All the items within the ISOs of a given version have been tested to work with each other and qualified for installation on your rack system.

ISO Naming

The number of ISOs and file naming differs depending on the appliance software version you download. In recent versions, the Oracle Cloud Infrastructure platform images are packaged in a secondary ISO.

- **Version 3.0.2-b1325160 and earlier:** single ISO

  File name: `pca-3.0.2-<build_no>.iso`
- **Version 3.0.2-b1392231 and later:** two ISOs

  File names: `pca-3.0.2-<build_no>.iso` and `pca-3.0.2-oci-images-<build_no>.iso`

To be able to upgrade your appliance from the ISOs, you need to download the image files to a location from where a web server can make them available to the Private Cloud Appliance management nodes. If multiple ISOs are associated with a release, store them together. Upgrade commands take the primary ISO path as a parameter; it is implied that a secondary ISO is located in the same directory.

If you have set up a bastion host connected to the internal administration network of the appliance, it is convenient to store the ISO images on that machine and run a web server to make the ISOs accessible over HTTP.

Before the Upgrade

Before you start an upgrade procedure, ensure that you have performed these steps:

1. Download the new software ISO image files to a suitable location.
2. Verify that you have the necessary permissions to perform an upgrade.
3. Back up the current system configuration.
4. Preconfigure the upgrade environment.
5. Set up the new appliance software sources to run component upgrade procedures.

Tools are provided to verify the status of the appliance before, during, and after an upgrade. The upgrade workflows follow the upgrade plan, which you can consult at any time to track progress.

Verifying Private Cloud Appliance Administrator Permissions

To run upgrade workflows, you must have an administrator account to log in to the Service Enclave. You must be a member of one of these authorization groups: SuperAdmin, Admin, or DR Admin.

For more information about administrator access, see #unique_84.

When you log in to the Service CLI, you can verify that the upgrade and patch commands are available to you by displaying all custom commands, or custom commands in these object categories: `UpgradeRequest`, `PatchRequest`, `UpgradeJob`, `UpgradeJobList`. The list of commands is filtered based on your access profile. If the upgrade and patch commands are listed, it means you have permission to run them.

```
PCA-ADMIN> showallcustomcmds
    Operation Name: <Related Object(s)>
    ----------------------------------
    getUpgradeHistory:  UpgradeRequest
    getUpgradeHistoryDetails:  UpgradeRequest
    getUpgradeJob:  UpgradeJob
    getUpgradeJobs:  UpgradeJobList
    getUpgradePlan:  UpgradeRequest
    getUpgradeRequests:  UpgradeRequest
    getUpstreamUlnChannel:  PatchRequest
    getUpstreamUlnChannels:  PatchRequest
[...]

PCA-ADMIN> showcustomcmds PatchRequest
    patchCN
    patchIlom
    patchOCIImages
    patchSwitch
[...]
    setUpstreamUlnMirror
    syncUpstreamUlnMirror
    getUpstreamUlnChannels
    getUpstreamUlnChannel
    addUpstreamUlnChannel
    removeUpstreamUlnChannel

PCA-ADMIN> showcustomcmds UpgradeJob
    getUpgradeJob
    killUpgradeJob

PCA-ADMIN> showcustomcmds UpgradeJobList
    getUpgradeJobs

PCA-ADMIN> showcustomcmds UpgradeRequest
    getUpgradeRequests
    upgradeFullMN
    upgradeCN
    upgradeIlom
    upgradeSwitch
[...]
```

Creating a Backup Before a Private Cloud Appliance Upgrade

Before upgrading any component of Private Cloud Appliance, create a backup of the latest state of the system database, storage appliance, and Vault.

For system-critical components and services, Private Cloud Appliance runs a scheduled backup service that allows the appliance to be restored to its last known healthy state in case of a catastrophic failure. It is recommended that you implement a backup strategy for the users' cloud resources in the Compute Enclave as well.

Before upgrading or patching any component of the Private Cloud Appliance, you should create a backup of the latest state of the MySQL database, the ZFS Storage Appliance, and the Secret Service (Vault). The backup commands

leverage the existing backup service but create an additional restore point that includes the most recent changes from right before you start an upgrade or patch workflow.

> **Caution:**
>
> In appliance software version 3.0.2-b1081557 and newer, the monitoring data from Prometheus is not included in automated backups. To preserve your Prometheus data, create a backup and restore it manually. For more information, refer to the note with Doc ID 3021643.1.

Using the Service CLI

1. Start the three required backup tasks.

```
PCA-ADMIN> backup target=vault
Status: Success
Data:
  Type = BackupJob
  Job Id = ocid1.brs-
job.PCA3X62D9C1.mypca.joopwuv9403uzbfrh4x9mprmoduh3ljais6ex233v1b21ccqywu4a3vqyk
  Display Name = brs-job-1668419778-backup
  Profile Id =
 ocid1.backup_profile.PCA3X62D9C1.mypca.wrxfwtxwxw6ydp2mwnypcaaxxzmwpuhsc33gcm3d
  Time Created = 2022-11-06T09:56:18Z
  Lifecycle State = CREATING
  Retention = 14

PCA-ADMIN> backup target=zfs
Status: Success
Data:
  Type = BackupJob
  Job Id = ocid1.brs-
job.PCA3X62D9C1.mypca.9oaeaa2kw5crqfcjkh8kyhbxcv8bwh0f4ud6n3lucf802oj15ss3k39874
  Display Name = brs-job-1668419842-backup
  Profile Id =
 ocid1.backup_profile.PCA3X62D9C1.mypca.p7w0tgbvhtjqsgc8rllca2cvotkpgrtf4huiph74
  Time Created = 2022-11-06T09:57:22Z
  Lifecycle State = CREATING
  Retention = 14

PCA-ADMIN> backup target=mysql
Status: Success
Data:
  Type = BackupJob
  Job Id = ocid1.brs-
job.PCA3X62D9C1.mypca.iew5tphpgr3h6mhliw2fai2ywvv386a0xc7isfo8kisj0wrcx114irnit6
  Display Name = brs-job-1668419850-backup
  Profile Id =
 ocid1.backup_profile.PCA3X62D9C1.mypca.henfqzdbafs4z3mxeuslb1c6f4t049w0pxvwf1gi
  Time Created = 2022-11-06T09:57:30Z
  Lifecycle State = CREATING
  Retention = 14
```

2. Use the backup job ID to check the status of the backups.

```
PCA-ADMIN> getBackupJobs
Data:
  id
                                displayName                  components
  --
                                -----------                  ----------
  ocid1.brs-
job.PCA3X62D9C1.mypca.iew5tphpgr3h6mhliw2fai2ywvv386a0xc7isfo8kisj0wrcx114irnit6
    brs-job-1668419850-backup   mysql
```

```
  ocid1.brs-
job.PCA3X62D9C1.mypca.9oaeaa2kw5crqfcjkh8kyhbxcv8bwh0f4ud6n3lucf802oj15ss3k39874
    brs-job-1668419842-backup   zfs
  ocid1.brs-
job.PCA3X62D9C1.mypca.joopwuv9403uzbfrh4x9mprmoduh3ljais6ex233v1b21ccqywu4a3vqyk
    brs-job-1668419778-backup   vault

PCA-ADMIN> getBackupJob backupJobId=ocid1.brs-
job.PCA3X62D9C1.mypca.iew5tphpgr3h6mhliw2fai2ywvv386a0xc7isfo8kisj0wrcx114irnit6
Status: Success
Data:
  Type = BackupJob
  Job Id = ocid1.brs-
job.PCA3X62D9C1.mypca.iew5tphpgr3h6mhliw2fai2ywvv386a0xc7isfo8kisj0wrcx114irnit6
  Display Name = brs-job-1668419850-backup
  Time Created = 2022-11-06T09:57:30Z
  Status = success
  Components = mysql
```

3. Confirm that all three backup operations have completed successfully. Then, proceed to the next upgrade preparation phase.

Preconfiguring the Private Cloud Appliance Upgrade Environment

Preconfiguration refers to a series of operations to ensure the upgrade workflow can be set up correctly.

When the ISO files have been downloaded from My Oracle Support, and made available to the appliance over HTTP, a series of preconfiguration tasks must be performed. Although these operations make no changes to the system, they ensure that the environment meets all requirements for a correct upgrade workflow setup in the next phase of upgrade preparation.

During preconfiguration, the new appliance software ISOs are downloaded to the appliance internal storage. The primary ISO is mounted so the `pca-prerequisite` script can be installed and run. When all script operations are done, the ISO is unmounted again.

> **Caution:**
>
> Until the appliance upgrade is fully completed, ensure that the latest ISOs remain available in the location you specify.

> **Caution:**
>
> The Upgrader allows up to 10 hours to download the ISO images before it times out. This corresponds with 50GB of data downloaded over a connection with at least 20Mbps of bandwidth. Ensure that the download connection meets this minimum requirement. For optimal performance, a higher bandwidth is recommended.

Operations performed by the `pca-prerequisite` script include:

- validating current and new software versions
- installing the latest `preUpgrade` functionality
- copying the latest Upgrader code to the appliance
- writing environment parameters to a yaml file for use in the next preparation phase
- generating a new upgrade history file

This entire preconfiguration process is launched by an appliance administrator with a single command.

Unprovisioned Compute Nodes

If the Private Cloud Appliance contains unprovisioned compute nodes, they can cause errors during upgrade or patching, and might experience provisioning issues later. Therefore, when a record exists in the node database, the

compute node must be provisioned and added to a fault domain, otherwise the upgrade or patching process cannot proceed.

If you need to upgrade or patch the appliance software on a system with compute nodes that cannot be provisioned, these database records must be removed. Contact Oracle for assistance.

For more information about provisioning, see Performing Administrative Operations on Compute Nodes on page 156.

Prerequisite Version

The latest Upgrader code automatically enforces prerequisite software versions on your Private Cloud Appliance. In the early stages of upgrade or patch preparation, the Upgrader service validates the currently installed appliance software version against the new target version. The preparation process (`upgradePreConfig`) documented in this section will only proceed if validation is successful.

If the appliance is not running at least the minimum required version, the Upgrader exits the process and rolls back the environment to its previous state. View the details of the failed upgrade job:

```
PCA-ADMIN> getupgradejob upgradeJobId=1700153626051-prepare-40046
Data:
  Log File = /nfs/shared_storage/pca_upgrader/log/pca-
upgrader_prepare_environment_2023_11_16-16.53.46.log
  Arguments = [...]
  Status = Failed
[...]
  Tasks 23 - Name = Check Prerequisite Build Version
  Tasks 23 - Description = Check current build version not lower than
 prerequisite version
  Tasks 24 - Name = Check Prerequisite Build Version
  Tasks 24 - Message = (("Caught exception while checking prerequisite build
 number
  Exception: Command: ['/usr/bin/python3', '/var/lib/pca-upgrader/
prerequisite_build_validator.py',
  'rack=PCA', 'upgrade=ISO'] failed (1): stderr: b'' stdout: b'PCA version
 is lower than prerequisite build,
  must upgrade to prerequisite build 3.0.2-b892153 to proceed further
 upgrade\\n'",), {})
  Tasks 24 - Status = Failed
```

You must first install the prerequisite version, which is indicated by the error message in the upgrade job output.

Performing the Preconfiguration Tasks

Run preconfiguration from the UI or CLI.

Using the Service Web UI

1. In the navigation menu, go to the Maintenance section and click Upgrade Plan. This provides an overview of current and target component versions.
2. Click Upgrade & Patching to display the Upgrade Jobs page.
3. In the top-right corner of the Upgrade Jobs page, click Create Upgrade or Patch.

   The Create Request window appears. Choose *Upgrade* as the Request Type.
4. Select the appropriate upgrade request type: Upgrade PreConfig.
5. Fill out the upgrade request parameters:
   - **Option:** Enter *ISO*.
   - **Location:** Enter the path to the location where the ISO image is stored. If this version consists of multiple ISOs, point to the primary `pca-3.0.2-<build_no>.iso` file.
   - **ISO Checksum:** Enter the checksum required to verify the integrity of the ISO image. The checksum is provided alongside the ISO image in a file named `<iso_image>.sha256sum`.

- **Log Level:** Optionally, select a specific log level for the upgrade log file. The default log level is "Information". For maximum detail, select "Debug".
- **Advanced Options JSON:** Optionally, add a JSON string to provide additional command parameters.
- **Alternative ULN Channel:** This parameter applies to patching and can be ignored.

6. Click Create Request.

   The new upgrade request appears in the Upgrade Jobs table. The preconfiguration steps are performed as described at the start of this section.

Using the Service CLI

1. Gather the information that you need to run the command:

   - the location of the ISO image to upgrade from

     Enter the path to where the ISO image is stored. If this version consists of multiple ISOs, point to the primary `pca-3.0.2-<build_no>.iso` file.

     All ISO content will be unpacked on the shared storage accessible from the management nodes. Secondary ISO images are detected and processed automatically, but they must be stored in the same source directory.
   - the checksum used to verify the ISO image

     The checksum is provided alongside the ISO image. Its file name is the ISO image name with `.sha256sum` appended. The system uses the checksum to verify that the data in the ISO images is intact and valid for this upgrade.

2. Enter the upgrade preconfiguration command.

   Syntax (entered on a single line):

   ```
   upgradePreConfig
   option=ISO
   location=<path-to-iso>
   isoChecksum=<iso-file-checksum>
   ```

   Example:

   ```
   PCA-ADMIN> upgradePreConfig  option=ISO \
   location=http://host.example.com/pca-<version>-<build>.iso \
   isoChecksum=4a7778d45b3996319ee67c772fbbc0320a9806a942490a1cfe6b37930372809e
   Status: Success
   JobId: 105524a2-24a2-4b52-91c4-5cd8832552d2
   Data: Service request has been submitted. Upgrade Job Id =
    1737483518274-prepare-28393 Upgrade Request Id = UWS-e37208d4-
   d433-4e16-8140-de38c0757c9d
   ```

3. Use the request ID and the job ID to check the status of the upgrade process.

   ```
   PCA-ADMIN> getUpgradeJobs
   Data:
     id                            Upgrade Request Id
                Command Name   Result
     --                            ------------------
                ------------   ------
     1737483518274-prepare-28393   UWS-e37208d4-d433-4e16-8140-
   de38c0757c9d          prepare        Passed

   PCA-ADMIN> getUpgradeJob upgradeJobId=1737483518274-prepare-28393
   Status: Success
   Data:
     Upgrade Request Id = UWS-e37208d4-d433-4e16-8140-de38c0757c9d
     Name = prepare
     Pid = 28393
   ```

```
  Host = pcamn02
  Log File = /nfs/shared_storage/pca_upgrader/log/pca-
upgrader_prepare_environment_2025_01_21-18.18.38.log
  Arguments =
 {"component_names":null,"diagnostics":false,"display_task_plan":false,"dry_run_
host.example.com/pca-<version>-
<build>.iso","online_upgrade":null,"precheck_status":false,"repo_config_override
e37208d4-d433-4e16-8140-de38c0757c9d","uln":null,"patch":null}
  Status = Passed
  Execution Time(sec) = 776
  Tasks 1 - Name = Validate Image Location
  Tasks 1 - Description = Verify that the image exists at the
 specified location and is correctly named
[...]
```

4. Proceed to the next upgrade preparation phase.

Setting Up New Software Sources for Private Cloud Appliance Upgrade

This preparation task installs the latest version of the Upgrader, unpacks the new version of the Private Cloud Appliance software, and generates the upgrade plan.

The code of the Upgrader is changed regularly, as is the case with any other system service. The setup phase of the upgrade preparation ensures that the Upgrader is up-to-date first, so that all upgrade commands are run with the latest version. Upgrade operations will not be allowed to run if the system detects that the Upgrader is not the latest available version.

When the latest version of the Upgrader is up and running, the sources of the new appliance software are set up for use in the system upgrade procedures. The current software image is backed up, and the new software ISOs are unpacked on the management cluster shared storage in the appropriate directory structure: OS base image, appliance software RPM packages, new compute instance images, new firmware, and so on.

At the end of the setup phase, an *upgrade plan* is generated. The upgrade plan is based on a comparison between the current installation and the new version that was unpacked. It determines which upgrade procedures need to be performed in the next phase. For more information, see Checking Status and Progress of Private Cloud Appliance Upgrade Plan on page 357.

This entire setup process is launched by an appliance administrator with a single command.

> **Caution:**
>
> Ensure that the preconfiguration process has been completed first. See Preconfiguring the Private Cloud Appliance Upgrade Environment on page 343.
>
> To prevent inconsistencies while executing the upgrade plan later on, it is critical that both commands in the preparation process, `upgradePreConfig` (preconfiguration) and `preUpgrade` (setup), are completed together in the specified order. If at any time you need to rerun the `preUpgrade` command, you must rerun the preceding command first.

Using the Service Web UI

1. In the navigation menu, go to the Maintenance section and click Upgrade Plan. This provides an overview of current and target component versions.
2. Click Upgrade & Patching to display the Upgrade Jobs page.
3. In the top-right corner of the Upgrade Jobs page, click Create Upgrade or Patch.

   The Create Request window appears. Choose *Upgrade* as the Request Type.
4. Select the appropriate upgrade request type: preUpgrade.
5. For the upgrade request type, enter *ISO*. The required files are retrieved from the unpacked ISO images.
6. Click Create Request.

The new upgrade request appears in the Upgrade Jobs table. When the job has completed successfully, the Upgrader is up-to-date and the new software version sources are ready for use.

Using the Service CLI

1. Start the installation of the latest Upgrader version, and setup of the new appliance software sources.

```
PCA-ADMIN> preUpgrade type=ISO
Status: Running
JobId: aa33246e-fd79-460f-a191-cc97f8be04be
Data: in progress
```

Upgrader operations performed with this command:

a. Install the new Upgrader version on the management nodes, then restart the Upgrader systemd service for the changes to take effect.
b. Update the Service CLI and Service Web UI to make the latest commands and functions available.

Setup operations performed with this command:

a. Back up the current software image.
b. Unpack the ISO images, store new software RPMs and installation files (images, firmware) in correct directories.
c. Remove obsolete backups and images from the appliance internal shared storage.
d. Generate the upgrade plan.

2. Check the status of the preUpgrade process at any time using this command:

```
PCA-ADMIN> getPreUpgradeStatus
Data:
  status = IN-PROGRESS
  message =  PREUPGRADE-c0f0b38d-64d2-42bb-8d96-de39d1218b24: A pre-
upgrade task is running!

PCA-ADMIN> getPreUpgradeStatus
Data:
  status = SUCCESS
  message =  The previous pre-upgrade task succeeded!
```

3. Confirm that the latest version of the Upgrader has been installed successfully.

```
PCA-ADMIN> getUpgradeJobs
Data:
  id                              Upgrade Request Id
             Command Name   Result
  --                              ------------------
             ------------   ------
  1737484792498-setup-42073     PREUPGRADE-44ec9a80-b3b5-4029-
bd77-8178d62173a8   setup          Passed
  1737483518274-prepare-28393   UWS-e37208d4-d433-4e16-8140-
de38c0757c9d          prepare        Passed
```

In case the preUpgrade process fails, check the upgrade job for troubleshooting information.

```
PCA-ADMIN> getUpgradeJob upgradeJobId=1737484792498-setup-42073
Data:
  Upgrade Request Id = PREUPGRADE-44ec9a80-b3b5-4029-
bd77-8178d62173a8
  Name = setup
  Pid = 25325
  Host = pcamn02
  Log File = /nfs/shared_storage/pca_upgrader/log/pca-
upgrader_setup_upgrade_environment_2025_01_21-18.39.52.log
```

```
  Arguments =
 {"component_names":null,"diagnostics":false,"display_task_plan":false,"dry_run_
b3b5-4029-bd77-8178d62173a8","uln":null,"patch":null}
  Status = Failed
  Execution Time(sec) = 1069
  Tasks 1 - Name = Validate Setup Image Location
  Tasks 2 - Name = Validate Upgrader Version
  Tasks 3 - Name = Validate Upgrade Order
[...]
  Tasks 22 - Name = Delete Old backups
  Tasks 22 - Message = [Errno 2] No such file or directory: '/nfs/
shared_storage/yum/pca_upgrader'
  Tasks 22 - Status = Failed
```

4. Ensure that the system is in ready state. Then, proceed with the rack upgrade workflow.
5. Check which components need to be upgraded by displaying the upgrade plan. Verify that the target build number matches the appliance software version you are planning to upgrade to.

```
PCA-ADMIN> getUpgradePlan
Data:
  id                          component    currentBuild
 targetBuild      currentVersion                  targetVersion
                           requireReboot   timeEstimation (min)
 requireUpgrade   impactedInfra
  --                          ---------    ------------
 -----------      -------------                  -------------
                           -------------   --------------------
 -------------   -------------
  generic                     zfssa        3.0.2-b1300385
   3.0.2-b1300385   2013.06.05.8.73.1-2.73.5701.1
 2013.06.05.8.73.1-2.73.5701.1              false           45
               false            host,compute
  100.96.2.64                 compute      3.0.2-b1300130
   3.0.2-b1300385   oraclelinux-release-7.9-1.0.9
 3.0.202.3.11.0.0.0.5.7ea3f4ccbf3e17fc...   true            50
               true             compute
  100.96.2.65                 compute      3.0.2-b1300130
   3.0.2-b1300385   oraclelinux-release-7.9-1.0.9
 3.0.202.3.11.0.0.0.5.7ea3f4ccbf3e17fc...   true            50
               true             compute
  100.96.2.66                 compute      3.0.2-b1300130
   3.0.2-b1300385   oraclelinux-release-7.9-1.0.9
 3.0.202.3.11.0.0.0.5.7ea3f4ccbf3e17fc...   true            50
               true             compute
  100.96.2.33                 host         3.0.2-b1300130
   3.0.2-b1300385   oraclelinux-release-7.9-1.0.13
 3.0.202.3.11.0.0.0.5.7ea3f4ccbf3e17fc...   true            40
               true             host
  100.96.2.34                 host         3.0.2-b1300130
   3.0.2-b1300385   oraclelinux-release-7.9-1.0.13
 3.0.202.3.11.0.0.0.5.7ea3f4ccbf3e17fc...   true            40
               true             host
  100.96.2.35                 host         3.0.2-b1300130
   3.0.2-b1300385   oraclelinux-release-7.9-1.0.13
 3.0.202.3.11.0.0.0.5.7ea3f4ccbf3e17fc...   true            40
               true             host
  generic                     mysql        3.0.2-b1300385   3.0.2-
b1300385   8.0.36-1.1                      8.0.36-1.1
                  false           15                     false
       host
  generic                     etcd         3.0.2-b1300385   3.0.2-
b1300385   3.5.6                           3.5.6
```

```
                    false           5                       false
       host
  generic                       vault        3.0.2-b1300385
   3.0.2-b1300385   v1.7.1-3.10.0.0.0.4.g5f2d5af
 v1.7.1-3.10.0.0.0.4.g5f2d5af               false           5
               false           host
  generic                       kubernetes   3.0.2-b1300385   3.0.2-
b1300385   1.25.16-2                        1.25.16-2
                    false           60                      false
       host,compute
  generic                       platform     3.0.2-b1300130   3.0.2-
b1300385   None                             None
                    false           40                      true
       host,compute
  Oracle-Linux-7.9              ociImages    3.0.2-b1300130   3.0.2-
b1300385   None                             2024.07.31_0
                    false           5                       true
       host
  Oracle-Linux8-OKE-1.28.8      ociImages    3.0.2-b1300130   3.0.2-
b1300385   None                             20241201
                    false           5                       true
       host
  Oracle-Solaris-11             ociImages    3.0.2-b1300130   3.0.2-
b1300385   None                             2024.08.26_0
                    false           5                       true
       host
  Oracle-Linux8-OKE-1.27.12     ociImages    3.0.2-b1300130   3.0.2-
b1300385   None                             20241201
                    false           5                       true
       host
  Oracle-Linux8-OKE-1.26.15     ociImages    3.0.2-b1300130   3.0.2-
b1300385   None                             20241201
                    false           5                       true
       host
  Oracle-Linux-9                ociImages    3.0.2-b1300130   3.0.2-
b1300385   None                             2024.07.31_0
                    false           5                       true
       host
  Oracle-Linux-8                ociImages    3.0.2-b1300130   3.0.2-
b1300385   None                             2024.07.31_0
                    false           5                       true
       host
  100.96.0.33                   ilom         3.0.2-b1300130   3.0.2-
b1300385   5.1.4.20                         5.1.4.25
                    true            10                      true
       host,compute
  100.96.0.34                   ilom         3.0.2-b1300130   3.0.2-
b1300385   5.1.4.20                         5.1.4.25
                    true            10                      true
       host,compute
  100.96.0.35                   ilom         3.0.2-b1300130   3.0.2-
b1300385   5.1.4.20                         5.1.4.25
                    true            10                      true
       host,compute
  100.96.0.64                   ilom         3.0.2-b1300130   3.0.2-
b1300385   5.1.4.20                         5.1.4.25
                    true            10                      true
       host,compute
  100.96.0.65                   ilom         3.0.2-b1300130   3.0.2-
b1300385   5.1.4.20                         5.1.4.25
                    true            10                      true
       host,compute
  100.96.0.66                   ilom         3.0.2-b1300130   3.0.2-
b1300385   5.1.4.20                         5.1.4.25
```

```
                    true            10                      true
       host,compute
  leaf                          switch       3.0.2-b1300385   3.0.2-
b1300385   10.3.4a                            10.3.4a
                    false           60                      false
       host,compute
  spine                         switch       3.0.2-b1300385   3.0.2-
b1300385   10.3.4a                            10.3.4a
                    false           60                      false
       host,compute
  mgmt                          switch       3.0.2-b1300385   3.0.2-
b1300385   10.3.4a                            10.3.4a
                    false           60                      false
       host,compute
```

Checking Upgrade Readiness and Status

Private Cloud Appliance provides a set of commands and tools to estimate the impact of an upgrade and monitor the progress of the entire workflow.

It is generally recommended to run the most recent version of the Private Cloud Appliance operating software on your system, in order to take advantage of the latest functionality as well as security and stability features. The process to bring the appliance software to the latest version, whether through upgrading or patching, is designed for minimal disruption: the system might experience a temporary reduction in performance or availability, but workloads continue to run without downtime.

To help the appliance administrators prepare for upgrade or patching, scope the work required to take the appliance from its current software version to the latest, and monitor progress of an ongoing upgrade or patch workflow, the system provides a set of commands and tools that are described in the topics that follow. To make the process as fluent as possible, with minimal risk of delays or interruptions, we recommend running commands in verification-only mode first. The administrator's most important tool is the upgrade plan, which lays out all the required steps in the correct order and provides progress information from start to completion.

Checking Current Version of Private Cloud Appliance Components

To evaluate upgrade requirements and the impact of a Private Cloud Appliance upgrade you can view the current state of the top-level rack components. The system lists the most important build and version numbers of all the components for which an upgrade procedure is documented.

The component version list can be viewed in two ways:

- In the Service Web UI, go to Maintenance in the navigation menu and select Component Version.
- In the Service CLI, enter the following command:

```
PCA-ADMIN> getComponentVersions
Data:
  id                        component    iso              version
  --                        ---------    ---              -------
  100.96.2.64               compute      3.0.2-b1046481   3.0.2-687
  100.96.2.65               compute      3.0.2-b1046481   3.0.2-687
  100.96.2.66               compute      3.0.2-b1046481   3.0.2-687
  100.96.2.67               compute      3.0.2-b1046481   3.0.2-687
  100.96.2.68               compute      3.0.2-b1046481   3.0.2-687
  100.96.2.69               compute      3.0.2-b1046481   3.0.2-687
  generic                   etcd         3.0.2-b1049367   3.3.10
  100.96.2.33               host         3.0.2-b1049367   oraclelinux-
release-7.9-1.0.9
  100.96.2.34               host         3.0.2-b1049367   oraclelinux-
release-7.9-1.0.9
  100.96.2.35               host         3.0.2-b1049367   oraclelinux-
release-7.9-1.0.9
  100.96.0.33               ilom         3.0.2-b1049367   5.1.1.21
  100.96.0.34               ilom         3.0.2-b1049367   5.1.1.21
```

```
  100.96.0.35                 ilom         3.0.2-b1049367   5.1.1.21
  100.96.0.64                 ilom         3.0.2-b1049367   5.1.2.20.a
  100.96.0.65                 ilom         3.0.2-b1049367   5.1.1.21
  100.96.0.66                 ilom         3.0.2-b1049367   5.1.1.21
  100.96.0.67                 ilom         3.0.2-b1049367   5.1.2.20.a
  100.96.0.68                 ilom         3.0.2-b1049367   5.1.1.21
  100.96.0.69                 ilom         3.0.2-b1049367   5.1.1.21
  generic                     kubernetes   3.0.2-b1049367   1.25.15-1
  generic                     mysql        3.0.2-b1049367   8.0.33-1.1
  Oracle-Linux-7.9            ociImages    3.0.2-b1049367   2023.09.26_0
  Oracle-Linux-8              ociImages    3.0.2-b1049367   2023.09.26_0
  Oracle-Linux-9              ociImages    3.0.2-b1049367   2023.09.26_0
  Oracle-Linux8-OKE-1.26.6    ociImages    3.0.2-b1049367   20240210
  Oracle-Linux8-OKE-1.27.7    ociImages    3.0.2-b1049367   20240209
  Oracle-Linux8-OKE-1.28.3    ociImages    3.0.2-b1049367   20240210
  Oracle-Solaris-11           ociImages    3.0.2-b1049367   2023.10.16_0
  generic                     platform     3.0.2-b1046481   None
  leaf                        switch       3.0.2-b1049367   10.3.4a
  mgmt                        switch       3.0.2-b1049367   10.3.4a
  spine                       switch       3.0.2-b1049367   10.3.4a
  generic                     vault        3.0.2-b1049367   v1.7.1-3
  generic                     zfssa        3.0.2-b1046481
 2013.06.05.8.57.1-2.57.5501.1
```

Checking Private Cloud Appliance Upgrade History

The upgrade history provides a clear way to drill down into the details of the upgrade and patching activity on Private Cloud Appliance.

> **Note:**
>
> ULN based patching has been deprecated. However, if the system was patched in the past, those records appear in the upgrade history.

Information about all component upgrades and patches is stored in upgrade jobs, which can be consulted in the Service Web UI and Service CLI. Over time, as the system goes through multiple upgrades, the large number of entries might make the list difficult to interpret.

The upgrade history presents the information from all upgrade and patch jobs in a categorized way so you can see which version upgrades have been performed, which jobs have been run for each of those upgrades, and from which source (ISO upgrade or ULN patch). Details include build versions, component versions before and after, job completion, success or failure, time stamps, and duration.

Appliance software builds are installed onto the appliance either through upgrade from ISO content or patching from ULN. Display the build history by running the following command.

```
PCA-ADMIN> getUpgradeHistory
Data:
  id                                    From Build       To Build
  Status       Type   Start Time            End Time              Total
 Upgrade Time (min)   Actual Upgrade Time (min)
  --                                    ----------       --------
     ------       ----   ----------            --------
 ------------------------   -------------------------
  pca-upgrade-history-3.0.2-b1300385   3.0.2-b1300130   3.0.2-b1300385
 incomplete   ISO    2025-01-21T18:31:33   N/A                    N/A
               N/A
  pca-upgrade-history-3.0.2-b1300130   3.0.2-b1261765   3.0.2-b1300130
 complete     ISO    2025-01-06T05:11:24   2025-01-06T11:35:16   384
               369
  pca-upgrade-history-3.0.2-b1261765   3.0.2-b1010555   3.0.2-b1261765
 complete     ULN    2024-12-21T04:22:41   2024-12-21T12:59:35   517
               377
```

```
  pca-upgrade-history-3.0.2-b1010555   3.0.2-b892153    3.0.2-b1010555
 complete     ULN    2024-05-18T22:32:13   2024-05-19T06:01:51   449
                341
  pca-upgrade-history-3.0.2-b892153    3.0.2-b799577    3.0.2-b892153
 complete     ISO    2023-10-10T18:10:37   2023-10-11T05:17:40   667
                462
```

If necessary, you can filter results by type.

```
PCA-ADMIN> GetUpgradeHistory type=uln
Data:
  id                                    From Build       To Build
  Status       Type   Start Time            End Time              Total
 Upgrade Time (min)   Actual Upgrade Time (min)
  --                                    ----------       --------
     ------       ----   ----------            --------
 ------------------------   ------------------------
  pca-upgrade-history-3.0.2-b1261765   3.0.2-b1010555   3.0.2-b1261765
 complete     ULN    2024-12-21T04:22:41   2024-12-21T12:59:35   517
                378
  pca-upgrade-history-3.0.2-b1010555   3.0.2-b892153    3.0.2-b1010555
 complete     ULN    2024-05-18T22:32:13   2024-05-19T06:01:51   449
                341
```

To display the upgrade job list related to a particular build, copy the build ID and run this command:

```
PCA-ADMIN> getUpgradeHistoryDetails id=pca-upgrade-history-3.0.2-b1300385
Data:
  id            Component    Timestamp             From Build
        To Build                                   Status   Job ID
                   Time Taken (min)
  --            ---------    ---------             ----------
        --------                                   ------   ------
                   ----------------
  generic       prepare      2025-01-21T18:31:34   3.0.2-b1300130
         3.0.2-b1300385                             Passed   1737483518274-
prepare-28393       13
  generic       setup        2025-01-21T19:49:36   3.0.2-b1300130
         3.0.2-b1300385                             Passed   1737484792498-
setup-42073         70
  generic       preupgrade   2025-01-21T19:49:44   3.0.2-b1300130
        3.0.2-b1300385                             Passed   N/A
                   1
  generic       zfssa        2025-01-21T19:57:45
 2013.06.05.8.73.1-2.73.5701.1    2013.06.05.8.73.1-2.73.5701.1
 Passed   1737489464069-zfssa-1898          0
  100.96.2.64   compute      2025-01-21T19:58:21   oraclelinux-
release-7.9-1.0.9    3.0.202.3.11.0.0.0.5.7ea3f4ccbf3e17fc...   Passed
 1737489497566-compute-8759        52
  100.96.2.65   compute      2025-01-21T20:50:34   oraclelinux-
release-7.9-1.0.9    3.0.202.3.11.0.0.0.5.7ea3f4ccbf3e17fc...   Passed
 1737492632323-compute-49553       0
  100.96.2.66   compute      2025-01-21T20:51:36   oraclelinux-
release-7.9-1.0.9    3.0.202.3.11.0.0.0.5.7ea3f4ccbf3e17fc...   Passed
 1737492695074-compute-61798       0
  100.96.2.33   host         2025-01-21T20:52:56   oraclelinux-
release-7.9-1.0.13   3.0.202.3.11.0.0.0.5.7ea3f4ccbf3e17fc...   Passed
 1737492758087-host-71365          29
  100.96.2.34   host         2025-01-21T21:22:22   oraclelinux-
release-7.9-1.0.13   3.0.202.3.11.0.0.0.5.7ea3f4ccbf3e17fc...   Passed
 1737494531521-host-162838         29
```

```
  100.96.2.35   host         2025-01-21T21:51:45   oraclelinux-
release-7.9-1.0.13   3.0.202.3.11.0.0.0.5.7ea3f4ccbf3e17fc...   Passed
 1737496294531-host-396629         24
  generic       kubernetes   2025-01-21T22:15:53   1.25.16-2
         1.25.16-2                                   Passed   1737497752826-
kubernetes-637866   0
  generic       platform     2025-01-21T22:16:58   3.0.2-b1300130
         3.0.2-b1300385                              Passed   1737497814071-
platform-646623     43
  generic       ociImages    2025-01-21T23:00:33   3.0.2-b1300130
         3.0.2-b1300385                              Passed   1737500431705-
oci-979395          18
  100.96.0.33   ilom         2025-01-21T23:19:03   5.1.4.20
         5.1.4.25                                    Passed   1737501529548-
ilom-1060489        13
  100.96.0.34   ilom         2025-01-21T23:32:31   5.1.4.20
         5.1.4.25                                    Passed   1737502338638-
ilom-1133349        13
  100.96.0.35   ilom         2025-01-21T23:46:10   5.1.4.20
         5.1.4.25                                    Passed   1737503158143-
ilom-1197515        13
  100.96.0.64   ilom         2025-01-21T23:59:46   5.1.4.20
         5.1.4.25                                    Passed   1737503974834-
ilom-1266068        13
  100.96.0.65   ilom         2025-01-22T00:13:30   5.1.4.20
         5.1.4.25                                    Passed   1737504796873-
ilom-1361643        13
  100.96.0.66   ilom         2025-01-22T00:27:01   5.1.4.20
         5.1.4.25                                    Passed   1737505608602-
ilom-1435802        13
  leaf          switch       2025-01-22T00:40:22   10.3.4a
         10.3.4a                                     Passed   1737506421692-
cisco-1507991       0
  spine         switch       2025-01-22T00:41:23   10.3.4a
         10.3.4a                                     Passed   1737506482956-
cisco-1524202       0
  mgmt          switch       2025-01-22T00:42:25   10.3.4a
         10.3.4a                                     Passed   1737506544387-
cisco-1532342       0
```

Note:

Alternatively, you can view this information in the Service Web UI. In the navigation menu, go to the Maintenance section, and click Upgrade History. Click an upgrade workflow entry in the table to view the job details of the upgrade workflow.

Ensuring Private Cloud Appliance Is In Ready State

If issues are detected, either from the fault log or the health checks, you can resolve them before the planned Private Cloud Appliance upgrade window, and keep the actual system upgrade as fluent and short as possible. It is considered good practice to ensure that backups are created of the system and the resources in your environment.

Private Cloud Appliance can be upgraded with limited impact on the system. No downtime is required, and user workloads continue to run while the underlying infrastructure is being upgraded in stages.

Important:

Concurrent upgrade operations are not supported. An upgrade job must be completed before a new one can be started. Consult the upgrade history (using the `getUpgradeHistory` command) and the upgrade plan to confirm there is no ongoing or pending upgrade activity.

Fault Log

When preparing for an upgrade, check the system fault log. Issues with potential impact on upgrade and patching are flagged as *upgrade faults*, and will prevent any upgrade or patch command from running. The Service CLI provides filtering options for these faults, as shown in these examples:

```
PCA-ADMIN> list fault fields upgradeFault,Status,Severity
Data:
  id                                     Upgrade Fault   Status    Severity
  --                                     -------------   ------    --------
  37f6cefe-f7d5-49a8-adff-76c1a020bcc8   False           Cleared   Critical
  2be57600-4dbd-40f0-a0f8-e4ffbb2c8468   True            Active    Critical
  297c770e-16e1-11ef-9e78-a8698c107234   True            Cleared   Major
  d27e0895-eb87-4e66-bd4b-f6500153cf64   True            Cleared   Critical
  77752a35-4cf7-49ed-88df-06158d846358   True            Active    Critical
  9ff4fe22-8ed6-463f-95b5-458d6a76d185   False           Active    Critical
  0a09204c-953e-4df3-912b-a2a175ce8c1a   True            Cleared   Critical
[...]

PCA-ADMIN> list fault where upgradeFault EQ True
Data:
  id                                     Name
         Status    Severity
  --                                     ----
         ------    --------
  2be57600-4dbd-40f0-a0f8-e4ffbb2c8468   pcamn01--PCA-8000-44--asrclient
         Active    Critical
  297c770e-16e1-11ef-9e78-a8698c107234   ilom-pcacn001--PCA-8000-EA--ilom-
pcacn001   Cleared   Major
  0a09204c-953e-4df3-912b-a2a175ce8c1a   pcamn01--PCA-8000-AH--mysql_cluster
         Cleared   Critical
  f7da7c82-03bc-4383-95cf-542cefbb5d39   pcamn01--PCA-8000-CD--mysql_cluster
         Cleared   Critical
  74fd23ac-3006-4ab6-9c50-544869ba78f9   pcamn02--PCA-8000-6C--registry
         Cleared   Critical
  dc77b088-939d-4725-b45f-1072024b02ba   pcamn02--PCA-8000-0E--etcd
         Cleared   Critical
  d27e0895-eb87-4e66-bd4b-f6500153cf64   pcamn01--PCA-8000-22--vault
         Cleared   Critical
  77752a35-4cf7-49ed-88df-06158d846358   pcamn01--PCA-8000-93--mysql_cluster
         Active    Critical
```

Pre-Checks

Every upgrade operation is preceded by a set of pre-checks. These are built into the upgrade code and will report an error if the system is not in the required state for the upgrade. The upgrade will only begin if all pre-checks are passed.

You can use the pre-checks to test in advance for any system health issues that would prevent a successful upgrade. After preparing the upgrade environment, run any or all of the upgrade commands with the "verify only" option.

> **Caution:**
>
> Oracle strongly recommends testing that the Private Cloud Appliance is ready for upgrading, by executing the full rack upgrade command in verify-only mode. The output provides a readiness report you can use to plan any corrective actions as well as the upgrade.
>
> This verification can take a long time to complete. Run it long enough in advance so the actual upgrade can be completed within the scheduled window.

In the Service Web UI the verify-only option is activated with a checkbox when you create the upgrade request; in the Service CLI you use the optional upgrade command parameter shown in this example:

```
PCA-ADMIN> upgradeRack type=ISO action=APPLY component=KUBERNETES
 verifyOnly=True
JobId: d14646fb-b2a9-445d-9702-b99ac605a993
Data: Service request has been submitted. Upgrade Job Id = 1744286305509-
kubernetes_verify-2265363 Upgrade Request Id = UWS-7a4a52a4-
dc7c-439c-9e32-9e61358d4e56

PCA-ADMIN> getUpgradeJobs
Data:
  id                                          Upgrade Request Id
                  Command Name        Result
  --                                          ------------------
                  ------------        ------
  1744286305509-kubernetes_verify-2265363   UWS-7a4a52a4-
dc7c-439c-9e32-9e61358d4e56          kubernetes_verify   Passed
```

Rack-Wide Health Check

The full rack upgrade workflow provides a command option to run a suite of health checks across the entire appliance. The latest checks are available after the setup phase of upgrade preparation (preUpgrade) is completed. The rack-wide health check is designed to run before an upgrade, but you can use it to scan for potential issues at any time, as long as no upgrade job is in progress.

Using the Service Web UI

1. In the navigation menu, go to the Maintenance section and click Upgrade & Patching.
2. In the top-right corner of the Upgrade Jobs page, click Create Upgrade or Patch.

   The Create Request window appears. Choose *Upgrade* as the Request Type.
3. Select Upgrade Rack and enter these upgrade request parameters:

   - **Type:** For upgrade, select ISO. The ULN option applies to patching.
   - **Action:** Select Health Check.
4. Click Create Request.

   The new upgrade request appears in the Upgrade Jobs table.
5. Click the Job ID to display its detail page.

   - General upgrade job information – such as IDs, time stamps, log location and so on – is shown in the top panel of the page.
   - The Resources section of the page provides a table view of all tasks, in this case health checks, that are part of the job. The table describes each individual task and the result of its run.
6. In case a health check returns an error, ensure it's corrected before you start an upgrade.

Using the Service CLI

1. Start the rack-wide health check workflow.

   ```
   PCA-ADMIN> upgradeRack action=HEALTHCHECK
   JobId: 6cc1b68e-3b39-4e72-9917-eb5f161e7078
   Data: Service request has been submitted. Upgrade Job Id =
    1765790185978-setup_verify-1411430 Upgrade Request Id = UWS-
   a598fc4d-1fac-46a9-908a-2335c8a73d8b
   ```
2. Use the request ID to check the status of the health check workflow. When the workflow is complete, an end time is displayed and the status changes to Passed or Failed.

   ```
   PCA-ADMIN> getUpgradeStatus requestId=UWS-
   a598fc4d-1fac-46a9-908a-2335c8a73d8b
   ```

```
Data:
  Request id = UWS-a598fc4d-1fac-46a9-908a-2335c8a73d8b
  Status = Running
  Type = ISO
  Previous Build = 3.0.2-b1325160
  Target Build = 3.0.2-b1443573
  Jobs 1 = 1765790185978-setup_verify-1411430
  Start Time = 2025-09-15T09:16:25
```

3. Check the status of the individual tasks. Use the upgrade job id to display the tasks in detail.

```
PCA-ADMIN> getUpgradeJob upgradeJobId=1765790185978-
setup_verify-1411430
Data:
  Upgrade Request Id = UWS-a598fc4d-1fac-46a9-908a-2335c8a73d8b
  Name = setup_verify
  Start Time = 2025-09-15T09:16:25
  End Time = 2025-09-15T09:22:28
  Pid = 1411430
  Host = pcamn02
  Log File = /nfs/shared_storage/pca_upgrader/log/pca-
upgrader_setup_upgrade_environment_verify_2025_09_15-09.16.25.log
  Arguments =
 {"component_names":null,"diagnostics":false,"display_task_plan":false,"dry_run_
a598fc4d-1fac-46a9-908a-2335c8a73d8b","uln":null,"patch":null}
  Status = Failed
  Execution Time(sec) = 362
  Tasks 1 - Name = UpgradePlanUpdateTask
  Tasks 1 - Message = Task did not run. This task only runs when
 SetupEnvConfigurationTask's upgrade is True AND Setup Upgrade
 Environment has status of Passed.
  Tasks 1 - End Time = 2025-09-15T09:16:26
  Tasks 1 - Status = Not Run
  Tasks 1 - Reason Phrase = Not Required
  Tasks 1 - Execution Time(sec) = 0
  Tasks 2 - Name = Validate Setup Image Location
  Tasks 2 - Description = Verify that the image exists at the
 downloaded location and is correctly named
  Tasks 2 - Start Time = 2025-09-15T09:16:26
  Tasks 2 - End Time = 2025-09-15T09:16:26
  Tasks 2 - Status = Passed
  Tasks 2 - Reason Phrase = Passed
  Tasks 2 - Execution Time(sec) = 0
  Tasks 3 - Name = Validate Upgrader Version
  Tasks 3 - Description = Ensures the upgrader does not require an
 upgrade
  Tasks 3 - Start Time = 2025-09-15T09:16:26
  Tasks 3 - End Time = 2025-09-15T09:16:26
  Tasks 3 - Status = Passed
  Tasks 3 - Reason Phrase = Passed
  Tasks 3 - Execution Time(sec) = 0
[...]
  Tasks 12 - Name = Check Pods Status
  Tasks 12 - Description = Checks all pods are in healthy status
  Tasks 12 - Message = The following pods are in unhealthy status:
                       ['vcn1302908-meta-6bdd5d499b-cwnf8']
  Tasks 12 - Start Time = 2025-09-15T09:16:27
  Tasks 12 - End Time = 2025-09-15T09:20:27
  Tasks 12 - Status = Failed
  Tasks 12 - Reason Phrase = Failed
  Tasks 12 - Execution Time(sec) = 240
[...]
```

The output of the `getUpgradeJob` command provides detailed information about the tasks performed during the health check procedure. It displays descriptions, time stamps, duration, and success or failure. Whenever a task fails – as is the case with Task 12 in the example – the command output indicates which check has failed. For in-depth troubleshooting you can search the log file at the location provided near the start of the command output.

4. In case a health check returns an error, ensure it's corrected before you start an upgrade.

Checking Status and Progress of Private Cloud Appliance Upgrade Plan

The Upgrader in Private Cloud Appliance generates an upgrade plan and uses it to orchestrate all required upgrade operations. You can view the current upgrade status at any time.

In software versions 3.0.2-b892153 and later, the Upgrader uses an *upgrade plan* as a kind of checklist to perform all upgrade operations, which implies full management cluster upgrades as well as individual component upgrades. The concept of this approach is described in more detail in Upgrade Plan and History on page 335.

The upgrade plan is generated when the Upgrader itself is upgraded to the latest version. The plan is based on a comparison of the currently installed components on the rack, and the target component versions and latest packages downloaded to shared storage during the preparation of the upgrade environment. The resulting upgrade plan shows for which components an upgrade procedure needs to be executed in the next phase.

All components must be upgraded in a prescribed order. The upgrade plan will prevent a component upgrade procedure from starting if the preceding upgrades have not been completed. An error message informs the administrator which components need to be upgraded first.

> **Note:**
>
> In general, firmware may be upgraded whenever new versions are made available for your system. Firmware upgrades can be applied in no particular order and independently of other components. The ZFS Storage Appliance is an exception: its firmware must be upgraded before all other components.

This is the order of operations enforced through the upgrade plan:

1. Prepare upgrade environment (Upgrade PreConfig)
2. Upgrade the Upgrader (PreUpgrade)
3. ZFS Storage Appliance firmware
4. Compute nodes
5. Host operating system of management nodes
6. MySQL cluster database
7. Secret service (including Etcd and Vault)
8. Kubernetes container orchestration packages (platform layer)
9. Containerized microservices
10. Oracle Cloud Infrastructure images

Check the Upgrade Plan

After the upgrade environment has been prepared, all upgrade operations required to bring the system up-to-date are listed in the upgrade plan. Whenever an upgrade procedure has been completed successfully, the upgrade plan is updated with the latest status: for upgraded components the source and target versions are identical and the "upgrade required" flag is disabled.

At any point in time you can check how far the system has progressed through the upgrade plan. It indicates which components are already up-to-date and which still require upgrading. The upgrade plan provides a snapshot view, which is accurate and useful at the time you are consulting it. To see detailed progress of upgrade activity, use the job framework instead.

The upgrade plan can be viewed in two ways:

- In the Service Web UI, go to Maintenance in the navigation menu and select Upgrade Plan.

- In the Service CLI, enter the following command:

```
PCA-ADMIN> getUpgradePlan
Data:
  id                          component    currentBuild     targetBuild
   currentVersion                    targetVersion
     requireReboot   timeEstimation (min)   requireUpgrade   impactedInfra

  --                          ---------    ------------     -----------
   --------------                    -------------
     -------------   --------------------   --------------   -------------

  generic                     zfssa        3.0.2-b1300385   3.0.2-b1300385
   2013.06.05.8.73.1-2.73.5701.1    2013.06.05.8.73.1-2.73.5701.1
     false           45                     false            host,compute

  100.96.2.64                 compute      3.0.2-b1300130
   3.0.2-b1300385   oraclelinux-release-7.9-1.0.9
 3.0.202.3.11.0.0.0.5.7ea3f4ccbf3e17fc...   true            50
         true            compute
  100.96.2.65                 compute      3.0.2-b1300130
   3.0.2-b1300385   oraclelinux-release-7.9-1.0.9
 3.0.202.3.11.0.0.0.5.7ea3f4ccbf3e17fc...   true            50
         true            compute
  100.96.2.66                 compute      3.0.2-b1300130
   3.0.2-b1300385   oraclelinux-release-7.9-1.0.9
 3.0.202.3.11.0.0.0.5.7ea3f4ccbf3e17fc...   true            50
         true            compute
  100.96.2.33                 host         3.0.2-b1300130
   3.0.2-b1300385   oraclelinux-release-7.9-1.0.13
 3.0.202.3.11.0.0.0.5.7ea3f4ccbf3e17fc...   true            40
         true            host
  100.96.2.34                 host         3.0.2-b1300130
   3.0.2-b1300385   oraclelinux-release-7.9-1.0.13
 3.0.202.3.11.0.0.0.5.7ea3f4ccbf3e17fc...   true            40
         true            host
  100.96.2.35                 host         3.0.2-b1300130
   3.0.2-b1300385   oraclelinux-release-7.9-1.0.13
 3.0.202.3.11.0.0.0.5.7ea3f4ccbf3e17fc...   true            40
         true            host
  generic                     mysql        3.0.2-b1300385   3.0.2-b1300385
   8.0.36-1.1                        8.0.36-1.1
     false           15                     false            host

  generic                     etcd         3.0.2-b1300385   3.0.2-b1300385
   3.5.6                             3.5.6
     false           5                      false            host

  generic                     vault        3.0.2-b1300385   3.0.2-b1300385
   v1.7.1-3.10.0.0.0.4.g5f2d5af     v1.7.1-3.10.0.0.0.4.g5f2d5af
     false           5                      false            host

  generic                     kubernetes   3.0.2-b1300385   3.0.2-b1300385
   1.25.16-2                         1.25.16-2
     false           60                     false            host,compute

  generic                     platform     3.0.2-b1300130   3.0.2-b1300385
   None                              None
     false           40                     true             host,compute

  Oracle-Linux-7.9            ociImages    3.0.2-b1300130   3.0.2-b1300385
   None                              2024.07.31_0
```

```
    false           5                    true             host

Oracle-Linux8-OKE-1.28.8    ociImages    3.0.2-b1300130   3.0.2-b1300385
 None                             20241201
    false           5                    true             host

Oracle-Solaris-11           ociImages    3.0.2-b1300130   3.0.2-b1300385
 None                             2024.08.26_0
    false           5                    true             host

Oracle-Linux8-OKE-1.27.12   ociImages    3.0.2-b1300130   3.0.2-b1300385
 None                             20241201
    false           5                    true             host

Oracle-Linux8-OKE-1.26.15   ociImages    3.0.2-b1300130   3.0.2-b1300385
 None                             20241201
    false           5                    true             host

Oracle-Linux-9              ociImages    3.0.2-b1300130   3.0.2-b1300385
 None                             2024.07.31_0
    false           5                    true             host

Oracle-Linux-8              ociImages    3.0.2-b1300130   3.0.2-b1300385
 None                             2024.07.31_0
    false           5                    true             host

100.96.0.33                 ilom         3.0.2-b1300130   3.0.2-b1300385
 5.1.4.20                         5.1.4.25
    true            10                   true             host,compute

100.96.0.34                 ilom         3.0.2-b1300130   3.0.2-b1300385
 5.1.4.20                         5.1.4.25
    true            10                   true             host,compute

100.96.0.35                 ilom         3.0.2-b1300130   3.0.2-b1300385
 5.1.4.20                         5.1.4.25
    true            10                   true             host,compute

100.96.0.64                 ilom         3.0.2-b1300130   3.0.2-b1300385
 5.1.4.20                         5.1.4.25
    true            10                   true             host,compute

100.96.0.65                 ilom         3.0.2-b1300130   3.0.2-b1300385
 5.1.4.20                         5.1.4.25
    true            10                   true             host,compute

100.96.0.66                 ilom         3.0.2-b1300130   3.0.2-b1300385
 5.1.4.20                         5.1.4.25
    true            10                   true             host,compute

leaf                        switch       3.0.2-b1300385   3.0.2-b1300385
 10.3.4a                          10.3.4a
    false           60                   false            host,compute

spine                       switch       3.0.2-b1300385   3.0.2-b1300385
 10.3.4a                          10.3.4a
    false           60                   false            host,compute

mgmt                        switch       3.0.2-b1300385   3.0.2-b1300385
 10.3.4a                          10.3.4a
    false           60                   false            host,compute
```

Browse the Job Framework

The Upgrader is integrated with the job framework of the appliance controller software. Each operation results in a work request that triggers one or more jobs. These work requests and jobs can be queried. Their status and detailed properties provide insights into the Upgrader activity on the system.

When you enter a Service CLI command, a job ID and work request ID are returned. These allow you to check status, retrieve more details, find related jobs in the same workflow, and so on. When you create an upgrade (or patch) request using the Service Web UI, those IDs are not displayed on screen, but you can find them in the Upgrade Jobs table. From there, you can also drill down into the work requests and jobs.

Using the Service Web UI

To perform an upgrade operation from the Service Web UI, you use the main navigation menu, go to Maintenance, and click Upgrade & Patching. This opens the Upgrade Jobs page, which displays all recent upgrade jobs in a table, and allows you to create a new upgrade (or patch) request by clicking the button in the top-right corner.

The Upgrade Jobs table lets you browse all the Upgrader requests and jobs, so you can use it to track progress of current, past, and scheduled Upgrader activity.

- The default table view shows jobs with their associated command, workflow request, time stamps, and result. The Job ID and Request ID are hyperlinks to pages with more detailed information.

  The Actions menu on the right hand side of each table row has options equivalent to the hyperlinks, and also lets you copy the job or request ID.
- When filtered by request ID, the Upgrade Jobs table shows only the jobs that were created as part of the selected upgrade request. Use it to track multicomponent workflows and check which jobs are completed or pending. If a failure occurred, you can see which job did not complete as expected.

  To return to the default table view, click Clear Filter.
- To view the details of a job, click its job ID in the Upgrade Jobs table. Besides the information shown in the jobs table, the Upgrade Job detail page contains the log file location, job execution details, and the list of tasks performed as part of the upgrade job.

  If the upgrade job has not completed, you can kill it by clicking the button in the top right corner of the page.
- To view the details of an upgrade request, click the request ID in the Upgrade Jobs table. Besides the information shown in the jobs table, the Upgrade Status detail page contains source and target builds, workflow details, and a list of associated upgrade jobs.

  The upgrade job list is the same as the main Upgrade Jobs table filtered by request ID. It also lets you access each job's detail page by clicking the job ID.

  In addition to the job list, the Resources section of the Upgrade Status page lets you display for which components the job has been completed, and which components are still pending.

Using the Service CLI

When you perform an upgrade operation from the Service CLI, a request is created. The job ID and upgrade request ID are returned immediately. Use these IDs to query the jobs and requests, and find the details you need.

- When working in the Service CLI, the request IDs from the Upgrader Workflow Service are the primary element to access information about Upgrader activity. In case you did not note the request ID when entering the command, or you are searching for data from previous workflows, use the `getUpgradeRequests` command to look up the one you want to investigate.

```
PCA-ADMIN> getUpgradeRequests
Data:
  id
  --
  PREUPGRADE-61b56867-7966-4096-9c74-08efe8d2d0ad
```

```
  UWS-496e26aa-2923-46f1-a485-c086a005077a
  UWS-da5d70ba-09ea-4ef0-b8ea-84d9f44c5e62
  PREUPGRADE-a6aa2c0e-ceff-4de4-aa23-de7d21b06668
[...]
```

- With the correct upgrade request ID you can check the status and progress of the upgrade workflow. Use the `getUpgradeStatus` command to display the workflow details, including the job list, and completed and pending components.

```
PCA-ADMIN> getUpgradeStatus requestId=UWS-910dd8fd-28b3-4f97-
a909-02c26085266c
Data:
  Request id = UWS-910dd8fd-28b3-4f97-a909-02c26085266c
  Status = Failed
  Composition ID = fullrack
  Type = ISO
  Previous Build = 3.0.2-b1333709
  Target Build = 3.0.2-b1337789
  Jobs 1 = 1741252377283-zfssa-81712
  Jobs 2 = 1741253904337-compute-72870
[...]
  Jobs 13 = 1741269765197-platform-663307
  Completed Components 1 = zfssa(generic)
  Completed Components 2 = compute(100.96.2.64)
[...]
  Pending Components 1 = compute(100.96.2.68)
  Pending Components 2 = host(100.96.2.33)
[...]
```

- Using the upgrade request ID you can also list the jobs associated with the request, and scan it for status, progress, and potential issues. Use the `getUpgradeJobs` command to display the job list.

```
PCA-ADMIN> getUpgradeJobs requestId=UWS-910dd8fd-28b3-4f97-
a909-02c26085266c
Data:
  id                                Upgrade Request Id
           Command Name   Result
  --                                ------------------
           ------------   ------
  1741269765197-platform-663307     UWS-910dd8fd-28b3-4f97-
a909-02c26085266c   platform       Failed
  1741269703036-kubernetes-656961   UWS-910dd8fd-28b3-4f97-
a909-02c26085266c   kubernetes     Passed
  1741267992739-host-422432         UWS-910dd8fd-28b3-4f97-
a909-02c26085266c   host           Passed
  1741266155984-host-161315         UWS-910dd8fd-28b3-4f97-
a909-02c26085266c   host           Passed
  1741264266152-host-11547          UWS-910dd8fd-28b3-4f97-
a909-02c26085266c   host           Passed
  1741264201476-compute-87720       UWS-910dd8fd-28b3-4f97-
a909-02c26085266c   compute        Passed
  1741264136546-compute-81124       UWS-910dd8fd-28b3-4f97-
a909-02c26085266c   compute        Passed
  1741264070151-compute-74130       UWS-910dd8fd-28b3-4f97-
a909-02c26085266c   compute        Passed
  1741252377283-zfssa-81712         UWS-910dd8fd-28b3-4f97-
a909-02c26085266c   zfssa          Passed
```

- To investigate a failure, or display more detailed information about a particular job, use the `getUpgradeJob` command. The job details include the path to the log file, the complete task list, and error messages captured during the run.

```
PCA-ADMIN> getUpgradeJob upgradeJobId=1741269765197-platform-663307
```

```
Data:
  Upgrade Request Id = UWS-910dd8fd-28b3-4f97-a909-02c26085266c
  Composition Id = fullrack
  Name = platform
  Pid = 663307
  Host = pcamn01
  Log File = /nfs/shared_storage/pca_upgrader/log/pca-
upgrader_platform_services_2025_03_06-14.02.45.log
  Arguments =
 {"component_names":null,"diagnostics":false,"display_task_plan":false,"dry_run_
[...]}
  Status = Failed
  Execution Time(sec) = 4068
  Tasks 1 - Name = Validate Upgrader Version
[...]
  Tasks 31 - Name = Network Controllers Upgrade
  Tasks 31 - Description = Upgrade Network Controllers.
  Tasks 31 - Message = [{'name': 'auxiliary-metadata', 'version': 2,
 [...]
      Error: UPGRADE FAILED: release pcaswitchmgr failed, and has
 been rolled back due to atomic being set:
      pre-upgrade hooks failed: timed out waiting for the
 condition }]
  Tasks 31 - Status = Failed
  Tasks 31 - Reason Phrase = Failed
  Tasks 31 - Execution Time(sec) = 1834
[...]
```

Performing a Private Cloud Appliance Full Rack Upgrade

A full rack upgrade is the recommended way to upgrade all components of Private Cloud Appliance. The tools required for this approach are installed along with the latest Upgrader during the upgrade preparation stages.

> **Caution:**
>
> Ensure that all preparation steps for system upgrade have been completed. For instructions, see Preparing for Private Cloud Appliance Upgrade on page 339.

A single administrator command initiates a fully orchestrated workflow. The Upgrader Workflow Service (UWS) autonomously performs all tasks defined in the upgrade plan, which is generated during the pre-upgrade process. It determines which specific components need to be upgraded, and in what order.

The rack upgrade workflow includes these components:

1. ZFS Storage Appliance firmware (including ILOMs of storage controllers)
2. Compute nodes
3. Management nodes
   - Host operating system of the 3 nodes
   - MySQL cluster database
   - Secret service (including Etcd and Vault)
   - Kubernetes container orchestration packages (platform layer)
   - Containerized microservices
4. Oracle Cloud Infrastructure images
5. ILOM firmware (all nodes, includes internal hardware components)
6. Switch firmware (all switches)

The order shown applies to software version 3.0.2-b1443573. Components could be handled in a different order depending on the specific release. Consult the upgrade plan for the correct order.

Components are skipped if they are already at the required version, but a same-version upgrade can be forced using the Service Web UI or Service CLI command option, if necessary. In case a component must be upgraded separately from the full rack upgrade workflow, see Upgrading Private Cloud Appliance Components Individually on page 371.

Although the full rack upgrade workflow is launched with one convenient command, the internal orchestration of many small commands requires intricate logic and precise timing. To become familiar with the key operations in the workflow, and their impact on system operation, we recommend that administrators read and understand the following topics:

- Full Management Cluster Migration to Oracle Linux 8 on page 369
- End-to-End Compute Node Upgrade on page 370

Launching the Full Rack Upgrade

Run the full rack upgrade from the UI or CLI.

Using the Service Web UI

1. In the navigation menu, go to the Maintenance section and click Upgrade Plan. This provides an overview of current and target component versions.
2. Perform a rack-wide health check to ensure the appliance is ready to accept a new software version. See Rack-Wide Health Check on page 355.

   > **Caution:**
   >
   > Oracle strongly recommends verifying in advance that the appliance is ready for upgrade. The preferred way is to perform a rack-wide health check. The alternate option is to run the upgrade command in verify-only mode. It is allowed to do both.

3. Click Upgrade & Patching to display the Upgrade Jobs page.
4. In the top-right corner of the Upgrade Jobs page, click Create Upgrade or Patch.

   The Create Request window appears. Choose *Upgrade* as the Request Type.
5. Select the appropriate upgrade request type.

   For a full rack upgrade, select Upgrade Rack.
6. Fill out the upgrade request parameters:

   - **Action:** To perform the upgrade or patch operation, select Apply.
   - **Type:** For upgrade, select ISO. The ULN option applies to patching.
   - **Component:** For full rack upgrade, select All.
   - **Qualifier:** This parameter is not required when upgrading all components.
   - **Verify Only:** Enable this option to run the operation in verification only mode.
   - **Force:** Enable this option to force the operation. Use only when instructed by Oracle.
7. Click Create Request.

   All jobs related to the new upgrade request appear in the Upgrade Jobs table. Click the upgrade request ID to display a filtered job list. Click a job name to see the details of a job.

   > **Note:**
   >
   > After upgrade, if the upgrade plan specifies that a component must be rebooted for the changes to take effect, a reboot is performed as part of the upgrade process. No administrator action is required.

8. Confirm that the upgrade workflow is completed without errors.

   Consult the Upgrade Jobs table. Click the upgrade request ID to display a filtered job list. Click a job name to see the details of a job. For more information, see Browse the Job Framework on page 360.

After successful completion, all compute nodes, the entire management node cluster, and all platform and infrastructure services are on Oracle Linux 8. Any required firmware updates have been applied. The system is fully operational again.

Using the Service CLI

1. Perform a rack-wide health check to ensure the appliance is ready to accept a new software version. See Rack-Wide Health Check on page 355.

```
PCA-ADMIN> upgradeRack action=HEALTHCHECK
JobId: de9eca17-e357-42c4-8aab-c73c311f787f
Data: Service request has been submitted. Upgrade Request Id = UWS-
ab2e72d9-d937-4afb-b6eb-112c87988f56
```

> **Caution:**
>
> Oracle strongly recommends verifying in advance that the appliance is ready for upgrade. The preferred way is to perform a rack-wide health check. The alternate option is to run the upgrade command in verify-only mode. It is allowed to do both.

2. Optionally, as an alternative to the health check, run the upgrade command in verify-only mode. This applies no changes but indicates whether the command will succeed based on the current status of the appliance. Before you start the actual upgrade, all component verification jobs must be completed, which could take around 45 minutes in total.

   a. Start the upgrade request in verify-only mode.

```
PCA-ADMIN> upgradeRack type=ISO action=APPLY verifyOnly=True
JobId: 4e99c667-e3df-42c4-4a37-c73c311f49f1
Data: Service request has been submitted. Upgrade Request Id =
 UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056
```

   b. Find the associated upgrade jobs and note the IDs.

```
PCA-ADMIN> getUpgradeJobs requestId=UWS-9776b0fe-7f5f-4e46-9e3f-
ceb4b1702056
Data:
  id                                      Upgrade Request Id
                    Command Name        Result
  --                                      ------------------
                    ------------        ------
  1721306505338-cisco_verify-1482721
 UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056   cisco_verify
 Passed
  1721306428955-cisco_verify-1460275
 UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056   cisco_verify
 Passed
  1721306388669-cisco_verify-1447173
 UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056   cisco_verify
 Passed
  1721305430640-ilom_verify-1425717
 UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056   ilom_verify
 Passed
  1721304295877-ilom_verify-1413464
 UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056   ilom_verify
 Passed
  1721303749853-ilom_verify-1264181
 UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056   ilom_verify
 Passed
  1721303241144-ilom_verify-1197515
 UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056   ilom_verify
 Passed
```

```
 1721302164633-ilom_verify-1133965
UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056   ilom_verify
Passed
 1721301487584-ilom_verify-1019683
UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056   ilom_verify
Passed
 1721300618710-oci_verify-971867
UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056   oci_verify
Passed
 1721297014027-platform_verify-325471
UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056   platform_verify
Passed
 1721296952862-kubernetes_verify-387991
UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056   kubernetes_verify
Passed
 1721295894573-host_verify-125736
UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056   host_verify
Passed
 1721295631562-host_verify-366123
UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056   host_verify
Passed
 1721295558048-host_verify-81361
UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056   host_verify
Passed
 1721294395027-compute_verify-76178
UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056   compute_verify
Passed
 1721294332372-compute_verify-55394
UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056   compute_verify
Passed
 1721290297566-compute_verify-4534
UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056   compute_verify
Passed
 1721290264046-zfssa_verify-1752
UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056   zfssa_verify
Passed
```

c. Check the status in the upgrade job details.

```
PCA-ADMIN> getUpgradeJob upgradeJobId=1721295631562-
host_verify-366123
Data:
  Upgrade Request Id = UWS-9776b0fe-7f5f-4e46-9e3f-ceb4b1702056
  Name = host_verify
  Pid = 366123
  Log File = /nfs/shared_storage/pca_upgrader/log/pca-
upgrader_host_os_pcamn03_verify_2025_01_21-19.38.47.log
  Status = Passed
```

3. When the upgrade verification has completed successfully, start the actual full rack upgrade.

```
PCA-ADMIN> upgradeRack type=ISO action=APPLY
JobId: de9eca17-e357-42c4-8aab-c73c311f787f
Data: Service request has been submitted. Upgrade Request Id =
 UWS-8cbb37e3-1d9b-43bc-a793-2a84633fa718
```

Note the IDs of the upgrade request and upgrade job.

4. Use the request ID to check the status of the upgrade workflow.

```
PCA-ADMIN> getUpgradeStatus requestId=UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718
Data:
  Request id = UWS-8cbb37e3-1d9b-43bc-a793-2a84633fa718
```

```
  Status = Running
  Composition ID = fullrack
  Type = ISO
  Previous Build = 3.0.2-b1300130
  Target Build = 3.0.2-b1300385
  Jobs 1 = 1737489464069-zfssa-1898
  Jobs 2 = 1737489497566-compute-8759
  Jobs 3 = 1737492632323-compute-49553
  Jobs 4 = 1737492695074-compute-61798
  Jobs 5 = 1737492758087-host-71365
  Completed Components 1 = zfssa(generic)
  Completed Components 2 = compute(100.96.2.64)
  Completed Components 3 = compute(100.96.2.65)
  Completed Components 4 = compute(100.96.2.66)
  Completed Components 5 = kubernetes(generic)
  Completed Components 6 = switch(leaf)
  Completed Components 7 = switch(spine)
  Completed Components 8 = switch(mgmt)
  Pending Components 1 = host(100.96.2.33)
  Pending Components 2 = host(100.96.2.34)
  Pending Components 3 = host(100.96.2.35)
  Pending Components 4 = platform(generic)
  Pending Components 5 = ociImages(Oracle-Linux-7.9)
  Pending Components 6 = ociImages(Oracle-Linux8-OKE-1.28.8)
  Pending Components 7 = ociImages(Oracle-Solaris-11)
  Pending Components 8 = ociImages(Oracle-Linux8-OKE-1.27.12)
  Pending Components 9 = ociImages(Oracle-Linux8-OKE-1.26.15)
  Pending Components 10 = ociImages(Oracle-Linux-9)
  Pending Components 11 = ociImages(Oracle-Linux-8)
  Pending Components 12 = ilom(100.96.0.33)
  Pending Components 13 = ilom(100.96.0.34)
  Pending Components 14 = ilom(100.96.0.35)
  Pending Components 15 = ilom(100.96.0.64)
  Pending Components 16 = ilom(100.96.0.65)
  Pending Components 17 = ilom(100.96.0.66)
```

5. Optionally, check the status of an individual upgrade job.

   Because the full rack upgrade is a multicomponent upgrade process, there are multiple upgrade jobs associated with the upgrade request. You can filter for those jobs based on the request ID. Using the job ID, you can drill down into the details of each upgrade job.

```
PCA-ADMIN> getUpgradeJobs requestId=UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718
Data:
  id                                Upgrade Request Id
          Command Name   Result
  --                                ------------------
          ------------   ------
  1737496294531-host-396629         UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   host           Passed
  1737494531521-host-162838         UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   host           Passed
  1737492758087-host-71365          UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   host           Passed
  1737492695074-compute-61798       UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   compute        Passed
  1737492632323-compute-49553       UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   compute        Passed
  1737489497566-compute-8759        UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   compute        Passed
  1737489464069-zfssa-1898          UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   zfssa          Passed
```

```
PCA-ADMIN> getUpgradeJob upgradeJobId=1737494531521-host-162838
Data:
  Upgrade Request Id = UWS-8cbb37e3-1d9b-43bc-a793-2a84633fa718
  Composition Id = fullrack
  Name = host
  Pid = 162838
  Host = pcamn01
  Log File = /nfs/shared_storage/pca_upgrader/log/pca-
upgrader_host_os_pcamn02_2025_01_21-21.22.11.log
  Arguments =
 {"component_names":null,"diagnostics":false,"display_task_plan":false,"dry_run_
a793-2a84633fa718","uln":null,"patch":"false"}
  Status = Passed
  Execution Time(sec) = 1736
  Tasks 1 - Name = Validate Image Location
  Tasks 2 - Name = Check Registry Service Running Status
  Tasks 3 - Name = Validate Upgrader Version
  Tasks 4 - Name = Validate Upgrade Order
[...]
```

The output of the `getUpgradeJob` command provides detailed information about the tasks performed during the upgrade procedure. It displays descriptions, time stamps, duration, and success or failure. Whenever an upgrade operation fails, the command output indicates which task has failed. For in-depth troubleshooting you can search the log file at the location provided near the start of the command output.

> **Note:**
>
> After upgrade, if the upgrade plan specifies that a component must be rebooted for the changes to take effect, a reboot is performed as part of the upgrade process. No administrator action is required.

6. Confirm that the upgrade workflow is completed without errors.

   For more information, see Browse the Job Framework on page 360.

   > **Note:**
   >
   > In case you forgot the ID of the upgrade request or upgrade job, you can search using the commands `getUpgradeRequests` and `getUpgradeJobs`.

   Check the job associated with the `upgradeRack` command.

```
PCA-ADMIN> show job id=de9eca17-e357-42c4-8aab-c73c311f787f
Data:
  Id = de9eca17-e357-42c4-8aab-c73c311f787f
  Type = Job
  Associated Work Request Id = UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718
  Done = true
  Name = OPERATION-upgradeRack
  Progress Message = Done
  Run State = Succeeded
  Transcript = Created job OPERATION
  Username = admin
  WorkItemIds 1 = id:4fd5d71b-81b3-4a37-a0df-c4149e9c6609
 type:WorkItem  name:
```

   Use the request ID to check the status of the upgrade workflow.

```
PCA-ADMIN> getUpgradeStatus requestId=UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718
Data:
  Request id = UWS-8cbb37e3-1d9b-43bc-a793-2a84633fa718
```

```
  Status = Passed
  Composition ID = fullrack
  Type = ISO
  Previous Build = 3.0.2-b1300385
  Target Build = 3.0.2-b1301270
  Jobs 1 = 1737489464069-zfssa-1898
  Jobs 2 = 1737489497566-compute-8759
  Jobs 3 = 1737492632323-compute-49553
  Jobs 4 = 1737492695074-compute-61798
  Jobs 5 = 1737492758087-host-71365
  Jobs 6 = 1737494531521-host-162838
  Jobs 7 = 1737496294531-host-396629
  Jobs 8 = 1737497752826-kubernetes-637866
  Jobs 9 = 1737497814071-platform-646623
  Jobs 10 = 1737500431705-oci-979395
  Jobs 11 = 1737501529548-ilom-1060489
  Jobs 12 = 1737502338638-ilom-1133349
  Jobs 13 = 1737503158143-ilom-1197515
  Jobs 14 = 1737503974834-ilom-1266068
  Jobs 15 = 1737504796873-ilom-1361643
  Jobs 16 = 1737505608602-ilom-1435802
  Jobs 17 = 1737506421692-cisco-1507991
  Jobs 18 = 1737506482956-cisco-1524202
  Jobs 19 = 1737506544387-cisco-1532342
  Completed Components 1 = zfssa(generic)
  Completed Components 2 = compute(100.96.2.64)
  Completed Components 3 = kubernetes(generic)
  Completed Components 4 = ociImages(Oracle-Linux-8)
  Completed Components 5 = ociImages(Oracle-Linux-7.9)
  Completed Components 6 = ociImages(Oracle-Linux8-OKE-1.27.12)
  Completed Components 7 = ociImages(Oracle-Linux8-OKE-1.28.8)
  Completed Components 8 = ociImages(Oracle-Linux-9)
  Completed Components 9 = ociImages(Oracle-Linux8-OKE-1.26.15)
  Completed Components 10 = ociImages(Oracle-Solaris-11)
  Completed Components 11 = ilom(100.96.0.33)
  Completed Components 12 = ilom(100.96.0.34)
  Completed Components 13 = ilom(100.96.0.35)
  Completed Components 14 = ilom(100.96.0.64)
  Completed Components 15 = ilom(100.96.0.65)
  Completed Components 16 = ilom(100.96.0.66)
  Completed Components 17 = switch(leaf)
  Completed Components 18 = switch(spine)
  Completed Components 19 = switch(mgmt)
  Start Time = 2025-01-21T19:57:44
  End Time = 2025-01-22T00:42:25
```

Check the upgrade jobs associated with the upgrade request.

```
PCA-ADMIN> getUpgradeJobs requestId=UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718
Data:
  id                                Upgrade Request Id
          Command Name   Result
  --                                ------------------
          ------------   ------
  1737506544387-cisco-1532342       UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   cisco          Passed
  1737506482956-cisco-1524202       UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   cisco          Passed
  1737506421692-cisco-1507991       UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   cisco          Passed
  1737505608602-ilom-1435802        UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   ilom           Passed
```

```
  1737504796873-ilom-1361643        UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   ilom           Passed
  1737503974834-ilom-1266068        UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   ilom           Passed
  1737503158143-ilom-1197515        UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   ilom           Passed
  1737502338638-ilom-1133349        UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   ilom           Passed
  1737501529548-ilom-1060489        UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   ilom           Passed
  1737500431705-oci-979395          UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   oci            Passed
  1737497814071-platform-646623     UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   platform       Passed
  1737497752826-kubernetes-637866   UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   kubernetes     Passed
  1737496294531-host-396629         UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   host           Passed
  1737494531521-host-162838         UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   host           Passed
  1737492758087-host-71365          UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   host           Passed
  1737492695074-compute-61798       UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   compute        Passed
  1737492632323-compute-49553       UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   compute        Passed
  1737489497566-compute-8759        UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   compute        Passed
  1737489464069-zfssa-1898          UWS-8cbb37e3-1d9b-43bc-
a793-2a84633fa718   zfssa          Passed
```

After successful completion, all compute nodes, the entire management node cluster, and all platform and infrastructure services are on Oracle Linux 8. Any required firmware updates have been applied. The system is fully operational again.

Full Management Cluster Migration to Oracle Linux 8

Software release 3.0.2-b1443573 migrates the entire Private Cloud Appliance to Oracle Linux 8. It affects not only the operating system of components, but also the platform layer, container images, microservices, firmware, and RPM packages. With the new version of Oracle Linux, several improvements in the upgrade process are introduced, as well as some extra requirements.

System Backup and Cluster Teardown

The host OS is technically not upgraded but installed from scratch. First, a backup of critical data and files is created on the shared storage in the ZFS Storage Appliance, so that the system configuration can be restored at the end of the upgrade workflow. One by one, the management nodes must be removed from the clusters they belong to, so that both the nodes and the clusters can be migrated to Oracle Linux 8.

Fresh Installation from Image

Each management node is rebooted from a new image, which allows a new disk layout and partitioning to be applied. When the OS and all other required tools are installed in the new partitions, the backups are restored and the node joins the new clusters (PCS, MySQL, Kubernetes). Checks are performed to ensure that the node can resume all its functions. At this point, all platform and infrastructure services are on Oracle Linux 8.

Cluster Reconstruction

To reduce complexity and risk of failure, the management nodes are migrated to Oracle Linux 8 in a fixed order: pcamn01, pcamn02, pcamn03. If the first node owns the cluster virtual IP, it is reassigned to another

node. When the upgrade workflow is completed, the first node takes back the virtual IP, joins the new cluster, and waits for the other nodes to complete the new cluster setup.

Alternate Boot Partitions

During the installation of Oracle Linux 8, the root disk of the physical nodes is set up with two equivalent sets of partitions (A and B). On a running host, if the A partitions are in use, then the B partitions can be manipulated without affecting the operating state of the host. The Upgrader takes advantage of the alternate partitions on Oracle Linux 8 hosts by preparing the next OS installation on the available partitions. When ready, the boot manager is reconfigured so the host can reboot from the partition with the updated OS. As a result, the host can remain fully operational for a longer time, and the window of reduced availability is reduced to a minimum. This mechanism is used for management nodes and compute nodes.

The full management node cluster workflow includes these components:

1. Host operating system
2. Clustered MySQL database
3. Secret Service (including Etcd and Vault)
4. Kubernetes container orchestration packages
5. Containerized microservices
6. Oracle Cloud Infrastructure images

> **Caution:**
>
> When the new images have been imported, a background job is launched to ensure that running OKE clusters receive the latest available CVE fixes delivered with the new images. For more information, see Upgrading Oracle Cloud Infrastructure Images on page 384.

End-to-End Compute Node Upgrade

The compute node upgrade is designed as another orchestrated workflow within the full rack upgrade workflow. Each compute node goes through its own end-to-end process. All nodes are handled in the same way one after the other. Assuming the nodes are running Oracle Linux 8, the Upgrader takes advantage of the alternate partitions – as described for the management cluster – for maximum efficiency and minimum inpact on availability.

To allow all compute nodes to be upgraded without intervention from an administrator, the workflow takes care of compute instance migration and displaced instances autonomously. If the compute nodes or instances are in a state that prevents a successful end-to-end run, the workflow is stopped and a notification is returned, so the administrator can take corrective measures.

An end-to-end compute node upgrade consists of these steps:

1. Check the status of the compute node and instances.
   - If instances are running that cannot be migrated, the workflow is stopped.
   - If the compute node is locked for maintenance, or otherwise not ready for upgrade, the workflow is stopped.
2. Set a provisioning lock on the compute node.

   > **Caution:**
   >
   > Depending on the high-availability configuration of the Compute service, automatic instance migrations can prevent the Upgrader from successfully locking a compute node. See Monitoring Displaced Instances on page 371.
3. Migrate all running compute instances away from the compute node. If compute node evacuation fails, the workflow is stopped and the operation is rolled back.

> **Caution:**
>
> If the workflow fails at this stage, the provisioning lock is not automatically removed. The lock prevents further instance placement changes.

4. Deprovision and rediscover the compute node.
5. Upgrade the host: reboot from the new OS base image and install any additional tools required.
6. Provision the compute node so it can resume normal operation.

Monitoring Displaced Instances

During compute node upgrade or patching, no active compute instances can be present, so the node must be evacuated and placed under a provisioning lock. To evacuate a compute node, the Compute Service live-migrates instances to another compute node in the same fault domain. If the fault domain does not have sufficient capacity, high-availability configuration settings might cause instances to be live-migrated to another fault domain, and migrated back to their selected fault domain when the required capacity is available again.

Compute instances that have been migrated away from their assigned fault domain, are called *displaced instances*. Their migrations can interfere with compute node upgrade or patching. When the locks on a given compute node are released, its displaced instances start migrating back, during which time it might be impossible to lock the next compute node for maintenance.

Before upgrading or patching a compute node, monitor the status of displaced instances. **Do not** proceed with the next compute node until the list is empty.

- In the Service CLI, use the command `getDisplacedInstances`. In the following example, two instances have been migrated away from fault domain 1.

```
PCA-ADMIN> getDisplacedInstances
Data:
 id                          displayName  compartmentId
 faultDomain     faultDomainSelected
 --                          -----------  -------------
 -----------     -------------------
 ocid1.instance.unique_ID  inst-name    ocid1.compartment.unique_ID
 FAULT-DOMAIN-3  FAULT-DOMAIN-1
 ocid1.instance.unique_ID  inst-name    ocid1.compartment.unique_ID
 FAULT-DOMAIN-2  FAULT-DOMAIN-1
```

- In the Service Web UI, click the navigation menu, click FD Instances, and then click Displaced Instances.

For more information, see the following infrastructure administration topics:

- Performing Administrative Operations on Compute Nodes on page 156 (migrating instances and locking a compute node)
- Configuring High Availability in the Compute Service on page 169

Upgrading Private Cloud Appliance Components Individually

Upgrades were made granular by design. All Private Cloud Appliance components have their own upgrade procedure that is independent of other components. In a full rack upgrade, however, these component procedures are consolidated into one large workflow, directed by the upgrade plan. This is convenient, and eliminates issues resulting from poorly timed commands or operations performed in the wrong order.

The normal and recommended way to upgrade the appliance software is to perform a full rack upgrade. However, deviations from the full rack upgrade workflow might be needed to work around an unforeseen orchestration problem, or to meet a particular component requirement. If this is the case, Oracle will provide guidance on how to upgrade components individually.

Before starting any procedure in this chapter, ensure that all preparation steps for system upgrade have been completed. For instructions, see Preparing for Private Cloud Appliance Upgrade on page 339.

Individual component upgrades must also comply with the prescribed order of operations. A component can only be upgraded on condition that the preceding components in the list are already at the required version. For more information, see Checking Status and Progress of Private Cloud Appliance Upgrade Plan on page 357.

In general, firmware may be upgraded whenever new versions are made available for the system. Firmware upgrades can be applied in no particular order and independently of other components.

Components are skipped if they are already at the required version, but a same-version upgrade can be forced using the Service Web UI or Service CLI command option, if necessary.

Upgrading Compute Nodes

The compute node upgrade on Private Cloud Appliance ensures that the latest Oracle Linux kernel and user space packages are installed, as well as the appliance-specific tools and optimizations.

Compute nodes must be locked and upgraded one at a time; concurrent upgrades are not supported. Before performing a compute node upgrade, ensure that you understand its key operations and their impact on system operation. See End-to-End Compute Node Upgrade on page 370.

To upgrade one or more compute nodes outside of the full rack upgrade workflow, follow these instructions. You can either upgrade a single compute node by providing its host IP address, or all compute nodes installed in the appliance.

Upgrading a Single Compute Node

Using the Service Web UI

1. Get the IP address of the compute node you intend to upgrade.
2. In the navigation menu, go to the Maintenance section and click Upgrade Plan. This provides an overview of current and target component versions.
3. Click Upgrade & Patching to display the Upgrade Jobs page.
4. In the top-right corner of the Upgrade Jobs page, click Create Upgrade or Patch.

   The Create Request window appears. Choose *Upgrade* as the Request Type.
5. Select the appropriate upgrade request type: Upgrade Rack.
6. Fill out the upgrade request parameters:

   - **Action:** To perform the upgrade or patch operation, select Apply.
   - **Type:** For upgrade, select ISO. The ULN option applies to patching.
   - **Component:** For a single compute node, select Compute.
   - **Qualifier:** Enter the compute node's assigned IP address in the internal administration network. This is an IP address in the internal 100.96.2.0/23 range.
   - **Verify Only:** Enable this option to run the operation in verification only mode.
   - **Force:** Enable this option to force the operation. Use only when instructed by Oracle.
7. Click Create Request.

   The new upgrade request appears in the Upgrade Jobs table.
8. Confirm that the upgrade workflow is completed without errors.

   For more information, see Browse the Job Framework on page 360.

   After successful rediscovery, reprovisioning, and reboot, the compute node runs the updated version of the operating system and tools, and is fully operational again.

Using the Service CLI

1. Get the IP address of the compute node you intend to upgrade.

To obtain the host IP address and other information relevant to the upgrade procedure, use the Service CLI command provided in the following example. You can run the command as often as needed to check and confirm status as you proceed through the upgrade of all compute nodes.

```
PCA-ADMIN> list computeNode fields
 hostname,ipAddress,ilomIp,state,firmwareVersion,provisioningLocked,maintenanceL
 orderby hostname ASCENDING
Data:
  id                                     Hostname   Ip Address
 ILOM Ip Address   State   Firmware Version           Provisioning
 Locked   Maintenance Locked
  --                                     --------   ----------
    ---------------   -----   ----------------
 ------------------   ------------------
  cf488903-fef8-4a51-8a41-c6990e4755c5   pcacn001   100.96.2.64
 100.96.0.64       On      PCA Hypervisor:3.0.2-859   false
        false
  42a7594d-1173-4dbd-4755-07810cc2d527   pcacn002   100.96.2.65
 100.96.0.65       On      PCA Hypervisor:3.0.2-859   false
        false
  bc0f37d5-ba77-423e-bc11-017704b47e59   pcacn003   100.96.2.66
 100.96.0.66       On      PCA Hypervisor:3.0.2-859   false
        false
  2e5ac527-01f5-4230-ae41-0522fcb57c9a   pcacn004   100.96.2.67
 100.96.0.67       On      PCA Hypervisor:3.0.2-859   false
        false
  5a6b61cf-7e99-4df2-87e4-b37c5fb0bfb8   pcacn005   100.96.2.68
 100.96.0.68       On      PCA Hypervisor:3.0.2-859   false
        false
  885f2aa4-f017-41e8-b2bc-e588cc0c6162   pcacn006   100.96.2.69
 100.96.0.69       On      PCA Hypervisor:3.0.2-859   false
        false
```

2. Enter the compute node upgrade command.

   Syntax (entered on a single line):

   ```
   upgradeRack type=ISO action=APPLY component=COMPUTE
    qualifier=<compute-node-ip>
   ```

   Example:

   ```
   PCA-ADMIN> upgradeRack type=ISO action=APPLY component=COMPUTE
    qualifier=100.96.2.66
   JobId: 375371c3-1cf6-4866-9293-315e3de4966a
   Data: Service request has been submitted. Upgrade Job Id =
    1737619948523-compute-598228 Upgrade Request Id = UWS-ebdef142-
   db7a-478a-9f95-fe10714d8f44
   ```

3. Use the job ID to check the status of the upgrade process.

   ```
   PCA-ADMIN> getUpgradeJob upgradeJobId=1737619948523-compute-598228
   Data:
     Upgrade Request Id = UWS-ebdef142-db7a-478a-9f95-fe10714d8f44
     Name = compute
     Pid = 598228
     Host = pcamn01
     Log File = /nfs/shared_storage/pca_upgrader/log/pca-
   upgrader_compute_pcacn003_2025_01_23-08.12.28.log
     Arguments =
    {"component_names":null,"diagnostics":false,"display_task_plan":false,"dry_run_
   ebdef142-db7a-478a-9f95-fe10714d8f44","uln":null,"patch":"false"}
   ```

```
  Status = Passed
  Execution Time(sec) = 3130
[...]
```

4. Ensure that the upgrade job passes the initial checks performed before the actual node upgrade starts.
   - If the Upgrader detects that the node is not in an appropriate state for upgrade, take corrective measures and run the upgrade command again.
   - If the Upgrader detects instances that cannot be migrated, shut down these instances and run the upgrade command again.
5. Confirm that the upgrade is completed without errors.

   For more information, see Browse the Job Framework on page 360.

   After successful rediscovery, reprovisioning, and reboot, the compute node runs the updated version of the operating system and tools, and is fully operational again.

Upgrading All Compute Nodes

Using the Service Web UI

1. In the navigation menu, go to the Maintenance section and click Upgrade Plan. This provides an overview of current and target component versions.
2. Click Upgrade & Patching to display the Upgrade Jobs page.
3. In the top-right corner of the Upgrade Jobs page, click Create Upgrade or Patch.

   The Create Request window appears. Choose *Upgrade* as the Request Type.
4. Select the appropriate upgrade request type: Upgrade Rack.
5. Fill out the upgrade request parameters:
   - **Action:** To perform the upgrade or patch operation, select Apply.
   - **Type:** For upgrade, select ISO. The ULN option applies to patching.
   - **Component:** For all installed compute nodes, select All Computes.
   - **Qualifier:** This parameter is not required when upgrading all compute nodes.
   - **Verify Only:** Enable this option to run the operation in verification only mode.
   - **Force:** Enable this option to force the operation. Use only when instructed by Oracle.
6. Click Create Request.

   The new upgrade request appears in the Upgrade Jobs table.
7. Confirm that the upgrade workflow is completed without errors.

   For more information, see Browse the Job Framework on page 360.

   After successful rediscovery, reprovisioning, and reboot, the compute nodes run the updated version of the operating system and tools, and are fully operational again.

Using the Service CLI

1. Enter the compute upgrade command.

```
PCA-ADMIN> upgradeRack type=ISO action=APPLY component=ALL_COMPUTES
JobId: 60f72145-deb9-469e-8d46-b449a7da11fe
Data: Service request has been submitted. Upgrade Request Id =
 UWS-606c2c4e-c94e-4607-b1df-f4cbd4b0edff
```

2. Use the request ID to check the status of this multicomponent upgrade process.

```
PCA-ADMIN> getUpgradeStatus requestId=UWS-606c2c4e-c94e-4607-b1df-
f4cbd4b0edff
Data:
  Request id = UWS-606c2c4e-c94e-4607-b1df-f4cbd4b0edff
  Status = Running
  Composition ID = all_computes
  Type = ISO
```

```
  Previous Build = 3.0.2-b1300385
  Target Build = 3.0.2-b1301270
  Jobs 1 = 1737621345177-compute-860804
  Jobs 2 = 1737621353957-compute-861842
  Jobs 3 = 1737621415558-compute-868540
  Completed Components 1 = compute(100.96.2.64)
  Pending Components 1 = compute(100.96.2.65)
  Pending Components 2 = compute(100.96.2.66)
```

3. Optionally, check the details of an individual compute node upgrade job.

```
PCA-ADMIN> getUpgradeJobs requestId=UWS-606c2c4e-c94e-4607-b1df-
f4cbd4b0edff
Data:
  id                                Upgrade Request Id
      Command Name   Result
  --                                ------------------
      ------------   ------
  1737621415558-compute-868540   UWS-606c2c4e-c94e-4607-b1df-
f4cbd4b0edff   compute        None
  1737621353957-compute-861842   UWS-606c2c4e-c94e-4607-b1df-
f4cbd4b0edff   compute        Passed
  1737621345177-compute-860804   UWS-606c2c4e-c94e-4607-b1df-
f4cbd4b0edff   compute        Passed

PCA-ADMIN> getUpgradeJob upgradeJobId=1737621345177-compute-860804
Data:
  Upgrade Request Id = UWS-606c2c4e-c94e-4607-b1df-f4cbd4b0edff
  Composition Id = all_computes
  Name = compute
  Pid = 860804
  Host = pcamn01
  Log File = /nfs/shared_storage/pca_upgrader/log/pca-
upgrader_compute_pcacn001_2025_01_23-08.35.45.log
  Arguments =
 {"component_names":null,"diagnostics":false,"display_task_plan":false,"dry_run_t
c94e-4607-b1df-f4cbd4b0edff","uln":null,"patch":"false"}
  Status = Passed
  Execution Time(sec) = 2976
[...]
```

4. Ensure that each upgrade job passes the initial checks performed before the actual node upgrade starts.

   - If the Upgrader detects that the node is not in an appropriate state for upgrade, take corrective measures and run the upgrade command again.
   - If the Upgrader detects instances that cannot be migrated, shut down these instances and run the upgrade command again.

5. Confirm that the entire upgrade is completed without errors. Use the `getUpgradeStatus` and `getUpgradeJob` commands.

   For more information, see Browse the Job Framework on page 360.

   After successful rediscovery, reprovisioning, and reboot, the compute nodes run the updated version of the operating system and tools, and are fully operational again.

Upgrading the Management Node Cluster

A full management node cluster upgrade includes all the required components on all three Private Cloud Appliance management nodes: the host OS and the services running on the management cluster.

The three-node cluster is upgraded with a single command, as described in this topic. It combines the component upgrade procedures described in the topics that follow:

- Upgrading the Management Node Host Operating System on page 378
- Upgrading Services on the Management Cluster on page 381

Performing a Full Management Node Upgrade

Using the Service Web UI

1. In the navigation menu, go to the Maintenance section and click Upgrade Plan. This provides an overview of current and target component versions.
2. Click Upgrade & Patching to display the Upgrade Jobs page.
3. In the top-right corner of the Upgrade Jobs page, click Create Upgrade or Patch.

   The Create Request window appears. Choose *Upgrade* as the Request Type.
4. Select the appropriate upgrade request type.

   For a full management node upgrade, select Upgrade Rack.
5. Fill out the upgrade request parameters:

   - **Action:** To perform the upgrade or patch operation, select Apply.
   - **Type:** For upgrade, select ISO. The ULN option applies to patching.
   - **Component:** For the management node cluster, select Full MN.
   - **Qualifier:** This parameter is not required when upgrading the full management node cluster.
   - **Verify Only:** Enable this option to run the operation in verification only mode.
   - **Force:** Enable this option to force the operation. Use only when instructed by Oracle.
6. Click Create Request.

   The new upgrade request appears in the Upgrade Jobs table.

   > **Note:**
   >
   > After upgrade, if the upgrade plan specifies that the management nodes must be rebooted for the changes to take effect, a reboot is performed as part of the upgrade process. No administrator action is required.
7. Confirm that the upgrade workflow is completed without errors.

   For more information, see Browse the Job Framework on page 360.

   After successful completion, all management nodes run the updated operating system and tools. The cluster is fully operational and all platform and infrastructure services are at the latest version.

Using the Service CLI

1. Enter the upgrade command.

```
PCA-ADMIN> upgradeRack type=ISO action=APPLY component=FULL_MN
JobId: 08e269eb-64ee-4521-9c57-f3bc3832b86c
Data: Service request has been submitted. Upgrade Request Id =
 UWS-915f4e0b-938b-41b1-b5e3-c624258fd616
```

2. Use the request ID to check the status of this multi-component upgrade process.

```
PCA-ADMIN> getUpgradeStatus requestId=UWS-915f4e0b-938b-41b1-b5e3-
c624258fd616
Data:
  Request id = UWS-915f4e0b-938b-41b1-b5e3-c624258fd616
  Status = Running
  Composition ID = full_mn
  Type = ISO
  Previous Build = 3.0.2-b1300385
  Target Build = 3.0.2-b1301270
  Jobs 1 = 1738138865639-host-4035603
  Jobs 2 = 1738138888311-host-4037912
  Jobs 3 = 1738138951952-host-4048064
  Jobs 4 = 1738139013828-kubernetes-4059059
  Jobs 5 = 1738139074871-platform-4068124
```

```
  Jobs 6 = 1738139135906-oci-4077263
  Completed Components 1 = host(100.96.2.33)
  Completed Components 2 = host(100.96.2.34)
  Completed Components 3 = host(100.96.2.35)
  Completed Components 4 = kubernetes(generic)
  Completed Components 5 = platform(generic)
  Pending Components 1 = ociImages(Oracle-Solaris-11)
  Pending Components 2 = ociImages(Oracle-Linux8-OKE-1.26.15)
  Pending Components 3 = ociImages(Oracle-Linux8-OKE-1.27.12)
  Pending Components 4 = ociImages(Oracle-Linux8-OKE-1.28.8)
  Pending Components 5 = ociImages(Oracle-Linux-8)
  Pending Components 6 = ociImages(Oracle-Linux-9)
  Pending Components 7 = ociImages(Oracle-Linux-7.9)
```

3. Optionally, check the details of an individual upgrade job.

```
PCA-ADMIN> getUpgradeJobs requestId=UWS-915f4e0b-938b-41b1-b5e3-
c624258fd616
Data:
  id                                  Upgrade Request Id
            Command Name   Result
  --                                  ------------------
            ------------   ------
  1738139135906-oci-4077263           UWS-915f4e0b-938b-41b1-b5e3-
c624258fd616   oci            None
  1738139074871-platform-4068124      UWS-915f4e0b-938b-41b1-b5e3-
c624258fd616   platform       Passed
  1738139013828-kubernetes-4059059    UWS-915f4e0b-938b-41b1-b5e3-
c624258fd616   kubernetes     Passed
  1738138951952-host-4048064          UWS-915f4e0b-938b-41b1-b5e3-
c624258fd616   host           Passed
  1738138888311-host-4037912          UWS-915f4e0b-938b-41b1-b5e3-
c624258fd616   host           Passed
  1738138865639-host-4035603          UWS-915f4e0b-938b-41b1-b5e3-
c624258fd616   host           Passed

PCA-ADMIN> getUpgradeJob upgradeJobId=1738138951952-host-4048064
Data:
  Upgrade Request Id = UWS-915f4e0b-938b-41b1-b5e3-c624258fd61
  Composition Id = full_mn
  Name = host
  Pid = 4048064
  Host = pcamn01
  Log File = /nfs/shared_storage/pca_upgrader/log/pca-
upgrader_host_os_pcamn03_2025_01_29-08.22.31.log
  Arguments =
 {"component_names":null,"diagnostics":false,"display_task_plan":false,"dry_run_
b5e3-c624258fd616","uln":null,"patch":"false"}
  Status = Passed
  Execution Time(sec) = 1682
[...]
```

4. Confirm that the upgrade workflow is completed without errors.

   For more information, see Browse the Job Framework on page 360.

   After successful completion, all management nodes run the updated operating system and tools. The cluster is fully operational and all platform and infrastructure services are at the latest version.

Upgrading the Management Node Host Operating System

To upgrade the Oracle Linux host OS of the Private Cloud Appliance management nodes outside of the full rack upgrade workflow, follow the instructions in this topic.

The management nodes operate in a cluster, distributing the load of requests, services, and many different types of operations across the three nodes. While the host OS of a management node is upgraded, the operation of the cluster is affected: the node is temporarily unavailable, the cluster configuration might be changed, core appliance functions (database, platform and infrastructure services) are reduced in availability and performance, and so on. The Upgrader ensures that the management cluster remains operational during an upgrade, using the orchestration logic of its workflow service.

You can either upgrade the host OS of a single management node by providing its host IP address, or all three management nodes in a single workflow.

Upgrading the OS of a Single Management Node

Using the Service Web UI

1. Get the IP address of the management node for which you intend to upgrade the host OS.
2. In the navigation menu, go to the Maintenance section and click Upgrade Plan. This provides an overview of current and target component versions.
3. Click Upgrade & Patching to display the Upgrade Jobs page.
4. In the top-right corner of the Upgrade Jobs page, click Create Upgrade or Patch.

   The Create Request window appears. Choose *Upgrade* as the Request Type.
5. Select the appropriate upgrade request type: Upgrade Rack.
6. Fill out the upgrade request parameters:
   - **Action:** To perform the upgrade or patch operation, select Apply.
   - **Type:** For upgrade, select ISO. The ULN option applies to patching.
   - **Component:** For a single management node, select Host.
   - **Qualifier:** Enter the management node's assigned IP address in the internal administration network. This is an IP address in the internal 100.96.2.0/23 range.
   - **Verify Only:** Enable this option to run the operation in verification only mode.
   - **Force:** Enable this option to force the operation. Use only when instructed by Oracle.
7. Click Create Request.

   The new upgrade request appears in the Upgrade Jobs table.

   > **Note:**
   >
   > After upgrade, if the upgrade plan specifies that the management nodes must be rebooted for the changes to take effect, a reboot is performed as part of the upgrade process. No administrator action is required.
8. Confirm that the upgrade process is completed without errors.

   For more information, see Browse the Job Framework on page 360.

   After successful OS installation, reboot, and cluster reconstruction, the management node returns to normal operation. All three management nodes must run the same versions of host OS, services, and tools to ensure optimal cluster operation.

Using the Service CLI

1. Get the IP address of the management node for which you intend to upgrade the host OS.

```
PCA-ADMIN> list managementNode fields
 hostname,ipAddress,ilomIp,state,firmwareVersion orderby hostname
 ASCENDING
Data:
```

```
  id                                       Hostname   Ip Address
 ILOM Ip Address   State   Firmware Version
  --                                       --------   ----------
 ---------------   -----   ----------------
  22ae47d8-a57a-433c-988d-df62fd3548e1   pcamn01    100.96.2.33
 100.96.0.33       On      5.0.2.23
  042693b6-3ddb-4fb0-914d-f3deea838c8f   pcamn02    100.96.2.34
 100.96.0.34       On      5.0.2.23
  bd2563b8-e310-4fca-ba1b-0e19b8040fc6   pcamn03    100.96.2.35
 100.96.0.35       On      5.0.2.23
```

2. Enter the host OS upgrade command.

   Syntax (entered on a single line):

   ```
   upgradeRack type=ISO action=APPLY component=HOST
    qualifier=<management-node-ip>
   ```

   Example:

   ```
   PCA-ADMIN> upgradeRack type=ISO action=APPLY component=HOST
    qualifier=100.96.2.35
   JobId: f2b6e533-a0a5-4423-9a49-1cb21ed0eb9e
   Data: Service request has been submitted. Upgrade Job
    Id = 1737620045150-host-616754 Upgrade Request Id =
    UWS-9e03c924-628b-4584-a77d-75ae71bf8b8b
   ```

3. Use the job ID to check the status of the upgrade process.

   ```
   PCA-ADMIN> getUpgradeJob upgradeJobId=1737620045150-host-616754
   Data:
     Upgrade Request Id = UWS-9e03c924-628b-4584-a77d-75ae71bf8b8b
     Name = host
     Pid = 616754
     Host = pcamn01
     Log File = /nfs/shared_storage/pca_upgrader/log/pca-
   upgrader_host_os_pcamn03_2025_01_23-08.14.05.log
     Arguments =
    {"component_names":null,"diagnostics":false,"display_task_plan":false,"dry_run_
   a77d-75ae71bf8b8b","uln":null,"patch":"false"}
     Status = Passed
     Execution Time(sec) = 1736
   [...]
   ```

4. Confirm that the upgrade process is completed without errors.

   For more information, see Browse the Job Framework on page 360.

   After successful OS installation, reboot, and cluster reconstruction, the management node returns to normal operation. All three management nodes must run the same versions of host OS, services, and tools to ensure optimal cluster operation.

Upgrading the OS of All Management Nodes

Using the Service Web UI

1. In the navigation menu, go to the Maintenance section and click Upgrade Plan. This provides an overview of current and target component versions.
2. Click Upgrade & Patching to display the Upgrade Jobs page.
3. In the top-right corner of the Upgrade Jobs page, click Create Upgrade or Patch.

   The Create Request window appears. Choose *Upgrade* as the Request Type.
4. Select the appropriate upgrade request type: Upgrade Rack.

5. Fill out the upgrade request parameters:
   - **Action:** To perform the upgrade or patch operation, select Apply.
   - **Type:** For upgrade, select ISO. The ULN option applies to patching.
   - **Component:** For all three management nodes, select All Hosts.
   - **Qualifier:** This parameter is not required when upgrading the OS of all management nodes.
   - **Verify Only:** Enable this option to run the operation in verification only mode.
   - **Force:** Enable this option to force the operation. Use only when instructed by Oracle.
6. Click Create Request.

   The new upgrade request appears in the Upgrade Jobs table.
7. Confirm that the upgrade workflow is completed without errors.

   For more information, see Browse the Job Framework on page 360.

   After successful OS installation and reboot of all three management nodes, the management cluster is fully restored and returns to normal operation.

Using the Service CLI

1. Enter the host upgrade command.

```
PCA-ADMIN> upgradeRack type=ISO action=APPLY component=ALL_HOSTS
JobId: 2792a532-7190-4919-9bc1-8c64f30126d8
Data: Service request has been submitted. Upgrade Request Id = UWS-
a5403cc7-90c8-4224-9e05-06f132561522
```

2. Use the request ID to check the status of this multicomponent upgrade process.

```
PCA-ADMIN> getUpgradeStatus requestId=UWS-
a5403cc7-90c8-4224-9e05-06f132561522
Data:
  Request id = UWS-a5403cc7-90c8-4224-9e05-06f132561522
  Status = Passed
  Composition ID = all_hosts
  Type = ISO
  Previous Build = 3.0.2-b1300385
  Target Build = 3.0.2-b1301270
  Jobs 1 = 1737622450559-host-1066058
  Jobs 2 = 1737622464489-host-1069024
  Jobs 3 = 1737622527066-host-1080160
  Completed Components 1 = host(100.96.2.33)
  Pending Components 2 = host(100.96.2.34)
  Pending Components 3 = host(100.96.2.35)
```

3. Optionally, check the details of an individual management node upgrade job.

```
PCA-ADMIN> getUpgradeJobs requestId=UWS-
a5403cc7-90c8-4224-9e05-06f132561522
Data:
  id                          Upgrade Request Id
      Command Name   Result
  --                          ------------------
      ------------   ------
  1737622527066-host-1080160   UWS-
a5403cc7-90c8-4224-9e05-06f132561522   host           None
  1737622464489-host-1069024   UWS-
a5403cc7-90c8-4224-9e05-06f132561522   host           Passed
  1737622450559-host-1066058   UWS-
a5403cc7-90c8-4224-9e05-06f132561522   host           Passed

PCA-ADMIN> getUpgradeJob upgradeJobId=1737622450559-host-1066058
Data:
```

```
  Upgrade Request Id = UWS-a5403cc7-90c8-4224-9e05-06f132561522
  Composition Id = all_hosts
  Name = host
  Pid = 1066058
  Host = pcamn02
  Log File = /nfs/shared_storage/pca_upgrader/log/pca-
upgrader_host_os_pcamn01_2025_01_23-08.54.24.log
  Arguments =
 {"component_names":null,"diagnostics":false,"display_task_plan":false,"dry_run_
a5403cc7-90c8-4224-9e05-06f132561522","uln":null,"patch":"false"}
  Status = Passed
  Execution Time(sec) = 1750
[...]
```

4. Confirm that the entire upgrade is completed without errors. Use the `getUpgradeStatus` and `getUpgradeJob` commands.

   For more information, see Browse the Job Framework on page 360.

   After successful OS installation and reboot of all three management nodes, the management cluster is fully restored and returns to normal operation.

Upgrading Services on the Management Cluster

The management cluster runs the services that provide the administrative and user-facing functionality of Private Cloud Appliance. Commands are provided to upgrade these outside of the full rack upgrade workflow, in case the circumstances require it.

Upgrading the Host Services

Several critical services of the platform layer – including MySQL, container registry, envoy, etcd and vault – run inside containers on the cluster of management nodes. When the latest controller software introduces a new version of such a service, it appears in the upgrade plan as a *hostservices* component. The new container image is loaded into the management node registries, which the Upgrader uses to orchestrate a rolling upgrade of the service container across the management cluster.

Using the Service Web UI

1. In the navigation menu, go to the Maintenance section and click Upgrade Plan. This provides an overview of current and target component versions.
2. Click Upgrade & Patching to display the Upgrade Jobs page.
3. In the top-right corner of the Upgrade Jobs page, click Create Upgrade or Patch.

   The Create Request window appears. Choose *Upgrade* as the Request Type.
4. Select the appropriate upgrade request type: Upgrade Rack.
5. Fill out the upgrade request parameters:

   - **Action:** To perform the upgrade or patch operation, select Apply.
   - **Type:** For upgrade, select ISO. The ULN option applies to patching.
   - **Component:** For all host services on the management cluster, select Hostservices.
   - **Qualifier:** This parameter is not required when upgrading all management node host services.
   - **Verify Only:** Enable this option to run the operation in verification only mode.
   - **Force:** Enable this option to force the operation. Use only when instructed by Oracle.
6. Click Create Request.

   The new upgrade request appears in the Upgrade Jobs table.
7. Confirm that the upgrade workflow is completed without errors.

   For more information, see Browse the Job Framework on page 360.

Using the Service CLI

1. Enter the upgrade command.

```
PCA-ADMIN> upgradeRack type=ISO action=APPLY component=HOSTSERVICES
JobId: 9bb6d947-e148-411f-903f-87b0130bafab
Data: Service request has been submitted. Upgrade Job Id
 = 1737625625214-hostservices-1654942 Upgrade Request Id =
 UWS-926f8050-a34f-480c-8760-dad689a3df49
```

2. Use the job ID to check the status of the upgrade process.

```
PCA-ADMIN> getUpgradeJob upgradeJobId=1737625625214-
hostservices-1654942
Data:
  Upgrade Request Id = UWS-926f8050-a34f-480c-8760-dad689a3df49
  Name = hostservices
  Pid = 1654942
  Host = pcamn01
  Log File = /nfs/shared_storage/pca_upgrader/log/pca-
upgrader_host_services_upgrade_2025_01_23-09.47.05.log
  Arguments =
 {"component_names":null,"diagnostics":false,"display_task_plan":false,"dry_run_
a34f-480c-8760-dad689a3df49","uln":null,"patch":"false"}
  Status = Passed
  Execution Time(sec) = 104
 [...]
```

3. Confirm that the upgrade process is completed without errors.

   For more information, see Browse the Job Framework on page 360.

Upgrading the Kubernetes Cluster

The Kubernetes container orchestration environment upgrade is decoupled from the host OS. With a single command, all Kubernetes packages, such as kubeadm, kubectl and kubelet, are upgraded on the three management nodes and all the compute nodes. For dependency reasons, Kubernetes must be upgraded after the management node host OS. Note that this upgrade does not include the microservices running within the Kubernetes cluster.

Using the Service Web UI

1. In the navigation menu, go to the Maintenance section and click Upgrade Plan. This provides an overview of current and target component versions.
2. Click Upgrade & Patching to display the Upgrade Jobs page.
3. In the top-right corner of the Upgrade Jobs page, click Create Upgrade or Patch.

   The Create Request window appears. Choose *Upgrade* as the Request Type.
4. Select the appropriate upgrade request type: Upgrade Rack.
5. Fill out the upgrade request parameters:

   - **Action:** To perform the upgrade or patch operation, select Apply.
   - **Type:** For upgrade, select ISO. The ULN option applies to patching.
   - **Component:** For the Kubernetes cluster, select Kubernetes.
   - **Qualifier:** This parameter is not required when upgrading the Kubernetes cluster.
   - **Verify Only:** Enable this option to run the operation in verification only mode.
   - **Force:** Enable this option to force the operation. Use only when instructed by Oracle.
6. Click Create Request.

   The new upgrade request appears in the Upgrade Jobs table.
7. Confirm that the upgrade process is completed without errors.

   For more information, see Browse the Job Framework on page 360.

Using the Service CLI

1. Enter the upgrade command.

```
PCA-ADMIN> upgradeRack type=ISO action=APPLY component=KUBERNETES
JobId: 6b796a4a-89ac-4fcc-a5c7-efb0eb9dceac
Data: Service request has been submitted. Upgrade Job Id =
 1737620352375-kubernetes-660839 Upgrade Request Id = UWS-
dd3cdaf7-5302-4b57-8858-caff924c2dcf
```

2. Use the job ID to check the status of the upgrade process.

```
PCA-ADMIN> getUpgradeJob upgradeJobId=1737620352375-
kubernetes-660839
Data:
  Upgrade Request Id = UWS-dd3cdaf7-5302-4b57-8858-caff924c2dcf
  Name = kubernetes
  Pid = 660839
  Host = pcamn01
  Log File = /nfs/shared_storage/pca_upgrader/log/pca-
upgrader_kubernetes_cluster_2025_01_23-08.19.12.log
  Arguments =
 {"component_names":null,"diagnostics":false,"display_task_plan":false,"dry_run_
dd3cdaf7-5302-4b57-8858-caff924c2dcf","uln":null,"patch":"false"}
  Status = Passed
  Execution Time(sec) = 249
[...]
```

3. Confirm that the upgrade process is completed without errors.

   For more information, see Browse the Job Framework on page 360.

Upgrading the Microservices

The microservices upgrade covers both the internal services of the platform layer, and the administrative and user-level services exposed through the infrastructure services layer. The containerized microservices have their own separate upgrade mechanism. A service is upgraded if a new Helm deployment chart and container image are found in the ISO content. When a new deployment chart is detected during the upgrade process, the pods running the services are restarted with the new container image.

> **Note:**
>
> In specific circumstances it is possible to upgrade certain platform services individually, by adding an optional JSON string to the command. This option should not be used unless Oracle provides explicit instructions to do so.

Using the Service Web UI

1. In the navigation menu, go to the Maintenance section and click Upgrade Plan. This provides an overview of current and target component versions.
2. Click Upgrade & Patching to display the Upgrade Jobs page.
3. In the top-right corner of the Upgrade Jobs page, click Create Upgrade or Patch.

   The Create Request window appears. Choose *Upgrade* as the Request Type.
4. Select the appropriate upgrade request type: Upgrade Rack.
5. Fill out the upgrade request parameters:

   - **Action:** To perform the upgrade or patch operation, select Apply.
   - **Type:** For upgrade, select ISO. The ULN option applies to patching.
   - **Component:** For the platform and infrastructure microservices, select Platform.
   - **Qualifier:** This parameter is not required when upgrading the microservices.
   - **Verify Only:** Enable this option to run the operation in verification only mode.

- **Force:** Enable this option to force the operation. Use only when instructed by Oracle.

6. Click Create Request.

   The new upgrade request appears in the Upgrade Jobs table.
7. Confirm that the upgrade process is completed without errors.

   For more information, see Browse the Job Framework on page 360.

Using the Service CLI

1. Enter the upgrade command.

```
PCA-ADMIN> upgradeRack type=ISO action=APPLY component=PLATFORM
JobId: 30116885-57e3-4737-b6a2-9f429f084fb5
Data: Service request has been submitted. Upgrade Job
 Id = 1737620370636-platform-666460 Upgrade Request Id =
 UWS-26312768-855d-4f31-90b9-cbb0559d1d7f
```

2. Use the job ID to check the status of the upgrade process.

```
PCA-ADMIN> getUpgradeJob upgradeJobId=1737620370636-platform-666460
Data:
  Upgrade Request Id = UWS-26312768-855d-4f31-90b9-cbb0559d1d7f
  Name = platform
  Pid = 666460
  Host = pcamn01
  Log File = /nfs/shared_storage/pca_upgrader/log/pca-
upgrader_platform_services_2025_01_23-08.19.30.log
  Arguments =
 {"component_names":null,"diagnostics":false,"display_task_plan":false,"dry_run_
cbb0559d1d7f","uln":null,"patch":"false"}
  Status = Passed
  Execution Time(sec) = 536
[...]
```

3. Confirm that the upgrade process is completed without errors.

   For more information, see Browse the Job Framework on page 360.

Upgrading Oracle Cloud Infrastructure Images

When new OCI Images become available and supported for Private Cloud Appliance, you can make them available for use in all existing tenancies with a single upgrade command.

The images are stored in the `/nfs/shared_storage/oci_compute_images` directory on the ZFS Storage Appliance. If you perform a full rack or management cluster upgrade, the new images are automatically added to your environment, in which case this procedure can be skipped. The image versions are tracked through the upgrade plan. Review the upgrade plan to verify if the images need to be upgraded.

An upgrade adds new Oracle Cloud Infrastructure Images to your environment, but it never removes any existing images. If you no longer need an image, you have the option to delete it using the `deletePlatformImage` command.

Adding New Images

Using the Service Web UI

1. In the navigation menu, go to the Maintenance section and click Upgrade Plan. This provides an overview of current and target component versions.
2. Click Upgrade & Patching to display the Upgrade Jobs page.
3. In the top-right corner of the Upgrade Jobs page, click Create Upgrade or Patch.

   The Create Request window appears. Choose *Upgrade* as the Request Type.
4. Select the appropriate patch request type: Upgrade Rack.

5. Fill out the upgrade request parameters:
   - **Action:** To perform the upgrade or patch operation, select Apply.
   - **Type:** For upgrade, select ISO. The ULN option applies to patching.
   - **Component:** For Oracle Cloud Infrastructure images, select OCI Images.
   - **Qualifier:** This parameter is not required when adding Oracle Cloud Infrastructure images.
   - **Verify Only:** Enable this option to run the operation in verification only mode.
   - **Force:** Enable this option to force the operation. Use only when instructed by Oracle.
6. Click Create Request.

   The new upgrade request appears in the Upgrade Jobs table.
7. Confirm that the upgrade process is completed without errors.

   For more information, see Browse the Job Framework on page 360.

Using the Service CLI

1. Enter the upgrade command.

```
PCA-ADMIN> upgradeRack type=ISO action=APPLY component=OCIIMAGES
JobId: c21f2253-8ab9-4491-ab01-5d9702f96bc5
Data: Service request has been submitted. Upgrade Job Id =
 1737620389632-oci-668679 Upgrade Request Id = UWS-a2e17445-
eb02-466f-9d16-6b644b89866e
```

2. Use the job ID to check the status of the upgrade process.

```
PCA-ADMIN> getUpgradeJob upgradeJobId=1737620389632-oci-668679
Data:
  Upgrade Request Id = UWS-a2e17445-eb02-466f-9d16-6b644b89866e
  Name = oci
  Pid = 668679
  Host = pcamn01
  Log File = /nfs/shared_storage/pca_upgrader/log/pca-
upgrader_oci_instance_images_2025_01_23-08.19.49.log
  Arguments =
 {"component_names":null,"diagnostics":false,"display_task_plan":false,"dry_run_t
a2e17445-eb02-466f-9d16-6b644b89866e","uln":null,"patch":"false"}
  Status = Passed
  Execution Time(sec) = 1083
[...]
```

3. Confirm that the upgrade process is completed without errors.

   For more information, see Browse the Job Framework on page 360.

Resolving Security Vulnerabilities in OKE Clusters

At the end of the Oracle Cloud Infrastructure images upgrade or patch process, the Upgrader launches a background job to resolve any known CVEs that might affect existing clusters deployed through the Kubernetes Engine (OKE). When the new images have been imported, an OKE Service tool ensures that the running control plane nodes receive the latest available CVE fixes delivered with the new images.

The CVE fixes are applied in a fully automated way, but the process could be derailed by timing issues in the appliance upgrade or patching workflow. Thus, it is important for an appliance administrator to monitor the OKE background job and verify that CVE fixes have been applied successfully to all existing OKE clusters. Note that an error in OKE cluster patching will NOT cause the appliance upgrade or patching process to fail.

The status of the Oracle Cloud Infrastructure images upgrade or patch process indicates that the OKE Service tool has been run. It also provides the OCIDs of the OKE clusters found, and any work requests for cluster patching operations that should be tracked.

```
getUpgradeJob upgradeJobId=1724442488245-oci-35655
Data:
```

```
  Log File = /nfs/shared_storage/pca_upgrader/log/pca-
upgrader_oci_instance_images_<date>-<time>.log
[...]
  Tasks 12 - Message = OKE Clusters CVE Patching initiated:

 {"ocid1.cluster.<AK01234567>.<mypca>.63f7764a345d4d74a9abd5267ad55a28p6ixuw4ejzr73yugyr
 "No operations performed",

 "ocid1.cluster.<AK01234567>.<mypca>.ac198ab8583848e8947501f7061bde16mx17lm2u2rugld6u3uj
 "ocid1.workrequest.<AK01234567>.<mypca>.oke-
g8l7kh306zlt59zb9vc4yvo532b1j4jwtffnmel83v1qif0q93lum7er"}
```

In the example, two active OKE clusters are found. The first cluster is using the latest image and does not need to be patched. The second cluster is out of date and needs to be updated with the latest available image. Use the work request to track the cluster update status from the OCI CLI, using the command: `oci ce work-request get --work-request-id` ***<workrequest_OCID>***.

```
# oci ce work-request get --work-request-id
 ocid1.workrequest.<AK01234567>.<mypca>.oke-
g8l7kh306zlt59zb9vc4yvo532b1j4jwtffnmel83v1qif0q93lum7er
{
  "data": {
    "compartment-id":
 "ocid1.compartment.<AK01234567>.<mypca>.ezbf00rrfc0qnoi8rofk3yzcbq0yeg9ly0gzf6caebv3ugc
    "id": "ocid1.workrequest.<AK01234567>.<mypca>.oke-
g8l7kh306zlt59zb9vc4yvo532b1j4jwtffnmel83v1qif0q93lum7er",
    "operation-type": "UNKNOWN_ENUM_VALUE",
    "resources": [
      {
        "action-type": "UPDATED",
        "entity-type": "CLUSTER",
        "entity-uri": null,
        "identifier":
 "ocid1.cluster.<AK01234567>.<mypca>.ac198ab8583848e8947501f7061bde16mx17lm2u2rugld6u3uj
      }
    ],
    "status": "SUCCEEDED",
    "time-accepted": "2024-09-03T11:18:29.750438+00:00",
    "time-finished": "2024-09-03T11:36:19.313926+00:00",
    "time-started": "2024-09-03T11:18:36.451513+00:00"
  },
  "etag": "00fa0a51-a9dd-5455-f390-429a20817d6d"
}
```

If errors have occurred, and certain clusters were not updated based on the latest available image, first ensure that the cluster is in a good working state, then run the following command from one of the management nodes:

```
# kubectl exec -it -n oke <oke_pod_name> -c oke -- pca-oke-cluster-tool --
action patch-cluster-cve
```

Upgrading Firmware

Firmware may be upgraded whenever new versions are made available for Private Cloud Appliance components.

Firmware upgrades can be applied in no particular order and independently of other components. Firmware is included in the ISO content for all component ILOMs, for the ZFS Storage Appliance, and for the switches. Select the instructions for the component type you want to upgrade.

Use the upgrade plan to check the progress of the appliance upgrade, and determine which firmware upgrades still need to be performed.

Upgrading ILOMs

To upgrade one or more ILOMs outside of the Private Cloud Appliance full rack upgrade workflow, follow these instructions.

ILOM upgrades can be applied to management nodes and compute nodes; the firmware packages might be different per component type. You can either upgrade a single ILOM by providing its IP address, or the ILOMs of all management and compute nodes installed in the appliance.

If new firmware is available for the internal drives and network interface cards installed in the management and compute nodes, the updates are applied as part as the ILOM upgrade process.

> **Caution:**
>
> You must NOT upgrade the ILOM of the management node that holds the management virtual IP address, and thus the primary role in the cluster. To determine which management node has the primary role in the cluster, and make another node the primary, use the following Service CLI commands:

```
PCA-ADMIN> getPrimaryMgmtNode
Data:
  data = pcamn01

PCA-ADMIN> updatePrimaryNode node=pcamn02
Data:
  message = Successfully issued update primary
 node command

PCA-ADMIN> getPrimaryMgmtNode
Data:
  data = pcamn02
```

Obtaining an ILOM IP Address

Using the Service Web UI

1. In the navigation menu, click Rack Units.
2. Click the name of the component that requires new ILOM firmware.
3. Select the Rack Unit Information tab.
4. Record the IP Address listed under ILOM IPs.

Using the Service CLI

- To list the ILOM IP addresses of all management nodes or compute nodes, use these commands:

```
PCA-ADMIN> getCNIloms
Data:
  data 1 = 100.96.0.66
  data 2 = 100.96.0.64
  data 3 = 100.96.0.65
[...]

PCA-ADMIN> getMNIloms
Data:
  data 1 = 100.96.0.33
  data 2 = 100.96.0.34
  data 3 = 100.96.0.35
```

- To obtain the ILOM IP address that corresponds with a specific node host name, use the following command:

```
PCA-ADMIN> getServerILOMIP hostName=<node_name>
```

For example:

```
PCA-ADMIN> getServerILOMIP hostName=pcacn002
Data:
  data = 100.96.0.65

PCA-ADMIN> getServerILOMIP hostName=pcamn03
Data:
  data = 100.96.0.35
```

- To obtain the ID, IP addresses, and other key information about compute nodes or management nodes, use the following commands:

```
PCA-ADMIN> list managementNode fields
 hostname,ipAddress,ilomIp,state,firmwareVersion orderby hostname
 ASCENDING
Data:
  id                                     Hostname   Ip Address
 ILOM Ip Address   State   Firmware Version
  --                                     --------   ----------
 ---------------   -----   ----------------
  22ae47d8-a57a-433c-988d-df62fd3548e1   pcamn01    100.96.2.33
 100.96.0.33       On      5.1.4.25
  042693b6-3ddb-4fb0-914d-f3deea838c8f   pcamn02    100.96.2.34
 100.96.0.34       On      5.1.4.25
  bd2563b8-e310-4fca-ba1b-0e19b8040fc6   pcamn03    100.96.2.35
 100.96.0.35       On      5.1.4.25

PCA-ADMIN> list computeNode fields
 hostname,ipAddress,ilomIp,state,firmwareVersion,provisioningLocked,maintenanceL
 orderby hostname ASCENDING
Data:
  id                                     Hostname   Ip Address
 ILOM Ip Address   State   Firmware Version           Provisioning
 Locked   Maintenance Locked
  --                                     --------   ----------
    ---------------   -----   ----------------
 -------------------   ------------------
  cf488903-fef8-4a51-8a41-c6990e4755c5   pcacn001   100.96.2.64
 100.96.0.64       On      PCA Hypervisor:3.0.2-859   false
        false
  42a7594d-1173-4dbd-4755-07810cc2d527   pcacn002   100.96.2.65
 100.96.0.65       On      PCA Hypervisor:3.0.2-859   false
        false
  bc0f37d5-ba77-423e-bc11-017704b47e59   pcacn003   100.96.2.66
 100.96.0.66       On      PCA Hypervisor:3.0.2-859   false
        false
  2e5ac527-01f5-4230-ae41-0522fcb57c9a   pcacn004   100.96.2.67
 100.96.0.67       On      PCA Hypervisor:3.0.2-859   false
        false
  5a6b61cf-7e99-4df2-87e4-b37c5fb0bfb8   pcacn005   100.96.2.68
 100.96.0.68       On      PCA Hypervisor:3.0.2-859   false
        false
  885f2aa4-f017-41e8-b2bc-e588cc0c6162   pcacn006   100.96.2.69
 100.96.0.69       On      PCA Hypervisor:3.0.2-859   false
        false
```

Upgrading a Single ILOM

Using the Service Web UI

1. Get the IP address of the component ILOM you intend to upgrade.

2. In the navigation menu, go to the Maintenance section and click Upgrade Plan. This provides an overview of current and target component versions.
3. Click Upgrade & Patching to display the Upgrade Jobs page.
4. In the top-right corner of the Upgrade Jobs page, click Create Upgrade or Patch.

   The Create Request window appears. Choose *Upgrade* as the Request Type.
5. Select the appropriate upgrade request type: Upgrade Rack.
6. Fill out the upgrade request parameters:
   - **Action:** To perform the upgrade or patch operation, select Apply.
   - **Type:** For upgrade, select ISO. The ULN option applies to patching.
   - **Component:** For a single ILOM, select ILOM.
   - **Qualifier:** Enter the component's assigned IP address in the ILOM network. This is an IP address in the internal 100.96.0.0/23 range.
   - **Verify Only:** Enable this option to run the operation in verification only mode.
   - **Force:** Enable this option to force the operation. Use only when instructed by Oracle.
7. Click Create Request.

   The new upgrade request appears in the Upgrade Jobs table.
8. Confirm that the upgrade workflow is completed without errors.

   For more information, see Browse the Job Framework on page 360.

Using the Service CLI

1. Enter the ILOM upgrade command.

   Syntax (entered on a single line):

   ```
   upgradeRack type=ISO action=APPLY component=ILOM
    qualifier=<component-ip>
   ```

   Example:

   ```
   PCA-ADMIN> upgradeRack type=ISO action=APPLY component=ILOM
    qualifier=100.96.0.34
   JobId: caf53268-bfb5-4e63-b96d-5446728a1617
   Data: Service request has been submitted. Upgrade Job Id
    = 1737620133356-ilom-632638 Upgrade Request Id = UWS-
   ffc59a65-04d1-49a3-b194-4055a34fd30f
   ```
2. Use the job ID to check the status of the upgrade process.

   ```
   PCA-ADMIN> getUpgradeJob upgradeJobId=1737620133356-ilom-632638
   Data:
     Upgrade Request Id = UWS-ffc59a65-04d1-49a3-b194-4055a34fd30f
     Name = ilom
     Pid = 632638
     Host = pcamn01
     Log File = /nfs/shared_storage/pca_upgrader/log/pca-
   upgrader_ilom_firmware_ilom-pcamn02_2025_01_23-08.15.33.log
     Arguments =
    {"component_names":null,"diagnostics":false,"display_task_plan":false,"dry_run_
   ffc59a65-04d1-49a3-b194-4055a34fd30f","uln":null,"patch":"false"}
     Status = Passed
     Execution Time(sec) = 627
   [...]
   ```
3. Confirm that the upgrade process is completed without errors.

   For more information, see Browse the Job Framework on page 360.

Upgrading All ILOMs

Using the Service Web UI

1. In the navigation menu, go to the Maintenance section and click Upgrade Plan. This provides an overview of current and target component versions.
2. Click Upgrade & Patching to display the Upgrade Jobs page.
3. In the top-right corner of the Upgrade Jobs page, click Create Upgrade or Patch.

   The Create Request window appears. Choose *Upgrade* as the Request Type.
4. Select the appropriate upgrade request type: Upgrade Rack.
5. Fill out the upgrade request parameters:

   - **Action:** To perform the upgrade or patch operation, select Apply.
   - **Type:** For upgrade, select ISO. The ULN option applies to patching.
   - **Component:** For all ILOM in the system, select All ILOMs.
   - **Qualifier:** This parameter is not required when upgrading all ILOMs.
   - **Verify Only:** Enable this option to run the operation in verification only mode.
   - **Force:** Enable this option to force the operation. Use only when instructed by Oracle.
6. Click Create Request.

   The new upgrade request appears in the Upgrade Jobs table.
7. Confirm that the upgrade workflow is completed without errors.

   For more information, see Browse the Job Framework on page 360.

Using the Service CLI

1. Enter the ILOM upgrade command.

```
PCA-ADMIN> upgradeRack type=ISO action=APPLY component=ALL_ILOMS
JobId: 948c4347-42d4-4d30-9f7e-bda0811786cf
Data: Service request has been submitted. Upgrade Request Id =
 UWS-3c4705f9-0018-4de8-832a-a48988f87ed4
```

2. Use the request ID to check the status of this multicomponent upgrade process.

```
PCA-ADMIN> getUpgradeStatus requestId=UWS-3c4705f9-0018-4de8-832a-
a48988f87ed4
Data:
  Request id = UWS-3c4705f9-0018-4de8-832a-a48988f87ed4
  Status = Passed
  Composition ID = all_iloms
  Type = ISO
  Previous Build = 3.0.2-b1300385
  Target Build = 3.0.2-b1301270
  Jobs 1 = 1737622769421-ilom-1113777
  Jobs 2 = 1737622777728-ilom-1115513
  Jobs 3 = 1737622846917-ilom-1136625
  Jobs 4 = 1737622914588-ilom-1152088
  Jobs 5 = 1737622981976-ilom-1165085
  Jobs 6 = 1737623062171-ilom-1180415
  Completed Components 1 = ilom(100.96.0.33)
  Completed Components 2 = ilom(100.96.0.34)
  Completed Components 3 = ilom(100.96.0.35)
  Completed Components 4 = ilom(100.96.0.64)
  Pending Components 1 = ilom(100.96.0.65)
  Pending Components 2 = ilom(100.96.0.66)
```

3. Optionally, check the details of an individual ILOM upgrade job.

```
PCA-ADMIN> getUpgradeJobs requestId=UWS-3c4705f9-0018-4de8-832a-
a48988f87ed4
```

```
Data:
  id                            Upgrade Request Id
      Command Name   Result
  --                            ------------------
      ------------   ------
  1737623062171-ilom-1180415   UWS-3c4705f9-0018-4de8-832a-
a48988f87ed4   ilom           None
  1737622981976-ilom-1165085   UWS-3c4705f9-0018-4de8-832a-
a48988f87ed4   ilom           None
  1737622914588-ilom-1152088   UWS-3c4705f9-0018-4de8-832a-
a48988f87ed4   ilom           Passed
  1737622846917-ilom-1136625   UWS-3c4705f9-0018-4de8-832a-
a48988f87ed4   ilom           Passed
  1737622777728-ilom-1115513   UWS-3c4705f9-0018-4de8-832a-
a48988f87ed4   ilom           Passed
  1737622769421-ilom-1113777   UWS-3c4705f9-0018-4de8-832a-
a48988f87ed4   ilom           Passed

PCA-ADMIN> getUpgradeJob upgradeJobId=1737622846917-ilom-1136625
Data:
  Upgrade Request Id = UWS-3c4705f9-0018-4de8-832a-a48988f87ed4
  Composition Id = all_iloms
  Name = ilom
  Pid = 1136625
  Host = pcamn01
  Log File = /nfs/shared_storage/pca_upgrader/log/pca-
upgrader_ilom_firmware_ilom-pcamn03_2025_01_23-09.00.46.log
  Arguments =
 {"component_names":null,"diagnostics":false,"display_task_plan":false,"dry_run_
a48988f87ed4","uln":null,"patch":"false"}
  Status = Passed
  Execution Time(sec) = 767
[...]
```

4. Confirm that the entire upgrade is completed without errors. Use the `getUpgradeStatus` and `getUpgradeJob` commands.

   For more information, see Browse the Job Framework on page 360.

Rebooting Nodes After ILOM Upgrade

At the end of the upgrade, the ILOM itself is rebooted automatically. However, the server component also needs to be rebooted for all changes to take effect. Wait 5 minutes to allow the ILOM workflow to complete first.

For minimum operational impact, schedule the compute node and management node reboot operations after all ILOMs have been upgraded.

Take into account that rebooting the compute nodes requires migrating the compute instances. For more information, see Performing Administrative Operations on Compute Nodes on page 156.

To reboot a compute node, get its ID and run the `reset` command.

```
PCA-ADMIN> list ComputeNode
PCA-ADMIN> reset id=<compute-node-id>
```

A management node reboot is a delicate and lengthy process that involves several cluster operations. Ensure that operations on the entire management cluster are fully restored before rebooting another node.

To reboot a management node, use the `restartServer` command with the node host name.

> **Caution:**
>
> Always verify the cluster state before rebooting a management node. Consult the Oracle Private Cloud Appliance Release Notes for more information:

> see known issue “Rebooting a Management Node while the Cluster State is Unhealthy Causes Platform Integrity Issues”.

```
Syntax: restartServer serverName=<pcamn01|pcamn02|pcamn03>

Example:
PCA-ADMIN> restartServer serverName=pcamn03
Data:
  status = true
  message = Successfully added pcamn03 to the queue to restart
```

Upgrading the ZFS Storage Appliance

To upgrade the ZFS Storage Appliance outside of the Private Cloud Appliance full rack upgrade workflow, follow these instructions.

To upgrade the operating software of the system's ZFS Storage Appliance, you enter the upgrade command with no additional parameters. The IP addresses of the storage controllers and the location of the firmware package are known, and a single upgrade command initiates a rolling upgrade of both controllers. If a new ILOM firmware version is included for the two controllers, it will be installed as part of this upgrade process.

> **Caution:**
>
> Ensure that no users are logged in to the ZFS Storage Appliance or the storage controller ILOMs during the upgrade process.
>
> Do not make storage configuration changes while an upgrade is in progress. While controllers are running different software versions, configuration changes made to one controller are not propagated to its peer controller.
>
> During firmware upgrade the storage controllers are placed in active/passive mode. They automatically return to active/active after the upgrade is completed.

Before You Begin

Before you initiate a ZFS Storage Appliance upgrade, you must disable the node state service to prevent errors in node states after the upgrade.

1. From a management node, set the provisioning lock by issuing this command:

   ```
   pca-admin locks set system provisioning
   ```

2. Perform the ZFS Storage Appliance upgrade using either the Service Web UI or the Service CLI procedure below.
3. Release the provisioning lock.

   ```
   pca-admin locks unset system provisioning
   ```

4. Confirm the lock state.

   ```
   pca-admin locks show system
   ```

Upgrading the Operating Software

Using the Service Web UI

1. In the navigation menu, go to the Maintenance section and click Upgrade Plan. This provides an overview of current and target component versions.
2. Click Upgrade & Patching to display the Upgrade Jobs page.
3. In the top-right corner of the Upgrade Jobs page, click Create Upgrade or Patch.

   The Create Request window appears. Choose *Upgrade* as the Request Type.

4. Select the appropriate upgrade request type: Upgrade Rack.
5. Fill out the upgrade request parameters:
   - **Action:** To perform the upgrade or patch operation, select Apply.
   - **Type:** For upgrade, select ISO. The ULN option applies to patching.
   - **Component:** For the ZFS Storage Appliance operating software, select ZFSSA.
   - **Qualifier:** This parameter is not required when upgrading the ZFS Storage Appliance.
   - **Verify Only:** Enable this option to run the operation in verification only mode.
   - **Force:** Enable this option to force the operation. Use only when instructed by Oracle.
6. Click Create Request.

   The new upgrade request appears in the Upgrade Jobs table.
7. Confirm that the upgrade process is completed without errors.

   For more information, see Browse the Job Framework on page 360.

Using the Service CLI

1. Enter the upgrade command.

```
PCA-ADMIN> upgradeRack type=ISO action=APPLY component=ZFSSA
JobId: 6e6afce0-31bf-47fa-9d82-04e0634cc22b
Data: Service request has been submitted. Upgrade Job
 Id = 1737620335404-zfssa-658852 Upgrade Request Id =
 UWS-19412b1f-002d-44cd-bc7c-9c8b5de14e2d
```

2. Use the job ID to check the status of the upgrade process.

```
PCA-ADMIN> getUpgradeJob upgradeJobId=1737620335404-zfssa-658852
Data:
  Upgrade Request Id = UWS-19412b1f-002d-44cd-bc7c-9c8b5de14e2d
  Name = zfssa
  Pid = 658852
  Host = pcamn01
  Log File = /nfs/shared_storage/pca_upgrader/log/pca-
upgrader_zfssa_ak_2025_01_23-08.18.55.log
  Arguments =
 {"component_names":null,"diagnostics":false,"display_task_plan":false,"dry_run_
bc7c-9c8b5de14e2d","uln":null,"patch":"false"}
  Status = Passed
  Execution Time(sec) = 739
[...]
```

3. Confirm that the upgrade process is completed without errors.

   For more information, see Browse the Job Framework on page 360.

Upgrading Private Cloud Appliance Switches

To upgrade the Cisco Nexus switches outside of the Private Cloud Appliance full rack upgrade workflow, follow these instructions.

The appliance rack contains three categories of Cisco Nexus switches: a management switch, two leaf switches, and two spine switches. They all run the same Cisco NX-OS network operating software. **You must perform the upgrades in this order: leaf switches first, then spine switches, and finally the management switch**.

You can either upgrade a single switch category, or all switches installed in the appliance. Only one command per switch category is required. The leaf switches and the spine switches are upgraded in pairs.

Some versions of the network operating software consist of two files: a binary file and an additional EPLD (electronic programmable logic device) image. Both are automatically retrieved from their designated location during the upgrade process, and applied in the correct order.

> **Caution:**
>
> Verify that all required switch ports are operational. If ports are down, an outage will occur during switch upgrade/patching. Consult the Oracle Private Cloud Appliance Release Notes for more information: see known issue "Switch Upgrade or Patch Procedure Not Blocked When Ports Are Down ".

Upgrading a Single Switch Type

Using the Service Web UI

1. In the navigation menu, go to the Maintenance section and click Upgrade Plan. This provides an overview of current and target component versions.
2. Click Upgrade & Patching to display the Upgrade Jobs page.
3. In the top-right corner of the Upgrade Jobs page, click Create Upgrade or Patch.

   The Create Request window appears. Choose *Upgrade* as the Request Type.
4. Select the appropriate upgrade request type: Upgrade Rack.
5. Fill out the upgrade request parameters:
   - **Action:** To perform the upgrade or patch operation, select Apply.
   - **Type:** For upgrade, select ISO. The ULN option applies to patching.
   - **Component:** For a single switch type, select Switch.
   - **Qualifier:** Enter the switch type you intend to upgrade: `leaf` - `spine` - `mgmt`. The upgrade order is as follows: leaf switches first, then spine switches, and finally the management switch.
   - **Verify Only:** Enable this option to run the operation in verification only mode.
   - **Force:** Enable this option to force the operation. Use only when instructed by Oracle.
6. Click Create Request.

   The new upgrade request appears in the Upgrade Jobs table.
7. Confirm that the upgrade process is completed without errors.

   For more information, see Browse the Job Framework on page 360.

Using the Service CLI

1. Enter the switch upgrade command.

   Syntax (entered on a single line):

   ```
   upgradeRack type=ISO action=APPLY
   component=SWITCH qualifier=[leaf | spine | mgmt]
   ```

   Example:

   ```
   PCA-ADMIN> upgradeRack type=ISO action=APPLY component=SWITCH
    qualifier=leaf
   JobId: 8855dcd2-5f24-41e9-8ad6-3c12ddec1458
   Data: Service request has been submitted. Upgrade Job
    Id = 1737620242708-cisco-646097 Upgrade Request Id =
    UWS-7cf998f6-7077-4137-a325-02dd990ea64c
   ```
2. Use the job ID to check the status of the upgrade process.

   ```
   PCA-ADMIN> getUpgradeJob upgradeJobId=1737620256788-cisco-646097
   Data:
     Upgrade Request Id = UWS-7cf998f6-7077-4137-a325-02dd990ea64c
     Name = cisco
     Pid = 646097
     Host = pcamn01
   ```

```
  Log File = /nfs/shared_storage/pca_upgrader/log/pca-
upgrader_cisco_firmware_leaf_2025_01_23-08.17.36.log
  Arguments =
 {"component_names":null,"diagnostics":false,"display_task_plan":false,"dry_run_
a325-02dd990ea64c","uln":null,"patch":"false"}
  Status = Passed
  Execution Time(sec) = 624
[...]
```

3. Confirm that the upgrade process is completed without errors.

   For more information, see Browse the Job Framework on page 360.

Upgrading All Switches

Using the Service Web UI

1. In the navigation menu, go to the Maintenance section and click Upgrade Plan. This provides an overview of current and target component versions.
2. Click Upgrade & Patching to display the Upgrade Jobs page.
3. In the top-right corner of the Upgrade Jobs page, click Create Upgrade or Patch.

   The Create Request window appears. Choose *Upgrade* as the Request Type.
4. Select the appropriate upgrade request type: Upgrade Rack.
5. Fill out the upgrade request parameters:

   - **Action:** To perform the upgrade or patch operation, select Apply.
   - **Type:** For upgrade, select ISO. The ULN option applies to patching.
   - **Component:** For all the switches installed in the system, select All Switches.
   - **Qualifier:** This parameter is not required when upgrading all switches.
   - **Verify Only:** Enable this option to run the operation in verification only mode.
   - **Force:** Enable this option to force the operation. Use only when instructed by Oracle.
6. Click Create Request.

   The new upgrade request appears in the Upgrade Jobs table.
7. Confirm that the upgrade workflow is completed without errors.

   For more information, see Browse the Job Framework on page 360.

Using the Service CLI

1. Enter the switch upgrade command.

```
PCA-ADMIN> upgradeRack type=ISO action=APPLY component=ALL_SWITCHES
JobId: 7c39653d-b490-4c7f-8a05-d7f137ed168e
Data: Service request has been submitted. Upgrade Request Id = UWS-
ba897ac1-934d-476f-b3b5-be713cfbcdab
```

2. Use the request ID to check the status of this multi-component upgrade process.

```
PCA-ADMIN> getUpgradeStatus requestId=UWS-ba897ac1-934d-476f-b3b5-
be713cfbcdab
Data:
  Request id = UWS-ba897ac1-934d-476f-b3b5-be713cfbcdab
  Status = Running
  Composition ID = all_switches
  Type = ISO
  Previous Build = 3.0.2-b1300385
  Target Build = 3.0.2-b1301270
  Jobs 1 = 1737622021772-cisco-979985
  Jobs 2 = 1737622029474-cisco-981089
  Jobs 3 = 1737622098759-cisco-989501
  Completed Components 1 = switch(leaf)
```

```
  Completed Components 2 = switch(spine)
  Pending Components 1 = switch(mgmt)
```

3. Optionally, check the details of an individual switch upgrade job.

```
PCA-ADMIN> getUpgradeJobs requestId=UWS-ba897ac1-934d-476f-b3b5-
be713cfbcdab
Data:
  id                            Upgrade Request Id
      Command Name   Result
  --                            ------------------
      ------------   ------
  1737622098759-cisco-989501   UWS-ba897ac1-934d-476f-b3b5-
be713cfbcdab   cisco          None
  1737622029474-cisco-981089   UWS-ba897ac1-934d-476f-b3b5-
be713cfbcdab   cisco          Passed
  1737622021772-cisco-979985   UWS-ba897ac1-934d-476f-b3b5-
be713cfbcdab   cisco          Passed

PCA-ADMIN> getUpgradeJob upgradeJobId=1737622021772-cisco-979985
Data:
  Upgrade Request Id = UWS-ba897ac1-934d-476f-b3b5-be713cfbcdab
  Composition Id = all_switches
  Name = cisco
  Pid = 979985
  Host = pcamn01
  Log File = /nfs/shared_storage/pca_upgrader/log/pca-
upgrader_cisco_firmware_leaf_2025_01_23-08.47.01.log
  Arguments =
 {"component_names":null,"diagnostics":false,"display_task_plan":false,"dry_run_
ba897ac1-934d-476f-b3b5-be713cfbcdab","uln":null,"patch":"false"}
  Status = Passed
  Execution Time(sec) = 2209
[...]
```

4. Confirm that the entire upgrade is completed without errors. Use the `getUpgradeStatus` and `getUpgradeJob` commands.

   For more information, see Browse the Job Framework on page 360.

Accessing Private Cloud Appliance User Interfaces

Use the Private Cloud Appliance Compute Web UI, OCI CLI, and API to create and manage resources on the appliance.

See the following topics:

- Signing in to the Private Cloud Appliance Compute Web UI on page 396
- Installing and Using the OCI CLI on page 397
- Using the API to Manage Private Cloud Appliance Resources on page 413

> **Note:**
>
> To access the Service Enclave, see Private Cloud Appliance Infrastructure Administration on page 145.

Signing in to the Private Cloud Appliance Compute Web UI

Use the Private Cloud Appliance Compute Web UI to create and manage compute, storage and other resources on a Private Cloud Appliance.

The Private Cloud Appliance Compute Web UI is an intuitive, graphical interface to create and manage resources on the appliance.

Prerequisites

You need to know the following Private Cloud Appliance information:

- System name
- Domain name
- Tenancy name
- Your Private Cloud Appliance username and password.

1. From a browser, enter the URL of the Private Cloud Appliance.

   Syntax: `https://console.<pca_name>.<domain_name>`

   Example: `https://console.pcasys1.example.com`

   where:

   - `pcasys1` is the Private Cloud Appliance name.
   - `example.com` is the domain name.
2. Enter your tenancy name and select **Continue**.

   The Sign In page is displayed.
3. Enter your username and password, and then select **Sign In**.

   The dashboard is displayed and can be used to reach common resources.
4. Use the navigation menu ( ) to access the full list of manageable resources.

Installing and Using the OCI CLI

Learn how to install and use the OCI CLI to manage resources on Private Cloud Appliance.

The OCI CLI is the command line interface to the Compute Enclave. You can use the OCI CLI on its own or with the Compute Web UI to complete tasks. The OCI CLI provides the same core functionality as the Compute Web UI, plus additional commands, such as the ability to run scripts that extend the functionality. The OCI CLI functionality is based on REST APIs which you can access from a browser with this URL:

`https://console.pcasysname.example.com/api-reference`

where *pcasysname* is the name of your Private Cloud Appliance and *example.com* is your domain. You can find the system and domain names on the dashboard of the Compute Web UI or you can ask an administrator for this information.

Installing the OCI CLI

To use the OCI CLI to manage Private Cloud Appliance resources, install and configure the OCI CLI.

Before You Begin

To install and use the OCI CLI, you must have:

- A user account for the Compute Web UI.
- An API signing key pair, required to sign API requests. If you don't already have an RSA public/private key pair in PEM format, you can create one in the OCI CLI configuration steps. To add the public key to your user account, see Adding an API Public Key to a User Profile on page 406.
- A Private Cloud Appliance self-signed certificate.

  This requirement is satisfied during the configuration steps.

You can install the OCI CLI on macOS, Microsoft Windows, or any supported Linux/UNIX operating system:

- Oracle Linux 7 and Oracle Linux 8
- CentOS 7.0 and CentOS 8.x
- Ubuntu 16.04, Ubuntu 18.04, and Ubuntu 20.04

Install the OCI CLI

You can install the OCI CLI on Oracle Linux or macOS operating systems using a package manager. To install on Microsoft Windows or some other operating system, use the install script.

> **Note:**
>
> If you already have the OCI CLI installed and configured, you can skip to Configuring the OCI CLI on page 400 to learn how to further configure the OCI CLI for Private Cloud Appliance.

To install the OCI CLI, its dependencies, and Python, follow the steps for your OS. During installation, respond to the prompts for information as described in Responding to the Install Script Prompts on page 400.

Oracle Linux 8

Run the following commands to install the OCI CLI:

```
$ sudo dnf -y install oraclelinux-developer-release-el8
$ sudo dnf install python36-oci-cli
```

To uninstall the OCI CLI, run:

```
$ sudo dnf remove python36-oci-cli
```

Oracle Linux 7

Run the following command to install the OCI CLI:

```
$ sudo yum install python36-oci-cli
```

To uninstall the OCI CLI, run:

```
$ sudo yum remove python36-oci-cli
```

macOS

You can use Homebrew to install, upgrade, and uninstall the OCI CLI on macOS.

> **Note:**
>
> Optionally, you can install the OCI CLI using the install script. See Using the Install Script for Other Operating Systems on page 399 in this section for details.

- To install the OCI CLI, run:

  ```
  $ brew update && brew install oci-cli
  ```

- To upgrade the OCI CLI, run:

  ```
  $ brew update && brew upgrade oci-cli
  ```

- To uninstall the OCI CLI, run:

  ```
  $ brew uninstall oci-cli
  ```

Microsoft Windows

You can use Microsoft Windows PowerShell to install the OCI CLI.

1. Open the PowerShell console using the Run as Administrator option.
2. Set the `http_proxy` and `https_proxy` environment variables.

> **Important:**
>
> The value of `https_proxy` is the hostname or IP address of your HTTP proxy server.

```
$Env:http_proxy="http://www-proxy.example.com:80"
$Env:https_proxy="http://www-proxy.example.com:80"
```

If your proxy server requires a username and password, or uses a port number other than 80, include that information, as shown in the following example:

```
$Env:https_proxy=http://username:password@proxy.example.com:port
```

Check that your proxy variables are set correctly. Ensure you can connect to internet locations.

```
$Env:http_proxy
$Env:https_proxy
ping https://raw.githubusercontent.com
```

3. The installer enables auto-complete by installing and running a script. To allow this script to run, you must enable the RemoteSigned execution policy.

   To configure the remote execution policy for PowerShell, run the following command:

   ```
   $ Set-ExecutionPolicy RemoteSigned
   ```

4. Force PowerShell to use TLS 1.2 for Microsoft Windows 2012 and Microsoft Windows 2016:

   ```
   $ [Net.ServicePointManager]::SecurityProtocol =
    [Net.SecurityProtocolType]::Tls12
   ```

5. Download the install script:

   ```
   $ Invoke-WebRequest ^
   https://raw.githubusercontent.com/oracle/oci-cli/master/scripts/install/
   install.ps1 ^
   -OutFile install.ps1
   ```

6. Run the install script with or without prompts.

   Run the `install.ps1` script that you downloaded in the previous step.

   To avoid prompts and accept the default values, run the script with the following option:

   ```
   $ install.ps1 -AcceptAllDefaults
   ```

7. Unset the proxy environment variables.

   ```
   $Env:http_proxy=""
   $Env:https_proxy=""
   ```

Using the Install Script for Other Operating Systems

For any other OS, run the following install script to install the OCI CLI, its dependencies, and Python.

```
$ bash -c "$(curl -L https://raw.githubusercontent.com/oracle/oci-cli/
master/scripts/install/install.sh)"
```

To avoid prompts and accept the default values, add the `--accept-all-defaults` option.

Responding to the Install Script Prompts

- If you don't have a compatible version of Python installed in Linux or Microsoft Windows, you are prompted to provide a location for installing the binaries and executables. The script installs Python for you.
- If you don't have a compatible version of Python installed in macOS, you're notified that your version of Python is incompatible. You must upgrade before you can proceed with the installation. The script doesn't install Python for you.
- When prompted to upgrade the OCI CLI to the newest version, respond with **Y** to overwrite an existing installation.
- When prompted to update your PATH, respond with **Y** to be able to invoke the `oci` command without providing the full path to the executable.

Configuring the OCI CLI

On Private Cloud Appliance, before using the OCI CLI, you must configure it for working with Private Cloud Appliance and obtain the system's certificate authority (CA) chain. You can complete the configuration manually or use the config setup tool to help you.

> **Important:**
>
> If you're already using the OCI CLI and have it configured for other purposes, read this section entirely before proceeding with any of the configuration steps.

Obtain the Required Information

Whether you manually configure the OCI CLI or use the setup config tool, there is required information you must provide for the configuration file. Before you begin the configuration process, ensure you have the following:

- User OCID

  The user's OCID is in `ocid1.user.`*`unique_ID`* format. You can copy the user's OCID from the user details page in the Compute Web UI. To navigate to your user details page, select your user name in the Compute Web UI dashboard and then select My Profile.
- Tenancy OCID

  The tenancy OCID is in `ocid1.tenancy..`*`unique_ID`* format. You can copy the tenancy OCID from the tenancy details page in the Compute Web UI. To navigate to the tenancy details page, select your username in the Compute Web UI dashboard and then select Tenancy.
- Region name

  The region name is in *`pcasys1.example.com`* format, where *`pcasys1`* is the name of your Private Cloud Appliance and *`example.com`* is your domain.

  If you have access to the Service Web UI, you can find the system and domain names on the dashboard. Otherwise, ask a Service Web UI administrator for the information.

If you don't already have an existing API public and private key pair, we recommend that you create them as part of the manual or automated OCI CLI configuration. For more information, see Using the Automated Configuration Method on page 401 or Manually Configuring the OCI CLI on page 402.

If you have an existing API public and private key pair that you want to use, ensure they meet the following requirements:

- They're in PEM format.
- Your public key is added to your user profile.
- You know the full path and file name of the private key. For example, `~/.oci/oci_api_key.pem` .

- You have your public key fingerprint, which is in ***xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx*** format. You can find this fingerprint on your profile page in the Compute Web UI or through a terminal using a command. For example:

```
openssl rsa -pubout -outform DER -in ~/.oci/oci_api_key.pem | openssl md5
 -c
```

Using the Automated Configuration Method

If this is the first time using the OCI CLI, the setup config tool walks you through setup process. When you enter the `oci setup config` command it prompts you for the information required for the config file and the API public/private keys and then generates an API key pair and creates the config file.

> **Note:**
>
> Alternatively, you can manually configure the OCI CLI. See Manually Configuring the OCI CLI on page 402.

To configure the OCI CLI using the setup config tool, follow these instructions:

1. From a command window, enter `oci setup config` and follow the prompts. For example:

```
oci setup config
This command provides a walkthrough of creating a valid OCI CLI config
 file.
Enter a location for your config [/home/myuserdir/.oci/config]:
Enter a user OCID: ocid1.user.unique_ID
Enter a tenancy OCID: ocid1.tenancy.unique_ID
```

> **Important:**
>
> For the step *Enter a region by index or name*, you can't enter the region in the required `system.domain` format. Rather, enter any value from the list because the value is meaningless to Private Cloud Appliance. In Step 2 you change the config file to provide the information needed by Private Cloud Appliance.

```
Enter a region by index or name(e.g.
1: ap-chiyoda-1, 2: ap-chuncheon-1, 3: ap-hyderabad-1, 4: ap-melbourne-1,
 5: ap-mumbai-1,
6: ap-osaka-1, 7: ap-seoul-1, 8: ap-sydney-1, 9: ap-tokyo-1, 10: ca-
montreal-1,
11: ca-toronto-1, 12: eu-amsterdam-1, 13: eu-frankfurt-1, 14: eu-zurich-1,
 15: me-dubai-1,
16: me-jeddah-1, 17: sa-santiago-1, 18: sa-saopaulo-1, 19: sa-vinhedo-1,
 20: uk-cardiff-1,
21: uk-gov-cardiff-1, 22: uk-gov-london-1, 23: uk-london-1, 24: us-
ashburn-1,
25: us-gov-ashburn-1, 26: us-gov-chicago-1, 27: us-gov-phoenix-1, 28: us-
langley-1,
29: us-luke-1, 30: us-phoenix-1, 31: us-sanjose-1): 24
```

```
Do you want to generate a new API Signing RSA key pair?
(If you decline you will be asked to supply the path to an existing key.)
 [Y/n]: Y
Enter a directory for your keys to be created [/home/myuserdir/.oci]:
Enter a name for your key [oci_api_key]:
Public key written to: /home/myuserdir/.oci/oci_api_key_public.pem
Enter a passphrase for your private key (empty for no passphrase):
Private key written to: /home/myuserdir/.oci/oci_api_key.pem
Fingerprint: xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx
Config written to /home/myuserdir/.oci/config
```

2. Navigate to the `~/myuserdir/.oci` directory and modify the `config` file to use the correct region, for example:

```
[DEFAULT]
user=ocid1.user.unique_ID
fingerprint=xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx
key_file=/home/myuserdir/.oci/oci_api_key.pem
tenancy=ocid1.tenancy.unique_ID
region=pcasys1.example.com
```

   where *pcasys1* is the name of your Private Cloud Appliance and *example.com* is your domain.

   If you have access to the Service Web UI, you can find the system and domain names on the dashboard. Otherwise, ask a Service Web UI administrator for the information.
3. If you haven't already, upload your API Signing public key through the Compute Web UI. For more information, seeAdding an API Public Key to a User Profile on page 406.

What's Next?

See Obtaining the Certificate Authority Bundle on page 403.

Manually Configuring the OCI CLI

Complete the following steps to manually configure the OCI CLI for Private Cloud Appliance. Ensure you have gathered all the required information.

> **Note:**
>
> Alternatively, you can use the automated method to configure the OCI CLI. See Using the Automated Configuration Method on page 401.

The steps below assume you're on a Linux system and that you have already created a user through the Compute Web UI. However, the basic procedure is the same for other OSs.

1. From a terminal, log into the system where you installed the OCI CLI and create an API key pair. For example:

```
oci setup keys
Enter a passphrase for your private key (empty for no passphrase):
Public key written to: /home/username/.oci/oci_api_key_public.pem
Private key written to: /home/username/.oci/oci_api_key.pem
Public key fingerprint: xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx
```

2. From a browser, sign in to the Compute Web UI.
3. Navigate to your user details page. Select your username in the top right of the page, and then select My Profile. Your user details page is displayed.
4. In the Resources section of your user details page, select API Keys, and then select the Add API Key button.
5. Navigate to the location of your public key or paste the public key contents and then select Upload Key.
6. In your `/home/username/.oci` directory, create a file named `config`. Add a profile section with the required information:

```
[PCA1]
user=ocid1.user...unique_id
key_file=/home/username/.oci/oci_api_key.pem
tenancy=ocid1.tenancy.unique_ID
region=pcasys1.example.com
fingerprint=xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx
```

   In this example, *pcasys1* is the name of your Private Cloud Appliance and *example.com* is your domain.

   If you have access to the Service Web UI, you can find the system and domain names on the dashboard. Otherwise, ask a Service Web UI administrator for the information.

What's Next?

See Obtaining the Certificate Authority Bundle on page 403.

Obtaining the Certificate Authority Bundle

Whether you configured the OCI CLI manually or used the automated tool, you must obtain the Private Cloud Appliance external silo CA chain before you can run commands.

The external silo CA chain must be copied to the system where you're installing the CLI and referenced in the `oci_cli_rc` file.

1. Navigate to your `~/.oci` directory.
2. Copy the external silo CA chain from the following location:

   ```
   https://iaas.system-name.domain-name/cachain
   ```

   Save the CA chain in a file. In this example, the file is named `ca.crt` and is saved in the `~/.oci` directory.
3. In the `~/.oci` directory, create a file named `oci_cli_rc`. Add the profile name and the path to your copy of the external silo CA chain. For example:

   ```
   [PCA1]
   cert-bundle=/home/username/.oci/ca.crt
   ```
4. Set the `OCI_CLI_CERT_BUNDLE` environment variable to the same path as in the previous step.

What's Next?

See Testing the OCI CLI Configuration on page 403.

Testing the OCI CLI Configuration

> **Important:**
>
> If you followed the manual configuration process and attempt to test the configuration by running the commands in this section, you might encounter a warning message stating the permissions on the config file are too open. If this happens, follow the instructions in the warning message to resolve the issue.

After you have installed and configured the OCI CLI, enter a list command to verify that the OCI CLI is working correctly. For example:

```
oci iam user list
```

Using Multiple Profiles

The OCI CLI configuration files `config` and `oci_cli_rc` can define more than one profile. Each profile section in the `config` file references a tenancy within a Private Cloud Appliance. The tenancies can be on different appliances.

In the following example `~/.oci/config` file, the PCA1 profile is for a tenancy on the *pcasys1* appliance, the PCA2 profile is for a tenancy on the *pcasys2* appliance, and the DEFAULT profile is a copy of the PCA1 profile. The DEFAULT profile is used when you haven't specified which profile to use.

In this example, the key file and fingerprint are the same in each profile, but the user OCID is different on the two different appliances or in two different tenancies on the same appliance.

```
[DEFAULT]
user=ocid1.user.unique_ID_1
key_file=/home/username/.oci/oci_api_key.pem
tenancy=ocid1.tenancy.unique_ID_1
region=pcasys1.example.com
fingerprint=58:f8:69:13:e1:a8:51:4d:5a:a0:11:69:ca:09:48:73
[PCA1]
user=ocid1.user.unique_ID_1
```

```
key_file=/home/username/.oci/oci_api_key.pem
tenancy=ocid1.tenancy.unique_ID_1
region=pcasys1.example.com
fingerprint=58:f8:69:13:e1:a8:51:4d:5a:a0:11:69:ca:09:48:73
[PCA2]
user=ocid1.user.unique_ID_2
key_file=/home/username/.oci/oci_api_key.pem
tenancy=ocid1.tenancy.unique_ID_2
region=pcasys2.example.com
fingerprint=58:f8:69:13:e1:a8:51:4d:5a:a0:11:69:ca:09:48:73
```

> **Note:**
>
> If you don't specify a profile to use, the DEFAULT profile is used. If you don't specify a profile and don't have a DEFAULT profile, you must use the `--profile` option in your commands.

To specify a profile, set the profile name as the value of the `OCI_CLI_PROFILE` environment variable:

```
export OCI_CLI_PROFILE=PCA1
```

The `--profile` option is a global option, specified on `oci`, as in the following example:

```
oci --profile PCA2 iam user list
```

You must specify the same profiles in your `oci_cli_rc` file that you specified in your `config` file:

```
[DEFAULT]
cert-bundle=/home/username/.oci/pca1/ca.crt
[PCA1]
cert-bundle=/home/username/.oci/pca1/ca.crt
[PCA2]
cert-bundle=/home/username/.oci/pca2/ca.crt
```

If you have configured multiple profiles, consider creating subdirectories within the `.oci` directory to store the different API keys and external silo CA chains for each profile.

Consider creating a file of environment variables for each profile. In addition to setting `OCI_CLI_PROFILE`, set `OCI_CLI_CERT_BUNDLE` to the same path that you specified in your `oci_cli_rc` file. Set `OCI_CLI_TENANCY` to the OCID of the tenancy for this profile. Giving other compartments and resources names makes commands easier to enter and read. For example:

```
oci network subnet create -c $Networking --vcn-id $VCN1 ...
```

Working with API Signing Keys

On Private Cloud Appliance, if you need to use the OCI CLI or make REST API requests, you must have an RSA API signing public and private key pair in PEM format.

API requests are signed with the private key, and the public key is used to verify the authenticity of the request. The private key is stored locally and the public key is uploaded to a user account. You can have a maximum of three (3) public keys per user account.

> **Note:**
>
> The API signing key pair is **not** the SSH key that you use to access compute instances.

Generating an API Key Pair

If you don't already have an existing API signing public and private key pair, we recommend that you create the key pair as part of the manual or automated configuration. To do so, use the `oci setup keys` command as described

in Manually Configuring the OCI CLI on page 402, or follow the prompts in Using the Automated Configuration Method on page 401.

If you want to create a key pair independent of the OCI CLI configuration, the following sections show you how to do this on Linux, macOS, and Microsoft Windows operating systems. You can then use these keys when you configure the OCI CLI.

Using Linux or macOS

1. Generate the private key.
   - Generate the key encrypted with a passphrase:

     ```
     openssl genrsa -out ~/.oci/oci_api_key.pem -aes128 2048
     ```

     > **Note:**
     >
     > Use of a passphrase is strongly recommended.

   - Generate the key with no passphrase:

     ```
     openssl genrsa -out ~/.oci/oci_api_key.pem 2048
     ```

2. Check the permission on the private key file and change permissions if necessary.

   The file permission must be 600 or 400 to ensure that only you can read the private key file.
3. Generate the public key from your new private key:

   ```
   openssl rsa -pubout -in ~/.oci/oci_api_key.pem -out ~/.oci/
   oci_api_key_public.pem
   ```

   This public key file can have the same permissions as the private key file or can be readable by everyone.

Using Microsoft Windows

1. Install Git Bash for Microsoft Windows.

   See https://git-scm.com/download/win.
2. Include the OpenSSL binary in your Microsoft Windows path.

   On default installations, the `openssl.exe` binary is in the following directory:

   ```
   C:\Program Files\Git\mingw64\bin
   ```

3. Generate the private key.
   - Generate the key encrypted with a passphrase:

     ```
     openssl genrsa -out %HOMEDRIVE%%HOMEPATH%\.oci\oci_api_key.pem -aes128 -
     passout  ^
     stdin 2048
     ```

     > **Note:**
     >
     > Use of a passphrase is strongly recommended.

   - Generate the key with no passphrase:

     ```
     $ openssl genrsa -out %HOMEDRIVE%%HOMEPATH%\.oci\oci_api_key.pem 2048
     ```

4. Check the permission on the private key file and change permissions if necessary.

   The file permission must be set so that only you can read the private key file.

5. Generate the public key from your new private key:

```
openssl rsa -pubout -in %HOMEDRIVE%%HOMEPATH%\.oci\oci_api_key.pem -out  ^
%HOMEDRIVE%%HOMEPATH%\.oci\oci_api_key_public.pem
```

   This public key file can have the same permissions as the private key file or can be readable by everyone.

Adding an API Public Key to a User Profile

An API signing key is a PEM-format RSA public/private key pair, at least 2048 bits.

Use the Compute Web UI to add your own API public key to your profile. You can't use the OCI CLI until you have your API signing key pair in place. If you don't have a login and password for the Compute Web UI, contact an administrator.

A user can have a maximum of three (3) public keys added to their user account. If the user has more than one API public key, the user must specify the key's fingerprint to indicate which key they're using to sign the request.

Using the Compute Web UI

1. From a browser, sign in to the Compute Web UI.
2. Navigate to the user details page.
   - If you're adding a public key to your own user account, select your user icon in the upper right of the Compute Web UI, and then select My Profile.
   - If you're adding a public key to a different user account, select Identity on the navigation menu, select Users, and then select the name of the user in the user list.
3. On the user details page, scroll to the Resources section, select **API Keys**, and then select **Add API Key**.
4. In the **Add Public Key** dialog box, navigate to the location of the public key or paste the public key contents and then select **Upload Key**.

Using the OCI CLI

After you have installed and configured the OCI CLI, you can use the `api-key upload` command to upload additional keys for your user account or upload keys for another user.

1. Get the OCID of the user that needs an API signing key (`oci iam user list`).
2. Use the user API key list command to ensure that the account does not already have the maximum three API signing keys.

   Syntax:

```
oci iam user api-key list --user-id user_OCID
```

3. Run the API key upload command.

   Syntax:

```
oci iam user api-key upload --user-id ocid1.user.unique_ID \
{ --key public_key | --key-file file://public_keyfile.pem }
```

   - *`public_key`* – an RSA public key in PEM format
   - *`public_keyfile`*`.pem` – a file that contains an RSA public key in PEM format

Finding an SPI Public Key Fingerprint

Your public key fingerprint, which is in *`xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx`* format, can be found on your profile page in the Compute Web UI or through a terminal using the following OpenSSL commands:

Linux and macOS:

```
openssl rsa -pubout -outform DER -in ~/.oci/oci_api_key.pem | openssl md5 -c
```

Microsoft Windows:

```
openssl rsa -pubout -outform DER -in \.oci\oci_api_key.pem | openssl md5 -c
```

Deleting an API Signing Key from a User Profile

You can delete your own API signing keys, and tenancy administrators can delete API signing keys for any user in their tenancy.

Using the Compute Web UI

1. In the navigation menu, select **Identity**, and then select **Users**.
2. Select the name of the user account for which you want to delete an API signing key.
3. For the API key that you want to delete, select Actions menu (⋮) and then select **Delete**.

Using the OCI CLI

1. Get the following information:
   - The OCID of the user account from which you want to delete an API signing key.

     ```
     oci iam user list
     ```

   - The fingerprint of the API signing key that you want to delete.

     ```
     oci iam user api-key list --user-id user_OCID
     ```

2. Run the user API key delete command.

   Syntax:

   ```
   oci iam user api-key delete --user-id user_OCID --fingerprint fingerprint
   ```

Understanding Command Syntax and Finding Help

Learn some basic information to help you as you begin using the OCI CLI, such as command syntax, how to find OCIDs, and where to get help with commands for Private Cloud Appliance.

Command Syntax

In general, commands entered in the OCI CLI have the following syntax:

```
oci service type action required-parameters optional-parameters
```

For example, in the following command:

```
oci iam user create --name joeb --description "Product test" --email
 joeb@example.com
```

- `iam` is the service
- `user` is the resource type
- `create` is the action
- `name` and `description` are required parameters
- `email` is an optional parameter

Obtaining OCIDs

When you use the OCI CLI, the majority of commands require an Oracle Cloud ID (OCID):

- `list` commands require the OCID of the compartment where you are looking for the resource.
- `create` commands require the OCID of the compartment where you want to create the resource.
- `get`, `update`, and `delete` commands require the OCID of the resource.
- `move` commands require the OCID of the resource and the OCID of the destination compartment.

Some commands require the OCID of a different resource. For example, creating a route table for a DRG requires the OCID of the DRG.

You can find OCIDs using the OCI CLI or the Compute Web UI. The following lists show you how to find the most commonly needed OCIDs using the OCI CLI.

Block volume service OCIDs

- Boot volume

```
oci bv boot-volume list --availability-domain AD-1 --compartment-
id compartment_OCID
```

- Volume

```
oci bv volume list --compartment-id compartment_OCID
```

- Volume backup policy

```
oci bv volume-backup-policy list --compartment-id compartment_OCID
```

- Volume group

```
oci bv volume-group list --compartment-id compartment_OCID
```

Compute service OCIDs

- Instance

```
oci compute instance list --compartment-id compartment_OCID
```

- Instance VNIC

```
oci compute instance list-vnics --compartment-id compartment_OCID
```

- Volume attachment

```
oci compute volume-attachment list --compartment-id compartment_OCID
```

Identity and access management service OCIDs

- Compartments within a tenancy

```
oci iam compartment list
```

- Compartments within a tenancy and including the tenancy

```
oci iam compartment list --include-root
```

- Compartments and all sub-compartments in a tenancy

```
oci iam compartment list --compartment-id-in-subtree true
```

- Compartment including its sub-compartments

```
oci iam compartment list --compartment-id compartment_OCID --compartment-
id-in-subtree
```

- Group

```
oci iam group list
```

- Policy

```
oci iam policy list --compartment-id compartment_OCID
```

- Tag namespace

```
oci iam tag-namespace list --compartment-id compartment_OCID
```

- User

```
oci iam user list
```

Network service OCIDs

- DHCP options

```
oci network dhcp-options list --compartment-id compartment_OCID [--vcn-
id VCN_OCID]
```

- Route table

```
oci network route-table list --compartment-id compartment_OCID [--vcn-
id VCN_OCID]
```

- Subnet

```
oci network subnet list --compartment-id compartment_OCID [--vcn-
id VCN_OCID]
```

- VCN

```
oci network vcn list --compartment-id compartment_OCID
```

In the Compute Web UI, an OCID Copy button is available on the details page of a resource and often also on the Actions menu (⋮) for the resource in the resource list.

To more easily specify OCIDs, and to make your commands more readable, you might want to set often used OCIDs in environment variables. For example, you could set the tenancy to T. Compartments are the most frequently used OCIDs, and the `--compartment-id` option can be shortened to `-c`.

The following examples show listing all VCNs in the NET compartment and using the large shape instance configuration to launch an instance.

```
oci compute vcn list -c $NET
oci compute-management instance-configuration launch-compute-instance --
instance-configuration-id $INST_CFG_LRG
```

Getting Help with OCI CLI Commands

You can get inline help by appending `--help`, `-h` or `-?` to a command:

- `oci --help` returns a list of commands and global command options.

- `oci` ***`service`*** `--help` returns a summary of the command reference for a service. For example:

```
oci compute -h
Usage: oci compute [OPTIONS] COMMAND [ARGS]...
  Compute Service CLI

Options:
  -?, -h, --help  For detailed help on any of these individual commands,
 enter
                  <command> --help.
Commands:
  boot-volume-attachment          Represents an attachment between a...
  capacity-reservation            A template that defines the...
  console-history                 An instance's serial console data.
  dedicated-vm-host               A dedicated virtual machine host...
  dedicated-vm-host-instance      Condensed instance data when...
  device                          Device Path corresponding to the...
  global-image-capability-schema  Global Image Capability Schema
  global-image-capability-schema-version
                                  Global Image Capability Schema...
  image                           A boot disk image for launching an...
  image-capability-schema         Image Capability Schema
  image-shape-compatibility-entry An image and shape that are...
  instance                        A compute host.
  instance-console-connection     The `InstanceConsoleConnection`...
  measured-boot-report            The measured boot report for a...
  pic                             Partner image catalog (PIC).
  shape                           A compute instance shape that can...
  vnic-attachment                 Represents an attachment between a...
  volume-attachment               A base object for all types of...
```

- `oci` *`service resource_type`* `--help` returns a summary of the command reference for a resource type. For example:

```
oci compute image -h
Usage: oci compute image [OPTIONS] COMMAND [ARGS]...
  A boot disk image for launching an instance. For more information, see
  [Overview of the Compute Service].

  To use any of the API operations, you must be authorized in an IAM
 policy.
  If you're not authorized, talk to an administrator. If you're an
  administrator who needs to write policies to give users access, see
  [Getting Started with Policies].

  **Warning:** Oracle recommends that you avoid using any confidential
  information when you supply string values using the API.

Options:
  -?, -h, --help  For detailed help on any of these individual commands,
 enter
                  <command> --help.

Commands:
  change-compartment  Moves an image into a different compartment
 within...
  create              Creates a boot disk image for the specified
 instance...
  delete              Deletes an image.
  export              Exports an image...
  get                 Gets the specified image.
  import              Imports an exported image from the Oracle Cloud...
```

```
  list                Lists a subset of images available in the
specified...
  update              Updates the display name of the image.
```

- `oci service resource_type action --help` returns the complete command reference for the specified service resource action. For example, the following command displays a full description of creating a compute image, and describes all options:

```
oci compute image create -h
```

For more information, see the Oracle Cloud Infrastructure CLI Command Reference.

Using JSON for Complex Command Input

On Private Cloud Appliance, complex command input includes arrays and objects with more than one value. Complex input is passed as a block of key/value pairs in JSON format. The JSON-formatted input can be provided as a string in the command line or as a file that's referenced in the command line.

Using a JSON String

To pass a JSON block as a string in the Compute Web UI command line, remove the newlines. On macOS, Linux, or UNIX, enclose the entire JSON block in single quotation marks. In Microsoft Windows command lines, enclose the JSON block in double quotation marks, and escape the double quotation marks that are within the block (`\"`).

On any OS or shell, you might need to escape other characters such as dollar signs.

If you receive the message "Parameter '*parameter_name*' must be in JSON format," then your JSON formatting isn't correct. If you copied the JSON format as described in Generating Formatted JSON on page 411, then recheck your command-line format, especially for characters that might need to be escaped.

- macOS, Linux, or UNIX

```
oci compute instance update --instance-id ocid1.image.unique_ID --
freeform-tags '{"Department":"Finance"}'
```

- Microsoft Windows

```
oci compute instance update --instance-id ocid1.image.unique_ID --
freeform-tags "{\"Department\":\"Finance\"}"
```

Using a JSON File

An advantage of storing complex option arguments in files is that you can easily reuse arguments that you know are in correct format. You can store the data exactly the way you copy it from output of the `--generate-param-json-input` option or of the `get` command: You don't need to remove newlines and escape certain characters as you need to do for command-line strings.

When you pass input using a JSON file, the option argument is the file name prefixed with `file://`. The file name can be the name of the file in the same directory where you're running the command, the relative path to the file, or the full path to the file.

Generating Formatted JSON

If you receive the message "Parameter '*parameter_name*' must be in JSON format," then your JSON formatting isn't correct. The following methods help you format the data correctly.

JSON Format for a Single Complex Type Option Value

If an option is complex type, then you can generate the JSON format for that option value by using the `--generate-param-json-input` option. The argument for the `--generate-param-json-input` option is the name of the option for which you want the JSON format, without the `--` option specifier. For example, the

following command shows the JSON format to use to specify route rules (`--route-rules` option) for a route table:

```
network route-table update --generate-param-json-input route-rules
[
  {
    "cidrBlock": "string",
    "description": "string",
    "destination": "string",
    "destinationType": "string",
    "networkEntityId": "string"
  },
  {
    "cidrBlock": "string",
    "description": "string",
    "destination": "string",
    "destinationType": "string",
    "networkEntityId": "string"
  }
]
```

Values that you provide in the option argument replace any existing values. For example, if you already have an ingress rule and you want to add an egress rule, you must specify both the existing ingress rule and the new egress rule. If you specify only the rule that you want to add, any previously existing rules will be gone. Similarly, to add a policy statement or add values for a defined tag, for example, you must respecify the existing statements or values that you want to keep in addition to what you want to add.

Sometimes the output contains a message about choices, such as in the following example:

```
oci compute instance launch --generate-param-json-input source-details
[
  "This parameter should actually be a JSON object rather than an array -
 pick one of the
  following object variants to use",
  {
    "bootVolumeId": "string",
    "sourceType": "bootVolume"
  },
  {
    "bootVolumeSizeInGBs": 0,
    "imageId": "string",
    "kmsKeyId": "string",
    "sourceType": "image"
  }
]
```

When you use the Compute Web UI to create an instance, you select either a Custom Image or a Boot Volume. Similarly, the message in the preceding example tells you to specify either the boot volume block or the image block, not both.

JSON Format for Values of All Options of a Command

The `--generate-full-command-json-input` option shows JSON format for values of all options of the specified command, including values that aren't complex values.

```
oci network route-table update --generate-full-command-json-input >
 route_table_update.json
```

Use the `--from-json` option to pass in your customized version of this output.

```
oci network route-table update --from-json file://route_table_update.json
```

JSON Format of Existing Values

If resources of this type already exist, do a `get` or `list` of the resources to see the JSON formatting and also to check the current values.

```
oci network route-table get --rt-id ocid1.routetable.unique_ID
```

Copy the block you want directly from this output, and then change the values as needed.

In the case where multiple values are allowed, use this method to avoid overwriting values that you want to keep. Copy the block you want from the `get` or `list` output, and then change, add, or delete values as needed.

Formatting and Filtering Command Output

By default, all command output is in JSON format:

```
oci iam region list
{
  "data": [
    {
      "key": "pcasysname",
      "name": "pcasysname"
    }
  ]
}
```

If you prefer, command output can be formatted as a table:

```
oci iam region list --output table
+------------+------------+
| key        | name       |
+------------+------------+
| pcasysname | pcasysname |
+------------+------------+
```

You can filter output using the JMESPath query option for JSON. Filtering is useful when you have a large amount of output. The `--output table` option shows a column for each property of a resource. The output table for an instance, for example, has almost 30 columns and probably overflows the width of your display.

To output only the data that you want, use the `--query` option with the `--output table` option, as shown in the following example:

```
oci compute image list -c ocid1.compartment.unique_ID --output table --query
 "data [*].{ImageOCID:id, OperatingSystem:\"operating-system\"}"
+-------------------------+-----------------+
| ImageOCID               | OperatingSystem |
+-------------------------+-----------------+
| ocid1.image.unique_ID_1 | OracleLinux     |
| ocid1.image.unique_ID_2 | OracleLinux     |
+-------------------------+-----------------+
```

For more information about the JMESPath query language for JSON, see JMESPath.

Using the API to Manage Private Cloud Appliance Resources

The Private Cloud Appliance API is a subset of the Oracle Cloud Infrastructure (OCI) REST API that uses HTTPS requests and responses.

Endpoints follow this format

```
https://<service>.<system_name>.<domain_name>
```

Where <*service*> is the name of the OCI service providing the API you want to use. For example, *iaas* for the core services. <*system_name*> is the name of the infrastructure, typically the same as the display name. <*domain_name*> is the domain the infrastructure is in.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Identity and Access Management (IAM)

Oracle Private Cloud Appliance Identity and Access Management Service (IAM) enables you to control which users have what access to which resources in your appliance tenancy.

You can create users, user groups, and dynamic groups (instance groups) and create policies to allow different types of access to specified resources in specified compartments.

It's the task of a tenancy administrator to control what type of access a user group has, and to which specific resources that access applies. The responsibility to manage and maintain access control can be delegated to other privileged users, for instance by granting them full access to a subcompartment of the tenancy.

In addition to users, instance principals also have authorization to manage resources.

Creating and Managing Compartments

On Private Cloud Appliance, compartments are the primary building blocks for organizing and controlling access to cloud resources. Users can create compartments to separate access to resources in one compartment from access to resources in other compartments.

Compartments contain resources such as cloud instances, virtual cloud networks, and block volumes. Your tenancy is the root compartment where you can create cloud resources and other compartments. You can create hierarchies of compartments that are up to six levels deep. You can limit access to compartment resources to specified user groups. Most resources can be moved between compartments later if your business needs change.

Understanding the Tenancy

On Private Cloud Appliance, a tenancy is a special compartment. The tenancy is the root compartment where you create and administer all resources, including other compartments.

Users, groups, and identity providers are always attached directly to the tenancy, not to any child compartment of the tenancy. You cannot specify a different compartment when you create a user, group, or identity provider. When you use the OCI CLI to operate on a user, group, or identity provider, the OCID of the tenancy from the `config` file is used by default.

Other resources can reside in the tenancy or in any other compartment. Operating on these resources requires you to select the correct compartment in the Compute Web UI or specify the compartment OCID in the OCI CLI or API. Operating on these resources also requires also you to have the required authorization to those resources in that compartment. Check the policy statements of the groups where you are a member to see whether you have the authorizations that you require.

Use the following procedures to get the OCID of the tenancy.

Using the Compute Web UI

1. In the Compute Web UI, select your user profile menu in the upper right of the page.
2. Select the Tenancy option.
3. On the tenancy details page, copy the tenancy OCID.

Using the OCI CLI

Use the oci iam compartment list command.

```
oci iam compartment list [OPTIONS]
```

Look for the `ocid1.tenancy.`*`unique_ID`* OCID.

- With no options, the `compartment list` command lists all compartments that are direct child compartments of the tenancy. The tenancy is the value of the first property listed (`compartment-id`) for every compartment in the list.
- If you specify the `--include-root` option, the tenancy is listed first, and the tenancy OCID is the value of the `id` property (the value of the `compartment-id` property is `null`).

As is true for other resources, in a compartment `list` or `get`, the `compartment-id` compartment is the parent compartment of the `id` compartment.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListCompartments operation.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Listing Compartments

On Private Cloud Appliance, you can list a single compartment, all child compartments of a specified compartment, or all compartments in the tenancy. You can list only compartments for which you have `inspect` authorization on at least one resource.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Compartments**.

   The list shows all compartments that are direct child compartments of the tenancy.
2. To view a compartment that is a subcompartment of a listed compartment, select the name of the listed compartment. On the details page for that compartment, scroll to the Resources section, and select Child Compartments.

   You might need to select the name of a compartment in the Child Compartments list, and repeat this step.

To find the compartment where a particular resource is located, navigate to a list of those resources. Above the resource list, use the Compartment drop-down menu to select the compartment.

Using the OCI CLI

Use the oci iam compartment list command.

```
oci iam compartment list [OPTIONS]
```

- With no options, the `compartment list` command lists all compartments that are direct child compartments of the tenancy.

  If you specify the `--include-root` option, the tenancy is listed first.
- To list all compartments that are direct child compartments of another compartment, specify the OCID of that parent compartment:

  ```
  $ oci iam compartment list --compartment-id ocid1.compartment.unique_ID
  ```

  This output does not list the specified parent compartment and does not list compartments that are deeper in the hierarchy of this parent compartment. This output only lists the direct child compartments of the specified parent compartment.
- To list all compartments and subcompartments in the tenancy, specify the `--compartment-id-in-subtree` option with a value of `true`.

  ```
  $ oci iam compartment list --compartment-id-in-subtree true
  ```

  Specifying the `--compartment-id` option, described in the preceding step, does not change this output: You cannot list just the compartment tree of a particular compartment other than the tenancy.

- To list just one particular compartment, you can specify the compartment name.

  ```
  $ oci iam compartment list --name Acompartment
  ```

  The output is the same as a `get` of that compartment.

  ```
  $ oci iam compartment get --compartment-id OCID_of_Acompartment
  ```

As is true for other resources, in a compartment `list` or `get`, the `compartment-id` compartment is the parent compartment of the `id` compartment.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListCompartments operation.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Creating a Compartment

On Private Cloud Appliance, you can create a compartment in your tenancy or in another compartment. You can create hierarchies of compartments that are up to six levels deep.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Compartments**.

   The list shows all compartments that are direct child compartments of the tenancy.
2. Select the **Create Compartment** button above the list of compartments.
3. In the Create Compartment dialog box, enter the following information:
   - **Name**: A name for this compartment. Compartment names have the following characteristics:
     - Must be unique within the tenancy.
     - Are case insensitive.
     - Can be changed later.
     - Can be no more than 100 characters.
     - Can contain only alphanumeric characters, period (.), hyphen (-), and underscore (_).
   - **Description**: A description for this compartment. This description can be no more than 400 characters and can be changed later.
   - **Create in Compartment**: The compartment in which you want to create the new compartment. The new compartment will be a child compartment of the selected compartment.
   - **Tagging**: (Optional) Add defined or free-form tags for this compartment as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
4. Select the **Create Compartment** button on the dialog box.

Using the OCI CLI

Use the oci iam compartment create command. Specify the OCID of the parent compartment where you want to create this new compartment.

```
oci iam compartment create --compartment-id compartment_OCID --name "text"
 --description "text" [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateCompartment operation.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Applying Tag Defaults

On Private Cloud Appliance, compartments can have resources called tag defaults. Tag defaults are defined tags that are inherited by all resources and child compartments that are created after the tag default is added to the parent compartment.

To learn more about tag defaults and how to add a tag default to a compartment, see Configuring Tag Defaults (IAM on the Appliance) on page 455.

Adding Policies for Access Control

On Private Cloud Appliance, you can assign a new compartment different access control from the access control of the parent compartment.

Child compartments inherit access permissions from their parent compartments. If you want the access to a new compartment to be different from the access to the parent compartment, create an access policy for the new compartment. For example, grant group DevX permission to read all resources in compartment Products and permission to manage all resources in child compartment ProductX. Grant group DevY permission to read all resources in compartment Products and permission to manage all resources in child compartment ProductY. Because of inheritance, group DevX can read all resources in compartment ProductY, and group DevY can read all resources in compartment ProductX.

For information about creating and attaching policies, see "Managing Policies."

Adding Resources to a Compartment

On Private Cloud Appliance, you add resources to a compartment by specifying that compartment when you create the resource or by moving a resource to the compartment from a different compartment.

Before you move a resource to a new compartment, review the documentation for the particular resource for information such as whether attached resources move with the moved resource.

After you move a resource to a new compartment, check whether the moved resource has the correct tags and policies applied. You might need to manually delete and add tags and policies.

The Resources section on a compartment details page in the Compute Web UI, and the compartment `list` and `get` commands in the OCI CLI, do not show all of the resources that belong to a compartment. For resources that are not listed, go to the Compute Web UI page for that resource, such as VCNs or instances, and select the compartment from the Compartment drop-down menu above the resource list. In the OCI CLI, specify the compartment OCID when you list the resources.

Updating a Compartment

On Private Cloud Appliance, you can change the name, description, and tags of a compartment.

To add, change, or remove tags, see Applying Tags to an Existing Resource on page 459.

You cannot change the parent compartment. To change the parent compartment, see Moving a Compartment to a Different Compartment on page 418.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Compartments**.

   The list shows all compartments that are direct child compartments of the tenancy. If the compartment that you want to update is not listed, navigate to the compartment that you want to update, as described in Listing Compartments on page 415.
2. For the compartment that you want to update, select the Actions menu (⋮) and select the **Edit** option.
3. In the Editing *`compartment_name`* Compartment dialog box, make the changes.
4. Select the **Save Changes** button in the dialog box.

Using the OCI CLI

Use the oci iam compartment update command. Specify the OCID of the compartment that you want to update.

```
oci iam oci iam compartment update --compartment-id compartment_OCID
 [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateCompartment operation.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Moving a Compartment to a Different Compartment

On Private Cloud Appliance, you can move a compartment to a different parent compartment in the same tenancy.

When you move a compartment, all subcompartments of the compartment are moved. Some resources of the moved compartment are moved. You can separately move other resources as needed. See the documentation for the particular resource type for more information.

After you move a compartment to a new parent compartment, the access policies of the new parent take effect and the policies of the previous parent no longer apply. Groups who had access to the compartment and its resources in the previous parent compartment lose their access when the compartment is moved. Groups who have access in the new parent compartment gain access to the moved compartment and its resources.

Tag defaults that are automatically applied to all resources created in the new parent are not automatically applied to the newly moved compartment and its resources. You might need to separately delete and add tag defaults to the moved compartment and delete and add tags to moved resources.

To move a compartment, you must belong to a group that has `manage all-resources` permissions on the lowest shared parent compartment of the current compartment and the destination compartment.

Using the Compute Web UI

Moving a compartment to a new parent compartment using the Compute Web UI is not supported.

Using the OCI CLI

Use the oci iam compartment move command. Specify the OCID of the compartment that you want to move and the OCID of the new parent compartment.

```
oci iam oci iam compartment update --compartment-id compartment_OCID --
target-compartment-id destination_compartment_OCID
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the MoveCompartment operation.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Compartment

To delete a compartment on Private Cloud Appliance, you must first move, delete, or terminate all of the resources in the compartment, including any policies that are attached to the compartment.

Before you begin, check the move and delete capabilities for all resources in the compartment.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Compartments**.

The list shows all compartments that are direct child compartments of the tenancy. If the compartment that you want to delete is not listed, navigate to the compartment that you want to delete, as described in Listing Compartments on page 415.

2. For the compartment that you want to delete, select the Actions menu (⋮) and select the **Delete Compartment** option.

   If the **Delete Compartment** option is not selectable, then you might not have permission to delete this compartment.
3. In the Delete Compartment confirmation dialog box, select the **Delete** button.

   The compartment status changes to Deleting.

   In the Resources section on the compartment details page, select Work Requests and view the details of the compartment delete. When the work request is completed, the compartment is removed from the compartments list. If the work request fails, the compartment status returns to Active.

Using the OCI CLI

Use the oci iam compartment delete command. Specify the OCID of the compartment that you want to delete.

```
oci iam oci iam compartment delete --compartment-id compartment_OCID
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteCompartment operation.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Creating and Managing User Accounts

On Private Cloud Appliance, administrators can create users that have defined access to resources in a tenancy.

By default, a Private Cloud Appliance tenancy has an administrative user in an administrators group, and a policy enables the administrators group to manage the tenancy. To limit a user to managing only a subset of resources in the tenancy or another compartment, or to have less than full management access to some resources, the administrator can create a user account, add the user account to one or more groups, and create one or more policies for those groups.

A user account is not automatically a member of any group. A user that's not a member of any group is visible in the tenancy but doesn't have access to any resources.

Creating a User

When you create a user, the user is automatically created in the tenancy. You can't specify a different compartment for the user.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Users**.
2. Above the user list, select the **Create User** button.
3. In the Create User dialog box, enter the following information:
   - **Name**: A name for this user account. User names have the following characteristics:
     - Must be unique within the tenancy. You can create a user with the same name as a user that has been deleted.
     - Are case insensitive.
     - Cannot be changed later.
     - Must be at least two and no more than 100 characters.

    - Can contain only alphanumeric characters, period (.), hyphen (-), underscore (_), plus sign (+), and at sign (@).
  - **Description**: A description for this user, such as the full name of the person or a brief description of the account. The description is 1-400 characters.
  - **Email Address**: (Optional) The email address for the user. The email address can be added or changed later.
  - **Password**: (Optional) To enable this user to log in to the Compute Web UI, check the box labeled "Generate a temporary password for this user."
  - **Tagging**: (Optional) Add defined or free-form tags for this user account as described in Adding Tags at Resource Creation on page 458. Tags can be added or changed later.
4. Select the **Create User** button in the dialog box.

   If you checked the box labeled "Generate a temporary password for this user," a Temporary Password for New User dialog pops up, showing the temporary password. You cannot retrieve this password again after you close this dialog. Copy the temporary password, save the password to a safe place for delivery to the user, and select the "I have made a note of the password" button.

   The details page of the new user is displayed.

Next steps:

1. Provide the user with a temporary password so that the user can set their own permanent Compute Web UI password.
   - If you checked the box labeled "Generate a temporary password for this user," provide the temporary password that you copied from the Temporary Password for New User dialog.
   - If you did not check the box labeled "Generate a temporary password for this user," or did not save that password, follow the instructions in "Providing a Temporary Compute Web UI Password" to generate a temporary password for the user.
2. Add this user to at least one group. See "Adding a User to a Group by Updating the User."

Using the OCI CLI

Use the oci iam user create command and required parameters to create a new user.

```
oci iam user create --description "text" --name "text" [OPTIONS]
```

In the OCI CLI, a description must be provided but its value can be an empty string.

Optionally provide your tenancy's OCID for the compartment value. By default, the root compartment OCID from the config file is used.

See Adding Tags at Resource Creation on page 458 to add defined and free-form tags.

The output of this command is the same as the output of the `user get` command.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Next steps:

1. Provide the user with a temporary password so that the user can set their own permanent Compute Web UI password. See "Providing a Temporary Compute Web UI Password" to generate a temporary password for the user.
2. Add this user to at least one group. See "Adding a User to a Group by Updating the User."

Using the API

Use the CreateUser operation to create a new user.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Next steps:

1. Provide the user with a temporary password so that the user can set their own permanent Compute Web UI password. See "Providing a Temporary Compute Web UI Password" to generate a temporary password for the user.
2. Add this user to at least one group. See Adding a User to a Group by Updating the User on page 423.

Providing a Temporary Compute Web UI Password

On Private Cloud Appliance, provide a temporary password for new users and for users who forget their password.

This procedure generates a temporary one-time password. When the user signs in to the Compute Web UI using this password, they are required to change the password before proceeding. The generated temporary password expires after seven (7) days.

A tenancy administrator can provide a temporary password for any user. Users must set their own permanent passwords by following the instructions in Setting Your Own Compute Web UI Password.

> **Note:**
>
> Passwords for federated users are not managed through the IAM service. See information from your federated identity provider.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Users**.

   If the user that needs a new password is in the Inactive state, see "Unlocking a User."
2. For the user that needs a new password, select the Actions menu (⋮) and select the **Change Password** option.
3. In the Change Password dialog box, select the Create Temporary Password button.

   A Password Changed dialog box pops up. The New Password field contains the temporary password.
4. Copy and save this temporary password.

   You cannot retrieve this password again after you close this dialog box. Copy the temporary password, and save the password to a safe place for delivery to the user.
5. Select the Close button on the dialog.
6. Deliver this temporary one-time password to the user. The user must follow the rules stated in "Setting Your Own Compute Web UI Password" when setting their new password.

Using the OCI CLI

Use the oci iam user ui-password create-or-reset command and required parameters to generate a temporary Compute Web UI password for a user.

Use the oci iam user list command to confirm that the user is active. If the lifecycle-state of the user is `INACTIVE`, see "Unlocking a User."

```
oci iam user ui-password create-or-reset --user-id ocid1.user.unique_ID
 [OPTIONS]
```

Copy the password value from the command output and deliver this temporary one-time password to the user. The user must follow the rules stated in when setting their new password.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateOrResetUIPassword operation to generate a temporary Compute Web UI password for a user.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Setting Your Compute Web UI Password

On Private Cloud Appliance, users don't require an access policy to set their own Compute Web UI password.

Use this procedure to set your Compute Web UI password initially, or to reset your password if you forgot your password.

Using the Compute Web UI

1. Get the temporary password that was generated for you.
2. On the login screen for the Compute Web UI, enter your user name.
3. Enter the temporary password.

   A dialog box pops up that says your password has expired and you must create a new password.
4. Select the Change my password button.
5. In the Change My Password dialog box, enter the temporary password in the Current Password field.
6. Enter a new password in the New Password field and again in the Confirm New Password field.

   Passwords must be at least 12 characters in length and contain at least one of each of the following: uppercase character, lowercase character, number, and symbol.
7. Select the Save Changes button.

   A dialog box pops up that says your password has been successfully updated.
8. Select the Continue button.
9. Log in using your new password.

Changing Your Compute Web UI Password

On Private Cloud Appliance, users do not require an access policy to change their own Compute Web UI password.

Use this procedure to change your Compute Web UI password while your current password still works.

Using the Compute Web UI

1. In the top right corner of the Compute Web UI, select your user menu.
2. Select the Change My Password option.
3. In the Change My Password dialog box, enter your current password in the Current Password field.
4. Enter a new password in the New Password field and again in the Confirm New Password field.

   Passwords must be at least 12 characters in length and contain at least one of each of the following: uppercase character, lowercase character, number, and symbol.
5. Select the Save Changes button.

   A dialog box pops up that says your password has been successfully updated.
6. Select the Continue button.

Viewing User Information and Group Membership

On Private Cloud Appliance, you can show information about a user, including which groups the user is a member of.

When you know what groups a user belongs to, you can view policies that name those groups to find what authorizations the user has.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Users**.

   The Users list page shows all users of the tenancy because user accounts cannot be in different compartments. All user resources are in the tenancy.
2. Select the name of the user for which you want more information.
3. On the details page for that user account, scroll down to the Resources section and select Groups.

   The list of groups that this user is a member of is shown.

4. To see the full list of members of a group, select the name of the group in the Groups list.
5. To see what authorizations these users have, do the following:
   a. Select the Groups link in the breadcrumb at the top left of the group details page.
   b. In the left navigation panel on the groups list page, select Policies.
   c. Select a policy name from the list to view its details page.
   d. On the policy details page, scroll to the Resources section and view the policy statements to find statements that grant authorizations to the group that you are interested in. A particular group could be mentioned in more than one policy.

Using the OCI CLI

1. Use the oci iam user list command to list all users.

   Copy the OCID of the user that you are interested in.
2. Use the oci iam user list-groups command to list all groups that this user is a member of.

   Note the name of each group.
3. Use the oci iam policy list command to show the statements in each policy.

   In the policy statements, look for the group name or `any-user` to see what authorizations a particular user has.

To list all the members of a group, use the Compute Web UI or the API.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

1. Use the ListUsers operation to list all users.
2. Use the ListUserGroupMemberships operation to do the following:
   - List all groups that have the specified user as a member. Specify the `userId`.
   - List all members of the specified group. Specify the `groupID`.
   - Determine whether the specified user is a member of the specified group. Specify both the `userId` and `groupID`.

The ListPolicies operation does not show statements in policies. To see the statements in policies, use the Compute Web UI or the OCI CLI.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Adding a User to a Group by Updating the User

On Private Cloud Appliance, a user must be a member of at least one group to have access to any resources.

Using the Compute Web UI

As an alternative to using the Users Compute Web UI page, you can use the Groups page as described in "Adding a User to a Group by Updating the Group."

1. In the Compute Web UI navigation menu, select **Identity**, then select **Users**.
2. Select the name of the user that you want to add to a group.
3. On the details page for the user, scroll to the Resources section and select Groups.
4. Select the **Add User to Group** button.
5. In the Add User to Group dialog, select a group from the drop-down list, and then select the **OK** button.

   The user is added to the selected group.

Using the OCI CLI

Use the oci iam group add-user command to add the specified user to the specified group.

```
oci iam group add-user --group-id group_OCID --user-id user_OCID
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the AddUserToGroup operation to add the specified user to the specified group.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Removing a User from a Group by Updating the User

On Private Cloud Appliance, a user must be a member of at least one group to have access to any resources.

Using the Compute Web UI

As an alternative to using the Users Compute Web UI page, you can use the Groups page as described in "Removing a User from a Group by Updating the Group."

1. In the Compute Web UI navigation menu, select **Identity**, then select **Users**.
2. Select the name of the user that you want to remove from a group.
3. On the details page for the user, scroll to the Resources section and select Groups.
4. Select the Actions menu (⋮) and select the **Remove from Group** option.

   The selected group is removed from the user's Groups list.

User Using the OCI CLI

Use the oci iam group remove-user command to remove the specified user from the specified group.

```
oci iam group remove-user --group-id group_OCID --user-id user_OCID
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the RemoveUserFromGroup operation to remove the specified user from the specified group.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Modifying a User

On Private Cloud Appliance, you can change a user account's description and email address.

You can add, change, or remove tags as described in Applying Tags to an Existing Resource on page 459.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Users**.
2. For the user account that you want to modify, select the Actions menu (⋮) and select the **Edit** option.
3. In the Edit *username* dialog box, modify the account's description, email address, or tags.
4. Select the **Save Changes** button in the dialog box.

Using the OCI CLI

Use the oci iam user update command to modify the description, email address, or tags of the specified user.

```
oci iam user update --user-id user_OCID [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateUser operation to modify the description, email address, or tags of the specified user.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Unlocking a User

On Private Cloud Appliance, you can unlock a user that is in the Inactive state. A user might be in the Inactive state after too many incorrect login attempts.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Users**.
2. For the user account that you want to unlock, select the Actions menu (⋮) and select the **Unblock** option.

   The user transitions from the Inactive state to the Active state.

Using the OCI CLI

Use the oci iam user update-user-state command to modify the description, email address, or tags of the specified user.

```
oci iam user update-user-state --user-id user_OCID --blocked false
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateUserState operation to unlock a user that is in the Inactive state. Set the `blocked` attribute to `false`.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a User

On Private Cloud Appliance, you cannot delete a user if the user is a member of any group. You cannot delete your own user.

When you delete a user, all API keys associated with that user account are also deleted.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Users**.
2. Select the name of the user that you want to delete.
3. Ensure that the user is not a member of any group.

   On the user details page, scroll to the Resources section and select Groups. To remove this user from a group, select the Actions menu (⋮) for the group in the Groups list, and select the **Remove from Group** option.
4. At the top of the user details page, select the **Delete** button.
5. On the Delete User confirmation dialog, select the **Confirm** button.

Using the OCI CLI

Use the oci iam user delete command to delete the specified user.

Use the oci iam user list-groups command to ensure that the user is not a member of any group.

```
oci iam user delete --user-id user_OCID
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteUser operation to delete the specified user.

Use the ListUserGroupMemberships operation to ensure that the user is not a member of any group.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Creating and Managing User Groups

Access to cloud resources is granted to groups, not directly to users. A user account isn't automatically a member of any group. To enable a user to do any work with cloud resources, you must add the user to a group and then create an access policy for that group.

A group is a set of users who have the same type of access to the same set of cloud resources. Organize users into groups according to which compartments and resources they need to access and how they need to work with those resources. A user can be a member of more than one group.

Creating a Group

On Oracle Private Cloud Appliance, when you create a group, the group is automatically created in the tenancy. You can't specify a different compartment for the group.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Groups**.
2. Select **Create Group**.
3. Enter the following information:
   - **Name**: A name for this group. Group names have the following characteristics:
     - Must be unique within the tenancy. You can create a group with the same name as a group that has been deleted.
     - Are case insensitive.
     - Can't be changed later.
     - Can be no more than 100 characters.
     - Can contain only alphanumeric characters, period (.), hyphen (-), and underscore (_).
   - **Description**: A description for this group. The description has the following characteristics:
     - Must be 1-400 characters.
     - Doesn't need to be unique.
     - Can be changed later.
     - Avoid entering confidential information.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
4. Select **Create Group**.

   The details page of the new group is displayed.

   Next steps:

   - Create an access policy for this group or add this group to an existing policy. A group has no permissions unless it's the subject of at least one policy. See Managing Policies on page 435.
   - Add users to this group. See Adding a User to a Group on page 427.

Using the CLI

Use the oci iam group create command and required parameters to create a group.

```
oci iam group create --name text --description "text" [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Run the CreateGroup operation to create a group.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing Group Information and Membership

On Private Cloud Appliance, you can view group information and group members.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Groups**.
2. Select the name of the group about which you want more information.
3. Under **Resources**, select **Group Members**.

   The list of users that belong to this group is shown.
4. To see the full list of groups where a user is a member, select the name of the user in the **Group Members** list.

   Under **Resources**, select that user and select **Groups**.

Using the CLI

Use the oci iam group list-user command and required parameters to list users in a group.

```
oci iam group list-users --group-id group_OCID [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the GetGroup operation to list group information.

Use the ListUserGroupMemberships operation to list users in a group.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Adding a User to a Group

On Private Cloud Appliance, users must be members of groups to have access to resources.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Groups**.
2. Select the name of the group where you want to add users.
3. On the details page, under **Resources**, select **Group Members**.
4. At the top of the **Group Members** list, select **Add User to Group**.
5. In the **Add User to Group** dialog box, select a user from the drop-down list, and then select **OK**.

   The selected user is added to the Group Members list.

Using the CLI

Use the oci iam group add-user command and required parameters to add a user to a group.

```
oci iam group add-user --group-id group_OCID --user-id user_OCID [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the AddUserToGroup operation to add a user to a group.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Removing a User from a Group

On Private Cloud Appliance, you can remove a user from a group by updating the group.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Groups**.
2. Select the name of the group where you want to remove a user.
3. On the details page, under **Resources**, select **Group Members**.
4. In the **Group Members** list, select Actions menu (⋮) for the user that you want to remove from the group, and select **Remove from Group**.
5. At the confirmation prompt, select **OK**.

   The user is removed from the group.

Using the CLI

Use the oci iam group remove-user command and required parameters to remove a user from a group.

```
SYNTX [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the RemoveUserFromGroup operation to remove a user from a group.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Modifying a Group

On Private Cloud Appliance, You can change the description for a group and modified tags.

To add, change, or remove tags, see Applying Tags to an Existing Resource on page 459.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Groups**.
2. For the group that you want to modify, select Actions menu (⋮), and select **Edit**.
3. In the dialog box, modify the group's description or tags.
4. Select **Save Changes**.

Using the CLI

Use the oci iam group update command and required parameters to modify a group.

```
oci iam group update --group-id group_OCID [ --description desc ] [ --
defined-tags tags ] [ --freeform-tags tags ] [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateGroup operation to modify a group.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Group

On Private Cloud Appliance, you can delete a group that doesn't have any members.

The Console instructions in this section describe how to remove users. See also Removing a User from a Group on page 428.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Groups**.
2. Select the name of the group that you want to delete.
3. Ensure that the group doesn't have any members.

   On the group details page, under **Resources**, select **Group Members**. To remove a user from the group, select the Actions menu (⋮) for the user in the **Group Members** list, and select **Remove from Group**.
4. At the top of the group details page, select **Delete**.
5. On the **Delete Group** confirmation dialog box, select **Confirm**.

Using the CLI

Use the oci iam group delete command and required parameters to delete a group.

```
SYNTX [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteGroup operation to delete a group.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Identity Federation with Microsoft Active Directory for Tenancy User Accounts

Federating enables users at your company to use the same login credentials for the Private Cloud Compute Web UI that they already use for other logins in the company.

To *federate*, an administrator creates a trust relationship between the existing identity provider and Private Cloud Appliance Infrastructure. When this relationship is established, federated users are prompted with a *single sign-on* when accessing the Compute Web UI.

> **Tip:**
>
> We document Microsoft Active Directory as a reference implementation, but Private Cloud Appliance works with other SAML 2.0-compliant identity providers, including Oracle Access Manager. For information and instructions, see Using Oracle Access Manager to Sign On.

You can federate multiple Active Directory (AD) accounts with Private Cloud Appliance Infrastructure. Each federation trust is for a single AD account. To create a trust, you perform some tasks in the Private Cloud Appliance Compute Web UI and some tasks in Active Directory Federation Services (ADFS).

Before you begin federating, ensure you have completed the following tasks:

- Installed and configured Microsoft ADFS for your organization.
- Created groups in AD that will map to groups in Private Cloud Appliance.
- Created users in AD who will sign in to the Private Cloud Appliance Compute Web UI.

> **Note:**
>
> Consider using a common prefix to name AD groups that you intend to map to Private Cloud Appliance. For example, use AD group names such as <*name*>_Administrators, <*name*>_NetworkAdmins, <*name*>_InstanceLaunchers.

The federation procedure for tenancy user accounts is the same as for administrator accounts. See Identity Federation with Microsoft Active Directory for Administrator Accounts on page 185. Links to detailed instructions are provided at the end of that page.

Configuring Instances for Calling Services

A Private Cloud Appliance compute instance can be configured to enable applications running on the instance to call services and manage resources similar to the way Private Cloud Appliance users call services to manage resources.

An *instance principal* is a compute instance that is authorized to perform actions on service resources. Applications running on an instance principal can call services and manage resources similar to the way Oracle Private Cloud Appliance users call services to manage resources. The instance is a principal actor just as a user is a principal actor. When you use instance principals, you do not need to configure user credentials or a configuration file on the instance to run applications that need to manage service resources.

To grant authorizations to an instance principal, include the instance as a member of a dynamic group. A dynamic group provides authorizations to instances just as a user group provides authorizations to users.

The IAM service feature that enables instances to be authorized actors (or principals) to perform actions on service resources is called an *instance principal*.

Perform the following steps to set up and use an instance as a principal:

- Configure the instance firewall to enable the instance to access service endpoints. See Configuring Instance Firewalls to Allow Calling Services on page 430.
- Ensure the instance is included in a dynamic group that grants permissions to access the required resources. See Creating and Managing Dynamic Groups on page 433.

  The instance must be created in or moved to a compartment that is named in a matching rule of the dynamic group, or the instance must have a resource tag assigned that is named in a matching rule. See Writing Matching Rules on page 433.

Configuring Instance Firewalls to Allow Calling Services

This topic describes how to modify the instance firewall configuration and how to create a `systemd` service to restore the changes if the system reboots.

Modify the Firewall Configuration

As a privileged user, modify the instance firewall configuration to enable the instance to access service endpoints such as `iaas` and `identity`.

Use the `iptables` command to add the following `BareMetalInstanceServices` rules to the instance firewall:

```
iptables -I BareMetalInstanceServices 14 -p tcp -d 169.254.169.254 --dport
 443 -j ACCEPT
iptables -I BareMetalInstanceServices 14 -p tcp -d 169.254.240.254 --dport
 443 -j ACCEPT
```

The first entry is required for all endpoints. The second entry is required to contact the Object Storage endpoint.

Make the Configuration Changes Persistent

Use the following procedure to make these firewall configuration changes persist across instance reboots.

1. Save the updated IP tables configuration.

   ```
   iptables-save > /etc/sysconfig/iptables.rules
   ```

2. Create a script to automatically restore the current (modified) firewall configuration on reboot.

   In this example, the script is named `/sbin/restore-iptables.sh`. The following is the content of the file `/sbin/restore-iptables.sh`:

   ```
   #!/bin/sh
   /sbin/iptables-restore < /etc/sysconfig/iptables.rules
   ```

3. Set executable bit on the script.

```
chmod +x /sbin/restore-iptables.sh
```

4. Create a `systemd oneshot` service to execute the `/sbin/restore-iptables.sh` script at boot time.

   In this example, the service is named `/etc/systemd/system/restore-iptables.service`. The following is the content of the file `/etc/systemd/system/restore-iptables.service`:

```
[Unit]
Description=Restore IP Tables
After=cloud-final.service

[Service]
ExecStart=/sbin/restore-iptables.sh
User=root
Group=root
Type=oneshot

[Install]
WantedBy=multi-user.target
```

5. Reload the `systemd` manager configuration, and enable the service to run at boot time.

```
systemctl daemon-reload
systemctl enable restore-iptables
```

Configuring Instance Certificates to Allow Calling Services

By default, Private Cloud Appliance endpoints (such as `iaas` and `identity`) offer a certificate that is signed by a CA that is specific to that appliance. By default, operating systems do not trust certificates that are signed by a CA that is specific to this Private Cloud Appliance. If the OS does not trust the certificates that are offered, attempts to use the OCI SDK or OCI CLI will fail with a `CERTIFICATE_VERIFY_FAILED` error.

Implement one of the solutions described in this topic to successfully use the OCI SDK or OCI CLI on the instance.

> **Note:**
>
> Any user who can SSH to the instance automatically inherits the privileges granted to the instance.

Option 1: Bring Your Own Certificate (BYOC)

If you have a certificate that is signed by a CA that your OS trusts, configure the Private Cloud Appliance to offer that certificate.

On a Linux OS, the following command lists CAs that are trusted by default:

```
trust list --filter=ca-anchors
```

For information about how to provide your own certificate, see "Accessing External Interfaces with Your CA Trust Chain".

Option 2: Specify in the SDK Code the CA Bundle to Use

This method copies the appliance-specific CA bundle to the instance, but does not verify the server's certificate (`--insecure`). To ensure security, verify the content of the retrieved bundle (`external_ca.crt`).

1. Retrieve the certificate from the `iaas` endpoint of the appliance.

```
curl --insecure -sS -o external_ca.crt --noproxy "*" https://
iaas.pca_name.domain_name/cachain
```

   This command could be in a script that is passed to the instance at launch time by using either the `--user-data-file` option or the `--metadata` option with a `user_data` field. The script will be run by cloud-init inside the instance during init, saving the effort of manually retrieving this certificate file on a large number of instances.
2. Verify the content of the CA bundle saved in the `external_ca.crt` file.
3. Specify the CA bundle in the Python SDK code.

```
signer = oci.auth.signers.InstancePrincipalsSecurityTokenSigner(
    federation_client_cert_bundle_verify="/home/opc/external_ca.crt"
)
identity_client = oci.identity.IdentityClient(config={}, signer=signer)
identity_client.base_client.session.verify = "/home/opc/external_ca.crt"
```

Option 3: Globally Trust the Private Cloud Appliance CA Bundle

This method is the same as the preceding method with the following difference: Instead of specifying the CA bundle in the SDK code, this method adds the CA bundle to the trust chain.

> **Note:**
>
> When the CA bundle is added to the trust chain, every application on this compute instance will trust certificates signed with the CA specified in this bundle. Consider whether this is an acceptable security risk.

1. Retrieve the certificate from the `iaas` endpoint of the appliance.

```
curl --insecure -sS -o external_ca.crt --noproxy "*" https://
iaas.pca_name.domain_name/cachain
```

2. Verify the content of the CA bundle saved in the `external_ca.crt` file.
3. Update the global CA trust chain.

```
cp external_ca.crt /etc/pki/ca-trust/source/anchors/
update-ca-trust extract
```

Steps 1 and 3 in this method could be in a script that is passed to the instance at launch time by using either the `--user-data-file` option or the `--metadata` option with a `user_data` field. The script will be run by cloud-init inside the instance during init, saving the effort of performing these steps manually on a large number of instances.

Configuring Python SDK and OCI CLI for Instance Principals

This topic describes how to enable instance principal authorization for the Python SDK, OCI CLI, or Terraform.

Enabling Instance Principal Authorization for the Python SDK

In your SDK for Python, create an `oci.auth.signers.InstancePrincipalsSecurityTokenSigner` object as shown in the following example:

```
# By default this will hit the auth service in the region returned by
 http://169.254.169.254/opc/v1/instance/region on the instance.

signer = oci.auth.signers.InstancePrincipalsSecurityTokenSigner()
identity_client = oci.identity.IdentityClient(config={}, signer=signer)
```

To refresh the token without waiting, use the following command:

```
signer.refresh_security_token()
```

Enabling Instance Principal Authorization for the OCI CLI

Set the authorization option (`--auth`) for a command as shown in the following example:

```
oci os ns get --auth instance_principal
```

Alternatively, set the following environment variable:

```
OCI_CLI_AUTH=instance_principal
```

If both are set, the value set for `--auth` takes precedence over the environment variable.

Enabling Instance Principal Authorization for Terraform

Set the `auth` attribute to "InstancePrincipal" in the provider definition as shown in the following example:

```
variable "region" {}

provider "oci" {
   auth = "InstancePrincipal"
   region = "${var.region}"

}
```

When you use instance principal authorization you don't need to include the `tenancy_ocid`, `user_ocid`, `fingerprint`, and `private_key_path` attributes.

Creating and Managing Dynamic Groups

On Private Cloud Appliance, dynamic groups are groups of compute instances that meet the criteria defined for the group.

The membership of the dynamic group changes as instances newly satisfy the criteria or no longer satisfy the criteria. For example, if the members of a dynamic group are defined to be all instances in a specified compartment, the group membership changes when instances are created in or removed from that compartment.

Policies permit applications that are running on instances that are members of dynamic groups to make API calls to perform actions on service resources. Instances authenticate by using certificates that are assigned to each member instance.

Creating a Dynamic Group

When you create a dynamic group, the group is automatically created in the tenancy. You cannot specify a different compartment for the dynamic group.

Writing Matching Rules

A rule specifies one or more conditions for membership in the dynamic group. Type the rule into the text box. To add another rule, select the **+Another Matching Rule** button.

A rule with a single condition has the following syntax:

```
variable = | != value
```

A rule with multiple conditions has the following syntax:

```
any | all {variable =|!= value, variable =|!= value,...}
```

`variable` is one of the following:

- `instance.compartment.id`. The OCID of the compartment where the instance resides.

- `instance.id`. The OCID of the instance.
- `tag.tagnamespace.tagkey.value = tagvalue`. The *tagnamespace*, *tagkey*, and *tagvalue* all must match to include this resource in the group.
- `tag.tagnamespace.tagkey.value`. Only the *tagnamespace* and *tagkey* must match to include this resource in the group. The value of the tag is not considered.

The following matching rule includes all instances in the specified compartment:

```
instance.compartment.id = 'ocid1.compartment.unique_ID'
```

The following matching rule includes all instances in the specified compartment except for the two instances specified with `instance.id !=`:

```
all {instance.compartment.id = 'ocid1.compartment.unique_ID',
 instance.id != 'ocid1.instance.unique_ID', instance.id !=
 'ocid1.instance.unique_ID2'}
```

The following matching rule includes all instances that have a tag applied that has tag namespace Product1, key ProjectA, and value abc123:

```
tag.Product1.ProjectA.value = abc123
```

Updating a Dynamic Group

On Private Cloud Appliance, you can change the description and matching rules (members) for a dynamic group.

You can add, change, or remove tags as described in Applying Tags to an Existing Resource on page 459.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Dynamic Groups**.
2. (Optional) Modify the description.
   a. In the dynamic groups list, for the dynamic group that you want to modify, select the Actions menu (⋮), and select **Edit**.
   b. In the **Edit** dialog box, modify the dynamic group's description or tags.
   c. Select **Save Changes**.
3. (Optional) Modify the matching rules.
   a. In the dynamic groups list, select the name of the dynamic group that you want to modify.
   b. On the details page, select **Edit All Matching Rules**.
   c. Edit the matching rule in the text box.
   d. Select **Save Changes**.

Using the CLI

Use the oci iam dynamic-group update command and required parameters to update a dynamic group.

```
oci iam dynamic-group update --dynamic-group-id dynamic_group_OCID [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateDynamicGroup operation to update a dynamic group.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Dynamic Group

On Private Cloud Appliance, when you delete a dynamic group, the compute instances that had been members of the group no longer have the authorizations that they had through the policies applied to the dynamic group.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Dynamic Groups**.
2. For the dynamic group that you want to delete, select the Actions menu (⋮), and select **Delete**.
3. Confirm the deletion.

Using the CLI

Use the oci iam dynamic-group delete command and required parameters to delete a dynamic group.

```
SYNTX [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteDynamicGroup operation to delete a dynamic group.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing Policies

On Private Cloud Appliance, a policy is a named set of one or more policy statements. Policy statements grant permissions to users to access resources.

When designing access policies, remember the following policy characteristics:

- The policy will apply to the compartment where you attach the policy and to all subcompartments of that compartment. Permissions granted in a particular compartment, including the tenancy, are inherited by all subcompartments of that compartment.
- A user can be a member of more than one group. A group can be the subject of more than one policy. A policy can have up to 50 policy statements.
- If some users need full access to the named resources and other users only need to use the resources, you need to create multiple groups and multiple policies. A tenancy can have up to 100 policies.
- Users who have full access to resources in a subcompartment probably also need view or use access to related resources in that compartment and in parent compartments. For example, users who have access to create instances in a compartment might also need access to use tag namespaces to apply defined tags to the instances, or access to read images in a different compartment.

Policy Basics

To govern control of your resources, you need at least one policy. Each policy consists of one or more policy statements that follow this basic syntax:

```
Allow group <group_name> to <verb> <resource-type> in
 compartment <compartment_name>
```

Policy statements begin with the word `Allow`. Policies only allow access; they cannot deny it. Instead, all access is implicitly denied, meaning users can only do what they have been granted permission for. A tenancy administrator defines the groups and compartments; the available resource types are determined by Oracle.

If you want a policy to apply to the tenancy and not a compartment inside the tenancy, change the end of the policy statement as follows:

```
Allow group <group_name> to <verb> <resource-type> in tenancy
```

A basic feature of policies is the concept of *inheritance*: compartments inherit any policies from their parent compartment. If a group has a particular level of access to certain resource types in a compartment, then those same permissions apply in all subcompartments of the compartment where this policy is applied. The simplest example is the Administrators group in the tenancy: the built-in policy allows the administrators to manage all the resources in the tenancy root compartment. Because of policy inheritance, the administrators have full access to all operations and all resources in every compartment.

Resource Types

The resource types that you can use in policies are either *individual* or *family* types. The family resource types make policy writing easier, as they include multiple individual resource types that are often managed together. For example, the `virtual-network-family` type brings together a variety of types related to the management of VCNs: `vcns`, `subnets`, `route-tables`, `security-lists`, and so on. If you need to write a more granular policy, use an individual resource type to give access to only those specific resources. Note that there are other ways to make policies more granular, such as the ability to specify conditions under which the access is granted.

With future service updates, it is possible that resource type definitions are changed or added. These are typically reflected automatically in the resource family type for that service, so your policies remain current.

Some operations require access to multiple resource types. For example, launching an instance requires the permission to create instances and to work with a cloud network. Or creating a volume backup requires access to both the volume and the volume backup. That means you have separate statements to give access to each resource type.

These individual statements do not have to be in the same policy. A user can gain the required access from being in different groups. For example, a user could be in one group that gives the required level of access to the `volumes` resource type, and in another group that gives the required access to the `volume-backups` resource type. The sum of the individual statements, regardless of their location in the overall set of policies, allows the user to create a volume backup.

Policy Attachment

Another basic feature of policies is the concept of *attachment*. When you create a policy you must attach it to a compartment; or to the tenancy, which is the root compartment. Where you attach it controls who can then modify it or delete it. If you attach a policy to the tenancy, it can only be modified by the Administrators group, and not by users with only access to a subcompartment.

If you instead attach the policy to a child compartment, then anyone with access to manage the policies *in that compartment* can change or delete it. In practical terms, you can give compartment administrators – a group with access to manage all resources in the compartment – access to manage their own compartment's policies, without giving them broader access to manage policies that reside in the tenancy.

To attach a policy to a compartment, you must be in that compartment when you create the policy. As part of a policy statement you specify the compartment it applies to, so if you try to attach the policy to a different compartment you get an error. Policy attachment occurs at the time of creation, which means a policy can be attached to one compartment only.

Policy Statements

On Private Cloud Appliance, a policy can have up to 50 policy statements. A tenancy can have up to 100 policies. Decide what you want your policy to permit, and use the information in this section to write the necessary statements.

Ensure that the groups and compartments that you plan to name in the policy statements exist. Note the name or OCID of each group and compartment that you want to use.

> **Note:**
>
> If you use names in your policy statements instead of OCIDs, the policy will still be valid if the name of the group or compartment is subsequently changed. Internally, the OCID, not the name, is used. However, the policy could be more difficult for administrators to understand if the name of the group or compartment changes.

If you plan to use a tag to apply the policy to more than one group or more than one compartment, ensure that the tag exists. Note the name of the tag namespace, the name of the key, and the value that you want to use in the policy statement.

Policy Statement Syntax

The overall syntax of a policy statement is as follows:

```
Allow <subject> to <verb> <resource-type> in <location> where <conditions>
```

Additional spaces or line breaks in the statement have no effect.

Subject

Specify a group by name or OCID. You can specify multiple groups separated by commas. To cover all users in the tenancy, specify `any-user`.

These examples show how you can specify the subject in a policy statement.

- To specify a single group by name:

  ```
  Allow group A-admins to manage all-resources in compartment Project-A
  ```

- To specify multiple groups by name (a space after the comma is optional):

  ```
  Allow group A-admins, B-admins to manage all-resources in compartment
   Projects
  ```

- To specify a single group by OCID (the OCID is shortened for brevity):

  ```
  Allow group id ocid1.group.....<group1_unique_id>
  to manage all-resources in compartment Project-A
  ```

- To specify multiple groups by OCID (the OCIDs are shortened for brevity):

  ```
  Allow
  group id ocid1.group.....<group1_unique_id>,
  group id ocid1.group.....<group2_unique_id>
  to manage all-resources in compartment Projects
  ```

- To specify any user in the tenancy:

  ```
  Allow any-user to inspect users in tenancy
  ```

Verb

Specify a single verb.

```
Allow group A-admins to manage all resources in compartment Project-A
```

The policy syntax supports the following verbs, ordered by increasing permissions:

Verb	Type of Access	Target User
`inspect`	Ability to list resources, without access to any confidential information or user-specified metadata that might be part of that resource. **Notes:** • The operation to list policies includes the contents of the policies themselves. • The list operations for the Networking resource types return all the information, including the contents of security lists and route tables.	Third-party auditors
`read`	Includes `inspect` plus the ability to get user-specified metadata and the actual resource itself.	Internal auditors
`use`	Includes `read` plus the ability to work with existing resources. The actions vary by resource type. Includes the ability to update the resource, except for resource types where the "update" operation has the same effective impact as the "create" operation; for example `UpdatePolicy`, `UpdateSecurityList`, etc. In those cases the "update" ability is available only with the `manage` verb. In general, the verb `use` does not include the ability to create or delete that type of resource.	Day-to-day end users of resources
`manage`	Includes all permissions for the resource.	Administrators

The verb gives a certain general type of access. For example, `inspect` lets you list and get resources. You then join that type of access with a particular resource type in a policy. For example, allow group XYZ to `inspect compartments` in the tenancy. As a result, that group gains access to a specific set of permissions and API operations; for example `ListCompartments`, `GetCompartment`.

Resource Type

Specify a single resource-type, which can be:

- An individual resource type; for example: `vcns`, `subnets`, `instances`, `volumes`, etc.
- A family resource type; for example: `virtual-network-family`, `instance-family`, `volume-family`, etc.

  A family resource type covers a variety of individual resource types that are typically used together.
- `all-resources`: Covers all resources in the compartment or tenancy.

These examples show how you can specify the resource type in a policy statement.

- To specify a single resource type:

```
Allow group HelpDesk to manage users in tenancy
```

- To specify multiple resource types, use separate statements:

```
Allow group A-users to manage instance-family in compartment Project-A
Allow group A-users to manage volume-family in compartment Project-A
```

- To specify all resources in the compartment or tenancy:

```
Allow group A-admins to manage all-resources in compartment Project-A
```

Here is an overview of the family resource types can be used in policy statements:

Family Resource Type	Description
`compute-management-family`	This aggregate resource covers the following individual resource types: `instance-configurations`, `instance-pools`, `cluster-networks`.
`instance-family`	This aggregate resource covers the following individual resource types: `app-catalog-listing`, `console-histories`, `instances`, `instance-console-connection`, `instance-images`, `volume-attachments`.
`volume-family`	This aggregate resource covers all individual resource types related to block volumes: `volumes`, `volume-attachments`, `volume-backups`, `boot-volume-backups`, `backup-policies`, `backup-policy-assignments`, `volume-groups`, `volume-group-backups`.
`virtual-network-family`	This aggregate resource covers all individual resource types related to the networking service. For example: VCNs, subnets, route tables, gateways, VNICs, network security groups, and so on.
`file-family`	This aggregate resource covers all individual resource types related to the file storage service: `file-systems`, `mount-targets`, `export-sets`.
`object-family`	This aggregate resource covers all individual resource types related to the object storage service: `objectstorage-namespaces`, `buckets`, `objects`.
`compute-container-family`	This aggregate resource covers all individual resource types related to the container instances service: `compute-container-instances` and `compute-containers`.

Location

Specify a single compartment by name or OCID. Or simply specify `tenancy` to cover the entire tenancy. Remember that users, groups, and compartments reside in the tenancy. Policies can be attached to either the tenancy or a child compartment.

The location is required in the statement. To attach a policy to a compartment, you must be in that compartment when you create the policy.

These examples show how you can specify the location in a policy statement.

- To specify a compartment by name:

```
Allow group A-admins to manage all-resources in compartment Project-A
```

- To specify a compartment by OCID:

```
Allow group A-admins to manage all-resources
in compartment id ocid1.compartment.oc1..aaaaaaaaexampleocid
```

- To specify multiple compartments, use separate statements:

```
Allow group InstanceAdmins to manage instance-family in compartment
 Project-A
```

```
Allow group InstanceAdmins to manage instance-family in compartment
 Project-B
```

```
Allow group InstanceAdmins to manage instance-family
in compartment id ocid1.compartment.oc1..aaaaaaaayzexampleocid
Allow group InstanceAdmins to manage instance-family
in compartment id ocid1.compartment.oc1..abcabcabcexampledocid
```

Conditions

Specify one or more conditions. With multiple conditions, use `any` or `all` for a logical OR or AND, respectively.

These are the types of values you can use in conditions:

Value Type	Examples

String	Single quotation marks are required around the value.
	`'johnsmith@example.com'`
	`'ocid1.compartment.oc1..aaaaaaaaph...ctehnqg756a'`

Pattern	`/HR*/` - matches strings that start with "HR" `/*HR/` - matches strings that end with "HR" `/*HR*/` - matches strings that contain "HR"

These examples show how you can specify conditions in a policy statement.

> **Note:**
>
> In the example statements, the condition to match group names makes it impossible for GroupAdmins to list all users and groups. The list operation does not involve specifying a group, which means there is no value to match the condition variable `target.group.name`. To resolve this, a statement including the `inspect` verb is added.

- The following policy enables the GroupAdmins group to create, update, or delete any groups with names that start with "A-Users-":

```
Allow group GroupAdmins to manage groups in tenancy
 where target.group.name = /A-Users-*/
Allow group GroupAdmins to inspect groups in tenancy
```

- The following policy enables the NetworkAdmins group to manage cloud networks in any compartment except the one specified:

```
Allow group NetworkAdmins to manage virtual-network-family in tenancy
where target.compartment.id !=
 'ocid1.compartment.oc1..aaaaaaaaexampleocid'
```

- The following policy uses multiple conditions and lets GroupAdmins create, update, or delete any groups whose names start with "A-", except for the A-Admins group itself:

```
Allow group GroupAdmins to manage groups in tenancy
where all {target.group.name=/A-*/,target.group.name!='A-Admins'}
Allow group GroupAdmins to inspect groups in tenancy
```

Conditions That Aren't Applicable

On Private Cloud Appliance, if a condition isn't applicable to the rest of the policy statement, then that condition evaluates to false and access isn't granted.

A condition isn't applicable if it's testing information that is not available in the request. For example, the following policy statement grants `use` access to the resource `users`, but does not allow the requesting users to list or update users, even though those permissions are included in the `use` permission:

```
allow group GroupAdmins to use users in tenancy where target.group.name !=
 'Administrators'
```

The request to list users or update a user doesn't include information about groups. The list users and update user requests have no value for `target.group.name`. The test fails, and the request to list or update a user is denied.

To fix this example, you could remove the `where` clause and allow only `inspect` or `read` access.

Using Defined Tags in Conditions

On Private Cloud Appliance, certain conditions evaluate the value of a defined tag that has been applied to a user, compartment, or resource. In these conditions, the predefined variable can be called a tag variable.

Using conditions with tag variables enables you to do the following:

- Write a single policy statement that applies to multiple user groups, compartments, or resources.
- Change the permissions that are granted without changing the policy statement. Instead, to allow or revoke access, apply tags to different resources or remove tags from resources.

See Tagging Resources (IAM on the Appliance) on page 446 for information about how to create and apply defined tags.

The general syntax of a condition that uses tag variables is the same as the syntax of a condition that uses other condition variables:

`variable op 'value'`

The value of each of these three parts is specialized for tags.

`variable`

Tag condition variables include the name of the tag namespace and the name of the key in the variable name:

```
base_variable_name.tag_namespace_name.tag_key_name
```

`op`

One of `=`, `!=`, `in`, or `not in`.

The `in` and `not in` operations refer to members of the set of possible values for the tag.

`value`

The *`value`* is a value of the defined tag. The value can be a single value or a list of values.

The following tag variables are supported:

`request.principal.group.tag`

This variable potentially grants access to multiple groups in one statement. The following statement allows any user that is a member of a group that has been tagged with tag Operations>Project>ABC to manage instance resources in compartment ProdX:

```
allow any-user to manage instance-family in compartment ProdX
where request.principal.group.tag.Operations.Project='ABC'
```

If you replace 'ABC' in the preceding statement with '*' or /*/, a user that is a member of a group that has been tagged with any value of Operations>Project could manage instance resources in compartment ProdX.

`target.resource.compartment.tag`

This variable potentially grants access to multiple compartments in one statement. The following statement allows users in group NetAdmins to use network resources in any compartment that has been tagged with either tag Operations>Project>ABC or tag Operations>Personnel>Test:

```
allow group NetAdmins to use virtual-network-family in tenancy where
any { target.resource.compartment.tag.Operations.Project='ABC',
target.resource.compartment.tag.Operations.Personnel='Test' }
```

If you replace `any` with `all` in the preceding statement, the statement allows users in group NetAdmins to use network resources in any compartment that has been tagged with both tag Operations>Project>ABC and tag Operations>Personnel>Test.

The following statement allows users in group NetAdmins to use network resources in any compartment that has been tagged with either tag Operations>Personnel>Development or tag Operations>Personnel>Test:

```
allow group NetAdmins to use virtual-network-family in tenancy where
target.resource.compartment.tag.Operations.Personnel in ('Development',
 'Test')
```

`target.resource.tag`

This variable grants access to one or more resources of the specified type. The following statement allows group Xadmins to use any instance in compartment ProdX that is tagged with tag Operations>Project>XYZ.

```
allow group Xadmins to use instances in compartment ProdX
where target.resource.tag.Operations.Project = 'XYZ'
```

Writing Policies to Access Resources Across Tenancies

On Private Cloud Appliance, you can write policies to allow tenancy access from other tenancies so you can share resources across tenancies.

The administrators of both tenancies need to create special policy statements that explicitly state which resources can be accessed and shared. These special statements use the following special verbs:

- **Endorse**: This policy statement describes what work a group in a *source tenancy* can perform in other tenancies. You write the `endorse` statement for the tenancy that contains the group of users who need to work with another tenancy's resources.
- **Admit**: This policy statement describes what work a group from other tenancies can perform in a *destination tenancy*. You write the `admit` statement for the tenancy that is granting permission to access its resources. The `admit` statement identifies the group of users from the source tenancy that requires resource access in the destination tenancy.
- **Define**: This policy statement is used to assign an alias for a *source tenancy* OCID, a *source group* OCID, and a *destination tenancy* OCID. You define a *source tenancy* alias and a *source group* alias for use in `admit` policy statements. You define a *destination tenancy* alias for use in `endorse` policy statements.

  You must include a `define` statement in the same policy entity as the `endorse` or `admit` statement.

The `endorse` and `admit` statements work together. An `endorse` statement resides in the source tenancy while an `admit` statement resides in the destination tenancy. Without a corresponding statement that specifies access, a particular `endorse` or `admit` statement grants no access. *Both tenancies must agree on access and have policies that allow for access.*

In the source tenancy, you write `define` and `endorse` policy statements using the following syntax:

```
define tenancy destination-tenancy-alias as tenancy_ocid
```

```
endorse group group-name to verb resource in tenancy destination-tenancy-
alias
```

In the destination tenancy, you write two `define` policy statements and an `admit` policy statement using the following syntax:

```
define tenancy source-tenancy-alias as tenancy_ocid
```

```
define group source-group-alias as group_ocid
```

```
admit group source-group-alias of tenancy source-tenancy-alias
 to verb resource in compartment/tenancy
```

Source Tenancy Policy Statements

On Private Cloud Appliance, an administrator for the source tenancy creates policy statements that endorses a group in their tenancy to manage resources in a destination tenancy.

You can write broad policy statements for the source tenancy as in the following examples.

- To endorse a specific group, such as `StorageAdmins`, to do anything with all object storage resources in any tenancy:

  ```
  endorse group StorageAdmins to manage object-family in any-tenancy
  ```

- To endorse a specific group, such as `DNSAdmins`, to do anything with all DNS resources in any tenancy:

  ```
  endorse group DNSAdmins to manage dns in any-tenancy
  ```

You can write a policies that reduce the scope of access for a source tenancy group. For this type of policy, the administrator for the source tenancy must define an alias for the destination tenancy OCID and reference that alias in the policy's statements, for example:

- To endorse a specific group, such as `StorageAdmins`, to manage object storage resources in a specific tenancy, such as `DestinationTenancy`, you would use the following policy statements:

  ```
  define tenancy DestinationTenancy as ocid1.tenancy.oc1..<unique_ID>
  endorse group StorageAdmins to manage object-family in tenancy
   DestinationTenancy
  ```

- To endorse a specific group, such as `DNSAdmins`, to manage DNS resources in a specific tenancy, such as `DestinationTenancy`, you would use the following policy statements:

  ```
  define tenancy DestinationTenancy as ocid1.tenancy.oc1..<unique_ID>
  endorse group DNSAdmins to manage dns in tenancy DestinationTenancy
  ```

The destination tenancy must also have a policy that allows the source tenancy group access to the destination tenancy and its resources. Without this corresponding policy in the destination tenancy, the source tenancy group can't access the destination tenancy or its resources.

Destination Tenancy Policy Statements

On Private Cloud Appliance, an administrator for the destination tenancy creates policy statements that allows a group in a source tenancy access to the destination tenancy and its resources.

You can write broad policy statements for the destination tenancy as in the following examples.

- To allow a specific group in a source tenancy, such as `StorageAdmins`, to do anything with all object storage resources in the destination tenancy:

```
define tenancy SourceTenancy as ocid1.tenancy.oc1..<unique_ID>
define group StorageAdmins as ocid1.group.oc1..<unique_ID>
admit group StorageAdmins of tenancy SourceTenancy to manage object-family
 in tenancy
```

- To allow a specific group in a source tenancy, such as `DNSAdmins`, to do anything with all DNS resources in the destination tenancy:

```
define tenancy SourceTenancy as ocid1.tenancy.oc1..<unique_ID>
define group DNSAdmins as ocid1.group.oc1..<unique_ID>
admit group DNSAdmins of tenancy SourceTenancy to manage dns in tenancy
```

You can write policies that reduces the scope of access for a source tenancy group to destination tenancy resources. For these types of policies, an administrator for the destination tenancy must define aliases for the source tenancy OCID and source group OCID, and then reference those aliases in the policy's statements, for example:

- To allow a specific group in the source tenancy, such as `StorageAdmins`, to manage object storage resources in a specific compartment, such as `SharedBuckets`, you would use the following policy statements:

```
define tenancy SourceTenancy as ocid1.tenancy.oc1..<unique_ID>
define group StorageAdmins as ocid1.group.oc1..<unique_ID>
admit group StorageAdmins of tenancy SourceTenancy to manage object-family
 in compartment SharedBuckets
```

- To allow a specific group in the source tenancy, such as `DNSAdmins`, to manage DNS resources in a specific compartment, such as `SharedZomes`, you would use the following policy statements:

```
define tenancy SourceTenancy as ocid1.tenancy.oc1..<unique_ID>
define group DNSAdmins as ocid1.group.oc1..<unique_ID>
admit group DNSAdmins of tenancy SourceTenancy to manage dns in
 compartment SharedZones
```

The source tenancy must also have a policy that endorses the source tenancy group's access to the destination tenancy and its resources. Without this corresponding policy in the source tenancy, the source tenancy group would not be able to access the destination tenancy or its resources.

Creating a Policy

On Private Cloud Appliance, a policy must have at least one policy statement. You can't create an empty policy and add statements later.

Decide what you want your policy to allow, and see Policy Statements on page 436 to design the necessary statements.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Policies**.
2. Select **Create Policy**.
3. Enter the following information:
   - **Name**: The policy name. Policy names have the following characteristics:
     - Must be unique within the tenancy.
     - Are case insensitive.
     - Can't be changed later.
     - Can be no more than 100 characters.
     - Can't include spaces. Can include only letters, numbers, hyphens, periods, or underscores.
     - Avoid entering confidential information.

- **Description**: A description for the policy. This description can be no more than 400 characters.
- **Create in Compartment**: Select the compartment where you want to attach this policy. The policy will apply to this compartment and all child compartments of this compartment.
- **Statements**: Enter a policy statement. For information about how to write policy statements, see Policy Statements on page 436.

  To add a second policy statement, select the +Another Statement button. You can enter up to 50 statements. If you create more than one policy statement, you can select the X button next to a statement to delete that statement.
- **Tagging**: (Optional) Add defined or free-form tags for this policy as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.

4. Select **Create Policy**.

   The details page for the new policy is displayed. The Resources section of the page shows the policy statements.

Using the CLI

Use the oci iam policy create command and required parameters to create a policy.

```
oci iam policy create -c compartment_OCID --name text --description "text" \
{ --statements '["statement","statement"]' | --statements
 file://policy.json } [OPTIONS]
```

The *compartment_OCID* is the compartment where you want to attach this policy.

This command returns the same output as the `policy get` command.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreatePolicy operation to create a policy.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Updating a Policy

On Private Cloud Appliance, you can modify the description or tags.

Using the Compute Web UI

To Modify the Policy Description or Tags

1. In the Compute Web UI navigation menu, select **Identity**, then select **Policies**.
2. If the policy that you want to modify isn't listed, select the correct compartment from the Compartment drop-down menu above the policies list.
3. For the policy that you want to modify, select the Actions menu (⋮) for the policy, and select **Edit**.
4. Update the description or tags.

   To change tags, see Adding Tags at Resource Creation on page 458.
5. Select **Save Changes**.

To Modify Policy Statements

1. In the Compute Web UI navigation menu, select **Identity**, then select **Policies**.
2. If the policy that you want to modify isn't listed, select the correct compartment from the Compartment drop-down menu above the policies list.
3. Select the name of the policy that you want to modify.
4. Under **Resources**, in the **Statements** list, select **Configure Policy Statements**.

5. In **Edit Statements** in the *policy_name* **Policy** dialog box, change or add policy statements:
   - To change policy statements, see Policy Statements on page 436.
   - To add a policy statement, select **Another Statement**. You can enter up to 50 statements.
   - If more than one policy statement exists, you can select **X** next to a statement to delete that statement.
6. Select **Submit**.

Using the CLI

Use the oci iam policy update command and required parameters to update a policy.

```
oci iam policy update --policy-id policy_OCID [ --description desc ] \
[ --defined-tags tags ] [ --freeform-tags tags ] \
[ --statements policy_statements --version-date "" ] [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdatePolicy operation to update a policy.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Policy

On Private Cloud Appliance, you can delete policies.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Identity**, then select **Policies**.
2. If the policy that you want to delete isn't listed, select the correct compartment from the Compartment drop-down menu above the policies list.
3. For the policy that you want to delete, select the Actions menu (⋮), and select **Delete**.
4. In the confirmation dialog box, select **Delete**.

Using the CLI

Use the oci iam policy delete command and required parameters to delete a policy.

```
SYNTX [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeletePolicy operation to delete a policy.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Tagging Resources (IAM on the Appliance)

Private Cloud Appliance tagging enables you to add metadata to resources by applying key/value pairs called defined tags or free-form tags. You can create tag defaults on compartments, which are tags that are automatically applied to all newly created resources in the tagged compartment.

Uses for tags include the following. For more information about using tags, see Using Resource Tags (IAM on the Appliance) on page 458.

- Applying policies to resources. For example, if you apply access policies to resources, you can change ownership of a resource to a different product group by changing a tag value on the resource rather than changing the resource access policy directly.
- Filtering resource lists.
- Applying parameter values to some resources.

Private Cloud Appliance supports two types of tags:

- Free-form tags enable any user who has authorization to update a resource to quickly and conveniently include relevant information in the metadata of the resource.
  - Before applying a free-form tag, you cannot see a list of free-form tags that are already in use in your tenancy. You can use the resource get operation to view only the free-form tags that are applied to a specified resource.
  - Free-form tags provide unmanaged metadata. You cannot enforce use of particular keys or values for free-form tags.
  - Free-form tags cannot be used in policy statements.
- Defined tags enable you to enforce use of particular keys and values. Related tags can be grouped into a common tag namespace.
  - Tag namespaces with tag key definitions must exist in the tenancy before users can apply a defined tag to a resource. You can see them by listing tag namespaces.
  - You can allow users to enter a value, or you can require that users select from a list of values.
  - You can use a variable to automatically provide the tag value.
  - Defined tags can be used in policy statements.

Creating and Managing Tag Namespaces (IAM on the Appliance)

Private Cloud Appliance tag namespaces enable you to create collections of related tags by creating tag key definitions within a tag namespace.

Tag namespaces with tag key definitions must exist in the tenancy before users can apply a defined tag to a resource.

Creating a Tag Namespace

Learn how to create a tag namespace on Private Cloud Appliance to group related tags.

After you create a tag namespace in any compartment, that tag namespace is available in every compartment in the tenancy.

A tenancy can have at most 100 tag namespaces.

Tag namespace names have the following characteristics:

- Must be unique within the tenancy.
- Are case insensitive.
- Cannot be changed later.
- Can be no more than 100 characters.
- Cannot contain period (.) or space characters.

The tag namespace description can be no more than 256 characters.

Using the Compute Web UI

1. In the navigation menu, select Governance, and then select Tag Namespaces.
2. Select the Create Namespace Definitions button.
3. In the Create Namespace Definition dialog box, scroll down to the Tagging section and select the Tag Namespace field to verify that the namespace definition name that you plan to use does not already exist in this tenancy.

   If the namespace is listed, then it has already been created in a different compartment and is available to all compartments in the tenancy. You will receive an error message if you try to create this tag namespace when it is already available.
4. In the Create Namespace Definition dialog box, enter the following information:

- *Compartment*: The compartment in which you want to create the namespace definition.
- *Namespace Definition Name*: A name for this tag namespace. See the preceding list of characteristics.
- *Description*: A description for this set of tags.
- *Tagging*: (Optional) Add defined or free-form tags for this tag namespace as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.

5. Select the Create Namespace Definition button in the dialog box.

   The details page for the new tag namespace definition is displayed.

Using the CLI

Use the oci iam tag-namespace create command and required parameters to create a new tag namespace.

Use the following command to verify that the namespace name you plan to use is unique in the tenancy. If the namespace is listed, then it has already been created in a different compartment and is available to all compartments in the tenancy. You will receive an error message if you try to create this tag namespace when it is already available.

```
oci iam tag-namespace list --compartment-id tenancy_OCID
```

Example:

```
oci iam tag-namespace create --compartment-id compartment_OCID --
name tag_namespace_name --description "text"
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateTagNamespace operation to create a new tag namespace.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Updating a Tag Namespace

Learn how to change the description of a tag namespace on Private Cloud Appliance.

You can also and add, change, or delete tags that are on the tag namespace resource.

Using the Compute Web UI

1. In the navigation menu, select Governance, and then select Tag Namespaces.

   If the tag namespace that you want to update is not listed, use the Compartment drop-down menu above the tag namespaces list to select the correct compartment.
2. For the namespace that you want to update, select the Actions menu, and select the Edit option.
3. Update the tag namespace.

   In the Edit dialog box, you can change the description of the tag namespace and add or change the tags on the tag namespace resource.
4. Select the Update Tag Namespace button in the dialog box.

Using the CLI

Use the oci iam tag-namespace update command and required parameters to change the description of a tag namespace.

Example:

```
oci iam tag-namespace update --tag-namespace-id tag_namespace_OCID --
description "text"
```

To add or modify a tag on the tag namespace, add the options described in Applying Tags to an Existing Resource on page 459.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateTagNamespace operation to change the description of the tag namespace or change the tags on the tag namespace resource.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Retiring a Tag Namespace

Learn how to retire a tag namespace on Private Cloud Appliance.

When you retire a tag namespace, all tag key definitions in that tag namespace are retired, and you cannot create new tag key definitions in that tag namespace. Retired tags cannot be applied to resources. However, retired tags remain applied to any resources where they were already applied and can still be used in operations such as listing, sorting, and filtering.

Using the Compute Web UI

1. In the navigation menu, select Governance, and then select Tag Namespaces.

   If the tag namespace that you want to retire is not listed, use the Compartment drop-down menu above the tag namespaces list to select the correct compartment.
2. For the tag namespace that you want to retire, select the Actions menu, and select the Retire option.
3. At the Retire Tag Namespace confirmation prompt, select Confirm.

   The state of the tag namespace changes to Inactive. On the details page for the tag namespace, the tag key definitions are also in state Inactive.

Using the CLI

Use the oci iam tag-namespace retire command and required parameters to retire a tag namespace.

Example:

```
oci iam tag-namespace retire --tag-namespace-id tag_namespace_OCID
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

On Private Cloud Appliance, retiring a tag namespace by using the API is not supported.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Reactivating a Tag Namespace

Learn how to reactivate a retired tag namespace on Private Cloud Appliance.

You can reactivate a tag namespace that is retired. When you reactivate a tag namespace, you can create new tag key definitions in that tag namespace.

When you reactivate a retired tag namespace, the tag key definitions and tags are not reactivated. To use tag key definitions that were retired with the namespace, you must explicitly reactivate each tag key definition. See Reactivating a Tag Key Definition on page 454.

Using the Compute Web UI

1. In the navigation menu, select Governance, and then select Tag Namespaces.

   If the tag namespace that you want to reactivate is not listed, use the Compartment drop-down menu above the tag namespaces list to select the correct compartment.
2. For the tag namespace that you want to reactivate, select the Actions menu, and select the Reactivate option.
3. At the Retire Tag Namespace confirmation prompt, select Confirm.

The state of the tag namespace changes to Active. On the details page for the tag namespace, the tag key definitions are still shown as Inactive.

Using the CLI

Use the oci iam tag-namespace reactivate command and required parameters to reactivate a tag namespace.

Example:

```
oci iam tag-namespace reactivate --tag-namespace-id tag_namespace_OCID
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

On Private Cloud Appliance, reactivating a tag namespace by using the API is not supported.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Moving a Tag Namespace

Learn how to move a tag namespace to a different compartment on Private Cloud Appliance.

You can move an active or retired tag namespace and its tag key definitions to a different compartment within the same tenancy.

To move a tag namespace, you must be granted `manage tag-namespaces` access in both compartments.

Using the Compute Web UI

On Private Cloud Appliance, moving a tag namespace by using the Compute Web UI is not supported.

Using the CLI

Use the oci iam tag-namespace change-compartment command and required parameters to move a tag namespace to a different compartment.

Example:

```
oci iam tag-namespace change-compartment --compartment-
id destination_compartment_OCID --tag-namespace-id tag_namespace_OCID
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ChangeTagNamespaceCompartment operation to move a tag namespace to a different compartment.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Tag Namespace

Learn how to delete a tag namespace on Private Cloud Appliance.

A tag namespace must be retired before it can be deleted. See Retiring a Tag Namespace on page 449.

To delete a tag namespace, all tag key definitions in that namespace must be deleted. See Deleting a Tag Key Definition on page 455.

Using the Compute Web UI

1. In the navigation menu, select Governance, and then select Tag Namespaces.

   If the tag namespace that you want to delete is not listed, use the Compartment drop-down menu above the tag namespaces list to select the correct compartment.
2. Ensure that the tag namespace that you want to delete is retired (in state Inactive).

3. Select the name of the tag namespace that you want to delete.
4. On the tag namespace details page, scroll down to the Resources section and select Tag Key Definitions.

   Ensure that all tag key definitions are deleted.
5. On the Controls menu at the top of the details page, select the Delete option.
6. At the Delete Tag Namespace confirmation prompt, select Confirm.

   The tag namespace is removed from the Tag Namespace list, and tags in that tag namespace are removed from resources.

Using the CLI

Use the oci iam tag-namespace delete command and required parameters to delete a tag namespace.

First ensure that all tag key definitions for this tag namespace are deleted.

```
oci iam tag list --tag-namespace-id tag_namespace_OCID
```

Example:

```
oci iam tag-namespace delete --tag-namespace-id tag_namespace_OCID
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteTagNamespace operation to delete a tag namespace.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Creating and Managing Tag Key Definitions (IAM on the Appliance)

Private Cloud Appliance defined tags enable users to specify predefined metadata for resources. Predefined keys and values help to group and act on resources.

A tag namespace contains tag key definitions. Instances of tag key definitions that are applied to resources are called defined tags.

A tag key definition includes a tag key name and tag value type. A tag key definition might include a value, depending on the tag value type. The tag value type specifies whether the tag user enters a value or selects a predefined value when applying a tag to a resource. A value can be set automatically by specifying a variable.

Creating a Tag Key Definition

Learn how to create a tag key definition on Private Cloud Appliance.

Create a tag key definition within a tag namespace.

A tag namespace can have at most 100 tag key definitions.

Tag key names have the following characteristics:

- Must be unique within the tenancy.

  The same tag key name can be used in different tag namespaces.
- Are case insensitive.
- Cannot be changed later.
- Can be no more than 100 characters.
- Cannot contain period (.) or space characters.

The tag key description can be no more than 256 characters.

Using the Compute Web UI

1. In the navigation menu, select Governance, and then select Tag Namespaces.

If the tag namespace where you want to add the tag key definition is not listed, use the Compartment drop-down menu above the tag namespaces list to select the correct compartment.

2. Select the name of the tag namespace where you want to add a tag key definition.
3. On the tag namespace details page, scroll down to the Resources section and select Tag Key Definitions.
4. Select the Create Tag Key Definition button.
5. In the Create Tag Key Definition dialog box, enter the following information:
   - *Name*: A name for this tag key. See the preceding list of characteristics.
   - *Description*: A description for this tag key definition.
6. Select the Tag Value Type.
   - *Static Value*: Not populated. The user must enter a value when the tag is applied to a resource. This is the default selection.
   - *A List of Values*: A predefined list of values. Enter at least one value. The user must select one of these predefined values to apply to a resource.

     Separate multiple values with new lines. Duplicate values and blank lines are invalid. Values are case sensitive and can be no more than 256 characters.

     A value can include one of the following variables:

     `${iam.principal.name}`

     The name of the user that tagged the resource.

     `${iam.principal.type}`

     The type of principal that tagged the resource. One of: root user, IAM user, or Instance principal.

     `${oci.datetime}`

     The date and time that the tag was created.
7. Select the Create Tag Key Definition button in the dialog box.

   The details page for the new tag key definition is displayed.

Using the CLI

Use the oci iam tag create command and required parameters to create a new tag key definition.

Example for a static value (the default):

```
oci iam tag create --tag-namespace-id tag_namespace_OCID --name text --
description "text"
```

The following command creates a tag key definition with a list of values from which the user must select when applying the tag.

```
oci iam tag create --tag-namespace-id tag_namespace_OCID --name text --
description "text" --validator values
```

The value of the `--validator` option argument is a JSON definition of the tag values. This JSON definition can be provided as a string on the command line or in a file.

The following command displays a template of the correct JSON to provide:

```
oci iam tag create --generate-param-json-input validator
```

Example parameter for a list of values:

```
--validator '{"validator-type": "ENUM", "values":
 ["typeA","typeB","typeC"]}'
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateTag operation to create a new tag key definition.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Updating a Tag Key Definition

Learn how to update a tag key definition on Private Cloud Appliance.

You can update the description of a tag key definition and change the tag value type and value.

You cannot update a tag key definition that is retired.

If the value of the tag key definition that you are updating is a predefined list, you cannot remove or change any value that is the value of a tag default. To remove or change a tag key definition value that is the value of a tag default, first update the tag default to use a different value. See Configuring Tag Defaults (IAM on the Appliance) on page 455.

Using the Compute Web UI

1. In the navigation menu, select Governance, and then select Tag Namespaces.

   If the tag namespace where you want to update a tag key definition is not listed, use the Compartment drop-down menu above the tag namespaces list to select the correct compartment.
2. Select the name of the tag namespace where you want to update a tag key definition.
3. On the tag namespace details page, scroll down to the Resources section and select Tag Key Definitions.
4. For the tag key definition that you want to update, select the Actions menu and select the Edit option.
5. In the Edit Tag Key Definition dialog box, you can modify the description or change the tag value type. If you select A List of Values for the type, you must add at least one value in the Value box. Separate multiple values with new lines. Duplicate values and blank lines are invalid. Values are case sensitive and can be no more than 256 characters.
6. Select the Save Changes button in the dialog box.

Using the CLI

Use the oci iam tag update command and required parameters to update a tag key definition.

Example:

```
oci iam tag update --tag-namespace-id tag_namespace_OCID --tag-name text --
description "text" --validator file://values.json
```

To update the value type to allow the user to enter a value rather than select from a predefined list of values, specify an empty validator option: `--validator '{}'`

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateTag operation to update a tag key definition.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Retiring a Tag Key Definition

Learn how to retire a tag key definition on Private Cloud Appliance.

When you retire a tag key definition, you cannot apply tags that are based on this tag key definition to resources. Existing tag defaults that are based on this tag key definition will not be automatically applied to newly created resources. However, the tag is not removed from resources where it was already applied. The tag still exists as metadata on those resources and you can still use the retired tag in operations such as listing, sorting, and reporting.

Using the Compute Web UI

1. In the navigation menu, select Governance, and then select Tag Namespaces.

   If the tag namespace where you want to retire a tag key definition is not listed, use the Compartment drop-down menu above the tag namespaces list to select the correct compartment.
2. Select the name of the tag namespace where you want to retire a tag key definition.
3. On the tag namespace details page, scroll down to the Resources section and select Tag Key Definitions.
4. For the tag key definition that you want to retire, select the Actions menu and select the Retire option.
5. At the Retire Tag Key Definition confirmation prompt, select Confirm.

Using the CLI

Use the oci iam tag retire command and required parameters to retire a tag key definition.

Example:

```
oci iam tag retire --tag-name text --tag-namespace-id tag_namespace_OCID
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

On Private Cloud Appliance, retiring a tag key definition by using the API is not supported.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Reactivating a Tag Key Definition

Learn how to reactivate a tag key definition on Private Cloud Appliance.

When you reactivate a tag key definition, it is again available for you to apply to resources. You cannot reactivate a tag key definition if the parent tag namespace is retired.

Using the Compute Web UI

1. In the navigation menu, select Governance, and then select Tag Namespaces.

   If the tag namespace where you want to reactivate a tag key definition is not listed, use the Compartment drop-down menu above the tag namespaces list to select the correct compartment.
2. Select the name of the tag namespace where you want to reactivate a tag key definition.
3. On the tag namespace details page, scroll down to the Resources section and select Tag Key Definitions.
4. For the tag key definition that you want to reactivate, select the Actions menu and select the Reactivate option.
5. At the Reactivate Tag Key Definition confirmation prompt, select Confirm.

Using the CLI

Use the oci iam tag reactivate command and required parameters to reactivate a tag key definition.

Example:

```
oci iam tag reactivate --tag-name text --tag-namespace-id tag_namespace_OCID
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

On Private Cloud Appliance, reactivating a tag key definition by using the API is not supported.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Tag Key Definition

Learn how to delete a tag key definition on Private Cloud Appliance.

A tag key definition must be retired before it can be deleted. See Retiring a Tag Key Definition on page 453.

When you delete a tag key definition, tags that are based on this tag key definition are removed from all resources. Tag defaults that are based on this tag key definition are not removed from compartments.

You cannot restore a deleted tag key definition.

You can create a new tag key definition with the same name as the deleted tag key definition.

Using the Compute Web UI

1. In the navigation menu, select Governance, and then select Tag Namespaces.

   If the tag namespace where you want to delete a tag key definition is not listed, use the Compartment drop-down menu above the tag namespaces list to select the correct compartment.
2. Select the name of the tag namespace where you want to delete a tag key definition.
3. On the tag namespace details page, scroll down to the Resources section and select Tag Key Definitions.
4. Ensure that the tag key definition that you want to delete is retired (in state Inactive).
5. For the tag key definition that you want to delete, select the Actions menu and select the Delete option.
6. At the Delete Tag Key Definition prompt, select Confirm.

   The tag key definition status changes to Deleting, and all tags that are based on this tag key definition are removed from resources.

   When the tag removal process is finished, the tag key definition status changes to Deleted.

Using the CLI

Use the oci iam tag delete command and required parameters to delete a tag key definition.

Example:

```
oci iam tag delete --tag-name text --tag-namespace-id tag_namespace_OCID
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteTag operation to delete a tag key definition.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Configuring Tag Defaults (IAM on the Appliance)

On Private Cloud Appliance, a tag default is a defined tag that is automatically applied to resources that are created in the specified compartment.

Tag defaults have the following characteristics:

- The tag default is applied to all new resources that are created in that compartment, including in child compartments.
- The tag default is not applied to resources that already existed before the tag default was created.
- Tag defaults cannot be changed by creating or editing resources. With permission to use the tag namespace, you can change the value of the tag when you create or modify a resource. To change the tag default that will be applied to all new resources in the compartment, you must update the tag default on the compartment.
- If you change the default value of the tag default, existing occurrences of that tag default on resources are not updated.
- If you change a value of a tag key definition whose value is a defined list, or if you delete a tag key definition, existing occurrences of a tag default that is based on that tag key definition are not updated. Tag default values must be separately updated.

See Retiring a Tag Key Definition on page 453 and Deleting a Tag Key Definition on page 455 for information about the effect on tag defaults of retiring or deleting tag key definitions.

Creating a Tag Default

Learn how to set a tag default on a compartment on Private Cloud Appliance.

To create a tag default, specify a compartment, a tag key definition, and a value. If the value of the selected tag key definition is **Static Value**, then you can select **User-Defined Value** for the value of the tag default.

A compartment can have at most five tag defaults.

Using the Compute Web UI

1. In the navigation menu, select Identity, and then select Compartments.
2. If the compartment where you want to add a tag default is not listed, navigate to the correct compartment.

   Select the name of the top-level parent compartment. On the compartment details page, scroll to the Resources section and select Child Compartments. If necessary, select on the name of another compartment to view those child compartments.
3. Select the name of the compartment to which you want to add a tag default.
4. Scroll to the Resources section on the compartment details page and select Tag Defaults.

   Ensure that no more than four tag defaults already exist for this compartment.
5. Select the Create Tag Default button.
6. In the Tag Default dialog box, select the Tag Namespace and the Tag Key.
7. For Required Tag Value Options, select one of the following value types:
   - **Default Value**: Enter the value for this tag default. If the selected tag key has a defined list of values, then this Default Value must be a member of that list.
   - **User-Defined Value**: Users are required to enter the value when a resource is created. Selecting User-Defined Value is invalid if the selected tag key definition has a predefined list of values.
8. Select the Submit button in the dialog box.

   The new tag default is displayed on the compartment details page.

Using the CLI

Use the oci iam tag-default create command and required parameters to create a tag default on a compartment.

Example:

```
oci iam tag-default create --compartment-id compartment_OCID \
--tag-definition-id tag_definition_OCID --value text
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateTagDefault operation to create a tag default on a compartment.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Updating the Value of a Tag Default

Learn how to change the value of a tag default on a compartment on Private Cloud Appliance.

Using the Compute Web UI

1. In the navigation menu, select Identity, and then select Compartments.
2. If the compartment where you want to add a tag default is not listed, navigate to the correct compartment.

   Select the name of the top-level parent compartment. On the compartment details page, scroll to the Resources section and select Child Compartments. If necessary, select on the name of another compartment to view those child compartments.

3. Select the name of the compartment that has the tag default whose value you want to change.
4. Scroll to the Resources section on the compartment details page and select Tag Defaults.
5. For the tag default that you want to change, select the Actions menu and select the Edit option.
6. In the Tag Defaults dialog, specify the type of value you want the tag default to have:
   - **Default Value**: Enter the value for this tag default. If the selected tag key has a defined list of values, then this Default Value must be a member of that list.
   - **User-Defined Value**: Users are required to enter the value when a resource is created. Selecting User-Defined Value is invalid if the selected tag key definition has a predefined list of values.
7. Select the Submit button in the dialog box.

   The updated tag default is displayed on the compartment's details page.

Using the CLI

Use the oci iam tag-default update command and required parameters to update the value of a tag default on a compartment.

Example:

```
oci iam tag-default update --tag-default-id tag_default_OCID --value text
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateTagDefault operation to update the value of a tag default on a compartment.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Tag Default

Learn how to delete a tag default on a compartment on Private Cloud Appliance.

When you delete a tag default from a compartment, existing occurrences of the tag are not removed from resources.

Using the Compute Web UI

1. In the navigation menu, select Identity, and then select Compartments.
2. If the compartment where you want to add a tag default is not listed, navigate to the correct compartment.

   Select the name of the top-level parent compartment. On the compartment details page, scroll to the Resources section and select Child Compartments. If necessary, select on the name of another compartment to view those child compartments.
3. Select the name of the compartment that has the tag default whose value you want to change.
4. Scroll to the Resources section on the compartment details page and select Tag Defaults.
5. For the tag default that you want to delete, select the Actions menu and select the Delete option.
6. At the Delete Default Tag confirmation prompt, select Confirm.

   On the compartment details page, the state of the tag default is Deleting.

Using the CLI

Use the oci iam tag-default delete command and required parameters to delete a tag default on a compartment.

Example:

```
oci iam tag-default delete --tag-default-id tag_default_OCID
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteTagDefault operation to delete a tag default on a compartment.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Using Resource Tags (IAM on the Appliance)

On Private Cloud Appliance, tagging enables you to add metadata to resources by applying key/value pairs to resources.

You can add tags to resources during resource creation or update. See Adding Tags at Resource Creation on page 458.

Note the following requirements and limits:

- To apply, modify, or delete a tag on a resource, you must have authorization to update the resource.
- To apply, modify, or delete a defined tag on a resource, you must also have `use` access on the tag namespace.
- A resource can have at most 64 defined tags and ten free-form tags.

> **Note:**
>
> On Private Cloud Appliance you can't manage resource tags using bulk add, bulk update, or bulk remove operations.

Adding Tags at Resource Creation

On Private Cloud Appliance, you can add defined and free-form tags to a resource at the time that you create the resource.

To add a defined tag, the tag namespace and keys must be already defined.

Using the Compute Web UI

1. In the Create dialog for the resource, scroll to the Tagging section.
2. Select the Tag Namespace or select None. Select None to apply a free-form tag.
   - If you selected a Tag Namespace, then select the Tag Key, and enter a value or select a value from the list.
   - If you selected None (apply a free-form tag), then enter a Tag Key and enter a value.
3. To apply another tag, select the Additional Tag button.

   You can't specify more than one tag with the same tag namespace and the same tag key for a defined tag, or more than one tag with the same tag key for a free-form tag.
4. To review the tags on the resource, to go the details page for the new resource.

   On the resource details page, select the Tags tab to display the tags that are applied to this resource.

Using the CLI

Use the `--defined-tags` and `--freeform-tags` options with the `create` or `launch` command for the resource to add tags at resource creation.

Use the following commands to show the format for the argument for the tags options:

```
oci service resource create --generate-param-json-input defined-tags
oci service resource create --generate-param-json-input freeform-tags
```

The output of the `create` or `launch` command for the resource shows the defined and free-form tags that have been applied.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the `definedTags` and `freeformTags` attributes with the resource create operation to add tags at resource creation.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Applying Tags to an Existing Resource

On Private Cloud Appliance, you can add, change, and delete defined and free-form tags for a resource at the time that you update the resource.

To add a defined tag, the tag namespace and keys must be already defined.

Using the Compute Web UI

1. In the Edit dialog for the resource, scroll to the Tagging section.

   You can add tags, and you can modify or delete any tags that already exist.
2. To add tags, select the Additional Tag button if necessary, and select the Tag Namespace or select None. Select None to add a free-form tag.

   - If you selected a Tag Namespace, then select the Tag Key, and enter a value or select a value from the list.
   - If you selected None (apply a free-form tag), then enter a Tag Key and enter a value.

   You cannot specify more than one tag with the same tag namespace and the same tag key for a defined tag, or more than one tag with the same tag key for a free-form tag.
3. To modify existing tags, change the selections or enter different values.
4. To delete a tag, select the trash can icon.
5. To review the tags on the resource, to go the details page for the resource.

   On the resource details page, select the Tags tab to display the tags that are applied to this resource.

Using the CLI

Use the `--defined-tags` and `--freeform-tags` options with the `update` command for the resource to add, change, or delete tags at resource update.

Use the following commands to show the format for the argument for the tags options:

```
oci service resource create --generate-param-json-input defined-tags
oci service resource create --generate-param-json-input freeform-tags
```

You cannot specify more than one tag with the same tag namespace and the same tag key for a defined tag, or more than one tag with the same tag key for a free-form tag. Use the resource `get` command to show the current defined and free-form tags.

The output of the `update` command for the resource shows the defined and free-form tags that are applied.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the `definedTags` and `freeformTags` attributes with the resource update operation for the resource to add, change, or delete tags at resource update.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Filtering a List of Resources by Tag

In a list of resources on Private Cloud Appliance, learn how to show only resources that have a specified tag key or tag (tag key and value).

Using the Compute Web UI

1. Display a list of resources.
2. At the top of the list, select the Select Tag(s) button.
3. Do the following in the Filter by Tag dialog box:

   a. Select either **Defined Tag** or **Free-Form Tag**.

      - If you select Defined Tag, then select a Tag Namespace and a Tag Key.
      - If you select Free-Form Tag, then select a Tag Key.

b. Optionally enter values in the Select Values field.

- Leave the Select Values field blank. This option returns all resources that are tagged with the selected namespace and key (for defined tags) or returns all resources that are tagged with the specified key (for free-form tags), regardless of the tag value.
- Enter values in the Select Values field. This option returns all resources that are tagged with any of the tag value(s) that you enter. Select a value or enter a single value in the text box. To specify multiple values for the same namespace and key, enter Enter or Return, and then enter a new value. Each value is displayed below the text box.

c. Click Filter by Tag.

The filter that is currently applied is displayed on the Select Tag(s) button.

4. To filter by multiple tags, click the + on the Select Tag(s) button, and repeat the preceding step.
5. To remove a filter, click the filter definition on the Select Tag(s) button.

Using the CLI

1. Use the list command for the resource to show each resource in the specified compartment.

   The information for each resource shows the existing defined tags and free-form tags.
2. Use tools for your operating system to filter the list.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

1. Use the list operation for the resource to show each resource in the specified compartment.

   The information for each resource shows the existing defined tags and free-form tags.
2. Use tools for your operating system to filter the list.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Using Resource Tags to Extend Functionality (IAM on the Appliance)

On Private Cloud Appliance, special resource tags are available that enable you to specify resource parameters that are not available as CLI options or API attributes.

Some of these reserved tags are defined tags and some are free-form tags.

Use of these tags counts against your tag limit.

> **Note:**
>
> Don't use these tag names for other purposes.

See the following sections:

- Using OraclePCA Defined Tags for Extended Functionality on page 460
- Using Free-Form Tags for Extended Functionality on page 464

Using OraclePCA Defined Tags for Extended Functionality

On Private Cloud Appliance, defined tags are available that enable you to specify resource parameters that are not available as CLI options or API attributes. These defined tags are in the OraclePCA tag namespace.

> **Note:**
>
> When you use the CLI or API, you can specify the OraclePCA tag namespace, tag key, and values for the parameters that you want to set. You do not need to first create the OraclePCA tag namespace and tag keys.
>
> You only need to create the OraclePCA tag namespace, tag keys, and value choices if you need to use the Compute Web UI to set these parameters.

> **Important:**
>
> If you create these tag keys, do not delete them. Do not create this tag namespace and these keys unless you need to use the Compute Web UI to set these parameters. If you create this tag namespace and these keys, create them exactly as shown here, do not modify them, and do not delete them.

Use the information in the following tables to create the tag key definitions after you create the OraclePCA tag namespace.

Block Volume Attributes

For information about how to use these attributes, see Creating a Block Volume on page 716. For information about the block size attribute, see Using Free-Form Tags for Extended Functionality on page 464.

Attribute	Value

Block Volume Synchronous Write Bias Tag Key Definition	• **Tag Key:** Enter `logBias`.
	• **Description:**Enter a description. For example, `Control the use of the write cache flash devices for a share or LUN.` Avoid entering confidential information.
	• **Tag Value Type:** Select **A List of Values**.
	• **Values:** Enter the following values, each on a separate line:
	  • `LATENCY`
	  • `THROUGHPUT`

Block Volume Secondary Cache Tag Key Definition	• **Tag Key**: Enter `secondaryCache`.
	• **Description:** Enter a description. For example, `Control the use of the read cache flash devices for a share or LUN`. Avoid entering confidential information.
	• **Tag Value Type:** Select **A List of Values**.
	• **Values:** Enter the following values, each on a separate line:
	  • `ALL`
	  • `METADATA`
	  • `NONE`

File System Attributes

For information about how to use these attributes, see Creating a File System on page 766.

Attribute	Value
File System Quota	• **Tag Key**: Enter `quota`. • **Description**: Enter a description. For example, `The quota value in gigabytes includes the data in the file system and all snapshots created under the file system.` Avoid entering confidential information. • **Tag Value Type**: Select **Static Value**.

Attribute	Value

File System Database Record Size	• **Tag Key**: Enter `databaseRecordSize`.
	• **Description**: Enter a description. For example, `The size of a database record in bytes.`.
	• **Tag Value Type**: Select **A List of Values**.
	• **Values**: Enter the following values, each on a separate line:
	  • `512`
	  • `1024`
	  • `2048`
	  • `4096`
	  • `8192`
	  • `16384`
	  • `32768`
	  • `65536`
	  • `131072`
	  • `262144`
	  • `524288`
	  • `1048576`

File System Backing Store Pool	• **Tag Key**: Enter `poolName`.
	• **Description**: Enter a description. For example, `Whether to use the default storage pool, or use a high performance pool as the backing store pool.` Avoid entering confidential information.
	• **Tag Value Type**: Select **A List of Values**.
	• **Values**: Enter the following values, each on a separate line:
	  • `PCA_POOL`
	  • `PCA_POOL_HIGH`

SR-IOV Networking Attribute

For information about how to use this attribute, see Configuring SR-IOV for Virtual Networking on page 535.

Attribute	Value
Network Configuration for SR-IOV	• **Tag Key**: Enter `networkType`. • **Description**: Enter a description. For example, `DRG and VCN network configuration required for SR-IOV.` • **Tag Value Type**: Select **Static Value**.
Network Configuration for SR-IOV - Roving Edge Slot	• **Tag Key**: Enter `vfioPcieSlot`. • **Description**: Enter a description. For example, `Enabling slot for SR-IOV.` • **Tag Value Type**: Select **Static Value**. • **Values**: Enter a value for the slot number [0,2,3]:
Network Configuration for SR-IOV - Roving Edge Port	• **Tag Key**: Enter `vfioPortIndex`. • **Description**: Enter a description. For example, `Enabling port for SR-IOV.` • **Tag Value Type**: Select **Static Value**. • **Values**: Enter a value for the port number [0,1,2,3]

Kubernetes Engine (OKE) Attributes

For information about how to use these attributes, see Creating an OKE Worker Node Pool on page 932.

Attribute	Values
SSH Key	• **Name**: Enter `sshkey`. • **Description**: Enter a description. For example, "Your public SSH key." Avoid entering confidential information. • **Tag Value Type**: Select **Static Value**.

Number of Control Plane Nodes	• **Name**: Enter `cpNodeCount`.
	• **Description**: For example, "The number of nodes in the control plane." Avoid entering confidential information.
	• **Tag Value Type**: Select **A List of Values**.
	• **Values**: Enter the following numbers, each on a separate line:
	`1`
	`3`
	`5`

Shape of Control Plane Nodes	• **Name**: Enter `cpNodeShape`. • **Description**: For example, "The shape of the control plane nodes." Avoid entering confidential information. • **Tag Value Type**: Select **Static Value**.
Shape Configuration of Control Plane Nodes	• **Name**: Enter `cpNodeShapeConfig`. • **Description**: For example, "The number of OCPUs and optionally amount of memory for a flexible node shape." Avoid entering confidential information. • **Tag Value Type**: Select **Static Value**.

Creating the OraclePCA Tag Namespace and Keys

See the tables in the preceding sections for specifications for each tag key.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Governance**, and then select **Tag Namespaces**.
2. If OraclePCA is not shown in the Tag Namespaces list, select the **Create Namespace Definitions** button.
3. In the Create Namespace Definition dialog box, scroll down to the Tagging section and select the Tag Namespace field.
   - If the OraclePCA tag namespace is not listed, continue with Step 4 on page 463 of this procedure.
   - If the OraclePCA tag namespace is listed, then it has already been created in a different compartment and is available to all compartments.

     If the OraclePCA tag namespace is already available, select the Cancel button in the Create Namespace Definition dialog box. You will receive an error message if you try to create this tag namespace when it is already available.
4. In the Create Namespace Definition dialog box, enter the following information:
   - **Compartment**: Select the compartment where you want to create the OraclePCA tag namespace definition.
   - **Name**: Enter OraclePCA.
   - **Description**: Enter a description for the tag namespace. For example, "Support resource attributes that are available on Private Cloud Appliance." Avoid entering confidential information.
   - Select the **Create Namespace Definition** button in the dialog box.

The details page for the OraclePCA tag namespace definition is displayed.

5. On the OraclePCA details page, scroll to the Resources section and select the **Create Tag Key Definition** button.
6. In the **Create Tag Key Definition** dialog box, enter the name, description, value type, and value if applicable for each tag key definition that you want to create. Specifications for each tag key definition are provided in the preceding tables.

   After each tag key definition is entered, select the **Create Tag Key Definition** button in the dialog box. To create another tag key, select the **Create Tag Key Definition** button on the tag key definitions list again.

Using the CLI

You do not need to define these tags to use them with OCI CLI commands. You might want to use OCI CLI to define these tags for yourself or others to use in the Compute Web UI.

1. Use the oci iam tag-namespace create command to create the OraclePCA tag namespace.

   The `--compartment-id` value is the compartment where you want to create the OraclePCA tag namespace and tags. If this tag namespace has already been created in a different compartment in the tenancy, then it is available to use in any other compartment in the tenancy, and you will receive an error message from this create tag namespace command.

   Example:

   ```
   oci iam tag-namespace create --compartment-id compartment_OCID --name
    OraclePCA --description "Support resource attributes that are available
    on Private Cloud Appliance."
   ```

2. Use the oci iam tag create command to create tags in the OraclePCA tag namespace. Specify the name, description, value type, and value if applicable for each tag key definition that you want to create. Specifications for each tag key definition are provided in the preceding tables.

   You do not need to specify the `--validator` option because you want the default value.

   Example:

   ```
   $ oci iam tag create --tag-namespace-id tag_namespace_OCID --name
    "cluster_id" --description "Required to create or update an OKE cluster
    or node pool."
   ```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

You do not need to define these tags to use them with API operations. You might want to use API to define these tags for yourself or others to use in the Compute Web UI.

Use the CreateTagNamespace operation to create the OraclePCA tag namespace. If this tag namespace has already been created in a different compartment in the tenancy, then it is available to use in any other compartment in the tenancy, and you will receive an error message from this create tag namespace operation.

Use the CreateTag operation to create tags in the OraclePCA tag namespace. Specify the name, description, value type, and value if applicable for each tag key definition that you want to create. Specifications for each tag key definition are provided in the preceding tables.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Using Free-Form Tags for Extended Functionality

On Private Cloud Appliance, special free-form resource tags are available that enable you to specify resource parameters that are not available as CLI options or API attributes.

Use of these tags counts against your tag limit.

The following sections describe the special purpose free-form tags.

> **Note:**
>
> Don't use these tag names for other purposes.

PCA_no_lm – Control Instance Live Migration

Use the `PCA_no_lm` free-form tag to instruct the Compute service not to live migrate an instance. The value can be either `True` or `False`.

By default, an instance can be live migrated, such as when you need to evacuate all running instances from a compute node. Live migration can be a problem for some instances. For example, live migration must not be used for instances in a Microsoft Windows cluster. To prevent an instance from being live migrated, set this tag to `True` on the instance.

Specify this tag in the Tagging section of the Create Instance or Edit *instance_name* dialog, in the `oci compute instance launch` or `oci compute instance update` command, or using the API.

The following is an example option for the CLI:

```
--freeform-tags '{"PCA_no_lm": "True"}'
```

Setting this tag to True on an instance doesn't prevent the instance from being moved when you change the fault domain. Changing the fault domain isn't a live migration. When you change the fault domain of an instance, the instance is stopped, moved, and restarted.

PCA_blocksize – Create New Volumes with a Specific Disk Block Size

Use this tag to change the LUN block size used by the underlying storage when a block volume is created.

> **Note:**
>
> Using this tag doesn't apply to the size of the block volume.

The default block size is 8192 bytes. To specify a different block size, specify the `PCA_blocksize` free-form tag in the Tagging section of the Create Block Volume dialog, in the `oci bv volume create` command, or using the API. Valid values are a power of 2 between 512 and 1M bytes, specified as a string and fully expanded.

The following is an example option for the CLI:

```
--freeform-tags '{"PCA_blocksize": "65536"}'
```

The block size can't be changed after the volume has been created.

Virtual Cloud Networks

On Private Cloud Appliance, networking enables you to set up virtual versions of traditional network components.

When you work with Private Cloud Appliance, one of the first steps is to set up a virtual cloud network (VCN) for your cloud resources.

The infrastructure that provides the necessary services to deploy cloud workloads is configured to operate within the network environment of your data center. During initialization, the appliance's core network components are integrated with your existing data center network design.

Virtual Network Interface Cards (VNICs)

The compute nodes in Private Cloud Appliance have physical network interface cards (NICs). When you create a compute instance on one of the servers, the Networking service ensures that a VNIC is created on top of a physical interface, so that the instance can communicate over the network. Each instance has a primary VNIC that is

automatically created and attached. The primary VNIC resides in the subnet you specify when creating the instance. It can't be removed from the instance.

A VNIC enables an instance to connect to a VCN and determines how the instance communicates with endpoints inside and outside of the VCN.

For more information, see Configuring VNICs on page 518.

IP Addressing

Instances use IP addresses for communication. Each instance has at least one private IP address and optionally one or more public IP addresses. A private IP address enables the instance to communicate with other instances inside the VCN. A public IP address enables the instance to communicate with hosts outside of the cloud network environment.

Certain types of resources are designed to be directly reachable from outside the secure Private Cloud Appliance network environment, and therefore automatically come with a public IP address. For example: a NAT gateway. Other types of resources are directly reachable only if you configure them to be. For example: specific instances in your VCN. Direct public connectivity also requires that the VCN has an internet gateway and that the public subnet has correctly configured route tables and security lists.

> **Note:**
>
> You can only configure IPv4 IP addresses. You can't configure IPv6 IP addresses.

For more information, see Managing Private IP Addresses on page 523 and Managing Public IP Addresses on page 528.

DHCP Options

The Networking service uses DHCP to automatically provide configuration information to instances when they boot up. Although DHCP lets you change some settings dynamically, others are static and never change. For example, when you create an instance, either you specify the instance's private IP address or the system chooses one for you. Each time the instance boots up or you restart the instance's DHCP client, DHCP passes that same private IP address to the instance. The address never changes during the instance's lifetime.

The Networking service provides DHCP options to let you control certain types of configuration on the instances in your VCN. You can change the values of these options at your discretion, and the changes take effect the next time you restart an instance's DHCP client or reboot the instance.

For more information, see Working with DHCP Options on page 489.

Traffic Management Steering Policies

Steering Policies enable you to configure policies to serve intelligent responses to DNS queries, meaning different answers (endpoints) may be served for the query depending on the logic defined in the policy.

For more information, see Managing Traffic with Steering Policies on page 542

Name Resolution

On Private Cloud Appliance, the Domain Name System (DNS) translates human-readable domain names to machine-readable IP addresses.

For example, when you type a domain name into a browser, the OS queries several DNS name servers until it finds the authoritative name server for that domain. The authoritative name server then responds with an IP address or other requested record data; the answer is relayed back to your browser and the DNS record is resolved to the web page.

For DNS name resolution within your VCN provides two options. You choose an option for each subnet in the VCN, using the subnet's set of DHCP options.

The default choice is the **Internet and VCN Resolver**. It consists of two parts: the Internet Resolver and the VCN Resolver. The Internet Resolver lets instances resolve host names that are publicly published on the internet, without

requiring the instances to have internet access by way of either an internet gateway or a connection to the on-premises network. The VCN resolver lets instances resolve host names, which you can assign, of other instances in the same VCN.

Or, you can use a **Custom Resolver**. This enables you to use up to three DNS servers of your choice for name resolution. These could be DNS servers that are available through the internet, in your VCN, or in your on-premises network, which is connected to your VCN through a DRG.

Domains and Host Names

When you create a VCN and subnets, you can specify DNS labels for each. Subnet DNS labels can only be set if the VCN itself is created with a DNS label. The labels, along with the parent domain of `oraclevcn.com` form the VCN domain name and subnet domain name. Next, when you create an instance, you can assign a host name. The host name is assigned to the primary VNIC that is automatically created during instance creation. Along with the subnet domain name, the host name forms the instance's fully qualified domain name (FQDN).

- **VCN domain name: `<VCN_DNS_label>`**`.oraclevcn.com`
- **Subnet domain name: `<subnet_DNS_label>.<VCN_DNS_label>`**`.oraclevcn.com`
- **Instance FQDN: `<host_name>.<subnet_DNS_label>.<VCN_DNS_label>`**`.oraclevcn.com`

The FQDN resolves to the instance's private IP address. The Internet and VCN Resolver also enables reverse DNS lookup, which lets you decide the host name corresponding to the private IP address.

If you add a secondary VNIC to an instance, you can specify a host name. The resulting FQDN resolves to the VNIC primary private IP address. The resulting FQDN resolves to that private IP address.

DNS labels and host names must meet these requirements:

- VCN and subnet labels must start with a letter and have a maximum length of 15 alphanumeric characters. Hyphens and underscores are not allowed. The value can't be changed later.
- Host names can be up to 63 characters in length and must be compliant with RFCs 952 and 1123. The value can be changed later.
- VCN DNS labels must be unique across the VCNs in your tenancy. Subnet DNS labels must be unique within the VCN, and host names must be unique within the subnet.

Host Name Validation and Generation

If you set DNS labels for VCN and subnet, the host name is validated for compliance during instance creation. If you haven't specified a host name, the system tries to use the instance display name. If the display name doesn't pass validation, or if you didn't specify one, the system generates a DNS-compliant host name.

If you create an instance using the CLI or SDK, and you haven't specified a host name or display name, the system doesn't generate them for you. This means the instance isn't resolvable using the Internet and VCN Resolver.

If you add a secondary VNIC to an instance, the system never generates a host name. You must provide a valid host name if you want the private IP address to be resolvable using the Internet and VCN Resolver.

Public DNS Zones

To enable DNS name resolution for instances deployed inside your VCNs, from outside the Private Cloud Appliance network environment, Private Cloud Appliance provides configuration support for public DNS zones. Within your tenancy, you create the DNS zones, or sections of the DNS namespace, that you intend to manage. The edge network of the Private Cloud Appliance handles all DNS queries for the domains.

For more information, see Managing Public DNS Zones on page 537.

Networking Scenarios

Learn about various networking scenarios for Private Cloud Appliance.

The following sections describe basic networking scenarios to help you understand the networking service and how networking components operate together.

> **Important:**
>
> Regardless of the selected networking scenario, always ensure that the IP address ranges configured as Private Cloud Appliance public IPs don't conflict with IP addresses used within the data center network.
>
> When configuring virtual cloud networks (VCNs) with a dynamic routing gateway (DRG), ensure that the VCN IP address ranges don't overlap with each other or with the on-premises network.

- Scenario A – Public Subnet on page 468 describes a setup consisting of a VCN and a public subnet on Private Cloud Appliance.
- Scenario B – Private Subnet on page 469 describes a setup consisting of a VCN and a private subnet on Private Cloud Appliance.
- Scenario C – Public and Private Subnets on page 470 describes a multitier setup consisting of a VCN with a public and a private subnet on Private Cloud Appliance.
- Scenario D - Transit Routing Through a Hub VCN on page 472 describes a network topology where your on-premises network connects to a VCN on Private Cloud Appliance to reach resources beyond that VCN.

Scenario A – Public Subnet

This scenario describes a setup consisting of a VCN and a public subnet on Private Cloud Appliance.

For external connectivity the VCN needs an internet gateway. Your on-premises network also uses this gateway to communicate with resources inside the VCN. The IP addresses used in this scenario must be public. In a private cloud context, this means a unique address directly reachable from the on-premises network.

The subnet uses the default security list, which has default rules that are designed to simplify getting started. The rules enable typical required access; for example inbound SSH connections and any type of outbound connections. Remember that security list rules only allow traffic. Any traffic not explicitly covered by a security list rule is implicitly denied. In this scenario, you add more rules to the default security list. You could instead create a custom security list for those rules. You would then set up the subnet to use both the default security list and the custom security list.

The subnet uses the default route table, which contains no rules when the VCN is created. In this scenario, the table has only a single rule: to route traffic intended for all destinations (0.0.0.0/0) through the internet gateway.

To set up this networking scenario, you perform the following steps:

1. Create the VCN.

   Choose a compartment you have permission to work in. Specify one or more non-overlapping CIDR blocks for the VCN; for example: 172.16.0.0/16. Optionally, enable DNS and specify a DNS label for the VCN.
2. Create the public subnet.

   Specify a single, contiguous CIDR block within the VCN CIDR block; for example: 172.16.10.0/24. Select the default route table. Ensure the subnet is a public subnet, so that instances can obtain public IP addresses. If you enabled DNS at the VCN level, you can choose to assign host names in the subnet and specify a subnet DNS label as well.
3. Create the internet gateway.

   When you create the internet gateway, it's enabled immediately. However, you must add a route rule to allow traffic to flow to the gateway.
4. Update the default route table to use the internet gateway.

   The default route table starts out with no rules. No route rule is required to route traffic within the VCN itself. You must add a rule that routes all traffic destined for addresses outside the VCN to the internet gateway. Enter these parameters:

   - Target Type: Internet Gateway
   - Destination CIDR block: 0.0.0.0/0

This means that all non-intra-VCN traffic that's not already covered by other rules in the route table goes to the target specified in this rule.

- Target: The internet gateway you created.

Because the subnet was set up to use the default route table, the resources in the subnet can now use the internet gateway. The existence of this rule also enables inbound connections to the subnet, through the internet gateway. The next step is to specify the types of traffic you want to allow into and out of the instances you later create in the subnet.

5. Update the default security list.

   You set up the subnet to use the VCN default security list. Now you add security list rules that allow the types of connections that the instances in the VCN need.

   For example, if the instances in your subnet are web servers, they likely need to receive inbound HTTPS connections. To enable that traffic, add an ingress rule to the default security list using these parameters:

   - Source Type: CIDR
   - Source CIDR: 0.0.0.0/0
   - IP Protocol: TCP
   - Source Port Range: All
   - Destination Port Range: 443

6. Create instances.

   Your next step is to create one or more instances in the subnet. Each instance automatically gets a private IP address. With the network setup in this scenario, you must give each instance a public IP address, otherwise you can't access them through the internet gateway.

Scenario B – Private Subnet

This scenario describes a setup consisting of a VCN and a private subnet on Private Cloud Appliance.

For connectivity to your on-premises network, the VCN needs a dynamic routing gateway (DRG).

The subnet uses the default security list, which has default rules that are designed simplify getting started. The rules enable typical required access; for example inbound SSH connections and any type of outbound connections. Remember that security list rules only allow traffic. Any traffic not explicitly covered by a security list rule is implicitly denied. In this scenario, you add more rules to the default security list. You could instead create a custom security list for those rules. You would then set up the subnet to use both the default security list and the custom security list.

The subnet uses the default route table, which contains no rules when the VCN is created. In this scenario, the table has only a single rule: to route traffic intended for all destinations (0.0.0.0/0) through the DRG.

To set up this networking scenario, you perform the following steps:

1. Create the VCN.

   Choose a compartment you have permission to work in. Specify one or more non-overlapping CIDR blocks for the VCN; for example: 172.16.0.0/16. Optionally, enable DNS and specify a DNS label for the VCN.

2. Create the private subnet.

   Specify a single, contiguous CIDR block within the VCN CIDR block; for example: 172.16.10.0/24. Make the subnet private; the instances you create can't obtain a public IP address. Select the default route table. If you enabled DNS at the VCN level, you can choose to assign host names in the subnet and specify a subnet DNS label as well.

3. Update the default security list.

   You set up the subnet to use the VCN default security list. Now you add security list rules that allow the types of connections that the instances in the VCN will need.

   For example, if your subnet contains Microsoft Windows instances and you intend to access them using RDP, add an ingress rule to the default security list using these parameters:

- Source Type: CIDR
- Source CIDR: 0.0.0.0/0
- IP Protocol: TCP
- Source Port Range: All
- Destination Port Range: 3389

4. Create a dynamic routing gateway (DRG) and attach it to your VCN.

   When you create the DRG, it is in "Provisioning" state for a short period. Ensure provisioning is done before continuing. Next, attach the DRG you just created to your VCN. For this scenario you can ignore the advanced attachment options. The DRG attachment will be in "Attaching" state for a short period before it's ready.

   To allow traffic to flow to the DRG, you must add a route rule.
5. Update the default route table to use the DRG.

   The default route table starts out with no rules. No route rule is required to route traffic within the VCN itself. You must add a rule that routes all traffic destined for addresses in your on-premises network to the DRG. Enter these parameters:

   - Target Type: Dynamic Routing Gateway.

     The VCN attached DRG is automatically selected as the target.
   - Destination CIDR block: 0.0.0.0/0

     This means that all non-intra-VCN traffic that is not already covered by other rules in the route table goes to the target specified in this rule.

   Because the subnet was set up to use the default route table, the DRG now enables traffic between the resources in the subnet and in your on-premises network.
6. Create instances.

   Your next step is to create one or more instances in the subnet. Each instance automatically gets a private IP address. With the network setup in this scenario, no additional configuration is required to access the instances from your on-premises network.

Scenario C – Public and Private Subnets

This scenario describes a multitier setup consisting of a VCN with a public and a private subnet on Private Cloud Appliance.

The public subnet holds public instances such as web servers, and the private subnet holds private instances such as database servers. The VCN has a dynamic routing gateway (DRG) for connectivity to your on-premises network. Instances in the public subnet have external access through an internet gateway.

> **Note:**
>
> In a public cloud environment, instances in the private subnet could be allowed to initiate external connections using a NAT gateway, for example to get software updates. However, in Private Cloud Appliance the NAT gateway would provide access to the on-premises network, which the DRG already enables for those instances. The combination of a NAT gateway and DRG could cause issues because of nondeterministic routing.

Each subnet uses the default security list, which has default rules that are designed to simplify getting started. The rules enable typical required access; for example inbound SSH connections and any type of outbound connections. Remember that security list rules only allow traffic. Any traffic not explicitly covered by a security list rule is implicitly denied.

Each subnet also has its own custom security list and custom route table with rules specific to the needs of the subnet's instances. In this scenario, the VCN default route table, which is always empty to start with, is not used.

To set up this networking scenario, you perform the following steps:

1. Create the VCN.

Choose a compartment you have permission to work in. Specify one or more nonoverlapping CIDR blocks for the VCN; for example: 172.16.0.0/16. Optionally, enable DNS and specify a DNS label for the VCN.

2. Add the necessary gateways to the VCN.

   The instances in the public subnet need an internet gateway for incoming and outgoing public traffic. The instances in the private subnet need a NAT gateway to be able to reach the data center network and the internet. These gateways are enabled immediately upon creation, but you must add route rules to allow traffic to flow to them.

   To reach the private instances in the VCN from your on-premises network, you create a DRG and attach it to the VCN. When you create the DRG, it's in "Provisioning" state for a short period. Ensure provisioning is done before attaching it to the VCN. For this scenario you can ignore the advanced attachment options. The DRG attachment will be in an "Attaching" state for a short period before it's ready. To allow traffic to flow to the DRG, you must also add a route rule.

3. Create custom route tables for the subnets that you create later.

   a. For the public subnet, create a route table and add a rule that routes all traffic destined for addresses outside the VCN to the internet gateway. Enter these parameters:

      - Target Type: Internet Gateway
      - Destination CIDR block: 0.0.0.0/0

        This means that all non-intra-VCN traffic that's not already covered by other rules in the route table goes to the target specified in this rule.
      - Target: The internet gateway you created.

   b. For the private subnet, create a route table and add two rules: one that routes traffic destined for the on-premises network to the DRG, and one that routes all other traffic leaving the VCN to the NAT gateway.

      Create the route rule for the NAT gateway with these parameters:

      - Target Type: NAT Gateway
      - Destination CIDR block: 0.0.0.0/0

        This means that all nonintra-VCN traffic that's not already covered by other rules in the route table goes to the target specified in this rule.
      - Target: The NAT gateway you created.

      Create the route rule for the DRG with these parameters:

      - Target Type: Dynamic Routing Gateway.

        The VCN's attached DRG is automatically selected as the target.
      - Destination CIDR block: 10.25.0.0/16

        This means that traffic intended for an address in the on-premises network (10.25.x.y) goes to the DRG target specified in this rule.

4. Update the default security list.

   Add security list rules that allow the types of connections that the instances in the VCN will need.

   Edit each of the existing stateful ingress rules so that the Source CIDR is **not** 0.0.0.0/0, but the CIDR for your on-premises network; in this example: 10.25.0.0/16.

5. Create custom security lists for the subnets you will create later.

   a. Create a custom security list for the public subnet and add rules to allow the types of connections that the public instances need. For example, web servers likely need to receive HTTP and HTTPS ingress traffic. For HTTP, use the following settings. For HTTPS, add another similar rule for TCP port 443.

      - Source Type: CIDR
      - Source CIDR: 0.0.0.0/0
      - IP Protocol: TCP
      - Source Port Range: All

- Destination Port Range: 80

b. Create a custom security list for the private subnet and add rules to allow the types of connections that the private instances need. For example, database servers likely need to receive SQL*Net (TCP port 1521) ingress traffic from clients in the private and the public subnet. For clients in the public subnet, use the following settings. For clients in the private subnet, add another similar rule for the CIDR of the private subnet (172.16.1.0/24).

- Source Type: CIDR
- Source CIDR: 172.16.2.0/24
- IP Protocol: TCP
- Source Port Range: All
- Destination Port Range: 1521

6. Create the subnets in the VCN.

- Public subnet:

  Specify a single, contiguous CIDR block within the VCN CIDR block; for example: 172.16.2.0/24. Ensure the subnet is a public subnet, so that instances can obtain public IP addresses. Select the custom public subnet route table you created earlier.

  Select two security lists: both the default security list and the public subnet security list you created earlier. If you enabled DNS at the VCN level, you can choose to assign host names in the subnet and specify a subnet DNS label as well.

- Private subnet:

  Specify a single, contiguous CIDR block within the VCN CIDR block; for example: 172.16.1.0/24. Make the subnet private; the instances you create in this subnet can't obtain a public IP address. Select the custom private subnet route table you created earlier.

  Select two security lists: both the default security list and the private subnet security list you created earlier. If you enabled DNS at the VCN level, you can choose to assign host names in the subnet and specify a subnet DNS label as well.

7. Create instances.

Your next step is to create instances in the subnets. Each instance automatically gets a private IP address. For each instance in the public subnet, ensure to assign the instance a public IP address. Otherwise, you won't be able to reach the instance from your on-premises network. The DRG you set up allows you to reach the instances in the private subnet from your on-premises network without any additional configuration.

Scenario D - Transit Routing Through a Hub VCN

This scenario describes a network topology where the on-premises network connects to a VCN on the Private Cloud Appliance to reach resources beyond that VCN. Routing rules in this VCN are configured so that traffic *transits through* the VCN to its destination.

As a basic example of why you might use transit routing: you have a large organization with different departments, each with their own VCN. The on-premises network needs access to the different VCNs, but you do not want the administration overhead of maintaining a secure connection from each VCN to the on-premises network. Instead, you use VCNs in a hub-and-spoke topology: one VCN acts as the hub to which the on-premises network connects, and the spoke VCNs are peered with the hub VCN. Traffic between the on-premises network and the peered VCNs transits through the hub VCN. The VCNs can be in different tenancies.

Important Aspects of Transit Routing

Transit routing introduces specific component configurations that a network administrator must take into account. These and other important aspects are discussed in this section.

Notable Concepts

If you are already familiar with the Networking service and VCN peering, these are the most important concepts to understand when implementing a transit routing scenario:

- There are several different gateways involved in transit routing. The hub VCN has a dynamic routing gateway (DRG), which is the communication path with the on-premises network. For each peered spoke VCN, there is a pair of local peering gateways (LPGs) that anchor the peering connection. One LPG is on the hub VCN, and the other is on the spoke VCN.
- The Networking service does not allow local VCN peering between two VCNs with overlapping CIDRs. That means each spoke must not overlap with the hub. However, the Networking service does not validate whether the spoke VCNs themselves overlap with each other, or if any of the VCNs overlap with the on-premises network. You must ensure that CIDRs for all the subnets that need to communicate with each other don't overlap. Otherwise, traffic might be dropped.
- For each spoke VCN subnet that needs to communicate with the on-premises network, you must update the subnet's route table with a rule that sets the target, or the next hop, as the spoke VCN's LPG for all traffic destined for the on-premises network.
- You must add a route table to the hub VCN and associate it with the DRG attachment. For each spoke VCN, add a route rule to the DRG attachment route table for all traffic destined for a given spoke VCN, or a subnet in the spoke VCN; set the target to the hub VCN's LPG for that spoke.
- You must add another route table to the hub VCN and associate it with the hub VCN's LPG for a given spoke. Add a route rule to the spoke LPG route table for all traffic destined for the on-premises network, or a subnet in that network; set the target to the DRG. The same route table setup is required for each spoke VCN that needs to communicate with the on-premises network.

Routing Restrictions

There are important restrictions to be aware of when implementing transit routing in your environment.

- **Route table for DRG attachment**

  A route table that is associated with a DRG attachment can have only rules that target an LPG or a private IP.

  A DRG attachment can exist without an associated route table. However, after you associate a route table with a DRG attachment, there must always be a route table associated with it. You can associate a different route table. You can also edit the table's rules, or delete some or all of the rules.
- **Route table for LPG**

  A route table that is associated with an LPG can have only rules that target a DRG.

  An LPG can exist without an associated route table. However, after you associate a route table with an LPG, there must always be a route table associated with it. You can associate a different route table. You can also edit the table's rules, or delete some or all of the rules.
- **Traffic through the hub VCN**

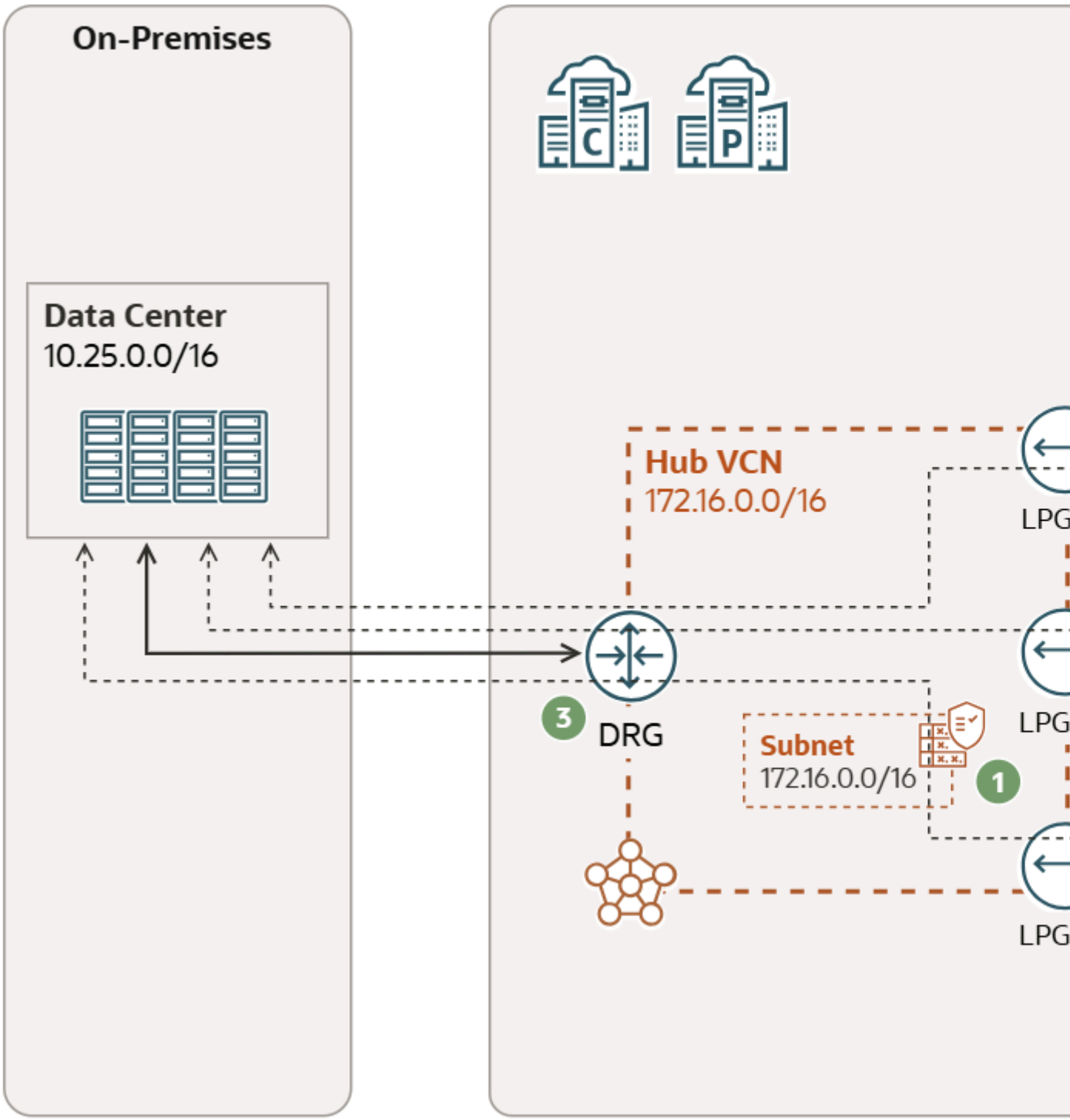
  The route tables discussed here are intended only for moving traffic *through* the hub VCN between locations in the on-premises network and locations in a spoke VCN. Routing traffic through a private IP in the hub is not supported on Private Cloud Appliance.
- **Inbound traffic to the hub VCN**

  Although the purpose of transit routing is to move traffic through the hub VCN, inbound traffic to subnets within the hub VCN is always allowed. No explicit rules are required in the DRG attachment route table or the hub LPG route table.

  When this kind of inbound traffic reaches the DRG or the hub LPG, the traffic is automatically routed to its destination in the hub VCN by the VCN local routing. Consequently, for any route table belonging to a given VCN, you cannot create a rule that lists that VCN's CIDR, or a sub-section, as the rule's destination.

Routing Paths in Detail

To further illustrate how transit routing takes place, the different paths of traffic are displayed in the diagram, and described in detail below.



The callouts in the diagram indicate the route tables to be added for this configuration. The route rules required in this example are listed in the table that follows.

Route Tables

Callout	Route Table	Route Rules

1	Hub subnet	`Destination CIDR Route Target`
		`10.25.0.0/16 DRG`
		`10.1.0.0/16 LPG-H1`
		`10.2.0.0/16 LPG-H2`
		`10.3.0.0/16 LPG-H3`

2	Spoke subnet	For all spoke subnets, the route target must be the LPG of the parent VCN.
		`Destination CIDR Route Target`
		`172.16.0.0/16 LPG-S1`
		`10.25.0.0/16 LPG-S1`

3	DRG attachment	`Destination CIDR Route Target`
		`10.1.0.0/16 LPG-H1`
		`10.2.0.0/16 LPG-H2`
		`10.3.0.0/16 LPG-H3`

4	Hub LPG	For each hub LPG (H1, H2, H3), traffic intended for the on-premises network must be sent through the DRG.
		`Destination CIDR Route Target`
		`10.25.0.0/16 DRG`

These are the networking elements used in the example:

- On-premises network: 10.25.0.0/16
- Hub VCN: 172.16.0.0/16

  The hub VCN has a DRG to connect with the on-premises network. It also has an LPG for each peering connection to a spoke VCN, for example: LPG-H1, LPG-H2 and so on.
- Spoke VCN-S1: 10.1.0.0/16

  The spoke VCN has an LPG for the peering connection to the hub VCN: LPG-S1

  The spoke VCN contains subnets, for example: 10.1.1.0/24, 10.1.2.0/24 and so on.
- Spoke VCN-S2: 10.2.0.0/16

  The spoke VCN has an LPG for the peering connection to the hub VCN: LPG-S2

  The spoke VCN contains subnets, for example: 10.2.1.0/24, 10.2.2.0/24 and so on.
- Additional spoke VCNs follow the same pattern as VCN-S1 and VCN-S2.

Traffic from the on-premises network to the spoke VCN

1. Traffic leaves the on-premises network and reaches the DRG. The traffic's destination is in Subnet-1 of VCN-S1, for example: 10.1.1.5.
2. The DRG attachment's associated route table has a rule for 10.1.0.0/16. It matches the destination and sends the traffic to the route target: LPG-H1.

3. LPG-H1 receives the traffic.

   Egress traffic leaving a VCN through an LPG is automatically routed to the LPG's peered LPG, which is LPG-S1 in this situation. That routing occurs automatically because of the peering connection between the two LPGs.
4. LPG-S1 receives the traffic.

   Traffic coming in to a VCN through the LPG is automatically routed to the destination within the VCN because of VCN local routing. No explicit route rules are required.

Traffic from the spoke VCN to the on-premises network

1. Traffic comes from an instance in Subnet-1 of spoke VCN-S1. The traffic's destination is in the on-premises network, for example: 10.25.112.17.
2. Subnet-1's associated route table has a rule for 10.25.0.0/16. It matches the destination and sends the traffic to the route target: LPG-S1.
3. LPG-S1 receives the traffic.

   Egress traffic leaving a VCN through an LPG is automatically routed to the LPG's peered LPG, which is LPG-H1 in this situation. That routing occurs automatically because of the peering connection between the two LPGs.
4. LPG-H1 receives the traffic.

   LPG-H1's associated route table has a rule for 10.25.0.0/16. It matches the destination and sends the traffic to the route target: the hub VCN's DRG.
5. The DRG receives the traffic.

   Egress traffic leaving a VCN through the DRG is automatically routed to the destination in the connected on-premises network. No explicit route rules in the DRG attachment's route table are required.

Subnet-1 in the spoke VCN and LPG-H1 both have route rules with 10.25.0.0/16 as the destination CIDR. Those rules are not required to use the exact same CIDR block, but you must ensure that both rules cover the traffic you want to route from the spoke to the on-premises network. The rule in Subnet-1's route table controls which traffic is routed from Subnet-1 to LPG-H1. The rule in LPG-H1's route table controls which traffic is routed from the spoke VCN to the on-premises network. If LPG-H1's route rule is more restrictive than Subnet-1's route rule, some traffic leaving the subnet could be dropped and not reach the DRG.

Traffic from the spoke VCN to a subnet in the hub VCN

Depending on your situation, you might want to enable traffic between instances in the hub VCN and a spoke VCN, in addition to the traffic between the on-premises network and a spoke VCN. This is possible as long as routing only occurs directly between gateways, and not through a private IP in the hub VCN.

This is how traffic would flow from the spoke VCN to a destination with an address in the hub VCN:

1. Traffic comes from an instance in Subnet-1 of spoke VCN-S1. The traffic's destination is in HubSubnet-A of the hub VCN, for example: 172.16.2.85.
2. Subnet-1's associated route table has a rule for 172.16.0.0/16. It matches the destination and sends the traffic to the route target: LPG-S1.
3. LPG-S1 receives the traffic.

   Egress traffic leaving a VCN through an LPG is automatically routed to the LPG's peered LPG, which is LPG-H1 in this situation. That routing occurs automatically because of the peering connection between the two LPGs.
4. LPG-H1 receives the traffic.

   Traffic coming in to a VCN through the LPG is automatically routed to the destination within the VCN because of VCN local routing. No explicit route rules are required.

A similar series of routing steps occurs for traffic going from HubSubnet-A to Subnet-1, but in the reverse direction. HubSubnet-A's route table has a rule that matches the spoke VCN's CIDR (10.1.0.0/16) and sends the traffic to LPG-H1, which forwards it on to LPG-S1.

Setting Up VCN Transit Routing

To set up a transit routing scenario using direct traffic between gateways, you perform the tasks described below.

Task 1: Set up the hub VCN

1. Create a new VCN.

   Specify one or more non-overlapping CIDR blocks for the VCN; for example: 172.16.0.0/16.
2. Optionally, create one or more subnets in the hub VCN. The subnets can contain cloud resources that your on-premises network or the spoke VCN need to use.

   Specify a single, contiguous CIDR block within the VCN's CIDR block; for example: 172.16.1.0/24.

Task 2: Connect the hub VCN with the on-premises network

1. Create a DRG and attach it to the hub VCN.
2. Update the hub VCN route table to allow traffic to flow to the DRG.

   Select the on-premises network as the destination CIDR; for example: 10.25.0.0/16.

   The route table to be associated with the DRG attachment is created in a later task.

Task 3: Set up a spoke VCN with at least one subnet

1. Create a new VCN. In this example it is named VCN-S1.

   Specify one or more non-overlapping CIDR blocks for the VCN; for example: 10.1.0.0/16.
2. Create one or more subnets in the spoke VCN.

   Specify a single, contiguous CIDR block within the VCN's CIDR block; for example: 10.1.1.0/24.

Task 4: Set up local peering between the hub VCN and the spoke VCN

1. Add an LPG to the hub VCN and spoke VCN. In this example they are named LPG-H1 and LPG-S1 respectively.
2. Establish the peering connection between the two LPGs.
3. Update the hub VCN route table to allow traffic to flow to LPG-H1.

   Select the spoke VCN-S1 as the destination CIDR: 10.1.0.0/16.

   The route table to be associated with LPG-H1 is created in a later task.
4. Create a route table for Subnet-1 in spoke VCN-S1.

   Add a route rule to direct traffic destined for the hub VCN (172.16.0.0/16) to LPG-S1.
5. Associate the route table with Subnet-1 in spoke VCN-S1.
6. If other subnets in the spoke VCN need to communicate with the on-premises network, perform steps 4 and 5 with different parameters for each of those subnets in order to set up and associate a similar route table.

Task 5: Add a route rule to the spoke VCN's subnet

1. Edit the route table associated with Subnet-1 in spoke VCN-S1. You created this route table in the previous task.
2. Add a route rule for traffic destined for the on-premises network (10.25.0.0/16). Set the rule target to the LPG of this spoke VCN: LPG-S1.
3. Repeat these steps for other subnets in the spoke VCN that need to communicate with the on-premises network.

Task 6: Set up ingress routing for the DRG and LPG on the hub VCN

1. Create a route table for the DRG attachment.

   Add a route rule for traffic destined for the spoke VCN-S1 (10.1.0.0/16). Set the rule target to the hub VCN's LPG for the spoke in question: LPG-H1.
2. Associate the DRG route table with the hub VCN's DRG attachment.
3. Create a route table for the hub VCN's LPG for this spoke: LPG-H1.

   Add a route rule for traffic destined for the on-premises network (10.25.0.0/16). Set the rule target to the hub VCN's DRG.
4. Associate the route table with the hub VCN's LPG for the spoke in question: LPG-H1.

Later: adding more spoke VCNs

1. Repeat tasks 3-5 for the new spoke VCN.
2. Repeat task 6 with these changes:
   - For Step 1: Instead of creating a new route table for the DRG attachment, update the existing route table to include a new rule for the new spoke VCN. The destination CIDR is the spoke VCN's CIDR, or a subnet within). The target is the hub VCN's LPG for the new spoke.
   - For Step 2: Skip this step entirely because the DRG attachment is already associated with its route table.
   - For the entire procedure: use convenient names in accordance with the spokes you are adding. For example, if you are adding spoke VCN-S4, it is helpful to call the two peering gateways LPG-S4 and LPG-H4, and use those gateway names to identify the route tables you associate with them.

Turning Off Transit Routing

To turn off transit routing, remove the rules from:

- The route table associated with the DRG attachment.
- The route table associated with each LPG on the hub VCN.

A route table can be associated with a resource but have no rules. Without at least one rule, a route table does nothing.

A DRG attachment or LPG can exist without a route table associated with it. However, after you associate a route table with a DRG attachment or LPG, there must always be a route table associated with it. But, you can associate a *different* route table. You can also edit the table's rules, or delete some or all rules.

Managing VCNs and Subnets

On Private Cloud Appliance, a virtual cloud network, or VCN, is a software-defined equivalent of a traditional network, with firewall rules and various types of communication gateways.

Virtual Cloud Network (VCN) Overview

A VCN resides in the single region of and covers one contiguous CIDR block of your choice. The size of a VCN is /16 to /30. The CIDR block can **NOT** be changed after the VCN is created. The maximum number of private IPs within a VCN is 64,000. Oracle recommends using a private IP address range as specified in RFC 1918 (10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16). This documentation uses the term private IP address when referring to IP addresses in your VCN CIDR.

You can privately connect a VCN to another VCN so that traffic doesn't leave the secure network environment of Private Cloud Appliance. However, the CIDRs of the two VCNs must not overlap. The concept of privately connecting VCNs is called *peering*. It involves setting up a type of virtual router known as a Local Peering Gateway. For more information about the use of gateways in VCNs, see Configuring VCN Gateways on page 505.

Subnets

A VCN is subdivided into subnets. Each subnet in a VCN consists of a contiguous range of IPv4 addresses that don't overlap with other subnets in the VCN. The first two IPv4 addresses and the last in the subnet CIDR are reserved by the Networking service. You can't change the size of the subnet after creation.

Subnets act as a unit of configuration: all instances in a particular subnet use the same route table, security lists, and DHCP options. Subnets can be either public or private. This is defined when the subnet is created, and can't be changed later. In a private subnet, instances can't be assigned a public IP address.

You can think of a compute instance as residing in a subnet. However, to be precise, each instance is attached to a virtual network interface card (VNIC), which in turn resides in the subnet and enables a network connection for that instance.

Creating a VCN

On Private Cloud Appliance, a *virtual cloud network* (VCN) a virtual, private network that closely resembles a traditional network. VCNs can be further divided into IP subnets. VCNs can communicate with each other through various types of gateways, each type intended for a particular purpose.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. Select **Create Virtual Cloud Network**.
3. Enter the following information:
   - **Name:** Enter a descriptive name for the VCN.
   - **Compartment:** Select the compartment in which to create the VCN.
   - **CIDR Block:** Specify which CIDR range can be used within the VCN.
   - **DNS:** If you check the box to use DNS host names in this VCN, then you can either enter a DNS label or leave the field blank to let the system generate a label for you. The first character of the label must be a letter. Only use letters and numbers. Up to 15 characters are allowed.
4. **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
5. Select **Create Virtual Cloud Network**.

   The details page of the new VCN is displayed.

Using the CLI

Use the oci network vcn create command and required parameters to create a VCN.

```
oci network vcn create --compartment-id compartment_OCID --cidr-block
 10.0.0.0/16 --dns-label vcn1 --display-name VCN1 [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Procedure

1. Gather the information that you need to run the command:
   - Compartment OCID (`oci iam compartment list --compartment-id-in-subtree true`)
2. Enter the `vcn create` command using at least the compartment OCID and CIDR block options.

   To use DNS host names in the VCN, include the DNS label in the create command. It can't be added later.

   Optionally, set a descriptive name for the VCN.

```
$ oci network vcn create --compartment-id <compartment_OCID> \
--cidr-blocks '["<cidr-block_eg-10.0.0.0/16>"]' --dns-label <dns_label> --
display-name <display_name>
{
  "data": {
    "cidr-block": "10.0.0.0/16",
    "cidr-blocks": [
      "10.0.0.0/16"
    ],
    "compartment-id": "ocid1.compartment.unique_ID",
    "default-dhcp-options-id": "ocid1.dhcpoptions.unique_ID",
    "default-route-table-id": "ocid1.routetable.unique_ID",
    "default-security-list-id": "ocid1.security_list.unique_ID",
    "defined-tags": {},
    "display-name": "VCN1",
    "dns-label": "vcn1",
    "freeform-tags": {},
    "id": "ocid1.vcn.unique_ID",
```

```
    "ipv6-cidr-block": null,
    "ipv6-private-cidr-block": null,
    "lifecycle-state": "PROVISIONING",
    "time-created": "2023-04-27T04:34:58.722835+00:00",
    "vcn-domain-name": "vcn1.oraclevcn.com"
  },
  "etag": "unique_ID"
}
```

Using the API

Use the CreateVCN operation to create a VCN.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Creating a Subnet

On Private Cloud Appliance, a subnet is a subdivision of a VCN. For each VCN, you create one or more subnets.

VCNs can be divided into subnets. Although it's possible to have an enormous VCN with a thousand IP addresses, it often makes sense from a performance and fault isolation standpoint to create multiple subnets within a VCN. The subnets can still communicate if configured properly.

IP subnet calculation can be a difficult task, especially when figuring out which IP addresses in the range are reserved. The wide range of allowable CIDR block addresses complicates the issue. Free subnet calculation tools available online can help, such as https://www.calculator.net/ip-subnet-calculator.html.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN in which you want to create a subnet.
3. Select the name of the VCN in which you want to create a new subnet.

   The VCN details page is displayed.
4. In the **Resources** section, select **Subnets**.
5. Select **Create Subnet**.
6. Enter the following information:

   - **Name:** Enter a descriptive name for the subnet.
   - **Create in Compartment:** Select the compartment where you want to create this subnet.
   - **CIDR Block:** Specify which CIDR range can be used within the subnet.

     > **Important:**
     >
     > The CIDR block range is an important subnet parameter. The range must be within the VCN CIDR block and must not overlap with other subnet CIDR block ranges. These IP addresses can't be shared.

   - **Route Table (Optional):** Select the route table to associate with this subnet. You might need to change the compartment selection. If you do not select a route table, the VCN default route table is used.
   - **Private or Public Subnet:** If you select Private Subnet, instances in this subnet are not allowed to obtain a public IP address.
   - **DNS Hostnames (Optional):** Check this box to assign a DNS hostname when you launch an instance in this subnet. If you check the box, enter a DNS label that's unique across the system.
   - **DHCP Options (Optional):** Select the set of DHCP options to associate with the subnet. You might need to change the compartment selection. If you do not select a set of options, the VCN default set is used.
   - **Security Lists (Optional):** If you want a security list for this subnet, select +Add Security List. Select a security list to associate with the subnet. You might need to change the compartment selection. If you want

another security list, select +Add Security List and select another security list. If you don't select a security list, the VCN default security list is used.

7. **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
8. Select **Create Subnet**.

   The details page of the new subnet is displayed.

Using the CLI

Use the oci network subnet create command and required parameters to create a subnet in a specified VCN.

```
oci network subnet create --compartment-id compartment_OCID --vcn-
id vcn_OCID --cidr-block cidr_block --dns-label dns_label --display-
name display_name [OPTIONS]
```

> **Important:**
>
> The CIDR block range is an important subnet parameter. The range must be within the VCN CIDR block and must not overlap with other subnet CIDR block ranges. These IP addresses can't be shared.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateSubnet operation to create a subnet in a specified VCN.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing a Subnet

On Private Cloud Appliance, you can edit subnets. You can change the subnet name, the route tables and security lists used by the subnet, and DHCP options.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN and subnet.
3. Select the name of the VCN that contains the subnet you want to edit.

   The VCN details page is displayed.
4. Under **Resources**, select **Subnets**.
5. Select the Actions menu (⋮), then select **Edit**.
6. Make the changes to the subnet. The following items can be changed:
   - **Name:** Change the name of the subnet.
   - **Route Table:** Select a different route table for this subnet. You might need to change the compartment selection.
   - **DHCP Options:** Select a different set of DHCP options for this subnet. You might need to change the compartment selection.
   - **Security Lists:** Select different or additional security lists for this subnet. You might need to change the compartment selection.
7. **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
8. Select **Save Changes**.

   The subnet properties are updated.

Using the CLI

Use the oci network subnet update command and required parameters to update a subnet.

```
oci network subnet update --subnet-id ocid1.subnet.unique_ID
 --dhcp-options-id ocid1.dhcpoptions.unique_ID  --route-table-id
 ocid1.routetable.unique_ID [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateSubnet operation to update a subnet.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Subnet

On Private Cloud Appliance, to delete a subnet, it must contain no resources (instances, VNICs).

For more information about deleting VCNs and Subnets, see Subnet or VCN Deletion

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN and subnet.
3. Select the name of the VCN that contains the subnet you want to delete.
4. Under **Resources**, select **Subnets**.
5. Select the Actions menu (⋮), select **Delete**.
6. Confirm the deletion.

Using the CLI

Use the oci network subnet delete command and required parameters to delete a subnet.

```
oci network subnet delete --subnet-id subnet_OCID [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteSubnet operation to delete a subnet.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a VCN

On Private Cloud Appliance, a VCN can only be deleted if it's empty. Before deleted a VCN, ensure that all subnets, route tables, gateways, and other resources have been deleted.

For more information about deleting VCNs and Subnets, see Subnet or VCN Deletion

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN.
3. Select the name of the VCN that you want to delete.

   The VCN details page is displayed.
4. Under **Resources**, ensure that the VCN doesn't have any resources.

   If all the resources such as Subnets, Route Tables, and so on include a zero in parenthesis, then the VCN doesn't have any resources.

If a resource exists, you must delete the resource before you can delete the VCN.

5. Select **Delete**.
6. Confirm the deletion.

Using the CLI

Use the oci network vcn delete command and required parameters to delete a VCN.

```
oci network vcn delete --vcn-id vcn_OCID [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteVCN operation to delete a VCN.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Configuring VCN Rules and Options

On Private Cloud Appliance, VCNs and their subnets have various rules and options associated with them. The main categories are the use of DHCP, route tables, and security. If you do not configure these rules and options explicitly, the system uses default values.

VCN Security Rules and Security Lists

On Private Cloud Appliance, VCN security involves creating and using rules that are gathered into security lists or network security groups (NSGs).

The rules can be stateful or stateless. Stateful rules presume various things about an instance interaction. For example, a stateful rule that applies to a request to an instance assumes that there is a response and automatically allows this response. In contrast, stateless rules don't presume anything about any interactions. Stateless rules are messages from one instance to another, and apply to everything regardless of what has gone before. Stateful rules require overhead processing to track states that stateless rules don't. Instances with heavy and widely varied traffic, such as web sites, should use stateless rules where possible to minimize system loads.

> **Note:**
>
> Care is needed when creating stateless and stateful rules. When a message appears that fits both stateful and stateless rule criteria, the stateless rule is applied. This results in responses being blocked to requests that seemingly should be allowed.

Your VCN might have subnets that use the default security list. Don't delete any of the list's default security rules unless you've first confirmed that resources in your VCN don't require them. Otherwise, you might disrupt VCN connectivity.

Remember that:

- A security rule is stateful by default, but can also be configured to be stateless. A common practice is to use stateless rules for high-performance applications. In a case where network traffic matches both stateful and stateless security lists, the stateless rule takes precedence. For more information about configuring VCN security rules, see Security Rules.
- To prevent unauthorized access or attacks on Compute instances, we recommend that you use a VCN security rule to allow SSH or RDP access only from authorized CIDR blocks rather than leave them open to the internet (0.0.0.0/0). For additional security, you can temporarily enable SSH (port 22) or RDP (port 3389) access on an as-needed basis using the VCN API `UpdateNetworkSecurityGroupRules` (if you're using network security groups) or `UpdateSecurityList`(if you're using security lists).

  For more information about enabling RDP access, see **To enable RDP access** in Recommended Networking Launch Types.

For performing instance health checks, we recommend that you configure VCN security rules to allow ICMP pings. For more information, see Rules to Enable Ping.

VCN network security groups (NSGs) and security lists enable security-critical network access control to Compute instances, and it's important to prevent any unintended or unauthorized changes to NSGs and security lists. To prevent unauthorized changes, we recommend that you use IAM policies to allow only network administrators to make NSG and security list changes.

The default security list for a VCN comes without stateless rules: all the default security list rules are stateful. In most cases, the default rules should be changed to allow only inbound traffic from authorized subnets.

Default Security List Rules

- **Stateful ingress:** Allow TCP traffic on destination port 22 (SSH) from authorized source IP addresses and any source port. This rule makes it easy for you to create a new cloud network and public subnet, open a Linux instance, and then immediately use SSH to connect to that instance without needing to write any security list rules yourself.
- **Stateful ingress:** Allow ICMP traffic type 3 code 4 from authorized source IP addresses. This rule enables instances to receive Path MTU Discovery fragmentation messages.
- **Stateful ingress:** Allow ICMP traffic type 3 (all codes) from your VCN CIDR block. This rule makes it easy for your instances to receive connectivity error messages from other instances within the VCN.
- **Stateful egress:** Allow all traffic. This allows instances to initiate traffic of any kind to any destination. Notice that this means the instances with public IP addresses can talk to any internet IP address if the VCN has a configured internet gateway. And because stateful security rules use connection tracking, the response traffic is automatically allowed regardless of any ingress rules.

Security List Rule Modifications

RDP

The default security list doesn't include a rule to allow Remote Desktop Protocol (RDP) access. If you're using Microsoft Windows images, ensure to add a stateful ingress rule for TCP traffic on destination port 3389 from authorized source IP addresses and any source port.

Instance Pings

The default security list doesn't include a rule to allow ping requests. If you plan to ping an instance, ensure that the instance's applicable security list includes an additional stateful ingress rule to allow ICMP traffic type 8 from the source network you plan to ping from. To allow ping access from the data center, use `0.0.0.0/0` for the source. Note that this rule for pinging is separate from the default ICMP-related rules in the default security list. Don't remove those rules.

Path MTU Discovery Fragmentation Messages

If you decide to use stateless security rules to allow traffic to and from endpoints outside the VCN, it's important to add a security rule that allows ingress ICMP traffic type 3 code 4 from source `0.0.0.0/0` and any source port. This rule enables instances to receive Path MTU Discovery fragmentation messages. This rule is critical for establishing a connection to instances. Without it, you can experience connectivity issues.

UDP

Instances can send and receive UDP traffic. In some cases, UDP packets might be fragmented on links with smaller MTU sizes. If a UDP packet is too large for the connection size limit, it's fragmented. However, only the first fragment from the packet contains the protocol and port information. If the security rules that allow this ingress or egress traffic specify a particular port number (source or destination), then the fragments after the first one are dropped. If you expect your instances to send or receive large UDP packets, set both the source and destination ports for the applicable security rules to ALL (instead of a particular port number).

Subnets and Security Rules

On their own, security lists define a set of security rules that apply to all the VNICs in an entire subnet. To use a specific security list with a particular subnet, associate the security list with the subnet either during subnet creation or later. A subnet can be associated with a maximum of five security lists. Any VNICs that are created in that subnet are subject to the security lists associated with the subnet.

Virtual Firewall

On Private Cloud Appliance, the Networking service offers two virtual firewall features that both use security rules to control traffic at the packet level – security lists and network security groups (NSGs). They offer different ways to apply security rules to a set of virtual network interface cards (VNICs).

- **Security lists:**

  A security list defines security rules at the subnet level, which means that all VNICs in a particular subnet are subject to the same rules. Each VCN comes with a default security containing default rules for essential traffic. The default security list is automatically used with all subnets, unless a custom security list is specified. A subnet can have up to five associated security lists.
- **Network security groups (NSGs):**

  A network security group defines security rules based on membership. Its security rules apply to resources that are explicitly added to the NSG. A VNIC can be added to a maximum of five NSGs. An NSG is intended to provide a virtual firewall for a set of cloud resources with the same security posture. For example: a group of instances that perform the same tasks and thus need to use the same set of ports.

Oracle recommends using NSGs instead of security lists because NSGs let you separate the VCN subnet architecture from your application security requirements. However, NSGs are only supported for specific services. It is possible to use both security lists and NSGs together, depending on your particular security needs.

If you have security rules that you want to enforce for all VNICs in a VCN, the easiest solution is to put the rules in one security list, and then associate that security list with all subnets in the VCN. This way you can ensure that the rules are applied, regardless of who in your organization creates a VNIC in the VCN. Or, you can add the required security rules to the VCN default security list.

If you choose to combine security lists and network security groups, the set of rules that applies to a particular VNIC is the union of these items:

- The security rules in the security lists associated with the VNIC subnet
- The security rules in all NSGs that the VNIC is in

A packet in question is allowed if *any rule* in any of the relevant lists and groups allows the traffic, or if the traffic is part of an existing connection being tracked because of a stateful rule.

Security Rules

On Private Cloud Appliance, you can configure security rules for security lists and network security groups (NSGs).

This section explains important aspects of security rules that you need to understand to implement them. How you create, manage, and apply security rules varies between security lists and network security groups.

Parts of a Security Rule

A security rule allows a particular type of traffic into or out of a VNIC. For example, a commonly used security rule allows ingress TCP port 22 traffic for establishing SSH connections to the instance. Without security rules, no traffic is allowed into and out of VNICs in the VCN.

Each security rule specifies the following items:

- **Direction (ingress or egress):** Ingress is inbound traffic to the VNIC; egress is outbound traffic from the VNIC.

  The REST API model for security lists is different from network security groups. With security lists, there is an `IngressSecurityRule` object and a separate `EgressSecurityRule` object. With network security groups, there is only a `SecurityRule` object, and the object's `direction` attribute determines whether the rule is for ingress or egress traffic.

- **Stateful or stateless:** If stateful, connection tracking is used for traffic matching the rule. If stateless, no connection tracking is used. See Stateful and Stateless Rules in this section.
- **Source type and source:** For ingress rules only; the source you provide depends on the source type you choose. These source types are allowed:

Source Type	Allowed Source
CIDR	The CIDR block where the traffic originates from. Use 0.0.0.0/0 to indicate all IP addresses. The prefix is required. For example, include the /32 if specifying an individual IP address.

- **Destination type and destination:** For egress rules only; the destination you provide depends on the destination type you choose. These destination types are allowed:

Destination Type	Allowed Destination
CIDR	The CIDR block that the traffic is destined for. Use 0.0.0.0/0 to indicate all IP addresses. The prefix is required. For example, include the /32 if specifying an individual IP address.

- **IP Protocol:** Either a single IPv4 protocol or "all" to cover all protocols.
- **Source port:** The port where the traffic originates from. For TCP or UDP, you can specify all source ports, or optionally specify a single source port number, or a range.
- **Destination port:** The port where the traffic is destined to. For TCP or UDP, you can specify all destination ports, or optionally specify a single destination port number, or a range.
- **ICMP type and code:** For ICMP, you can specify all types and codes, or optionally specify a single type with an optional code. If the type has multiple codes, create a separate rule for each code you want to allow.
- **Description:** NSG security rules contain an optional attribute to include a description of the rule. This is currently not supported for security list rules.

Stateful and Stateless Rules

When you create a security rule, you choose whether it is stateful or stateless. The default is stateful. Stateless rules are recommended if you have a high-volume internet-facing website, for the HTTP/HTTPS traffic.

Marking a security rule as **stateful** indicates that you want to use connection tracking for any traffic that matches that rule. This means that when an instance receives traffic matching the stateful ingress rule, the response is tracked and automatically allowed back to the originating host, regardless of any egress rules applicable to the instance. And when an instance sends traffic that matches a stateful egress rule, the incoming response is automatically allowed, regardless of any ingress rules.

Marking a security rule as **stateless** indicates that you do NOT want to use connection tracking for any traffic that matches that rule. This means that response traffic is not automatically allowed. To allow the response traffic for a stateless ingress rule, you must create a corresponding stateless egress rule.

If you use both stateful and stateless rules, and there is traffic that matches both a stateful and stateless rule in a particular direction, the stateless rule takes precedence and the connection is not tracked. You would need a corresponding rule in the other direction, either stateless or stateful, for the response traffic to be allowed.

If you decide to use stateless security rules to allow traffic to/from endpoints outside the VCN, it is important to add a security rule that allows ingress ICMP traffic type 3 code 4 from source 0.0.0.0/0 and any source port. This rule enables your instances to receive Path MTU Discovery fragmentation messages. This rule is critical for establishing a connection to your instances. Without it, you can experience connectivity issues.

Best Practices for Security Rules

- **Use network security groups**

Oracle recommends using NSGs for components that all have the same security posture. For example, in a multitier architecture, you would have a separate NSG for each tier. A tier's VNICs would all belong to that tier's NSG.

Within a tier, you might have a particular subset of the tier's VNICs that have additional, special security requirements. Therefore you would create another NSG for those additional rules, and place that subset of VNICs into both the tier's NSG and the NSG with additional rules.

- **Understand default security list rules**

  Each VCN automatically comes with a default security list that contains several default security rules to help you get started using the Networking service. Those rules exist because they enable basic connectivity.

  Even if you choose not to use security lists or the default security list, get familiar with the rules so you better understand the types of traffic that your networked cloud resources require. You might want to use those rules in your NSGs or any custom security lists that you set up.

  There is no default rule to allow ping requests. If you want to ping an instance, add a stateful ingress rule to specifically allow ICMP traffic type 8 from the source network you plan to ping from. To allow ping access from the internet, use 0.0.0.0/0 for the source. Note that this rule for pinging is separate from the default ICMP-related rules in the default security list. Don't remove those rules.
- **Do not delete default security rules indiscriminately**

  Your VCN might have subnets that use the default security list by default. Don't delete any of the list's default security rules unless you have first confirmed that resources in your VCN don't require them. Otherwise, you might disrupt your VCN connectivity.
- **If necessary, add rules to allow ping requests**

  There is no default rule to allow ping requests. If you want to ping an instance, add a stateful ingress rule to specifically allow ICMP traffic type 8 from the source network you plan to ping from. To allow ping access from the internet, use 0.0.0.0/0 for the source. Note that this rule for pinging is separate from the default ICMP-related rules in the default security list. Don't remove those rules.
- **If necessary, add rules to handle fragmented UDP packets**

  Instances can send or receive UDP traffic. If a UDP packet is too large for the connection, it's fragmented. However, only the first fragment from the packet contains the protocol and port information. If the security rules that allow this ingress or egress traffic specify a particular (source or destination) port number, the fragments after the first one are dropped. If you expect your instances to send or receive large UDP packets, set both the source and destination ports for the applicable security rules to ALL instead of a particular port number.
- **Align OS firewall rules with security rules**

  Your instances running images provided with Private Cloud Appliance also have OS firewall rules that control access to the instance. When troubleshooting access to an instance, ensure that all the following items are set correctly:

  - The rules in the network security groups that the instance is in
  - The rules in the security lists associated with the instance's subnet
  - The instance's OS firewall rules

Security Lists

On Private Cloud Appliance, a security list acts as a virtual firewall for an instance, with ingress and egress rules that specify the types of traffic allowed in and out.

Each security list is enforced at the VNIC level. However, you configure your security lists at the subnet level, which means that all VNICs in a particular subnet are subject to the same set of security lists.

The security lists apply to a particular VNIC whether it's communicating with another instance in the VCN or a host outside the VCN. Each subnet can have multiple security lists associated with it, and each list can have multiple rules.

Each VCN comes with a default security list. If you don't specify a custom security list for a subnet, the default security list is automatically used with that subnet. You can add and remove rules in the default security list. It has

an initial set of stateful rules, which should be changed to only allow inbound traffic from authorized subnets. The default rules are:

- **Stateful ingress:** Allow TCP traffic on destination port 22 (SSH) from authorized source IP addresses and any source port.

  This rule enables you to create a new cloud network and public subnet, create a Linux instance, and then immediately use SSH to connect to that instance without needing to write any security list rules yourself.

  The default security list doesn't include a rule to allow Remote Desktop Protocol (RDP) access. If you're using Private Cloud Appliance images, add a stateful ingress rule for TCP traffic on destination port 3389 from authorized source IP addresses and any source port.
- **Stateful ingress:** Allow ICMP traffic type 3 code 4 from authorized source IP addresses.

  This rule enables instances to receive Path MTU Discovery fragmentation messages.
- **Stateful ingress:** Allow ICMP traffic type 3 (all codes) from your VCN CIDR block.

  This rule enables instances to receive connectivity error messages from other instances within the VCN.
- **Stateful egress:** Allow all traffic.

  This allows instances to initiate traffic of any kind to any destination. This implies that instances with public IP addresses can talk to any internet IP address if the VCN has a configured internet gateway. And because stateful security rules use connection tracking, the response traffic is automatically allowed regardless of any ingress rules.

The general process for working with security lists is as follows:

1. Create a security list.
2. Add security rules to the security list.
3. Associate the security list with one or more subnets.
4. Create resources, such as compute instances, in the subnet.

   The security rules apply to all the VNICs in that subnet.

When you create a subnet, you must associate at least one security list with it. It can be either the VCN default security list or one or more other security lists that you already created. You can change which security lists the subnet uses at any time. You can add and remove rules in the security list. It's possible for a security list to contain no rules.

For more information, see Controlling Traffic with Security Lists on page 496.

Network Security Groups

On Private Cloud Appliance, a network security group (NSG) provides a virtual firewall for a set of cloud resources, within a single VCN, that all have the same security posture. For example: a group of compute instances that all perform the same tasks and thus all need to use the same set of ports.

Rules in an NSG are enforced on VNICs, but their NSG membership is determined through their parent resources. Not all cloud services support NSGs. Currently, the following types of parent resources support the use of NSGs:

- **Compute instances:** When you create an instance, you can specify one or more NSGs for the instance's primary VNIC. If you add a secondary VNIC to an instance, you can specify one or more NSGs for that VNIC. You can also change the NSG membership of existing VNICs.
- **Load balancers:** When you create a load balancer, you can specify one or more NSGs for the load balancer (not the backend set). You can also update an existing load balancer to use one or more NSGs.
- **Mount targets:** When you create a mount target for a file system, you can specify one or more NSGs. You can also update an existing mount target to use one or more NSGs.

For resource types that don't yet support NSGs, continue to use security lists to control traffic to and from those parent resources.

> **Note:**
>
> You can't associate an Internet Gateway with an NSG.

An NSG contains two types of elements:

- **VNICs:** One or more VNICs; for example, the VNICs attached to the set of compute instances that all have the same security posture. All the VNICs must be in the VCN that the NSG belongs to. A VNIC can be in a maximum of five NSGs.
- **Security rules:** Rules that define the types of traffic allowed into and out of the VNICs in the group. For example: ingress TCP port 22 SSH traffic from a particular source.

The general process for working with NSGs is as follows:

1. Create an NSG.

   When you create an NSG, it is initially empty, without any security rules or VNICs. After the NSG is created, you can add or remove security rules to allow the types of ingress and egress traffic that the VNICs in the group require.
2. Add security rules to the NSG.
3. Add parent resources, or more specifically VNICs, to the NSG.

   When you manage an NSG VNIC membership, you do it as part of working with the parent resource, not the NSG itself. You can do this when you create the parent resource, or you can update the parent resource and add it to one or more NSGs.

   When you create a compute instance and add it to an NSG, the instance's primary VNIC is added to the NSG. You can create secondary VNICs separately, and optionally add them to NSGs.

There are some differences in the REST API model for NSGs compared to security lists:

- With security lists, there is an `IngressSecurityRule` object and a separate `EgressSecurityRule` object. With network security groups, there is only a `SecurityRule` object, and the object's `direction` attribute determines whether the rule is for ingress or egress traffic.
- With security lists, the rules are part of the `SecurityList` object, and you work with the rules by calling the security list operations; for example: `UpdateSecurityList`. With NSGs, the rules aren't part of the `NetworkSecurityGroup` object. Instead you use separate operations to work with the rules for a particular NSG; for example: `UpdateNetworkSecurityGroupSecurityRules`.
- The model for updating existing security rules is different between security lists and NSGs. With NSGs, each rule in a particular group has a unique identifier. When you call `UpdateNetworkSecurityGroupSecurityRules`, you provide the IDs of the specific rules that you want to update. With security lists, the rules have no unique identifier. When you call `UpdateSecurityList`, you must pass in the entire list of rules, including rules that aren't being updated in the call.
- There is a limit of 25 rules when calling the operations to add, remove, or update security rules.

For more information, see Controlling Traffic with Network Security Groups on page 501.

Working with DHCP Options

On Private Cloud Appliance, when you create a subnet, you can specify the set of DHCP options for the subnet. A set of DHCP options is a resource with an OCID. If you don't specify a set of DHCP options, the default set for the VCN is used.

A subnet can only be assigned one set of DHCP options. You can edit a set of DHCP options, create a new set, and change which set is assigned to a subnet. The assigned DHCP option set applies to all the instances in that subnet.

Each subnet in a VCN can have a single set of DHCP options associated with it. That set of options applies to all instances in the subnet. Each VCN comes with a default set of DHCP options with initial values that you can change. Unless you create and assign a different set of DHCP options, every subnet uses the VCN default set.

These are the available DHCP options you can configure:

- **Domain Name Server**

  The default setting is DNS Type = Internet and VCN Resolver.

  If instead you set DNS Type = Custom Resolver, you can specify up to three DNS servers of your choice.
- **Search Domain**

If you set up your VCN with a DNS label, the default value for the Search Domain option is the VCN domain name: `<VCN_DNS_label>.oraclevcn.com`. Otherwise, the Search Domain option isn't present in the default set of DHCP options.

In general, when any set of DHCP options is initially created – the default set or a custom set –, the Networking service automatically adds the Search Domain option and sets it to the VCN domain name (`<VCN_DNS_label>.oraclevcn.com`) if all of these are true:

- The VCN has a DNS label.
- DNS Type = Internet and VCN Resolver.
- You did NOT specify a search domain of your choice during creation of the set of DHCP options.

After the set of DHCP options is created, you can always remove the Search Domain option or set it to a different value. You can specify only a single search domain in a set of DHCP options.

Important notes about your instances and DHCP options:

- Whenever you change the value of one of the DHCP options, you must either restart the DHCP client on the instance, or reboot the instance for the changes to take effect. This applies to all existing instances in the subnets associated with that set of DHCP options.
- Keep the DHCP client running so you can always access the instance. If you stop the DHCP client manually or disable NetworkManager, the instance can't renew its DHCP lease and will become inaccessible when the lease expires; typically within 24 hours. Do not disable NetworkManager unless you use another method to ensure renewal of the lease.
- Stopping the DHCP client might remove the host route table when the lease expires. Also, loss of network connectivity to your iSCSI connections might result in loss of the boot drive.
- Any changes you make to the `/etc/resolv.conf` file are overwritten whenever the DHCP lease is renewed or the instance is rebooted.
- Changes you make to the `/etc/hosts` file are overwritten whenever the DHCP lease is renewed or the instance is rebooted. To persist your changes to the `/etc/hosts` file in Oracle Linux or CentOS instances, add the following line to `/etc/oci-hostname.conf`: `PRESERVE_HOSTINFO=2`. If the `/etc/oci-hostname.conf` file doesn't exist, create it.

Creating a Set of DHCP Options

On Private Cloud Appliance, you can create a set of DHCP options.

For conceptual information, see:

- Name Resolution on page 466
- Working with DHCP Options on page 489

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN.
3. Select the name of the VCN for which you want to create a set of DHCP options.

   The VCN details page is displayed.
4. Under **Resources**, select **DHCP Options**.
5. Select **Create DHCP Options**.
6. Enter the following information:

   - **Name**: A descriptive name for the set of options. The name doesn't have to be unique, and you can change it later.
   - **Create in Compartment:** The compartment where you want to create the set of DHCP options.
   - **DNS Type**: If you want instances in the subnet to resolve internet hostnames and hostnames of instances in the VCN, select Internet and VCN Resolver. To use a DNS server of your choice, select Custom Resolver and then enter the IP address of the DNS server. You can enter up to three DNS server IP addresses.

- **Search Domain**: If you want instances in the subnet to append a particular search domain when resolving DNS queries, enter that domain here. Note that the Networking service automatically sets the search domain option in certain situations.
- **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.

7. Select **Create DHCP Options**.

   You can specify this set of options when creating or updating a subnet.

Using the CLI

Use the oci network dhcp-options create command and required parameters to create a new set of DHCP options for the specified VCN.

```
oci network dhcp-options create --compartment-id <compartment_OCID>  --vcn-
id <vcn_OCID> --options <JSON_formatted_values> [OPTIONS]
```

--options <*JSON_formatted_values*> is a set of DHCP options. This is a complex type whose value must be valid JSON. The value can be provided as a string on the command line or passed in as a file using the `file://path/to/file` syntax.

The --generate-param-json-input option can be used to generate an example of the JSON which must be provided. We recommend storing this example in a file, modifying it as needed and then passing it back in using the `file://` syntax.

Example:

```
'[{"type": "DomainNameServer", "customDnsServers": ["202.44.61.9"],
 "serverType": "CustomDnsServer"}]'
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateDhcpOptions operation to create a new set of DHCP options for the specified VCN.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing VCN DHCP Option Sets

On Private Cloud Appliance, every VCN has a default set of DHCP options. If you create more sets, then you can choose which set to assign to a subnet.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN.
3. Select the name of the VCN for which you want to list DHCP Options sets.
4. Under **Resources**, select **DHCP Options**.

   The list of DHCP options sets is displayed.
5. Select the Actions menu (⋮) for the a particular DHCP set, and select **Edit**.

   The options are displayed.

Using the CLI

Use the oci network dhcp-options list command and required parameters to list DHCP option sets.

```
oci network dhcp-options list --compartment-id ocid1.compartment.unique_ID
 --vcn-id ocid1.vcn.unique_ID [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListDhcpOptions operation to list DHCP option sets.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Updating a Set of DHCP Options

On Private Cloud Appliance, you can change DHCP options.

To update the DHCP options for the instances in a subnet, do one of the following:

- Update the DHCP Options object that's assigned to that subnet as described in this section.
- Update the subnet to assign a different DHCP Options object as described in Editing a Subnet on page 481.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN.
3. Select the name of the VCN for the DHCP options that you want to edit.
4. Under **Resources**, select **DHCP Options**.

   The list of DHCP options sets is displayed.
5. Select the Actions menu (⋮) for the set of options you want to change, then select **Edit**.
6. Change options.
7. Select **Save Changes**.

Using the CLI

Use the oci network dhcp-options update command and required parameters to update the specified set of DHCP options. You can update the display name or the options themselves.

> **Important:**
>
> The options object you provide replaces the entire existing set of options.

```
oci network dhcp-options update --dhcp-id <dhcp_OCID> <options_to_update>
 [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateDhcpOptoins operation to update the specified set of DHCP options. You can update the display name or the options themselves. Avoid entering confidential information.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Set of DHCP Options

On Private Cloud Appliance, you can delete a set of DHCP options as long as the DHCP options are not assigned to any subnet.

To unassign the DHCP options set from a subnet, update the subnet to assign a different set of DHCP options. See Editing a Subnet on page 481. You can't delete a VCN default set of DHCP options.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN.
3. Select the name of the VCN for which you want to delete a DHCP Options set.
4. Under **Resources**, select **DHCP Options**.

5. Select the Actions menu (⋮) for the set you want to delete, then select **Delete**.
6. Confirm the deletion.

Using the CLI

Use the oci network dhcp-options delete command and required parameters to delete the specified set of DHCP options, but only if it's not associated with a subnet.

```
oci network dhcp-options delete --dhcp-id ocid1.dhcpoptions.unique_ID
 [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteDhcpOptions operation to delete the specified set of DHCP options, but only if it's not associated with a subnet.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Working with Route Tables

On Private Cloud Appliance, each VCN automatically comes with an empty default route table. Subnets use the default route table of the parent VCN, unless you explicitly assign them a different route table.

When you add route rules to a VCN, you can add them to the default table if that suits your needs. However, if you need both a public subnet and a private subnet, you instead create a separate custom route table for each subnet.

The primary routing scenario is for sending subnet traffic to destinations outside the VCN. A subnet has a single route table of your choice associated with it at the time of creation. All VNICs in that subnet are subject to the rules in the route table. Traffic within the VCN is automatically handled by the VCN internal routing. No route rules are required to enable that traffic. You can change which route table the subnet uses at any time. You can also edit a route table's rules, or remove all the rules from the table.

A route rule specifies a destination CIDR block and the target, or the next hop, for any traffic that matches that CIDR. When selecting the target you also specify its compartment. Supported target types for a route rule are the different VCN gateways.

Route rules must be configured carefully to ensure that the network traffic reaches the intended destination and isn't dropped. Moving route rules between compartments is not supported.

Creating a Route Table

On Private Cloud Appliance, route rules are required to send traffic outside the VCN. If you don't need to send traffic outside the VCN, you can use the default route table that was created when the VCN was created. The default route table has no rules.

Each route rule specifies a destination CIDR block and the target (the next hop) for any traffic that matches that CIDR. Before you can create a rule, you must create a target.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN in which you want to create a subnet.
3. Select the name of the VCN for which you want to create a route table.

   The VCN details page is displayed.
4. Under **Resources**, select **Route Tables**.
5. Select **Create Route Table**.
6. Enter the required information:

   - **Name:** Specify a user-friendly name for the route table. The name doesn't have to be unique, and you can change it later. Avoid entering confidential information

- **Create in Compartment:** Select the compartment where you want to create the route table. You're not required to create the route table in the same compartment as the VCN.

7. To add a rule, select **+New Rule**, and enter the following information:
   - **Target Type:** Select from the list.
   - **CIDR Block**: The destination CIDR block for the traffic. A value of 0.0.0.0/0 means that all non-intra-VCN traffic that is not already covered by other rules in the route table goes to the target specified in this rule.
   - **Target:** The label for this value is the value that you selected for Target Type. Select the arrow and select the target. You might need to change the compartment just above the arrow.
   - **Description:** An optional description of the rule.
8. **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
9. Select **Create Route Table**.

   The details page of the new route table is displayed. You can specify this route table when creating or updating a subnet.

Using the CLI

Use the oci network route-table create command and required parameters to create a new route table for the specified VCN. In the request you must also include at least one route rule for the new route table.

```
oci network route-table create --compartment-id compartment_OCID --vcn-
id vcn_OCID --route-rules route_rules_json [OPTIONS]
```

The --route-rules option includes the collection of rules used for routing destination IPs to network devices. This is a complex type whose value must be valid JSON. The value can be provided as a string on the command line or passed in as a file using the `file://path/to/file` syntax.

The --generate-param-json-input option can be used to generate an example of the JSON which must be provided. We recommend storing this example in a file, modifying it as needed and then passing it back in through the `file://` syntax.

Example:

```
'[{"cidrBlock":"0.0.0.0/0","networkEntityId":"ocid1.internetgateway.oc1.phx.UniqueID"}]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateRouteTable operation to create a new route table for the specified VCN.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing a Route Table

On Private Cloud Appliance, you can list the route tables in the specified VCN.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN in which you want to create a subnet.
3. Select the name of the VCN for which you want to view route tables.

   The VCN details page is displayed.
4. Under **Resources**, select **Route Tables**.

   The list of route tables is displayed.
5. Select the name of the route table to view its route rules.

Using the CLI

Use the oci network route-table list command and required parameters to list the route tables in the specified VCN and specified compartment. If the VCN ID is not provided, then the list includes the route tables from all VCNs in the specified compartment. The response includes the default route table that automatically comes with each VCN in the specified compartment, plus any route tables you've created.

```
oci network route-table list --compartment-id ocid1.compartment.unique_ID --
vcn-id ocid1.vcn.unique_ID [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListRoutTables operation to list the route tables in the specified VCN.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Updating Route Table Rules

On Private Cloud Appliance, you can change the name of a route table and add, edit, or delete rules in a route table.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN in which you want to create a subnet.
3. Select the name of the VCN for which you want to update the route table.

   The VCN details page is displayed.
4. Under **Resources**, select **Route Tables**.
5. Select the name of the route table that you want to update.
6. Edit the route rules:
   - To change the name of the route table, select **Edit**, change the name in the dialog box, and select **Save Changes**.
   - To create a route rule, select **Add Route Rules** and enter the information.
   - To edit an existing rule, select the Actions menu (⋮) for that rule, and then select **Edit**.
   - To delete a rule, select the Actions menu (⋮) for that rule, and then select **Delete**.

Using the CLI

Use the oci network route-table update command and required parameters to update the specified route table's display name or route rules. Avoid entering confidential information.

```
oci network route-table update --rt-id <route_table_OCID> --route-
rules <options_to_change> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateRouteTable operation to update the specified route table's display name or route rules. Avoid entering confidential information.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Route Table

On Private Cloud Appliance, you can delete a route table as long as it is not associated with a subnet. You can't delete a VCN default route table.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN in which you want to create a subnet.
3. Select the name of the VCN for which you want to delete a route table.

   The VCN details page is displayed.
4. Under **Resources**, select **Route Tables**.
5. Select the Actions menu (⋮) for the route table that you want to delete, then select **Delete**.
6. Confirm the deletion.

Using the CLI

Use the oci network route-table delete command and required parameters to delete the specified route table.

```
oci network route-table delete --rt-id ocid1.routetable.unique_ID [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteRouteTable operation to delete the specified route table.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Controlling Traffic with Security Lists

On Private Cloud Appliance, both security lists and network security groups (NSGs) are types of virtual firewalls for your compute instances. Both security lists and NSGs define network security rules that decide which types of traffic are allowed in and out of instances (VNICs).

Security lists provide virtual firewall rules to all the VNICs in a subnet. To provide a set of firewall rules for a set of VNICs of your choice in a VCN, you can create an NSG. See Controlling Traffic with Network Security Groups on page 501.

Security lists enable you to define network security rules that apply to all VNICs in a subnet. A default security list is automatically created for each VCN. That default security list is assigned to each subnet in the VCN if you don't assign a different security list. Up to five security lists can be associated with a subnet.

If you use both security lists and NSGs, traffic in or out of a given VNIC is allowed if any rule in any applicable security list or NSG allows the traffic:

- Any rule in any security list that's associated with the VNIC subnet
- Any rule in any NSG that the VNIC is in

For more information about security lists, see Security Lists on page 487.

Creating a Security List

On Private Cloud Appliance, you can create a security list for a VCN.

Before you create a security list, view the security rules that are already defined in the default security list and any other security list for this VCN. See Viewing Security Lists on page 499.

A security list must have at least one rule. A security list isn't required to have both ingress and egress rules.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN in which you want to create a subnet.
3. Select the name of the VCN for which you want to create a security list.

   The VCN details page is displayed.

4. Under **Resources**, select **Security Lists**.
5. Select **Create Security List**.
6. In the Create Security List dialog box, enter the following information:
   - **Name:** Provide a descriptive name for the security list. The name doesn't have to be unique. Avoid entering confidential information. (The name can't be changed later in the Compute Web UI but it can be changed with the CLI).
   - **Create in Compartment:** Select the compartment where you want to create the security list.
7. Add at least one rule.

   To add one or more ingress rules, select **+New Rule** in the **Allow Rules for Ingress** box. Enter the following information:

   - **Stateless:** If you want the new rule to be stateless, check this box. By default, security list rules are stateful and apply to both a request and its coordinated response.
   - **CIDR**: The CIDR block for the ingress or egress traffic.
   - **IP Protocol:** The rule can apply to all IP protocols, or choices such as ICMP, TCP, or UDP. Select the protocol from the drop-down list.
     - **Port Range**: For some protocols, such as TCP or UDP, you can supply a source port range and destination port range.
     - **Parameter Type and Code**: For ICMP, you can select a parameter type and corresponding parameter code.
   - **Description:** An optional description of the rule.
8. **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
9. Select **Create Security List**.

   The details page of the new security list is displayed. You can specify this security list when creating or updating a subnet.

Using the CLI

Use the oci network security-list create command and required parameters to create a new security list for the specified VCN.

```
oci network security-list create --compartment-id <compartment_OCID> --vcn-
id <vcn_OCID> --ingress-security-rules <ingress_rules> --egress-security-
rules <egress_rules> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Procedure

1. Gather the information you need to run the command:
   - The OCID of the compartment where you want to create this security list (`oci iam compartment list`)
   - The OCID of the VCN for this security list (`oci network vcn list --compartment-id` ***`compartment_OCID`***)
2. Construct arguments for the `--ingress-security-rules` and `--egress-security-rules` options.

   Security rules are in JSON format. To see how to format a rule, use the following command:

   ```
   oci network security-list create --generate-param-json-input ingress-
   security-rules > ingress.json
   ```

   Use the same command with `egress-security-rules`.

   Ingress and egress security rules are the same except that ingress rules have `source` and `sourceType` properties while egress rules have `destination` and `destinationType` properties.

The value of the `protocol` property is `all` or one of the following numbers: 1 for ICMP, 6 for TCP, or 17 for UDP.

Or, you can `list` or `get` the default security list or another security list and copy the values of the `egress-security-rules` and `ingress-security-rules` properties.

Put the information for rules for this new security list in the appropriate places in the format, or replace the information in the rules that you copied.

The value of both rules options is either a string between single quotation marks or a file specified as `file://`***`path_to_file`***`.json`.

Egress and ingress rules must be in a list. If the list of egress rules or the list of ingress rules has only one item, that single rule must be enclosed in square brackets just as multiple rules would be. See the command in the next step for an example showing only one ingress rule.

Both egress rules and ingress rules must be specified. See the command in the next step for an example showing no egress rules.

3. Run the security list create command.

   Syntax:

   Example:

```
$ oci network security-list create --compartment-id
 ocid1.compartment.unique_ID \
--vcn-id ocid1.vcn.unique_ID --display-name "Limited Port Range" \
--egress-security-rules [] \
--ingress-security-rules '[{"source": "10.0.2.0/24", "protocol": "6",
 "isStateless": true, \
"tcpOptions": {"destinationPortRange": {"max": 1521, "min": 1521}, \
"sourcePortRange": {"max": 1521, "min": 1521}}}]'
{
  "data": {
    "compartment-id": "ocid1.compartment.unique_ID",
    "defined-tags": {},
    "display-name": "Limited Port Range",
    "egress-security-rules": [],
    "freeform-tags": {},
    "id": "ocid1.securitylist.unique_ID",
    "ingress-security-rules": [
      {
        "description": null,
        "icmp-options": null,
        "is-stateless": true,
        "protocol": "6",
        "source": "10.0.2.0/24",
        "source-type": "CIDR_BLOCK",
        "tcp-options": {
          "destination-port-range": {
            "max": 1521,
            "min": 1521
          },
          "source-port-range": {
            "max": 1521,
            "min": 1521
          }
        },
        "udp-options": null
      }
    ],
    "lifecycle-state": "PROVISIONING",
    "time-created": "unique_ID",
    "vcn-id": "ocid1.vcn.unique_ID"
```

```
  },
  "etag": "unique_ID"
}
```

Using the API

Use the CreateSecurityList operation to create a new security list for the specified VCN.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing Security Lists

On Private Cloud Appliance, you can view security lists that are associated with a VCN.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN in which you want to create a subnet.
3. Select the name of the VCN for which you want to view security lists.

   The VCN details page is displayed.
4. Under **Resources**, select **Security Lists**.

   The list of security lists is displayed.
5. Select the name of the security list to view its ingress and egress rules.

Using the CLI

Use the oci network security-list list command and required parameters to list security lists.

If you specify both the VCN OCID and the compartment OCID arguments, all security lists that belong to the specified VCN and are in the specified compartment are listed.

```
oci network security-list list --compartment-id ocid1.compartment.unique_ID
 --vcn-id ocid1.vcn.unique_ID [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListSecurityLists operation to list security lists.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Updating a Security List

On Private Cloud Appliance, you can edit the name of the security list and add, edit, or delete rules or tags in any security list, including the default security list.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN in which you want to create a subnet.
3. Select the name of the VCN for which you want to update a security list.

   The VCN details page is displayed.
4. Under **Resources**, select **Security Lists**.
5. For the security list that you want to update, do one of the following:

   - Select the Actions menu (⋮), then select **Edit** to open the **Edit Security List** dialog box. Update rules in the **Allow Rules for Ingress** and **Allow Rules for Egress** sections.

To delete a rule, select the trash can icon. To add a rule, select **+New Rule**. You can also update the security list name and tags. Select **Save**.

- Select the Actions menu (⋮), then select **View Details** to open the security list details page.
  - Select **Edit** to open the Edit Security List dialog box.
  - To edit only the rules, scroll to the **Resources** section and select either **Ingress Rules** or **Egress Rules**.

    To create a new rule, select **Create Security Rule**. To update a rule, select the Actions menu (⋮) for that rule and then select **Edit**.

    To delete a rule, select the Actions menu and then select **Delete**.

Using the CLI

Use the oci network security-list update command and required parameters to update the specified security list's display name or rules. Avoid entering confidential information.

```
oci network security-list update --security-list-id <security_list_OCID>
 [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateSecurityList operation to update the specified security list's display name or rules.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Security List

On Private Cloud Appliance, you can delete a security list as long as it isn't associated with a subnet. You can't delete a VCN default security list.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN in which you want to create a subnet.
3. Select the name of the VCN for which you want to delete a security list.

   The VCN details page is displayed.
4. Under **Resources**, select **Security Lists**.
5. Select the Actions menu (⋮) for the security list that you want to delete, then **Delete**.
6. Confirm the deletion.

Using the CLI

Use the oci network security-list delete command and required parameters to delete the specified security list, but only if it's not associated with a subnet.

```
oci network security-list delete [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteSecurityList operation to delete the specified security list, but only if it's not associated with a subnet.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Controlling Traffic with Network Security Groups

On Private Cloud Appliance, both network security groups (NSGs) and security lists are types of virtual firewalls for your compute instances. Both NSGs and security lists define network security rules that decide which types of traffic are allowed in and out of instances (VNICs).

NSGs provide virtual firewall rules for a set of VNICs of your choice in a VCN. To provide a set of firewall rules for all VNICs in a subnet, you can create a security list. See Controlling Traffic with Security Lists on page 496.

NSGs enable you to define network security rules for groups of instances, which can be in different subnets. For example, an NSG can apply to all the database servers, or to all the application servers running a certain application. Instead of applying security to a particular subnet, you create an NSG and then add the appropriate instances (VNICs) to the NSG.

When you create a VCN, a default security list is created. No default NSG is created because you must choose which VNICs to include in the group.

If you use both security lists and NSGs, traffic in or out of a particular VNIC is allowed if any rule in any applicable security list or NSG allows the traffic:

- Any rule in any security list that's associated with the VNIC subnet
- Any rule in any NSG that the VNIC is in

For more information, see Network Security Groups on page 488.

Creating a Network Security Group

On Private Cloud Appliance, you can create Network Security Groups (NSGs).

General Process for Working with NSGs

1. Create an NSG as described in this section.
2. Add security rules to the NSG. See Managing Network Security Group Rules on page 502.
3. Add parent resources (or more specifically, VNICs) to the NSG.

   You can do this when you create the parent resource, or you can update the parent resource and add it to one or more NSGs.

   When you create a Compute instance and add it to an NSG, the instance's primary VNIC is added to the NSG. Separately, you can create secondary VNICs and optionally add them to NSGs.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN in which you want to create a subnet.
3. Select the name of the VCN for which you want to create an NSG.

   The VCN details page is displayed.
4. Under **Resources**, select **Network Security Groups**.
5. Select **Create Network Security Group**.
6. In the **Create Network Security Group** dialog box, enter the following information:
   - **Name:** Enter a descriptive name for the NSG. The name doesn't have to be unique, and it can be changed later. Avoid entering confidential information.
   - **Create in Compartment:** Select the compartment where you want to create the NSG.
7. **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
8. Select **Create Network Security Group**.

   The details page for the new NSG is displayed. You can create security rules and select VNICs to add to the group now, or you can do these tasks later. See Configuring VCN Rules and Options on page 483 and Configuring VNICs on page 518.

Using the CLI

Use the oci network nsg create command and required parameters to create a new network security group for the specified VCN.

```
oci network nsg create --compartment-id <compartment_OCID> --vcn-
id <vcn_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateNetworkSecurityGroup operation to create a new network security group for the specified VCN.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing Network Security Groups

On Private Cloud Appliance, NSGs are available for you to create and use.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN in which you want to create a subnet.
3. Select the name of the VCN for which you want to view Network Security Groups.

   The VCN details page is displayed.
4. Under **Resources**, select **Network Security Groups**.

   The list of NSGs is displayed.
5. Select the name of the NSG to view its details, including security rules and attached VNICs.

Using the CLI

Use the oci network nsg list command and required parameters to list either the network security groups in the specified compartment, or those associated with the specified VLAN.

> **Note:**
>
> You must specify either a `vlanId` or a `compartmentId`, but not both. If you specify a `vlanId`, all other parameters are ignored.

```
oci network nsg list --compartment-id <compartment_OCID> | --VLAN-
id <vlan_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListNetworkSecurityGroups operation to list either the network security groups in the specified compartment, or those associated with the specified VLAN.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing Network Security Group Rules

On Private Cloud Appliance, you can add, update, and remove NSG rules.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN in which you want to create a subnet.
3. Select the name of the VCN for which you want to manage rules in an NSG.

The VCN details page is displayed.

4. Under **Resources**, select **Network Security Groups**.
5. In the list of NSGs, select the name of the NSG for which you want to manage rules. The NSG details page is displayed.
6. Under **Resources**, select **Security Rules**.
7. You can add, edit, and delete rules.

   - To add a rule, select **Create Security Rules**. To add one or more ingress rules, select +**New Rule** in the **Allow Rules for Ingress** box. To add one or more egress rules, select **+New Rule** in the **Allow Rules for Egress** box. Enter the following information:
     - **Stateless:** If you want the new rule to be stateless, check this box. By default, security list rules are stateful and apply to both a request and its coordinated response.
     - **CIDR**: The CIDR block for the ingress or egress traffic.
     - **IP Protocol:** The rule can apply to all IP protocols, or choices such as ICMP, TCP, or UDP. Select the protocol from the drop-down list.
       - **Port Range**: For some protocols, such as TCP or UDP, you can supply a source port range and destination port range.
       - **Parameter Type and Code**: For ICMP, you can select a parameter type and corresponding parameter code.
     - **Description:** An optional description of the rule. Avoid entering confidential information.
   - To edit a rule, select the Actions menu (⋮) for the Egress or Ingress rule, select **Edit**, make the necessary changes, and then select **Update**.
   - To delete a rule, select the Actions menu (⋮) for the Egress or Ingress rule, select **Remove**, and then select **Confirm**. While you're editing a rule, select the trash can icon to delete the rule.

Using the CLI

Use these CLI commands to manage NSG rules:

- Use the oci network nsg rules add command and required parameters to add one or more security rules to the specified network security group.

  ```
  oci network nsg rules add [OPTIONS]
  ```

- Use the oci network nsg rules list command and required parameters to list the security rules in the specified network security group.

  ```
  oci network nsg rules list [OPTIONS]
  ```

- Use the oci network nsg rules update command and required parameters to update one or more security rules in the specified network security group.

  ```
  oci network nsg rules update [OPTIONS]
  ```

- Use the oci network nsg rules remove command and required parameters to remove one or more security rules from the specified network security group.

  ```
  oci network nsg rules remove [OPTIONS]
  ```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the AddNetworkSecurityGroupSecurityRules operation to add one or more security rules to the specified network security group.

Use the ListNetworkSecurityGroupSecurityRules operation to list the security rules in the specified network security group.

Use the UpdateNetworkSecurityGroupSecurityRules operation to update one or more security rules in the specified network security group.

Use the RemoveNetworkSecurityGroupSecurityRules operation to remove one or more security rules from the specified network security group.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Attaching VNICs to a Network Security Group

On Private Cloud Appliance, an NSG has one or more VNICs. You can attach a VNIC to an NSG when you create an instance or when you create or update the VNIC.

See the following procedures:

- Creating an Instance on page 649
- Creating and Attaching a Secondary VNIC on page 520
- Updating a VNIC on page 522

Using the Compute Web UI

You can view NSGs and VNICs in different ways:

- To view the list of NSGs that a VNIC is attached to, perform the following steps.
  1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
  2. At the top of the page, select the compartment that contains the instance that has the VNIC.
  3. Select the name of the instance.

     The instance details page is displayed.
  4. Under **Resources**, and select **Attached VNICs**.
  5. Select the name of the VNIC.
  6. Under **Resources**, select **Network Security Groups**.
- To view the list of VNICs that are attached to an NSG, perform the following steps.
  1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
  2. At the top of the page, select the compartment that contains the VNC.
  3. Select the name of the VCN.

     The VCN details page is displayed.
  4. Under **Resources**, select **Network Security Groups**.
  5. In the list, select the name of the NSG.
  6. Under **Resources**, select **VNICs**.
- To change the list of NSGs that a VNIC is attached to, update the VNIC. See Updating a VNIC on page 522.

Using the CLI

Use the oci network vnic get command and required parameters to view the list of NSGs that a VNIC is attached to.

```
oci network vnic get [OPTIONS]
```

Use the oci network nsg list command and required parameters to view the list of VNICs that are attached to an NSG.

```
oci network nsg vnics list [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

To change the list of NSGs that a VNIC is attached to, update the VNIC.

Using the API

Use the GetVnic operation to view the list of NSGs that a VNIC is attached to.

Use the ListNetworkSecurityGroupVnics operation to view the list of VNICs that are attached to an NSG.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Network Security Group

On Private Cloud Appliance, you can delete Network Security Groups (NSGs) as long as the group doesn't contain any VNICs.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN in which you want to create a subnet.
3. Select the name of the VCN for which you want to delete an NSG.

   The VCN details page is displayed.
4. Under **Resources**, select **Network Security Groups**.
5. Select the Actions menu (⋮) for the NSG that you want to delete, then select **Delete**.
6. Confirm the deletion.

Using the CLI

Use the oci network nsg delete command and required parameters to delete the specified network security group.

```
oci network nsg delete [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteNetworkSecurityGroup operation to delete the specified network security group.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Configuring VCN Gateways

Learn about the different types of gateways you can configure on Private Cloud Appliance.

This section describes the role and usage of each gateway type.

Network Address Translation (NAT) Gateways

A NAT gateway gives cloud resources without a public IP access to the on-premises network, which is an external public network from the point of view of a VCN, without exposing those resources. You create a NAT gateway in the context of a specific VCN, so the gateway is automatically attached to that VCN upon creation.

See Enabling Public Connections through a NAT Gateway on page 506.

Internet Gateways

An internet gateway connects the edge of the VCN with the on-premises network. The ultimate target of the traffic routed through an internet gateway can be the internet.

See Providing Public Access through an Internet Gateway on page 507.

Local Peering Gateways

VCN peering is the process of connecting multiple virtual cloud networks (VCNs) so that resources can communicate using private IP addresses. You can use VCN peering to divide your network into multiple VCNs, for example, based on departments or lines of business, with each VCN having direct private access to the others. You can also place shared resources into a single VCN that all the other VCNs can access privately.

See Connecting VCNs through a Local Peering Gateway (LPG) on page 509.

Dynamic Routing Gateways (DRGs)

a dynamic routing gateway, or DRG, provides a path for private network traffic between the VCN and an on-premises network. This traffic is routed to the data center network and on to its destination.

See Creating a Dynamic Routing Gateway on page 511.

Service Gateways

Service Gateway (SG). Some services are isolated on their own network for security and performance reasons. The service gateway (SG) allows a VCN with no external access to privately access Service Network services (such as object storage) in a private subnet.

See Accessing Oracle Services through a Service Gateway on page 513.

Enabling Public Connections through a NAT Gateway

On Private Cloud Appliance, a NAT gateway gives cloud resources without a public IP access to the on-premises network, which is an external public network from the point of view of a VCN, without exposing those resources. You create a NAT gateway in the context of a specific VCN, so the gateway is automatically attached to that VCN upon creation.

A NAT gateway is used to translate IP addresses as traffic passes from one part of an IP network to another. This prevents sources and destinations from having identical IP addresses, and allows RFC 1918 private addresses used in Private Cloud Appliance traffic to communicate with on-premises data center networks.

The gateway allows hosts to initiate connections to the on-premises network and receive responses, but prevents them from receiving inbound connections initiated from the on-premises network. NAT gateways are highly available and support TCP, UDP, and ICMP ping traffic. The Networking service automatically assigns a public IP address to the NAT gateway. You can't choose its public IP address.

When a host in the private network initiates a connection to the on-premises network, the NAT device's public IP address becomes the source IP address for the outbound traffic. The response traffic from the on-premises network therefore uses that public IP address as the destination IP address. The NAT device then routes the response back to the private network, to the host that initiated the connection.

VCN routing is controlled at the subnet level, so you can specify which subnets use a NAT gateway. You can only configure one NAT gateway per VCN.

For the purpose of access control, when creating a NAT gateway, you must specify the compartment where you want the gateway to reside. If you're not sure which compartment to use, put the gateway in the same compartment as the VCN.

By default, a NAT gateway allows traffic at the time of creation. However, you can block or allow traffic through the gateway at any time. Blocking a NAT gateway prevents all traffic, regardless of any existing route rules or security rules in the VCN.

Configuring a NAT Gateway

On Private Cloud Appliance, you can configure NAT gateways for VCNs.

After created, the NAT Gateway is ready for the addition of route rules or security settings. Note that the name of the gateway (`natgatewayUniqueID`) is assigned automatically and not by a parameter, and that the IP address of the device is also assigned automatically.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN where the NAT gateway will be created.
3. Select the name of the VCN.

   The VCN details page is displayed.

4. Under **Resources**, select NAT Gateways.
5. Select **Create NAT Gateway**.
6. Enter the following information:
   - **Name:** Provide a descriptive name for the NAT gateway. Avoid entering confidential information
   - **Create in Compartment:** Select the compartment in which to create the NAT Gateway.
   - **Block Traffic** Choose whether to block traffic to this NAT Gateway.
     - **(Yes: Traffic Not Blocked)**: By default, the VCN uses the NAT gateway even if it's not configured.
     - **(No: Traffic Blocked)**: You can set the NAT gateway not see traffic until it's explicitly enabled to do so.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
7. Select **Create NAT Gateway**.

   The NAT Gateway is now ready for the addition of route rules or security settings. See Configuring VCN Rules and Options on page 483.

Using the CLI

Use the oci network nat-gateway create command and required parameters to create a new NAT gateway for the specified VCN. You must also set up a route rule with the NAT gateway as the rule's target.

```
oci network nat-gateway create [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateNatGateway operation to create a new NAT gateway for the specified VCN.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Providing Public Access through an Internet Gateway

On Private Cloud Appliance, an internet gateway connects the edge of the VCN with the on-premises network. The ultimate target of the traffic routed through an internet gateway can be the internet.

However, on Private Cloud Appliance, the internet gateway routes traffic to the on-premises network. Traffic to and from the internet is then managed by the routing configuration in the on-premises network.

You create an internet gateway in the context of a specific VCN, so the gateway is automatically attached to that VCN upon creation. To use the gateway, the hosts on both ends of the connection must have public IP addresses for routing. Connections that originate in the VCN and are destined for a public IP address, either inside or outside the VCN, go through the internet gateway. Connections that originate outside the VCN and are destined for a public IP address inside the VCN also go through the internet gateway.

A VCN can have only one internet gateway. You control which public subnets in the VCN can use the gateway by configuring the subnet associated route table. The public subnet security list rules determine the specific types of traffic that are allowed to and from the resources in the subnet. An internet gateway can be disabled, meaning no traffic flows to or from the internet, regardless of any existing route rules that enable such traffic.

For the purpose of access control, when creating an internet gateway, you must specify the compartment where you want the gateway to reside. If you're not sure which compartment to use, put the gateway in the same compartment as the VCN.

To configure an Internet gateway, see Configuring an Internet Gateway on page 507.

Configuring an Internet Gateway

On Private Cloud Appliance, you can configure an Internet Gateway (IGW) which provides the VCN with outside access through the on-premises data center network.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN where the internet gateway will be created.
3. Select the name of the VCN.

   The VCN details page is displayed.
4. Under **Resources**, select **Internet Gateways**.
5. Select **Create Internet Gateway**.
6. Enter the following information:
   - **Name:** Provide a descriptive name for the internet gateway. Avoid entering confidential information
   - **Create in Compartment:** Select the compartment in which to create the Internet Gateway.
   - **Enabled:** Use the toggle to decide if the gateway is enabled at creation or not. The default is to enable the gateway:
     - **Yes: Gateway Enabled**: By default, the VCN uses the gateway when created.
     - **(No: Gateway Disabled)**: You can set the gateway not see traffic until it's explicitly enabled to do so.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
7. Select **Create Internet Gateway**.

   The Internet Gateway is now ready for the addition of route rules or security settings. See Configuring VCN Rules and Options on page 483.

Using the CLI

Use the oci network internet-gateway create command and required parameters to create a new internet gateway for the specified VCN.

```
oci network internet-gateway create [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateInternetGateway operation to create a new internet gateway for the specified VCN.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Enable or Disable an Internet Gateway

On Private Cloud Appliance, you can enable or disable an Internet gateway.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN that has the internet gateway.
3. Select the name of the VCN.

   The VCN details page is displayed.
4. Under **Resources**, select **Internet Gateways**.
5. Select the Actions menu (⋮), then select **Edit**.
6. Set the **Enabled** toggle:
   - Slide the toggle right to enable the internet gateway.
   - Slide the toggle left to disable the internet gateway.
7. Select **Update**.

Using the CLI

Use the oci network internet-gateway update command and required parameters to update the specified internet gateway. You can disable/enable it, or change its display name or tags.

```
oci network internet-gateway update [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateInternetGateway command and required parameters to the specified internet gateway. You can disable/enable it, or change its display name or tags.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting an Internet Gateway

On Private Cloud Appliance, If you have previously configured an IGW, you can delete it.

The internet gateway doesn't have to be disabled, but there must not be a route table that lists it as a target.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN that has the internet gateway.
3. Select the name of the VCN.

   The VCN details page is displayed.
4. Under **Resources**, select **Internet Gateways**.
5. Select the Actions menu (⋮), then select **Delete**.
6. Confirm the deletion.

Using the CLI

Use the oci network internet-gateway delete command and required parameters to delete the specified internet gateway.

```
oci network internet-gateway delete [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteInternetGateway operation to delete the specified internet gateway.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Connecting VCNs through a Local Peering Gateway (LPG)

On Private Cloud Appliance, You can configure local peering gateways. VCN peering is the process of connecting multiple virtual cloud networks (VCNs) so that resources can communicate using private IP addresses.

You can use VCN peering to divide your network into multiple VCNs, for example, based on departments or lines of business, with each VCN having direct private access to the others. You can also place shared resources into a single VCN that all the other VCNs can access privately. Two peered VCNs can be in the same tenancy or different ones.

Policies

Peering between two VCNs requires explicit agreement from both parties in the form of IAM policies that each party implements for their own VCN compartment or tenancy. If the VCNs are in different tenancies, each administrator must provide their tenancy OCID and put in place special coordinated policy statements to enable the peering.

To implement the IAM policies required for peering, the two VCN administrators must designate one administrator as the requester and the other as the acceptor. The requester must be the one to initiate the request to connect the two LPGs. In turn, the acceptor must create a particular IAM policy that gives the requester permission to connect to LPGs in the acceptor's compartment. Without that policy, the requester's request to connect fails. Either VCN administrator can delete a peering connection by deleting their LPG.

Routing and Traffic Control

As part of configuring the VCNs, each administrator must update the VCN routing to enable traffic to flow between the VCNs. In practice, this looks just like routing you set up for any gateway, such as an internet gateway or dynamic routing gateway. For each subnet that needs to communicate with the other VCN, you update the subnet route table. The route rule specifies the destination traffic's CIDR and your LPG as the target. Your LPG routes traffic that matches that rule to the other LPG, which in turn routes the traffic to the next hop in the other VCN.

You can control packet flow over the peering connection with route tables in your VCN. For example, you can restrict traffic to only specific subnets in the other VCN. Without deleting the peering, you can stop traffic flow to the other VCN by removing route rules that direct traffic from your VCN to the other VCN. You can also effectively stop the traffic by removing any security list rules that enable ingress or egress traffic with the other VCN. This does not stop traffic flowing over the peering connection, but stops it at the VNIC level.

Security Rules

Each VCN administrator must ensure that all outbound and inbound traffic with the other VCN is intended, expected and well-defined. In practice, this means implementing security list rules that explicitly state the types of traffic your VCN can send to the other and accept from the other. If your subnets use the default security list, there are two rules allowing SSH and ICMP ingress traffic from anywhere, thus also the other VCN. Evaluate these rules and whether you want to keep or update them.

In addition to security lists and firewalls, evaluate other OS-based configuration on the instances in your VCN. There could be default configurations that do not apply to your own VCN CIDR, but inadvertently apply to the other VCN CIDR.

Connecting VCNs through a Local Peering Gateway

On Private Cloud Appliance, a Local Peering Gateway (LPG) is a way to connect VCNs so that elements in each VCN can communicate, even using private IP address.

The following components are required to set up a peering connection:

- Two VCNs with nonoverlapping CIDRs
- A local peering gateway (LPG) on each VCN in the peering relationship
- A connection between the two LPGs
- Route rules to enable traffic over the peering connection to and from the desired subnets in the respective VCNs
- Security rules to control the types of traffic allowed to and from the instances in the subnets in question

Using the Compute Web UI

1. In the Compute Web UI navigation menu, under **Networking**, select **Virtual Cloud Networks**.

   A list of previously configured VCNs in compartments is displayed. If the compartment you're creating the local peering gateway in isn't displayed, use the drop-down menu to select the correct compartment.
2. Select the name of the VCN.
3. Under the **Resources** menu, select **Local Peering Gateways**.
4. Select **Create Local Peering Gateway**.
5. Enter the required information:

   - **Name:** Enter a name. Avoid entering confidential information.
   - **Create in Compartment:** Select the compartment in which to create the local peering gateway.
   - **Route Table Association (Optional)** Optionally, you can associate a route table with the local peering gateway. A list of configured route tables for the selected compartment is in a drop-down menu. You can change the compartment by selecting **(change)** next to the compartment name.

- **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.

6. Select **Create Local Peering Gateway**.

   The Local Peering Gateway is now ready for connecting VCNs with Establish Peering Connection, and ready for the addition of route rules or security settings.

Using the CLI

Use the oci network local-peering-gateway create command and required parameters to create a new local peering gateway (LPG) for the specified VCN.

```
oci network local-peering-gateway create --compartment-id <compartment_OCID>
 --vcn-id <vcn_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateLocalPeeringGateway operation create a new local peering gateway (LPG) for the specified VCN.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Connecting to the On-Premises Network through a Dynamic Routing Gateway (DRG)

On Private Cloud Appliance, a dynamic routing gateway, or DRG, provides a path for private network traffic between the VCN and an on-premises network. This traffic is routed to the data center network and on to its destination.

The data center network is considered a public network because it's external to the Private Cloud Appliance environment. However, no form of tunneling is required because traffic doesn't travel over the internet. This is a significant difference with public cloud environments.

For the purpose of access control, when creating a DRG, you must specify the compartment where you want the DRG to reside. If you aren't sure which compartment to use, put the DRG in the same compartment as the VCN.

A DRG is a standalone object. To use it, you must attach it to a VCN. In the API, the process of attaching creates a DRG Attachment object with its own OCID. To detach the DRG, you delete the attachment object.

A VCN can be attached to only one DRG at a time, though a DRG can be attached to multiple VCNs. You can detach a DRG and reattach it at any time. After attaching a DRG, you must update the routing in the VCN to use the DRG. Otherwise, traffic from the VCN will not flow to the DRG.

The basic routing scenario sends traffic from a subnet in the VCN to the DRG. When sending traffic from the subnet to your on-premises network, you set up a rule in the subnet route table. The rule's destination CIDR is the CIDR for the on-premises network or a subnet within, and the rule's target is the DRG.

Creating a Dynamic Routing Gateway

On Private Cloud Appliance, a DRG is the equivalent of a general purpose router. A DRG is used to connect a VCN to the data center's IP address space. The router is configured separately from the VCNs, at the compartment level and is not required to be in the same compartment as the VCN (but it typically is).

After the DRG is configured, the DRG can be attached to more than one VCN and, similar to a physical router, can be attached and detached at any time, although perhaps with traffic loss. Also like a physical router, even when attached to a VCN, the DRG must have route table rules to steer traffic to the on-premises data center network's IP address space.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, under **Networking**, select **Dynamic Routing Gateway**s (DRGs).
2. Select **Create Dynamic Routing Gateway**.

3. Fill in the required information:
   - **Name:** Provide a name or description for the dynamic routing gateway. Avoid using any of the organization's confidential information.
   - **Create in Compartment:** Select the compartment in which to create the dynamic routing Gateway.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
4. Select **Create Dynamic Routing Gateway**.

   The Dynamic Routing Gateway is now ready for the addition of DRG attachments. See Attaching VCNs to a Dynamic Routing Gateway on page 512.

Using the CLI

Use the oci network drg create command and required parameters to create a dynamic routing gateway.

```
oci network drg create [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateDrg operation to create a dynamic routing gateway.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Attaching VCNs to a Dynamic Routing Gateway

On Private Cloud Appliance, you can connect many VCNs to a DRG, but each VCN can have only one DRG attached. Ensure that the route tables and security lists allow communication.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, under **Networking**, select **Dynamic Routing Gateway**s (DRGs).
2. At the top of the page, select the compartment that contains the DRG.
3. Select the name of the DRG that you plan to attach to a VCN.
4. Select **Attach to Virtual Cloud Network**.
5. From the list of VCNs in the drop-down list, select the VCN.
6. Select the VCN to attach the DRG to from the list of VCNs in the drop down list.
7. Select the correct compartment.
8. Select **Attach to DRG**.

   The DRG is attached to the selected VCN.
9. Repeat the process to attach the other VCNs to the DRG.

Using the CLI

Use the oci network drg-attachment create command and required parameters to attach the specified VCN to a DRG.

```
oci network drg-attachment create [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateDrgAttachment operation to attach the specified VCN to a DRG.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Accessing Oracle Services through a Service Gateway

On Private Cloud Appliance, a VCN can have only one service gateway. You create a service gateway in the context of a specific VCN, so the gateway is automatically attached to that VCN upon creation. A service gateway allows traffic to and from all subnets at the time of creation. There's no mechanism to block or disable this traffic.

Certain services, such as the Object Storage service, are exposed internally over a conceptual *Services Network*. Normally, these services would use public IP addresses that can be reached over a public network – or in a public cloud model, over the internet. Instead, the purpose of a service gateway is to enable a VCN to privately access the services in the Services Network, meaning resources in a private subnet, within a tightened down VCN with no external access, can still connect to those services.

In Private Cloud Appliance, these services are implemented at the infrastructure level through the management node cluster. Technically, the service endpoints, which are fully qualified domain names, are reachable from the entire on-premises network by default. This means the service gateway has no real function. Specifically for private cloud use, there's no need to configure a service gateway and associated route rules to enable private access to the service endpoints. However, to maintain compatibility with Oracle Cloud Infrastructure, the concept of a service gateway does exist.

The service gateway uses a service CIDR label, which is a string that represents the endpoints for the service or group of services of interest. This means that you don't have to know the specific endpoints. If the service's endpoint changes in the future, you don't need to make any adjustments. You use the service CIDR label when you configure the service gateway. You can create the service gateway and configure route rules involving a *"Service CIDR Block"*. However, remember they serve no purpose other than compatibility.

Security rules aren't implemented for service gateways.

Creating a Service Gateway

On Private Cloud Appliance, you can create one service gateway per VCN. The service gateway is automatically attached to the VCN it's created in. Services use CIDR labels, and are allowed by default.

For each enabled Service, you need a route rule with the Service object’s CIDR block as the rule destination and the service gateway as the rule target.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. At the top of the page, select the compartment that contains the VCN for which you want to create a Service Gateway.
3. Select the name of the VCN.
4. Under **Resources**, select **Service Gateways**.

   The number of configured Service Gateways in parentheses must be zero (0).
5. Select **Create Service Gateway**.
6. Fill in the required information:

   - **Name:** Provide a name or description for the Service Gateway.

     Avoid entering confidential information.
   - **Create in Compartment:** Select the compartment in which to create the Service Gateway.
   - **Services:** Select the service from the list.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
7. Select **Create Service Gateway**.

   The Service Gateway is now ready for the addition of route rules or security settings.

Using the CLI

Use the oci network service-gateway create command and required parameters to create a new service gateway in the specified compartment.

```
oci network service-gateway create [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateServiceGateway operation to create a new service gateway in the specified compartment.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Network Gateway Example: Internet Gateway

VCNs are the basic networking unit of Private Cloud Appliance, and can communicate with other processes through various types of gateways used for a particular purpose.

In this example, access to a VCN from outside the rack is established through an Internet gateway (IGW). All steps to produce a working IGW are detailed.

This example sets up an IGW to allow access to web servers on a public IP subnet running inside an instance. It also adds an ingress rule to the default security list to allow outside access to the web servers on the public subnet. Then this example allows ingress connections for HTTPS connections on TCP port 443, the standard port for HTTP encrypted traffic.

Without this ingress rule, inbound HTTPS connections aren't allowed. Make the new rule `stateful`, which is the default and allows a reply to an HTTPS request without creating an explicit rule for responses.

Next, this example adds the existing IGW target to the route rules. The route rules are added to the default route table for the VCN, or a new route table created for reaching the IGW specifically. The route rule uses CIDR block 0.0.0.0/0. This means that all traffic not already covered by other rules in the route table goes to the IGW target specified in this new rule.

This example includes steps to enable or disable the IGW, and how to delete it.

Overview

There are three major operations for establishing and using an IGW. Each step has its own set of prerequisites and can usually be configured with the Compute Web UI or the OCI CLI.

The three activities used to configure and operate an IGW are:

- Set up an IGW
- Create or update a route table to include a rule for an IGW
- Update the Security List or NSG

Internet Gateway Setup

To configure an internet gateway on Private Cloud Appliance, start by setting up the IGW.

There are certain considerations that need to be assessed before setting up and using an IGW:

- There are public subnets in the VCN that need internet access. (Only public subnets can use the IGW successfully.)
- Because the default condition is to deny access, the types of ingress and egress internet traffic that are allowed must be determined. These include ingress HTTPS connections, ingress ICMP pings, or other types of traffic. The IGW primarily responds to ingress network protocol requests.

To set up an IGW, see Providing Public Access through an Internet Gateway on page 507.

After the IGW is created, there are two additional steps needed to make the IGW work properly with route tables and security list for the VCN or instance port Network Security Groups (NSGs). First, route table entries for the subnet

must be configured to direct authorized traffic to the proper gateway destination. Second, the VCN containing to IGW must have the correct security rules to prevent unauthorized access and yet allow users to access resources they need.

Establish the Route Table Entries

For each public subnet that uses the internet gateway, you must update the subnet route table entry to include a route to the IGW.

Each route rule specifies a destination CIDR block and the target (the next hop) for any traffic that matches that CIDR. Before you can create a rule, you must create a target for the rule, in this case, an IGW.

This example adds the existing IGW target to the route rules of the default route table for the VCN. You can also create a new route table for reaching the IGW specifically, but that's not shown in this example. The route rule uses CIDR block 0.0.0.0/0 so that all traffic not covered by other rules in the route table go to the IGW target specified in the new rule.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. If needed, change the compartment to find the resource you want.
3. Select the name of the VCN for which you want to create a route table. The VCN details page is displayed.
4. Under **Resources**, select **Route Tables**.
5. Go to the details page of the Default Route Table and select **Add Route Rule**.
6. Select **+New Rule**, and enter the following information for this example:
   - **Target Type:** Select `Internet Gateway` from the list.
   - **CIDR Block**: Enter 0.0.0.0/0 as the destination CIDR block for the traffic.
   - **Target:** The target is the IGW. Select the arrow and select the target IGW. You might need to change the compartment just above the arrow.
   - **Description:** An optional description of the rule, such as "New rule for IGW."
7. Select **Create Route Table Rule**.

   The details page of the edited default route table is displayed. Because the subnet was set up to use the default route table, the resources in the subnet can now use the internet gateway.

Using the CLI

Use the oci network route-table update command and required parameters to update the route rules.

> **Note:**
>
> The `routeRules` object you provide replaces the entire existing set of rules.

```
oci network route-table update --rt-id <route-table_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Procedure

1. Gather the information you need to run the command:
   - The OCID of the compartment where you want to create this route table (`oci iam compartment list`)
   - The OCID of the VCN for this route table (`oci network vcn list --compartment-id compartment_OCID`)
2. Construct an argument for the `--route-rules` option.

   Route rules are specified in JSON format. To see how to format a rule, use the following command:

```
oci network route-table update --generate-param-json-input route-rules >
 route_rule_format.json
```

Example (put the following content into the `IGW_route_rule.json` file):

```
[
  {
    "cidr-block": "0.0.0.0/0",
    "description": null,
    "destination": null,
    "destination-type": "CIDR_BLOCK",
    "network-entity-id": "ocid1.internetgateway.unique_ID"
  }
]
```

3. Run the route table update command.

   Syntax:

```
oci network route-table update --compartment-id compartment_OCID \
--vcn-id vcn_OCID --route-rules file:///home/flast/IGW_route_rule.json
```

Using the API

Use the UpdateRouteTable operation to update the specified route table's display name or route rules. Avoid entering confidential information.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Establish the Internet Gateway Security Rules

After the IGW is created, the correct security setting must be established to prevent unauthorized access to the gateway. For example, all outside HTTPS access should only be allowed to access port 443 which is the default port of secure web page access. Without this explicit rule, the standard port isn't reachable.

This section uses security lists to accomplish this goal, but a similar result can be achieved using security rules in a Network Security Group (NSG).

For more information about security lists and NSGs, see Controlling Traffic with Security Lists on page 496 and Controlling Traffic with Network Security Groups on page 501.

> **Important:**
>
> If you have configured the public subnet to use the default security list, remember that the default includes several rules to enable basic access, such as ingress SSH and egress access to all destinations. We recommend that you become familiar with this basic access set of rules. If you don't use the default security list, ensure that basic access is still provided either in the customized security rules or in an NSG containing those modified rules.

This example adds an ingress rule to the default security list to allow ingress connections for HTTPS connections on TCP port 443, the standard port for HTTP encrypted traffic.

Without this ingress rule, inbound HTTPS connections aren't allowed. Make the new rule `stateful`, which allows a reply to an HTTPS request without creating an explicit rule for responses.

For information about creating a new security list instead of modifying the default or adding a rule to an existing security list, see Creating a Security List on page 496.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. If needed, change the compartment to find the resource you want.
3. Select the name of the VCN for which you want to add the rule to a security list. The VCN details page is displayed.
4. Under **Resources**, select **Security Lists**.

5. Update or add new rules as follows:
   - For the security list that you want to add the rule, select the Actions menu (⋮), then select **Edit**. Make changes as described below.
   - To add a new rule, in the **Allow Rules** sections, select **+New Rule**. You can also update the security list name and tags. Make changes as described below.
6. Select **Save Changes**.

For the HTTPS example using TCP port 443 ingress rule, enter the following information:

- **Stateless:** To allow for a response to the incoming HTTPS request, the new rule should be stateful. Ensure that the stateless box is unchecked. For more information about stateless and stateful rules, see Controlling Traffic with Security Lists on page 496.
- **CIDR**: The CIDR block for the example 0.0.0.0/0, which applies the rule to all IP source addresses.
- **IP Protocol:** Select the TCP protocol from the drop-down list.
- **Port Range**:
  - Source Port Range: Leave blank.
  - Destination Port Range: Enter 443.
- **Description:** An optional description of the rule, such as "Allow stateful traffic for HTTPS on TCP port 443."

You can always edit the new rule at any time.

Using the CLI

Use the oci network security-list update command and required parameters to update the specified security list's display name or rules. Avoid entering confidential information.

> **Note:**
>
> The `routeRules` object you provide replaces the entire existing set of rules.

```
oci network security-list update --security-list-id <security_list_OCID>
 [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Procedure

1. Get the OCID of the default security list of the VCN that you want to update (`oci network vcn list --compartment-id compartment_OCID`)
2. To update rules, construct arguments for the `--ingress-security-rules` and `--egress-security-rules` options as described in Creating a Security List on page 496. Arguments that you provide to these rules options overwrite any existing rules. If you want to keep some existing rules, use the following command to show the current rules, and then copy the rules that you want to keep into the new option arguments.

   ```
   oci network security-list get --security-list-id
    ocid1.securitylist.unique_ID
   ```

   Example (put the following content in the file `IGW_ingress_rule.json`):

   ```
   [
     {
       "description": null,
       "icmp-options": null,
       "is-stateless": false,
       "protocol": "6",
       "source": "0.0.0.0/0",
       "source-type": "CIDR_BLOCK",
       "tcp-options": {
   ```

```
      "destination-port-range": {
        "max": 22,
        "min": 22
      },
      "source-port-range": null
    },
    "udp-options": null
  },
  {
    "description": null,
    "icmp-options": null,
    "is-stateless": false,
    "protocol": "6",
    "source": "0.0.0.0/0",
    "source-type": "CIDR_BLOCK",
    "tcp-options": {
      "destination-port-range": {
        "max": 443,
        "min": 443
      },
      "source-port-range": null
    },
    "udp-options": null
  }
]
```

3. Run the security list update command to add the rule for HTTPS and TCP port 443 traffic.

   Example:

```
oci network security-list update  --security-list-id
 ocid1.securitylist.unique_ID --ingress-security-rules file:///home/flast/
IGW_ingress_rule.json

WARNING: Updates to defined-tags and egress-security-rules and freeform-
tags and
 ingress-security-rules will replace any existing values.
 Are you sure you want to continue? [y/N]: y
```

Using the API

Use the UpdateSecurityList operation to update the specified security list's display name or rules.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Configuring VNICs

On Private Cloud Appliance, the compute nodes have physical network interface cards (NICs). When you create a compute instance, the Networking service ensures that a VNIC is created on top of a physical interface, so that the instance can communicate over the network.

Each instance has a primary VNIC that's automatically created and attached. The primary VNIC resides in the subnet you specify when creating the instance. It can't be removed from the instance.

A VNIC enables an instance to connect to a VCN and determines how the instance communicates with endpoints inside and outside of the VCN. Each VNIC resides in a subnet within a VCN and includes these items:

- One primary private IPv4 address from the subnet the VNIC is in.
- Up to 31 optional private IPv4 addresses from the subnet the VNIC is in.
- An optional public IPv4 address for each private IP, assigned at your discretion.
- An optional host name for DNS for each private IP address.
- A MAC address.

- A flag to enable or disable the source/destination check on the VNIC network traffic.
- Optional membership of one or more network security groups (NSGs).
- An Oracle-assigned identifier (OCID).
- An optional friendly name you can choose and assign.

You can add secondary VNICs to an instance after the instance is created. To be able to use a secondary VNIC, you must also configure the instance OS for it. The maximum number of VNICs for an instance varies by shape. Each secondary VNIC can be in a different subnet than the primary VNIC, either within the same VCN or a different one. A secondary VNIC can also be in the same subnet as the primary VNIC. However, attaching multiple VNICs from the same subnet CIDR block to an instance can introduce asymmetric routing, especially on Linux instances.

> **Note:**
>
> To avoid asymmetric routing in a configuration with multiple IP addresses from the same subnet, Oracle recommends assigning multiple private IP addresses to one VNIC, or using policy-based routing.
>
> If traffic comes in to a service on the instance through a secondary VNIC and the service replies, the reply packets automatically have that VNIC IP address as the source IP address. Policy-based routing is required for that reply to go back out on the same interface and find the correct default gateway.

Secondary VNICs must always be attached to an instance and can't be moved. The process of creating a secondary VNIC automatically attaches it to the instance. The process of detaching a secondary VNIC automatically deletes it. They're automatically detached and deleted when you delete the instance. An instance's bandwidth is fixed regardless of the number of VNICs attached. You can't specify a bandwidth limit for a particular VNIC on an instance.

By default, every VNIC performs the source/destination check on its network traffic. The VNIC looks at the source and destination listed in the header of each network packet. If the VNIC isn't the source or destination, then the packet is dropped. If the VNIC needs to forward traffic – for example, if it needs to perform Network Address Translation (NAT) – you must disable the source/destination check on the VNIC.

VNICs reside in a subnet but attach to an instance. The VNIC attachment to the instance is a separate object from the VNIC or the instance itself. The VNIC and subnet always exist together in the same compartment, but the VNIC attachment to the instance always exists in the instance's compartment. This distinction affects your IAM policies if you set up an access control scenario where network administrators manage the network and other users manage instances.

VNICs are rate limited to prevent very active applications from reducing the available bandwidth of other applications to unacceptable levels. Rate limiting applies to each instance’s VNIC interface.

The rate is limited to a value below the maximum network bandwidth associated with each instance shape. Generally, the bandwidth limit increases with the maximum number of VNICs a shape uses, but there is no maximum bandwidth guarantee.

You can't specify the bandwidth limits directly. Rate limiting is a system function. The rate applied to each VNIC is based on the shape option during instance creation.

Viewing VNICs and VNIC Attachments

On Private Cloud Appliance, you can view VNIC attachments and for a particular instance and view VNIC details.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. At the top of the page, select the compartment that contains the instance in which you want to view VNIC attachments.
3. Select the name of the instance.

   The instance details page is displayed.
4. Under **Resources**, select **Attached VNICs**.

   The list of attached VNICs for that instance is displayed.

5. To view the details page for the VNIC, select the name of the attached VNIC.

Using the CLI

Use the oci compute vnic-attachment list command and required parameters to list the VNIC attachments in the specified compartment. A VNIC attachment resides in the same compartment as the attached instance.

```
oci compute vnic-attachment list [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListVnicAttachments operation to list the VNIC attachments in the specified compartment. A VNIC attachment resides in the same compartment as the attached instance.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Creating and Attaching a Secondary VNIC

On Private Cloud Appliance, you can add more VNICs to an instance.

The number of secondary VNICs that you can add to an instance depends on the shape of the instance. See Compute Shapes.

After you attach a secondary VNIC, log onto the instance to configure the instance OS to use the new interface. See Configuring the Instance OS for a Secondary VNIC on page 521.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. At the top of the page, select the compartment that contains the instance to which you want to add a secondary VNIC.
3. Select the name of the instance.
4. Under **Resources**, select **Attached VNICs**.

   The primary VNIC and any secondary VNICs attached to the instance are displayed.
5. Select **Create VNIC Attachment**.
6. In the **Subnet** section of the **Create VNIC Attachment** dialog box, specify the subnet to use for the VNIC. You might need to select a different compartment to find the VCN and subnet that you want.

   > **Note:**
   >
   > Specifying the same subnet for this VNIC that's specified for another VNIC for this instance can introduce asymmetric routing.
   >
   > Instead of creating a VNIC in the same subnet as an existing VNIC for this instance, consider creating a secondary private IP address for the existing VNIC that's in this subnet.
7. Specify whether to disable source/destination checks.

   By default, a VNIC looks at the source and destination listed in the header of each network packet. If the VNIC isn't the source or destination, then the packet is dropped.

   If the VNIC needs to forward traffic (for example, if the VNIC needs to perform Network Address Translation), check the box to disable this source/destination check.
8. If you selected a public subnet, you can specify whether to automatically assign a public IPv4 address object to the VNIC private IP address object.
9. (Optional) Specify the following private IP information.

- **Private IP Address**: An address that's within the CIDR block range assigned to the subnet and not already in use. If you don't enter an address, an IP address is automatically assigned.
- **Hostname**: A hostname to be used for DNS within the cloud network. This option is available only if the VCN and subnet both have DNS labels. The hostname can be up to 63 letters, numbers, and hyphens. No spaces are allowed.

10. (Optional) Add this VNIC to an NSG.

    By default, the new VNIC isn't attached to any NSG. Check the box labeled **Enable Network Security Groups** to add this VNIC to one or more NSGs.

    a. Select an NSG from the drop-down list. You might need to change the compartment to find the NSG you want.
    b. Select **Add Another NSG** to attach to another NSG.
    c. To remove an NSG from the list, select the trash can to the right of that NSG. To remove the last NSG or all NSGs, clear the **Enable Network Security Groups** box.

    See Controlling Traffic with Network Security Groups on page 501 for information about NSGs.

11. Select **Create Attachment**.

    The secondary VNIC is created and then displayed on the **Attached VNICs** list for the instance.

12. Configure the instance OS to use the secondary VNIC. See Configuring the Instance OS for a Secondary VNIC on page 521.

Using the CLI

Use the oci compute instance attach-vnics command and required parameters to create a secondary VNIC and attaches it to the specified instance. You can specify either –subnet-id or –vlan-id for this create request, but not both.

```
oci compute instance attach-vnic [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the AttachVnic operation to create a secondary VNIC and attaches it to the specified instance.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Configuring the Instance OS for a Secondary VNIC

On Private Cloud Appliance, after you create a secondary VNIC, log in to the instance OS and configure the OS to use the new VNIC.

For Oracle Linux and Microsoft Windows, you can use utilities and scripts that are provided by Oracle. The scripts use information from the instance metadata. See Retrieving Instance Metadata from Within the Instance on page 657 for instructions about how to view that data on the instance OS, including how to show the VNIC data.

For other OSs, consult the OS documentation.

Oracle Linux Instance OS Configuration

Use the oci-network-config utility to perform the OS configuration required for secondary VNICs on instances that run Oracle Linux.

Microsoft Windows Instance OS Configuration

You must configure the secondary VNIC within the OS. You can configure the secondary VNIC manually in the Microsoft Windows OS, or create a PowerShell script.

Ensure that the script includes these actions:

- Check if the network interface has an IP address and a default route.

- Enable the OS to recognize the secondary VNIC, the **script must overwrite the IP address and default route with static settings (which effectively disables DHCP).** The script should prompt you with a choice: to overwrite with the static settings, or exit.

Overall Process to Add a Secondary VNIC

1. Create and attach a secondary VNIC to the instance. Keep the VNIC OCID handy so you can provide it later when you run the configuration script.
2. Connect to the instance with Remote Desktop.
3. Run your script as an administrator.

Oracle Solaris Instance OS Configuration

Use the `ipadm` command to configure network interfaces persistently.

More Information about Configuring OSs for VNICs

See this OCI page: Virtual Network Interface Cards (VNICs).

Updating a VNIC

On Private Cloud Appliance, you can update the VNIC name, the host name, and whether to disable source/destination checks. You can add the VNIC to an NSG and remove the VNIC from an NSG.

Avoid entering confidential information in names.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. At the top of the page, select the compartment that contains the instance for which you want to update a VNIC.
3. Select the name of the instance.

   The instance details page is displayed.
4. Under **Resources**, select **Attached VNICs**.

   The list of attached VNICs for that instance is displayed.
5. For the VNIC that you want to update, select the Actions menu (⋮) and then select **Edit**.
6. In the **Update VNIC** dialog box, update the VNIC name, the host name, whether to disable source/destination checks, or whether to attach this VNIC to an NSG or detach this VNIC from an NSG.

   See Creating and Attaching a Secondary VNIC on page 520 for information about the Skip Source/Destination Check selection.

   If you change the Enable Network Security Groups box from cleared to selected, then you must select an NSG from the drop-down list. You might need to change the compartment to find the NSG you want.

   If the Enable Network Security Groups box is already checked, then you can select Add Another NSG to attach to another NSG.

   If more than one NSG is already listed, you can select the trash can next to an existing NSG to detach this VNIC from that NSG. To detach the last NSG or all NSGs, clear the Enable Network Security Groups box.

   See Controlling Traffic with Network Security Groups on page 501 for information about NSGs.
7. Select **Update VNIC**.

Using the CLI

Use the oci network vnic update command and required parameters to update the specified VNIC.

```
oci network vnic update [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateVnic operation to update the specified VNIC.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Secondary VNIC

On Private Cloud Appliance, you can delete a secondary VNIC.

This operation detaches and deletes the specified secondary VNIC.

You can't delete an instance's primary VNIC.

When you delete an instance, all attached VNICs (primary and secondary) are automatically detached and deleted.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. At the top of the page, select the compartment that contains the instance for which you want to delete a VNIC.
3. Select the name of the instance.

   The instance details page is displayed.
4. Under **Resources**, select **Attached VNICs**.

   The list of attached VNICs for that instance is displayed.
5. For the VNIC that you want to delete, select the Actions menu (⋮), then select **Delete**.
6. Select **Confirm**.

   The VNIC state changes to **Detached**. After a few seconds, the VNIC is removed from the list.
7. Log onto the instance and delete the configuration for the IP address from the instance OS.

   Undo the configuration you did when you added the VNIC. See Configuring the Instance OS for a Secondary IP Address on page 526.

Using the CLI

Use the oci compute instance detach-vnic command and required parameters to detach and delete the specified secondary VNIC.

```
oci compute instance detach-vnic [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DetachVnic operation to detach and delete the specified secondary VNIC.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing Private IP Addresses

On Private Cloud Appliance, a private IP address enables communication with resources on the VCN.

Each instance has at least one private IP address, and optionally, one or more public IP addresses.

The Networking service defines a private IP object, identified by an OCID, which consists of a private IPv4 address and optional host name for DNS. Each instance receives a primary private IP object during creation. The Networking service uses DHCP to assign a private IP address. This address doesn't change during the instance's lifetime and can't be removed from the instance. The private IP object is deleted when the instance is deleted.

If an instance has any secondary VNICs attached, each of those VNICs also has a primary private IP. A private IP can have a public IP assigned to it at your discretion. You can't use a private IP as the target of a route rule in a VCN.

Secondary Private IPs

After an instance is created, you can add a *secondary private IP* to one of its VNICs. You can add it to either the primary VNIC or a secondary VNIC on the instance. The secondary private IP address must be in the subnet to which the VNIC belongs. You can move a secondary private IP from a VNIC on one instance to a VNIC on another instance on condition that both VNICs belong to the same subnet.

Typical use cases for a secondary private IP include:

- **Instance failover:** You assign a secondary private IP to an instance. If a problem occurs with the instance, you can easily reassign its secondary private IP to a standby instance in the same subnet. In addition, if the secondary private IP has an associated public IP, that public IP moves along with the private IP.
- **Running multiple services or endpoints on a single instance:** For example, you could have an instance that runs multiple container pods, where each pod uses its own IP address from the VCN CIDR. The containers have direct connectivity to other instances and services in the VCN. Another example: you could run multiple SSL web servers with each one using its own IP address.

Secondary private IP addresses have the following properties:

- They're supported for all shapes and OS types.
- They can be assigned only after the instance is created, or the secondary VNIC is created and attached.
- They're automatically deleted when you delete the instance or detach and delete the secondary VNIC.
- Deleting a secondary private IP from a VNIC returns the address to the pool of available addresses in the subnet.
- A VNIC can have a maximum of 32 secondary private IPs.
- The instance's bandwidth is fixed regardless of the number of private IP addresses attached. You can't specify a bandwidth limit for a particular IP address on an instance.
- A secondary private IP can have a reserved public IP associated with it at your discretion.

> **Note:**
>
> After assigning a secondary private IP to a VNIC, you must configure the instance OS to use it. See Configuring the Instance OS for a Secondary IP Address on page 526

Viewing Private IP Addresses

On Private Cloud Appliance, you can see the private IP addresses for instances.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. At the top of the page, select the compartment that contains the instance for which you want to view IP address information.
3. Select the name of the instance.

   The instance details page is displayed.
4. View networking information or VNIC information.

   - Select the **Networking** tab. The primary private IP address and any attached public IP address are shown in the Instance Access column.
   - Under **Resources**, select **Attached VNICs**. Select the name of the VNIC for which you want to view IP addresses.

     On the VNIC details page, Under **Resources**, select **IP Addresses**. The primary private IP address and any secondary private IP addresses, and any attached public IP addresses, are shown in the table.

Using the CLI

Use the oci network private-ip list command and required parameters to list private IP address objects.

```
oci network private-ip list [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListPrivateIps operation to list private IP address objects.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Assigning a Secondary Private IP Address

On Private Cloud Appliance, when you create an instance, the instance automatically gets a VNIC, and the VNIC automatically gets a primary private IP address. You can add secondary private IP addresses to a VNIC. A VNIC can have up to 33 private IP addresses: one primary private IP address, and up to 32 secondary private IP addresses.

Creating a VNIC in the same subnet as another VNIC for the same instance can introduce asymmetric routing as described in Configuring VNICs on page 518. Instead, you can create a secondary private IP address for the existing VNIC that's in the subnet that you want.

See information about secondary private IP addresses, including use cases, in Managing Private IP Addresses on page 523.

After you perform the following procedure to assign a secondary private IP address, log onto the instance to configure the instance OS to use the new IP address. See Configuring the Instance OS for a Secondary IP Address on page 526.

Avoid entering confidential information in names and tags.

Moving a Secondary IP Address

In addition to adding a secondary private IP address, you can use this procedure to reassign (move) a currently assigned secondary private IP address to a different VNIC. Because the VNIC must be in the same subnet as the VNIC where the secondary private IP address is currently assigned, the new VNIC probably is attached to a different instance; as previously mentioned, having two VNICs in the same subnet in the same instance can introduce asymmetric routing.

To move a secondary private IP address, see the Unassign if assigned or `--unassign-if-already-assigned` options in the following procedures.

You can't move a VNIC primary private IP address.

If a public IP address object is assigned to a secondary private IP address object, and you move that secondary private IP address object to another VNIC, the public IP address object moves with it.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. At the top of the page, select the compartment that contains the instance for which you want to add a secondary private IP address.
3. Select the name of the instance to which you want to add a secondary private IP address.

   The instance details page is displayed.
4. Under **Resources**, select **Attached VNICs**.

   The primary VNIC and any secondary VNICs attached to the instance are displayed.
5. Select the name of the attached VNIC to which you want to add a secondary private IP address.
6. On the VNIC details page, under **Resources**, select **IP Addresses**.
7. Select **Assign Secondary Private IP Address**.
8. In the **Attach Private IP** dialog box, all input fields are optional.
   - **IP Address:** If you don't enter an address, an IP address from the subnet CIDR is automatically assigned.

If you enter an address, the IP address must be within the CIDR block for the subnet. You can enter a secondary private IP address that's already assigned to another VNIC in the subnet. You can't enter a primary private IP address.

If you enter an IP address that's already assigned, see the following option.

- **Unassign if assigned:** In the previous option, if you entered a secondary private IP address that's already assigned, select this item to *move* that private IP address. The address will be unassigned from the VNIC where it's currently assigned and reassigned to this VNIC.

  If you entered an IP address that's already assigned and you don't select this item, this secondary private IP assignment operation fails.
- **Hostname:** Enter the hostname to be used for DNS within the cloud network. This option is available only if the VCN and subnet both have DNS labels.

9. Select **Attach IP Address**.

   The new secondary private IP address is shown in the table.
10. Configure the new secondary private IP address in the instance. See Configuring the Instance OS for a Secondary IP Address on page 526.

Using the CLI

Use the network vnic assign-private-ip command and required parameters to assign a secondary private IP address to the specified VNIC. The secondary private IP must be in the same subnet as the VNIC.

```
oci network vnic assign-private-ip --vnic-id <vnic_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the PrivateIp operation to assign a secondary private IP address to the specified VNIC. The secondary private IP must be in the same subnet as the VNIC.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Configuring the Instance OS for a Secondary IP Address

On Private Cloud Appliance, after you create a secondary private IP address on a VNIC, sign in to the instance to configure the instance OS to use the new IP address.

Linux Instance OS Configuration

This configuration permits use of an IP address subnet, netmask, gateway, and DNS service that are entirely independent from the existing NIC. This configuration is persistent across reboots.

Create a new network interface configuration file to create a subinterface on the existing NIC. In this example, `ens03` is the name of the existing NIC and `ifcfg-ens3:0` is the name of the new configuration file.

1. Create the network configuration file `ifcfg-ens3:0` in the `/etc/sysconfig/network-scripts/` directory to create the first sub-interface (`:0`) on the existing `ens3` NIC.

   Include the following entries in `ifcfg-ens3:0`:

   ```
   TYPE=Ethernet
   BOOTPROTO=none
   IPADDR=a.b.c.d
   PREFIX=24
   GATEWAY=
   DNS=
   NAME=ens3:0
   DEVICE=ens3:0
   ```

2. Include the appropriate `IPADDR`, `PREFIX`, `GATEWAY`, and `DNS` entries for this new sub-interface.
3. Run the following command to start the new interface:

```
# ifup ens3:0
```

4. Run the following command to confirm that the new interface is operational:

```
# ifconfig -a
```

See also Linux: Details about Secondary IP Addresses.

Oracle Solaris Instance OS Configuration

Use the `ipadm` command to configure network interfaces persistently.

Microsoft Windows Instance OS Configuration

See Windows: Details about Secondary IP Addresses for information about how to either:

- Create a PowerShell script.
- Use the Network and Sharing Center user interface.

Updating a Secondary Private IP Address

On Private Cloud Appliance, you can update the hostname and change the IP address of a secondary private IP address object. You can't update a VNIC primary private IP address.

To change the hostname, use the Compute Web UI, CLI, or API as described in this section.

To change the IP address, delete the secondary IP address. Then create a new secondary IP address. See:

- Deleting a Secondary Private IP Address on page 528
- Assigning a Secondary Private IP Address on page 525

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. At the top of the page, select the compartment that contains the instance that has the secondary private IP address object that you want to update.
3. Select the name of the instance.

   The instance details page is displayed.
4. On the instance details page, under **Resources**, select **Attached VNICs**.

   The primary VNIC and any secondary VNICs attached to the instance are displayed.
5. Select the name of the attached VNIC that has the secondary private IP address object that you want to update.
6. On the VNIC details page, under **Resources**, select **IP Addresses**.
7. For the secondary private IP address object that you want to update, select the Actions menu (⋮), then select **Edit**.
8. In the **Attach Private IP** dialog box, update the host name.
9. Select **Attach IP Address**.

Using the CLI

Use the oci network private-ip update command and required parameters to change the display name or the hostname for a secondary private IP.

```
oci network private-ip update [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdatePrivateIp operation to change the display name or the hostname for a secondary private IP.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Secondary Private IP Address

On Private Cloud Appliance, you can delete a secondary IP address. You can't delete a VNIC primary private IP address.

On successful delete, the private IP address is returned to the pool of available addresses in the subnet. Any attached public IP address become available for assignment.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. At the top of the page, select the compartment that contains the instance for which you want to delete a secondary private IP address.
3. Select the name of the instance.

   The instance details page is displayed.
4. Under **Resources**, select **Attached VNICs**.

   The primary VNIC and any secondary VNICs attached to the instance are displayed.
5. Select the name of the attached VNIC for which you want to delete a secondary private IP address.
6. On the VNIC details page, under **Resources**, select **IP Addresses**.
7. For the secondary private IP address that you want to delete, select the Actions menu (⋮) and select **Delete**.

   Confirm the deletion.
8. Log onto the instance and delete the configuration for the IP address from the instance OS.

   Undo the configuration you did when you added the IP address. See Configuring the Instance OS for a Secondary VNIC on page 521.

Using the CLI

Use the network vnic unassign-private-ip command and required parameters to unassign a secondary private IP address from a VNIC.

```
oci network vnic unassign-private-ip [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Refer to the PrivateIp Reference for information about unassigning a secondary private IP address from a VNIC.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing Public IP Addresses

On Private Cloud Appliance, a public IP address enables communication outside the VCN, including to the data center network.

All of the following network configurations are required for an instance to communicate outside the VCN:

- The instance must be in a public subnet, which is configured when the subnet is created. Private subnets can't have a public IP address assigned to instances in the subnet.
- The instance must have a public IP address.
- The instance's VCN must have an internet gateway configured.
- The public subnet must have route table and security list entries that enable communications outside the VCN.

When you create an instance with a public IP address, the instance is assigned a public IP address from an address pool. Technically, the assignment is to a private IP object on the instance, and the VNIC that the private IP is assigned to must reside in a public subnet. An instance can have multiple secondary VNICs; each with a (primary) private IP. Consequently, you can assign an instance multiple public IPs across their VNICs.

The Networking service defines a public IP object, identified by an OCID, which consists of a private IPv4 address and additional properties that further define the public IP type and behavior. There are two types of public IPs:

- An **ephemeral** public IP is temporary and exists for the lifetime of the instance.
- A **reserved** public IP is persistent and exists beyond the lifetime of the instance it is assigned to. It can be unassigned and then reassigned to another instance.

The following table summarizes the differences between both types of IP addresses:

Characteristics	**Ephemeral Public IP**	**Reserved Public IP**
Allowed assignment	to a VNIC primary private IP limits: one per VNIC, two per instance	to a VNIC primary private IP limit: one per VNIC
Creation	Optionally created and assigned during instance creation or secondary VNIC creation. You can create and assign one later if the VNIC doesn't already have one.	You create one at any time. You can then assign it when you like.
Unassignment	You can unassign it at any time, which deletes it. You might do this if whoever created the instance included a public IP, but you do not want the instance to have one. When you stop an instance, its ephemeral public IPs remain assigned to the instance.	You can unassign it at any time, which returns it to the tenancy's pool of reserved public IPs.
Moving to a different resource	You can't move an ephemeral public IP to a different private IP.	You can move it at any time by unassigning and then reassigning it to another private IP. Can be in a different VCN or availability domain.
Automatic deletion	Its lifetime is tied to the private IP lifetime. Automatically unassigned and deleted when: • its private IP is deleted • its VNIC is detached or deleted • its instance is deleted	Never. Exists until you delete it.
Scope	Availability domain	Regional (can be assigned to a private IP in any availability domain in the region)
Compartment and availability domain	Same as the private IPs	Can be different from the private IPs

When you create an instance in a public subnet, the instance gets a public IP by default, unless you specify otherwise. After you create a public IP, you can't change which type it is. For example, if you create an instance that is assigned an ephemeral public IP with address 203.0.113.2, you can't convert the ephemeral public IP to a reserved public IP with address 203.0.113.2.

Resources that are designed to be directly publicly reachable automatically get a public IP address assigned from a pool upon creation. For NAT gateways, the assigned address is a regional ephemeral public IP. As with other

ephemeral public IPs, it is automatically unassigned and deleted when you delete its assigned resource. However, unlike other ephemeral public IPs, you can't edit it or unassign it yourself.

Viewing Public IP Addresses

On Private Cloud Appliance, you can use the Compute Web UI, CLI, and API to view the IP addresses assigned to instances.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. At the top of the page, select the compartment that contains the instance for which you want to view the private IP address.
3. Select the name of the instance.

   The instance details page is displayed.
4. View networking information or VNIC information:

   - Select the **Networking** tab. The primary private IP address and any attached public IP address are shown in the Instance Access column.
   - Under **Resources**, select **Attached VNICs**. Select the name of the VNIC for which you want to view IP addresses.

     On the VNIC details page, under **Resources**, select **IP Addresses**. The primary private IP address and any secondary private IP addresses, and any attached public IP addresses, are shown in the table.

Using the CLI

Use the oci network public-ip list command and required parameters to list the PublicIp objects in the specified compartment.

```
oci network public-ip list [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListPublicIps operation to list the PublicIp objects in the specified compartment.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Assigning an Ephemeral Public IP Address to an Instance

On Private Cloud Appliance, you assign a public IP address to an instance by assigning the public IP address object to a private IP address object.

An ephemeral public IP address is created and assigned in the same step.

An ephemeral public IP address can only be assigned to a primary private IP address: The value of the `is-primary` property of the private IP address object must be `true`. Every VNIC has one primary private IP address.

For secondary private IP addresses (the value of the `is-primary` property of the private IP address object is `false`), assign reserved public IP addresses.

An ephemeral public IP address can't be unassigned and can't be moved to a different private IP address.

An ephemeral public IP address object is deleted in the following cases:

- Its private IP address object is deleted.
- Its VNIC is detached or deleted.
- Its instance is deleted.

Avoid entering confidential information in names.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. At the top of the page, select the compartment that contains the instance for which you want to assign a public IP address.
3. Select the name of the instance.

   The instance detail page is displayed.
4. Under **Resources**, select **Attached VNICs**.
5. Select the name of the VNIC for which you want to assign a public IP address.
6. On the VNIC details page, under **Resources**, select **IP Addresses**. The primary private IP address and any secondary private IP addresses, and as any attached public IP addresses, are shown in the table.
7. If the primary private IP address doesn't already have a public IP address assigned, select the Actions menu (⋮) for the primary private IP address, and then select **Edit Public IP**.
8. In the dialog box, select **Ephemeral Public IP**.
9. (Optional) Give the ephemeral public IP address a name.
10. Select **Reserve Public IP**.

    You might need to refresh the page to see the new public IP address.

    The new public IP address is displayed in the **IP Addresses** table in **Resources**, in the **Primary IP Information** column for the VNIC, and in the **Instance Access** column of the **Networking** tab of the instance.

Using the CLI

Use the oci network public-ip create command and required parameters to create a public IP. Use the *lifetime* property to specify whether it's an ephemeral or reserved public IP.

```
oci network public-ip create [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreatePublicIp operation to create a public IP.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Reserving a Public IP Address

On Private Cloud Appliance, you can reserve a public IP address that's available to assign to a private IP address object at a later time.

There are two types of public IPs:

- **Ephemeral:** Think of it as temporary and existing for the lifetime of the instance.
- **Reserved:** Think of it as persistent and existing beyond the lifetime of the instance it's assigned to. You can unassign it and then reassign it to another instance whenever you like.

For more information, see Public IP Addresses.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Reserved Public IPs**.
2. Select **Reserve Public IP Address**.
3. Enter the following information:

   - **Reserved Public IP Address Name:** (Required) Enter a name for the reserved address. Avoid entering confidential information.
   - **Create in Compartment:** Select the compartment in which to reserve the address.
   - **IP Address Source:** Select the source from which to reserve the address.

4. Select **Reserve Public IP**.

   The new reserved public IP is created and displayed on the page. You can assign it to an instance if you like. See Assigning a Reserved Public IP Address to an Instance on page 532.

Using the CLI

Use the oci network public-ip create command with the `--lifetime` option and required parameters to define how the public IP handled when it is deleted.

```
oci network public-ip create --lifetime EPHEMERAL|RESERVED [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreatePublicIp operation to define how the public IP handled when it is deleted.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Assigning a Reserved Public IP Address to an Instance

On Private Cloud Appliance, you can assign a public IP address to an instance, you assign the public IP address object to a private IP address object.

An ephemeral public IP address can be assigned only to the primary private IP address of a VNIC. See Assigning an Ephemeral Public IP Address to an Instance on page 530. A reserved public IP address can be assigned to any private IP address.

See Reserving a Public IP Address on page 531 to create a reserved public IP address that is available to assign to a private IP address object at a later time.

Use the procedures in Updating a Public IP Address on page 532 to assign an existing reserved public IP address object to the specified private IP address object or to create and assign a reserved public IP address in one step.

A reserved public IP address object remains available for reassignment when its private IP address object is deleted, its VNIC is detached or terminated, or its instance is terminated.

Avoid entering confidential information in names.

Updating a Public IP Address

On Private Cloud Appliance, you can manage public IPs.

You can use the public IP update command to do any of the following actions:

- Assign an existing reserved public IP address object to a private IP address object.
- Create and assign a reserved public IP address object in one step.
- Move a reserved public IP address object to a different private IP address object.
- Unassign a reserved public IP address object from a private IP address object.
- Change the display name or tags for a public IP address object.

Avoid entering confidential information in names.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. At the top of the page, select the compartment that contains the instance for which you want to update a public IP address.
3. Select the name of the instance.

   The instance details page is displayed.
4. Under **Resources**, select **Attached VNICs**.

   Select the name of the VNIC for which you want to assign a public IP address.

5. On the VNIC details page, under **Resources**, select **IP Addresses**.

   The primary private IP address and any secondary private IP addresses, and any attached public IP addresses, are shown in the table.

6. For the private IP address for which you want to add or update a public IP address, select the Actions menu (⋮), then select **Edit Public IP**.
7. In the **Reserve Public IP** dialog box, select one of the following choices:

   - **No public IP**

     Select Reserve Public IP in the dialog to unassign this public IP address from this private IP address. You might have to refresh the page to see that the public IP address is no longer assigned.
   - **Reserve public IP**

     Select one of the following choices:

     - **Reserve existing public IP**

       a. Select an existing public IP address. You might need to change the compartment.
       b. Select **Reserve Public IP** in the dialog.

          If the specified public IP address object is already assigned to a different private IP address object, the public IP address object will be unassigned (moved) from the current private IP address object and reassigned to the specified private IP address object.
     - **Create public IP**

       Create and assign a reserved public IP address in one step.

       a. (Optional) Provide a name for the new reserved public IP address object.
       b. Select the compartment where the new reserved public IP address object will be created.
       c. Select the **IP Address Source**.
       d. Select **Reserve Public IP**.

   The new reserved public IP address shows in the **IP Addresses** table in **Resources**. You might need to refresh the page to see the new public IP address.

Using the CLI

Use the oci network public-ip update command and required parameters to update the specified public IP.

```
oci network public-ip update [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdatePublicIp operation to update the specified public IP.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Public IP Address

On Private Cloud Appliance, in certain circumstances, you can delete public IP addresses.

An ephemeral public IP address object can't be unassigned and can't be directly deleted. An ephemeral public IP address object is deleted in the following cases:

- Its private IP address object is deleted.
- Its VNIC is detached or deleted.
- Its instance is deleted.

A reserved public IP address object is unassigned but remains available for reassignment when its private IP address object is deleted, its VNIC is detached or deleted, or its instance is deleted.

Use the procedures in this section to delete a reserved public IP address object.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the CLI

Use the oci network private-ip delete command and required parameters to unassign and delete the specified public IP.

```
oci network public-ip delete [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeletePublicIp operation to unassign and delete the specified public IP.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Enabling IPv6 Virtual Networking

Compute instances on Private Cloud Appliance can be configured for connectivity with the on-premises network using IPv6 addresses. This is a dual-stack implementation, meaning IPv6 is supported as an optional addition in parallel with IPv4.

> **Important:**
>
> IPv6 virtual network connectivity is currently only supported for compute instances with SR-IOV (single root I/O virtualization) VNICs. See Configuring SR-IOV for Virtual Networking on page 535.

Prerequisites

The system must meet these prerequisites before you can enable IPv6 connectivity on compute instances.

- The option to enable IPv6 traffic is introduced in software version 3.0.2-b1483396 (September 2025). If the active controller software is older, the appliance must be upgraded first.
- The appliance Network Environment must be reconfigured to route ingress and egress traffic between the system and the on-premises network. Configuration changes on the uplinks from the spine switches are required.
- The VCN and related networking resources must be configured to route traffic to and from compute instances with IPv6 VNICs. Instructions are provided further in this topic.

Configuring IPv6 on Virtual Network Resources

IPv6 addresses can optionally be added to both new and existing compute instances and network resources.

> **Note:**
>
> All IPv6 addresses are globally unique. Therefore, NAT is not used, and there is no concept of public and private IPs.
>
> In an SR-IOV VCN, the default route table allows all IPv4 and IPv6 traffic, and the default security list allows all egress and ingress traffic. The respective route rules and security rules cannot be updated or deleted.

1. Enable IPv6 for the VCN. Only one CIDR (IPv4 plus IPv6) is allowed. Bring Your Own IP address space isn't supported. The address range must be at least /64 with a maximum of /7.
   a. To create a new VCN, follow the instructions in Creating a VCN on page 479. In addition, enable IPv6 by selecting Enable IPv6 Protocol. Enter the IPv6 Prefix for this VCN.
   b. To enable IPv6 for an existing VCN, navigate to the VCN detail page.

In the top-right corner, select Controls, then select Add IPv6 CIDR. In the dialog box, enter the IPv6 Prefix for this VCN and confirm.

The VCN now has both an IPv4 and IPv6 address range.

2. Enable IPv6 for the subnets that require it.

   a. To create a new subnet, follow the instructions in Creating a Subnet on page 480. In addition, enable IPv6 by selecting the VCN's IPv6 Prefix and entering a valid IPv6 CIDR Block for this subnet.
   b. To enable IPv6 for an existing subnet, navigate to the subnet detail page. In the top-right corner, select Controls, then select Add IPv6 CIDR. In the dialog box, select the VCN's IPv6 Prefix, enter a valid IPv6 CIDR Block for this subnet, and select the Add button confirm.

   The subnet CIDR is always a /64 within the VCN address range. The first and last 4 addresses are reserved and cannot be assigned to compute instances.

3. If you plan to use a Dynamic Routing Gateway (DRG) with a new IPv6 enabled VCN, remember to create the SR-IOV compatible DRG and attach the VCN to it. See Connecting to the On-Premises Network through a Dynamic Routing Gateway (DRG) on page 511.

   If you are enabling IPv6 for an existing VCN that is already attached to a DRG, no additional configuration steps are required. The DRG supports both IPv4 and IPv6 traffic when it's enabled for the VCN.

   Ensure that route rules exist for both IPv4 and IPv6 traffic.

4. Assign IPv6 addresses to your compute instances.

   Compute instances are first created with a primary IPv4 address. An optional IPv6 addresses is assigned when creating and attaching a secondary VNIC to an SR-IOV subnet.

   a. Navigate to the compute instance detail page. Under Resources, select Attached VNICs.
   b. Select Create VNIC Attachment. Enter a name and select the VCN and subnet to attach the VNIC to.
   c. Under IPv6, select Assign IPv6 Address. Select the available IPv6 Prefix and select either automatic (default) or manual address assignment.
   d. Click Create Attachment to assign an address from the subnet IPv6 range to this VNIC. The compute instance is now attached to the subnet through this VNIC.

5. Configure the VNICs inside the compute instances.

   SR-IOV VNICs are set up in pairs and must form a bond port. Run the `configure_vfio.py` script provided in the Oracle Systems blog Automating SR-IOV/VFIO bond creation on Oracle Compute Cloud@Customer and Private Cloud Appliance. The script configures the bond interface (`vfio0`) and assigns the correct IPv4 and IPv6 addresses.

Configuring SR-IOV for Virtual Networking

On Private Cloud Appliance, single root I/O virtualization (SR-IOV) technology enables instances to achieve low latency and high throughput simultaneously on 1 or more physical links. This technology is ideal for low-latency workloads such as video streaming, real-time applications, and large or clustered databases. Hardware-assisted (SR-IOV) networking uses the VFIO driver framework.

VCNs, DRGs, and instances must be configured and enabled for SR-IOV.

You configure a VCN to support SR-IOV by adding the `networkType`: `VFIO` tag in the OraclePCA defined tag namespace. You can't add this tag to an existing VCN. This tag can't be changed or removed from the VCN. The only way to remove the tag is to delete the VCN.

DRGs compatible with SR-IOV support also use the `networkType`: `VFIO` tag in the OraclePCA defined tag namespace. You can't add this tag to an existing DRG. This tag can't be changed or removed from the DRG. The only way to remove the tag is to delete the DRG. Only SR-IOV DRGs can attach to SR-IOV VCNs.

Linux instances natively support SR-IOV, however, you must configure a secondary VNIC on your instance, in addition to the primary network interface, as a path to the physical NIC. Only secondary vNICs can be used for SR-IOV connectivity. The vNIC type is determined by the VCN's `networkType` tag. You can use the script available in the Oracle blog *Automating SR-IOV/VFIO bond creation on PCA/C3* available at https://blogs.oracle.com/oracle-

systems/post/automating-sriovvfio-bond-creation-on-oracle-compute-cloudcustomer-and-private-cloud-appliance to automate creation of the network bond interfaces taking into account the variability in hardware type and SR-IOV/VFIO type VNIC count.

> **Important:**
>
> Instances configured with SR-IOV networking are non-migratable instances. These types of instances can't be live migrated. Private Cloud Appliance infrastructure upgrades can't migrate SR-IOV instances. You must manually shut down these types of instance before an upgrade. If you have SR-IOV instances, inform Oracle before Private Cloud Appliance infrastructure is upgraded. See Creating a Support Request. To access support, sign in to the Oracle Cloud Console as described in Sign In to the OCI Console.

You can't create these VCN components in an SR-IOV VCN:

- Internet Gateways
- NAT Gateways
- Local Peering Gateways
- Service Gateways
- Security Lists. You can't add new entries to a default security list belonging to an SR-IOV VCN. By default, the SR-IOV VCN has open ingress and egress, with just 1 rule each.
- DHCP Options
- Network Security Groups
- Route Tables. You can only add a default route with the target as an SR-IOV DRG in the default route table of an SR-IOV VCN.
- You can't create the following objects using an SR-IOV VCN/subnet: Load Balancer, Network Load Balancer, Mount Targets, OKE clusters.

Guidelines and Limitations

- Private Cloud Appliance supports up to 84 Virtual Functions (VFs) per compute node.
- VFIO vNICs are limited to 6 per instance, assuming 1 paravirtualized vNIC.

> **Note:**
>
> Each paravirtualized vNIC requires a single PCIe slot, while each VFIO vNIC required 4 PCIe slots. Attaching multiple paravirtualized vNICs may reduce the number of slots available for VFIO vNICs. and vis-a-versa. As mentioned previously, each instance is required to have one paravirtualized vNIC attached.

- IPv4 and IPv6 SR-IOV vNICs are not supported on Suse Linux.
- GPU Shapes are limited to 5 SR-IOV vNICS,1 primary paravirtualized vNIC, 3 secondary vNICs.

Configure SR-IOV Networking

1. Ensure you have the OraclePCA.networkType tag defined on the system. See Using Resource Tags to Extend Functionality (IAM on the Appliance) on page 460.

   Setting the OraclePCA.networkType:VFIO tag enables SR-IOV functionality.

   > **Note:**
   >
   > When you update a VCN or DRG that has the OraclePCA.networkType:VFIO tag applied, that tag can't be changed or removed from the VCN or DRG. If you want this VCN or DRG to no longer be configured for SR-IOV, then delete the VCN or DRG and create new ones that don't have the OraclePCA.networkType:VFIO tag set.

2. Create a VCN with SR-IOV functionality enabled.

Create a VCN. See Creating a VCN on page 479. In the Tagging section, add the OraclePCA.networkType tag with the value VFIO.

You must create a VCN with SR-IOV support enabled, you can't convert an existing VCN to include SR-IOV functionality.

3. If you plan to use a DRG in your SR-IOV configuration, you must create a DRG with SR-IOV functionality. Only SR-IOV DRGs can attach to SR-IOV VCNs.

   a. Create a DRG as described in Creating a Dynamic Routing Gateway on page 511. In the Tagging section, add the OraclePCA.networkType tag with the value VFIO.

      You must create a DRG with SR-IOV support enabled: the OraclePCA.networkType tag applied with value VFIO. You can't add SR-IOV functionality to an existing DRG.

   b. Attach the SR-IOVs VCNs to the DRG as described in Attaching VCNs to a Dynamic Routing Gateway on page 512.

4. Prepare an instance for SR-IOV functionality.

   a. Create and launch an instance. See Creating an Instance on page 649.

   b. Create and attach a secondary VNIC to the instance to use as the SR-IOV network interface. The primary VNIC of the instance can't be the SR-IOV VNIC. See Creating and Attaching a Secondary VNIC on page 520.

   c. Configure the network bond interfaces, including the secondary IP address on an SR-IOV bond port, using the `configure_vfio` script provided in the Oracle Systems blog Automating SR-IOV/VFIO bond creation on Oracle Compute Cloud@Customer and Private Cloud Appliance.

Managing Public DNS Zones

On Private Cloud Appliance, the Domain Name System (DNS) lets computers use hostnames instead of IP addresses to communicate with each other.

In its most basic form, DNS returns an IP address (if known) when given a string in the DNS namespace for that zone. However, DNS is also the way that an IP host client application knows where to get its own configuration information using DHCP (DHCID records), go to send or receive email (MX records), and more. Without DNS, client devices would have to know the proper IP addresses not only for local servers, but for every server or application they interacted with, no matter where in the world they were located. With DNS, clients can always find the correct location of www.oracle.com or any other application.

After you create a DNS zone inside a compartment, you can't move the zone to another compartment.

When creating a DNS zone, you specify the name of the domain it manages – for example: *example.com*. You select whether the zone is primary or secondary. A primary zone contains its own DNS records, while a secondary zone retrieves its records from another zone. To access the external zone's records, the secondary zone needs at least one server IP address for the external zone. In addition, a TSIG (Transaction Signature) key may be required. TSIG keys are shared secrets used for authentication of secondary DNS zones. You can store these keys in the compartment of your choice.

Each DNS zone you create automatically contains two essential records:

- The SOA (Start of Authority) record specifies authoritative information about the DNS zone. This information includes the primary name server, the domain administrator email address, the domain serial number, and several timers related to refreshing the zone. For more information about SOA records, see RFC 1035.
- The NS (Name Server) record lists the authoritative name servers for a zone. For more information about NS records, see RFC 1035.

You configure the DNS zone by adding specific domain information in the form of resource records. For example, using an address record, you make a domain name resolve to the public IP address of an instance in a public subnet of a VCN. For a list of record types, see Public DNS Resource Record Types.

Creating a Public DNS Zone

On Private Cloud Appliance, DNS zones are created in a compartment to associate IP addresses with portions of the DNS namespace. Zones are created in a compartment using the DNS service.

After a public DNS zone is created, you can add resource tags. You can also edit the `externalMasters` field of a `SECONDARY` zone.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **DNS**, then select **Zones**.
2. Select **Create Zone**.
3. Enter the required zone information:
   - **Zone Name:** Provide a name or description for the DNS zone.
   - **Compartment:** Select the compartment in which to create the DNS zone.
   - **Zone Type:** Choose the type of DNS zone you are creating.
     - **Primary**: A primary DNS zone is the original authoritative DNS zone of a portion of the DNS namespace. When a DNS server hosts a primary zone, that DNS server is the Authoritative DNS Server and is considered the primary source of information in that zone.
     - **Secondary**: A secondary DNS zone is a read-only copy of a primary DNS zone or another secondary DNS zone. A secondary DNS zone is kept on a Secondary DNS Server and reduces the load on the primary DNS zone and eliminated a single point of failure risk to name resolution inside the zone.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
4. Select **Create Zone**.

   The zone is now ready for the addition of zone records or for the configuration of TSIG Keys or Steering Policies. See Zone Records on page 538, Transaction Signature (TSIG) Keys on page 540, and Managing Traffic with Steering Policies on page 542.

Using the CLI

Use the oci dns zone create command and required parameters to create a DNS zone.

```
oci dns zone create [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateZone operation to create a DNS zone.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Zone Records

On Private Cloud Appliance, creating a DNS zone is only the beginning of working with DNS. The zone is essentially empty when created, except for a basic Start of Authority (SOA) and Name Server (NS) record.

The SOA record provides a kind of history of this DNS zone and holds information such as when it was last updated and things like that. The NS record contains the fully-qualified name of the DNS server for the zone. The NS record is very important and therefore has a high TTL, usually 24 hours (86400 seconds).

To make the name server truly useful, the zone must be rounded out and filled with the DNS records that form the basis of responses to the kinds of queries that clients make. These queries include IP addresses for parts of the domain namespace, email server details, and so on.

Creating a Zone Record

On Private Cloud Appliance, you can create a DNS zone record.

The RDATA field is where the content of the zone record is entered. The format of the information varies according to the type of record you are creating. However, the data must be in one of the formats that DNS understands. For example, an A-type zone record RDATA is an IP address, and an MX record contains information on how to route email. Because of the authoritative nature of the zone records within a zone, RDATA is not editable. If DNS information in a zone changes, then the old record must be deleted and a new record created.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **DNS**, then select **Zones**.

   A list of previously configured zones in compartments is displayed.
2. At the top of the page, select the compartment that contains the designated DNS zone.
3. Select the name of the zone.

   The information screen contains general zone information such as type and compartment, OCID (which you can show in full or copy to the clipboard), and the date and time that the zone was created. The zone records that exist are also displayed, and initially only SOA and NS records.
4. Enter the required zone record information:

   - **Zone Record:** Select the type of zone record you're creating from the drop-down list.
     - **A - IPv4 Address**: A host record, which is used to point a hostname to an IPv4 address. This is the most basic DNS record type.
     - You can add many other types of zone records: any types in the drop down list.
   - **Domain (Optional)**: Type the name of the zone subdomain if used (this value is already filled in based on the zone itself: the initial dot (".") is used for adding the zone subdomain).
   - **TTL**: Check this box to set your own value for the TTL of that particular record type. If you don't check this box, the default TTL value for that record type is used (for example, 300 for SOA, 86400 for NS). The valid range is from 1 to 129540 seconds (from 1 second to about a day and a half).
   - **Edit RDATA**: Check this box if you want to edit the RDATA information, such as the IP address or Target established by the zone record type. This box is only displayed for some zone record types.
   - **(RDATA)**: This unlabeled field varies based on the type of zone record created. For example, you enter the 32-bit IP address that corresponds to the A-type DNS record, or Flags for a DNSKEY zone record, if that's what you are creating.
     - **A - IPv4 Address**: If you're creating an A type zone record, then the data is a formatted IPv4 address. This is the most basic DNS record, but there are many others.
     - The RDATA field reflects the correct information for the type of zone record selected.
5. Select **Create Record**.

   The zone record is now added to the zone. If you select the optional box to **Add another record**, then the screen stays at the **Create DNS Zone Record** state to make record entry more efficient.

Using the CLI

Use the oci dns record zone update command and required parameters to replace records in the specified zone with the records specified in the request body.

```
oci dns record zone update [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateZoneRecords operation to replace records in the specified zone with the records specified in the request body.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing a Zone Record

On Private Cloud Appliance, you can update a DNS zone record.

There is no "`edit record`" command. You can update a group of records, and if one of the records in the list is the same except for the `rdata` for example, in effect you have updated the record.

Deleting a Zone Record

On Private Cloud Appliance, you can delete many, but not all DNS zone records. The initial SOA and NS records created by default when the zone is created can't be deleted.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **DNS**, then select **Zones**.

   A list of previously configured zones in compartments appears.
2. At the top of the page, select the compartment that contains the DNS zone with the record you want to delete.
3. Select the name of the zone.

   The information screen contains general zone information such as type and compartment, OCID (which you can show in full or copy to the clipboard), and the date and time that the zone was created. The zone records that exist are also displayed.
4. Select the Actions menu (⋮) for the zone record that you're deleting.
5. Select **Delete**.

Using the CLI

Use the oci dns record rrset delete command and required parameters to delete all records in the specified RRSet.

```
oci dns record rrset delete [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteRRSet command and required parameters to delete all records in the specified RRSet.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing a Public DNS Zone

On Private Cloud Appliance, you can add resource tags to an existing zone.

You can also edit the `externalMasters` field of a `SECONDARY` zone.

Transaction Signature (TSIG) Keys

On Private Cloud Appliance, you can create, add, and delete TSIG keys.

A DNS transaction signature (TSIG) is a network protocol defined in RFC 2845. The main purpose of the TSIG is to allow DNS to authenticate updates to a DNS database, so that malicious users can't change name resolution records to point to a bogus IP address instead of (for example) the IP address of a bank. TSIG uses one-way hashing and shared secret keys to provide a secure means to authenticate the endpoints of a connection for processing (or responding to) DNS update requests.

The TSIG protocol uses timestamps to prevent replay of recorded responses. Therefore, DNS servers and TSIG clients need accurate clocks to provide the timestamps. Several extensions to the basic TSIG protocol have been made to extend the types of cryptography and hashing methods that are supported by TSIG.

To use TSIG for a DNS zone, add TSIG keys to the DNS zone. The TSIG key must be base64 encoded.

Creating a TSIG Key

On Private Cloud Appliance, you can create TSIG keys to ensure that DNS packets originate from an authorized sender by using shared secret keys and one-way hashing to add a cryptographic signature to the DNS packets.

To add a TSIG key to an existing list of TSIG keys, simply create another key with a unique TSIG key name and a new algorithm or a new key value. To modify fields in an existing TSIG key, use the `update` command.

A TSIG key is a separate object from a DNS zone. You can have a `SECONDARY DNS` zone reference a TSIG key as part of its `ExternalMaster` definition. But creating a new key doesn't do anything for a `PRIMARY` zone.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **DNS**, then select **TSIG Keys**.
2. Select **Create Key**.
3. Enter the required TSIG Key information:
   - **Name:** Provide a name or description for the TSIG key.
   - **Compartment:** Select the compartment in which to create the TSIG key.
   - **Algorithm:** Choose the security algorithm for the TSIG Key you are creating, such as **hmac-sha256**.
   - **Secret Key:** Provide the base64 string encoding the binary shared secret that corresponds to the key. The maximum is 255 characters. An example key in base64 encoding is shown in RFC3874. You can provide the key in one of two ways:
     - **Select the key file**: If you provide the TSIG shared secret key this way, you can drag and drop the key file into the space provided.
     - **Paste the key**: If you provide the TSIG shared secret key this way, you can copy and paste the contents of the key file into the space provided.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
4. Select **Create TSIG Key**.

   The TSIG key now available for use in the DNS zone between TSIG client and DNS server.

Using the CLI

Use the oci dns tsig-key create command and required parameters to create a new TSIG key in the specified compartment.

```
oci dns tsig-key create [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateTsigKey operation to create a new TSIG key in the specified compartment.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a TSIG Key

On Private Cloud Appliance, you can delete a TSIG key.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **DNS**, then select **TSIG Keys**.
2. Select the Actions menu (⋮) for the TSIG key that you want to delete, and select **Delete**.

   The TSIG key is removed from the list.

Using the CLI

Use the oci dns tsig-key delete command and required parameters to delete the specified TSIG key.

```
oci dns tsig-key delete [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteTsigKey operation to delete the specified TSIG key.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Public DNS Zone

On Private Cloud Appliance, you can delete a public DNS zone.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **DNS**, then select **Zones**.
2. At the top of the page, select the compartment that contains the public DNS zone you want to delete.
3. Select the zone name that you want to remove from the drop-down list of zones.
4. Select **Delete**.

   The DNS zone is removed from the list.

   The DNS zone is deleted.

Using the CLI

Use the oci dns zone delete command and required parameters to delete the specified DNS zone.

```
oci dns zone delete [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteZone operation to delete the specified DNS zone.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing Traffic with Steering Policies

On Private Cloud Appliance, offers two types of traffic steering policies based on load balancing and some value of the IP address prefix.

DNS can do more than return an IP address (if known) when given a string in the DNS namespace for that zone. DNS is also a part of a system of traffic management, where traffic is distributed among multiple servers depending on some criterion, such as location. Steering policies are a way to distribute access to a single fully-qualified name across multiple servers.

For example, the same content could be available from multiple source servers, whether it is a streaming video or records from a product database. One server might be in the United States, and the other in Europe. A traffic steering policy could distribute traffic based on IP address or CIDR. Other criteria can be used for this traffic distribution, such as load balancing, which strives to keep the load on multiple servers roughly equal.

Private Cloud Appliance supports these types of traffic management steering policies:

Policy Type	Description
Load Balancer	Load Balancer policies allow distribution of traffic across multiple endpoints. Endpoints can be assigned equal weights to distribute traffic evenly across the endpoints, or custom weights may be assigned for ratio load balancing.
IP Prefix Steering	IP Prefix steering policies enable you to steer DNS traffic based on the IP Prefix of the originating query. You can divide users into groups based on the subnets from where requests originate, and steer them to specific resources based on the subdivision you made. For example, you can serve different responses for your internal users as opposed to external users.

A steering policy contains rules to answer DNS queries. You use those rules to filter answers based on the properties of the DNS request. When multiple answers are served in response to DNS queries, that group of answers is called an answer pool. The answers within a pool are sorted by priority and are marked eligible or ineligible. Ineligible answers are omitted from the response.

When attached to a DNS zone, a steering policy takes precedence over all resource records of the zone it covers, and causes DNS responses to be constructed from the steering policy rules instead. For example, if a steering policy attached to the *example.com* DNS zone contains a rule covering the domain *application.example.com* and an answer for the A (address) record type, then the steering policy responds with that answer regardless of any relevant A records that exist in the zone. If a steering policy has no answer for the record type being requested, the DNS request is passed on to the next steering policy or the base zone records.

Steering policies only support records of the types A, AAAA, CNAME. A domain can have at most one steering policy attachment covering a given record type. This means a DNS zone (*example.com*) may have multiple attached steering policies covering different record types for a given domain – for example, one A record policy and one CNAME record policy for *application.example.com*. A DNS zone may also have multiple attached steering policies covering a given record type for different domains – for example, an A record policy for *application.example.com* and an A record policy for *database.example.com*. However, multiple steering policies for the same domain and record type are not supported, because the answer can be provided by only one policy.

Creating a Load Balancer with Steering Policies

On Private Cloud Appliance, if you have more than one DNS server, you can distribute traffic in a load balancing fashion, based on the weight you assign to each of them.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **DNS**, then select **Steering Policies**.
2. Select **Create Steering Policy**.
3. Select **Load Balancer** to create a load balancer steering policy.
4. Enter the required information:
   - **Name:** Enter a name to display for the load balancer steering policy. Do not use confidential information.
   - **Policy TTL:** Enter a TTL in seconds for responses to steering policy requests. The maximum is 604800 seconds (equal to 168 hours or 7 days).
   - **Answers:** Supply the answer or answers to the DNS request for FILTER, WEIGHED, and LIMIT rules. You don't have to specify which condition the answers is for: that's all done by the load balancer template.
     - **Name:** Enter a name for the RData returned, such as Server1.
     - **Type:** Choose the type of resource record to return for the request from the drop-down list. Choices are items such as A (IPv4 address) or CNAME (canonical name).

- **RData:** Enter the resource record RData that is returned that corresponds to the Type selected. For example, for Type = A, the RData would be an IPv4 address.
- **Weight:** Enter a weight for this policy to use for load balancing. Values up to 256 are supported. The default is 10. Higher weights mean that policy answer is used more often. For example, if `dns-server1` and `dns-server2` have equal weights, DNS requests are split evenly between them. If `dns-server1` has a weight twice that of `dns-server2`, then `dns-server1` is used twice as often as `dns-server2`.

5. **Disabled:** The steering policy answer is enabled at creation by default. To disable this steering policy answer, select the toggle to change the Disabled value to TRUE.
6. **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
7. Select **Save Changes**.

   The load balancing steering policy is created.

Using the CLI

Use the oci dns steering-policy create command and required parameters to create a new steering policy in the specified compartment.

```
oci dns steering-policy create [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateSteeringPolicy operation to create a new steering policy in the specified compartment.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Creating an IP Prefix Steering Policy

On Private Cloud Appliance, an IP prefix steering policy dynamically routes DNS request traffic to different servers based on the originating IP prefix.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu select **DNS**, then select **Steering Policies**.
2. Select **Create Steering Policy**.
3. Select IP Prefix Steering and enter the following information:
   - **Name:** The name for the new steering policy.
   - **Policy TTL:**The Time To Live (TTL) for responses from the steering policy, in seconds. The maximum value is 604800 (equal to 168 hours or 7 days).
4. In the Answers box, supply the following properties:
   - **Name:** The name for response to requests sent to the new steering policy.
   - **Type:** The type of request and response. The choices are A, AAA, or CNAME.
   - **RData:** The zone record data to return for the query. It must match the type expected by the type chosen.
   - **Pool:** Select the IP address pool to use of the policy from the drop-down list.
   - **+Add Answer:** Select this box to add more answers to the requests received by the steering policy.
   - **Disabled:** This toggle determines if the IP prefix answer is enabled at creation or not. The default is enabled.
5. In the IP Prefix Steering Rules box, supply the following properties:
   - **+Add Rule:** Select this box to add rules to the IP prefix steering policy.
   - **Order:** Use the directional arrows to order the rule in the sequence of configured rules.
   - **Subnet Address:** Enter the IP subnet prefix to apply to this steering policy.
   - You can add more rules to this steering policy by selecting **+Add Rule**.

6. **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
7. Select **Save Changes**.

   The IP prefix steering policy is created.

Using the CLI

Use the oci dns steering-policy create command and required parameters to create a new steering policy in the specified compartment.

```
oci dns steering-policy create [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateSteeringPolicy operation to create a new steering policy in the specified compartment.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing a Steering Policy

On Private Cloud Appliance, you can edit steering policies using the Compute Web UI, CLI, and API.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **DNS**, then select **Steering Policies**.
2. For the policy that you want to update, select the Actions menu (⋮), and select **Edit**.
3. Make the necessary changes on the **Edit Steering Policy** dialog box.
4. Select **Save Changes**.

   The details page for this steering policy is displayed with the updated information.

Using the CLI

Use the oci dns steering-policy update command and required parameters to update the configuration of the specified steering policy.

```
oci dns steering-policy update [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateSteeringPolicy operation to update the configuration of the specified steering policy.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Moving a Steering Policy to a Different Compartment

On Private Cloud Appliance, you can move a steering policy to a different compartment.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the CLI

Use the oci dns steering-policy change-compartment command and required parameters to move a steering policy to a different compartment.

```
oci dns steering-policy change-compartment -c <destination_compartment_OCID>
 \
--steering-policy-id <steering_policy_OCID>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

This task is not available in the API.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Attaching a Domain to a Steering Policy

On Private Cloud Appliance, a steering policy must be attached to a domain for the policy to answer DNS queries for that domain. The attachment is automatically placed into the same compartment as the domain's zone.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **DNS**, then select **Steering Policies**.
2. At the top of the page, select the compartment that contains the steering policy.
3. Select the name of the policy to which you want attach a domain.
4. Under **Resources**, select **Attached Domains**.
5. In the list of attached domains, select **Add Attached Domain**.
6. In the Add Attached Domain dialog box, enter the domain name and select a zone.
7. Select **Submit**.

   The new domain is added to the **Attached Domains** list for this steering policy.

Using the CLI

Use the oci dns steering-policy-attachment create command and required parameters to create a new attachment between a steering policy and a domain, giving the policy permission to answer queries for the specified domain.

```
oci dns steering-policy-attachment create [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateSteeringPolicyAttachment operation to create a new attachment between a steering policy and a domain, giving the policy permission to answer queries for the specified domain.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing an Attached Domain

On Private Cloud Appliance, you can update a steering policy attachment with new information.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **DNS**, then select **Steering Policies**.
2. At the top of the page, select the compartment that contains the steering policy.
3. Select the name of the policy for which you want to edit an attached domain.
4. Under **Resources**, select **Attached Domains**.
5. Select the name of the attached domain that you want to edit.
6. On the top of the details page for the attached domain, select **Edit**.
7. Make the necessary changes on the **Edit Steering Policy Attachment** dialog box.
8. Select **Save Changes**.

   The details page for this steering policy attachment is displayed with the updated information.

Using the CLI

Use the oci dns steering-policy-attachment update command and required parameters to update the specified steering policy attachment with new information.

```
oci dns steering-policy-attachment update [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateSteeringPolicyAttachment operation to update the specified steering policy attachment with new information.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Steering Policy Attachment

On Private Cloud Appliance, you can delete a steering policy attachment.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **DNS**, then select **Steering Policies**.
2. At the top of the page, select the compartment that contains the steering policy.
3. Select the name of the policy for which you want to delete an attachment.
4. Under **Resources**, select **Attached Domains**.
5. For the attached domain that you want to delete, select the Actions menu (⋮), then select **Delete**.
6. Confirm the deletion.

   The steering policy attachment is removed from the **Attached Domains** list.

Using the CLI

Use the oci dns steering-policy-attachment delete command and required parameters to delete the specified steering policy attachment. A *204* response indicates that the delete has been successful.

```
oci dns steering-policy-attachment delete [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteSteeringPolicyAttachment operation to delete the specified steering policy attachment.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Steering Policy

On Private Cloud Appliance, you can delete a steering policy that isn't attached to any zones.

To detach a policy from a zone, see Deleting a Steering Policy Attachment on page 547.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **DNS**, then select **Steering Policies**.
2. At the top of the page, select the compartment that contains the steering policy.
3. Select the name of the policy that you want to delete.
4. Under **Resources**, select **Attached Domains**, and ensure that this policy has no attached domains.

   To deleted attached domains, see Deleting a Steering Policy Attachment on page 547.
5. At the top of the steering policy details page, select **Delete**.
6. Confirm the deletion.

Using the CLI

Use the oci dns steering-policy delete command and required parameters to delete the specified policy.

```
oci dns steering-policy delete [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteSteeringPolicy operation to delete the specified policy.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Load Balancers

On Private Cloud Appliance, load balancing is the method of sharing a workload equally among servers. It prevents clients from overwhelming certain servers.

The Load Balancer service provides automated traffic distribution from one entry point to multiple servers reachable from your virtual cloud network (VCN). The service offers a load balancer with your choice of a public or private IP address.

Load Balancer Types

Two major types of load balancers are available on Private Cloud Appliance:

- Load Balancer as a Service on page 550 – This type of load balancer operates at all protocol layers, including the application. When the term "load balancer" (LB) appears without qualification, the statement refers to LBaaS.
- Network Load Balancers on page 601 – This type of load balancer operates on protocol layers below the application itself, at the Network Layer. The term "network load balancer" (NLB) always refers to a network load balancer, not to LBaaS.

The verb "load balancing" refers to the actions of both LBs and NLBs. The term "load balancer" can refer to both LBs and NLBs. When you need to be specific, use LB and NLB.

A load balancer (both LBs and NLBs) can be either private or public.

- **Private:** A private load balancer is isolated from the network outside the Private Cloud Appliance. A private load balancer is assigned a private IP address from the address block of the specified private subnet. This private IP address is used as a front end for incoming internal VCN traffic, to balance that traffic across all backend servers.

  For a private LB you need a VCN with at least one private subnet. The subnet must have security rules that allow the intended traffic. The backend servers must be reachable from the selected VCN.
- **Public:** A public load balancer accepts traffic from a network location outside of the appliance. A public load balancer can be assigned a public IP address from a public subnet of a VCN that has a NAT gateway and an internet gateway (IGW) configured, or you can select the public IP address from a list. This public IP address is used as the entry point for incoming traffic, to balance that traffic across all backend servers. You can associate the public IP address with a friendly DNS name through any DNS provider.

  For a public LB, you need a VCN with at least one public subnet. The subnet must have security rules that allow the intended traffic. The backend servers must be reachable from the selected VCN.

  You can select a public IP address from a list, or you can let the system assign an IP address.

Subnet Prerequisites

Load balancers in a Private Cloud Appliance environment require several IP addresses for their deployment. This applies to standalone configurations, and setups used for Kubernetes Engine (OKE) control plane and worker node load balancing.

In the subnet where a load balancer is deployed, an IP address must be assigned to the gateway and to each of the three management nodes. This is in addition to the standard network and broadcast addresses. Each load balancer in the subnet also needs three IP addresses for its operation.

Here's a practical example for illustration purposes:

- Subnet overhead:

```
cidrBlock:          10.196.43.64/28
ipRange:            10.196.43.64 - 10.196.43.79
netIp:              10.196.43.64
netBroadcast:       10.196.43.79

netGw:              10.196.43.65
mn1-ip:             10.196.43.69
mn2-ip:             10.196.43.70
mn3-ip:             10.196.43.71
```

- Load balancers:

```
lb1-ip1:            10.196.43.66
lb1-ip2:            10.196.43.67
lb1-ipv:            10.196.43.68

lb2-ip1:            10.196.43.72
lb2-ip2:            10.196.43.73
lb2-ipv:            10.196.43.74
```

As the example shows, the deployment of a single load balancer requires 7 IP addresses. A /29 subnet only has 6 usable IPs, so to provide the required address range the subnet must be /28 or larger.

Characteristics Compared

The two load balancer types apply to different use cases. This section compares their characteristics side by side.

The following table summarizes the major differences.

LBaaS and NLB Major Characteristics Compared

Major Characteristic	**LBaaS**	**NLB**
Visibility	Public or Private	Public or Private
IP Address	Ephemeral or Reserved IP address	Ephemeral or Reserved IP address
Policy Parameters	Weighted Round Robin, IP hash, Least Connections	IP hash mapped from 5, 3, or 2 header fields
Layer 4 Functioning	Yes	Yes
Layer 7 Functioning	Yes	**No**
TLS Support	Yes	**No**

If a VCN uses network security groups (NSGs), you can associate the load balancer with an NSG. An NSG has a set of security rules that controls allowed types of inbound and outbound traffic. The rules apply only to the resources in the group. An NSG isn't a security list, where the rules apply to all the resources in any subnet that uses the list. See Controlling Traffic with Network Security Groups on page 501.

If you prefer to use security lists for the VCN, see Controlling Traffic with Security Lists on page 496.

Other differences are of an operational nature, or involved in configuration limits. Many of the NLB limitations are because of the functioning at Layer 4 and no higher. These differences are listed in the following table.

Other LBaaS and NLB Characteristics Compared

Characteristic	LBaaS	NLB
Routing of Requests	Yes	No
Persistence of Sessions	Yes	No
SSL Certificates	Yes	No
Cipher Suites	Yes	No
Listener Protocol	HTTP, HTTP2, TCP, HTTPS	TCP
Health Check Protocol	HTTP, TCP	HTTP, HTTPS, TCP
Certificate Support	Yes	No

Depending on the type of load balancer you want to use, see one of these sections:

- Load Balancer as a Service on page 550
- Network Load Balancers on page 601

Load Balancer as a Service

On Private Cloud Appliance, you can configure the Load Balancing service (LBaaS) to automatically distribute network traffic.

The Load Balancing service provides automated traffic distribution from one entry point to multiple servers reachable from your virtual cloud network (VCN). The service offers a load balancer with your choice of a public or private IP address; provisioned bandwidth isn't user-configurable.

A load balancer improves resource usage, facilitates scaling, and helps ensure high availability. You can configure multiple load balancing policies and application-specific health checks to ensure that the load balancer directs traffic only to healthy instances.

Load Balancer Essentials

When working with load balancers, it's critical that you understand its building blocks and related key terminology.

Backend Server

An application server responsible for generating content in reply to the incoming TCP or HTTP traffic.

You typically identify application servers with a unique combination of private IPv4 address and port – for example: 10.25.4.101:8080, 10.25.4.102:8080, etc.

Backend Set

A logical entity defined by a list of backend servers, a load balancing policy, and a health check policy.

The backend set determines how the load balancer directs traffic to the collection of backend servers. SSL configuration is optional.

Load Balancing Policy

A load balancing policy tells the load balancer how to distribute incoming traffic to the backend servers.

Common load balancer policies include: round robin, least connections, IP hash.

Health Check

A test to confirm the availability of backend servers.

You configure the health check policy when you create a backend set. You can configure TCP-level or HTTP-level health checks for backend servers. Test results are reported through health status indicators.

Listener

A logical entity that checks for incoming traffic on the load balancer IP address.

You configure a listener's protocol and port number, and the optional SSL settings. To handle TCP and HTTP traffic, you must configure multiple listeners. Supported protocols include: TCP, HTTP/1.x, HTTP/2.

Cipher Suite

A logical entity for a set of algorithms, or *ciphers*, using Transport Layer Security (TLS) to determine the security, compatibility, and speed of HTTPS traffic.

Path Route Set

A set of path route rules to route traffic to the correct backend set without using multiple listeners or load balancers.

Shape

A template that determines the load balancer total pre-provisioned maximum capacity (bandwidth) for ingress plus egress traffic. Private Cloud Appliance only provides the 400 Mbps shape.

Pre-provisioned maximum capacity applies to aggregated connections, not to a single client attempting to use the full bandwidth.

SSL Certificate

If you use HTTPS or SSL for the listener, you must associate an SSL server certificate (X.509) with the load balancer. A certificate enables the load balancer to terminate the connection and decrypt incoming requests before passing them to the backend servers. Traffic between the load balancer and the backend can also be SSL-encrypted.

Work Request

An object that reports on the current state of a Load Balancing request.

The Load Balancing service handles requests asynchronously. Each request returns a work request ID (OCID) as the response. You can view the work request item to see the status of the request.

Implementation Options

> **Note:**
>
> The controller software on Roving Edge Infrastructure currently doesn't support session persistence. The option applies to Compute Cloud@Customer and Private Cloud Appliance.

The default implementation of application load balancers doesn't provide session persistence through cookies. However, you can optionally select a different implementation if you need a load balancer with cookie based session persistence. The choice is made for individual load balancers, and cannot be changed after creation. Both types can be deployed on the same system.

You deploy an application load balancer with session persistence by adding a specific defined tag during the creation steps.

1. Ensure that the OraclePCA tag namespace exists on your system. If not, create the namespace.

   For more information, see Using Resource Tags to Extend Functionality (IAM on the Appliance) on page 460.
2. In the OraclePCA tag namespace, create the defined tag key `sessionPersistence` with two possible values: `True` or `False`.
3. When creating an application load balancer, apply the defined tag:
   - Compute Web UI: in the Tagging section, select OraclePCA.sessionPersistence with value True.

- OCI CLI: in the `oci lb load-balancer create` command, include this parameter:

```
--defined-tags '{"OraclePCA": {"sessionPersistence": True}}'
```

Frontend Configuration

Private Cloud Appliance supports different types of load balancers.

In the context of load balancers, the term *frontend* refers to the components that a client can see and send requests to. The entry point of client requests is the outward-facing floating IP address of the load balancer. Incoming traffic is checked by listeners, which are configured for specific protocols and ports. Different types of rules can be defined to categorize incoming requests and route them accordingly to different sets of backend servers.

Load Balancer Types

The Load Balancing service allows you to create a public or private load balancer within a VCN. Load balancers are deployed in pairs: one active and one standby instance sharing a floating IP address. A public load balancer has a public IP address that's accessible from outside the Private Cloud Appliance network environment. A private load balancer has an IP address from the hosting subnet, which is visible only within the VCN.

You can configure multiple listeners for an IP address to load balance transport Layer 4 and Layer 7 (TCP and HTTP) traffic. Both public and private load balancers act as reverse proxies and can route data traffic to any backend server that's reachable from the VCN.

All load balancers have a backend set to route incoming traffic to compute instances. The backend set is a logical entity that includes:

- a list of backend servers
- a load balancing policy
- a health check policy
- (optional) SSL handling

The backend servers – compute instances – associated with a backend set can exist anywhere, as long as the associated network security groups (NSGs), security lists, and route tables allow the intended traffic flow.

Private Load Balancer

To isolate a load balancer from the external network and simplify its security posture, you can create a private load balancer. The Load Balancing service assigns it a private IP address that serves as the entry point for incoming traffic.

When you create a private load balancer, the service requires only one subnet to host both the primary and standby load balancers. The load balancer is accessible only from within the VCN that contains the host subnet, or as further restricted by security rules.

The assigned floating private IP address is local to the host subnet. The primary and standby load balancers each require an extra private IP address from the host subnet.

See Creating a Load Balancer on page 566.

Public Load Balancer

To accept traffic from a network location outside of Private Cloud Appliance, you create a public load balancer. A public load balancer must be deployed in a public subnet of a VCN that has an internet gateway configured. The service assigns the load balancer a public IP address that serves as the entry point for incoming traffic. You can associate the public IP address with a friendly DNS name through any DNS provider.

A Private Cloud Appliance is a single Availability Domain by definition, which implies that the primary and standby load balancers can't be deployed in separate Availability Domains. Therefore, both the primary and standby load balancer are deployed in the same subnet, and they each are assigned a private IP address from that host subnet. In addition, a load balancer uses one floating public IP address that can be reassigned between standby and primary as required.

See Creating a Load Balancer on page 566.

Load Balancing Policies

On Private Cloud Appliance, after you create a load balancer, you can apply policies to control traffic distribution to your backend servers. The primary policy types that the Load Balancing service supports, are: Round Robin, Least Connections, and IP Hash.

Round Robin

The Round Robin policy is the default load balancer policy. This policy distributes incoming traffic sequentially to each server in a backend set list. After each server has received a connection, the load balancer repeats the list in the same order.

Round Robin is a simple load balancing algorithm. It works best when all the backend servers have similar capacity and the processing load required by each request doesn't vary significantly.

Least Connections

The Least Connections policy routes incoming nonsticky request traffic to the backend server with the fewest active connections. This policy helps you maintain an equal distribution of active connections with backend servers. As with the round robin policy, you can assign a weight to each backend server and further control traffic distribution.

This policy works best with protocols using long sessions, such as LDAP and SQL. It's not recommended for short sessions as typically seen in HTTP use cases. If you select the Least Connections policy for HTTP connections, the load balancer might instead apply Round Robin internally, and even adjust server weighting dynamically.

> **Note:**
>
> In TCP use cases, a connection can be active but have no current traffic. Such connections don't serve as a good load metric.

IP Hash

The IP Hash policy uses an incoming request's source IP address as a hashing key to route nonsticky traffic to the same backend server. The load balancer routes requests from the same client to the same backend server as long as that server is available. This policy honors server weight settings when establishing the initial connection.

> **Caution:**
>
> Multiple clients that connect to the load balancer through a proxy or NAT router appear to have the same IP address. If you apply the IP Hash policy to your backend set, the load balancer routes traffic based on the incoming IP address and sends these proxied client requests to the same backend server. If the proxied client pool is large, the requests could flood a backend server.

When processing load or capacity varies among backend servers, you can refine each of these policy types with backend server *weighting*. Weighting affects the proportion of requests directed to each server. For example, a server weighted '3' receives three times the number of connections as a server weighted '1.' You assign weights based on criteria of your choosing, such as each server's traffic-handling capacity. Weight values must be from 1 to 100.

Load balancer policy decisions apply differently to TCP load balancers, cookie-based session persistent HTTP requests – also known as sticky requests –, and nonsticky HTTP requests.

- A TCP load balancer considers policy and weight criteria to direct an initial incoming request to a backend server. All subsequent packets on this connection go to the same endpoint.
- For nonsticky HTTP requests, the load balancer applies policy and weight criteria to every incoming request and determines an appropriate backend server. Multiple requests from the same client could be directed to different servers.

Listeners

On Private Cloud Appliance,

The listener is a logical entity at the ingress side of the load balancer setup. It's the key component that detects incoming traffic on the load balancer IP address. It listens for requests from clients using a particular protocol and port. Requests are subsequently routed to the appropriate backend servers based on the rules defined in the load balancer configuration. To handle TCP, HTTP, and HTTPS traffic, you must configure at least one listener per traffic type.

When you create a listener, ensure that your VCN security rules allow the listener to accept traffic.

> **Note:**
>
> To accommodate high-volume traffic, we recommend that you use *stateless* security rules for your load balancer subnets.

You can have one SSL certificate bundle per listener. You can configure two listeners, one each for ports 443 and 8443, and associate SSL certificate bundles with each listener. For more information about SSL certificates for load balancers, see Load Balancer SSL Certificates on page 573.

Cipher Suites

On Private Cloud Appliance, a cipher suite is a set of algorithms or ciphers that help secure network connections using Transport Layer Security (TLS). You configure cipher suites for a load balancer to determine the security, compatibility and speed of HTTPS traffic. All ciphers are associated with at least one version of TLS (1.0, 1.1, 1.2).

Predefined Cipher Suites

On Private Cloud Appliance, the Load Balancing service supports predefined cipher suites. Note that different ciphers are supported when session persistence is enabled on the load balancer.

oci-default-ssl-cipher-suite-v1 (without session persistence)

This cipher suite contains a restricted set of ciphers that are only supported in TLS version 1.2 and meet stricter compliance requirements.

- ECDHE-RSA-AES128-GCM-SHA256
- ECDHE-RSA-AES256-GCM-SHA384

oci-default-ssl-cipher-suite-v1 (with session persistence)

This cipher suite contains a restricted set of ciphers that are only supported in TLS version 1.2 and meet stricter compliance requirements.

- ECDHE-RSA-AES128-GCM-SHA256
- ECDHE-RSA-AES128-SHA256
- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-SHA384
- DHE-RSA-AES256-GCM-SHA384
- DHE-RSA-AES256-SHA256
- DHE-RSA-AES128-GCM-SHA256
- DHE-RSA-AES128-SHA256

oci-modern-ssl-cipher-suite-v1 (without session persistence)

This cipher suite offers a wider set of ciphers, but still limited to TLS version 1.2 only.

- ECDHE-ECDSA-AES128-GCM-SHA256
- ECDHE-RSA-AES128-GCM-SHA256
- ECDHE-ECDSA-AES128-SHA
- ECDHE-RSA-AES128-SHA
- ECDHE-ECDSA-AES256-GCM-SHA384

- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-SHA
- ECDHE-ECDSA-AES256-SHA
- AES128-GCM-SHA256
- AES128-SHA
- AES256-GCM-SHA384
- AES256-SHA

oci-modern-ssl-cipher-suite-v1 (with session persistence)

This cipher suite offers a wider set of ciphers, but still limited to TLS version 1.2 only.

- ECDHE-ECDSA-AES128-GCM-SHA256
- ECDHE-RSA-AES128-GCM-SHA256
- ECDHE-ECDSA-AES128-SHA256
- ECDHE-RSA-AES128-SHA256
- ECDHE-ECDSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-ECDSA-AES256-SHA384
- ECDHE-RSA-AES256-SHA384
- AES128-GCM-SHA256
- AES128-SHA256
- AES256-GCM-SHA384
- AES256-SHA256
- DHE-RSA-AES256-GCM-SHA384
- DHE-RSA-AES256-SHA256
- DHE-RSA-AES128-GCM-SHA256
- DHE-RSA-AES128-SHA256

oci-compatible-ssl-cipher-suite-v1 (without session persistence)

This cipher suite supports the broadest set of ciphers. It contains ciphers supported by TLS versions 1.1 and 1.2.

- ECDHE-ECDSA-AES128-GCM-SHA256
- ECDHE-RSA-AES128-GCM-SHA256
- ECDHE-ECDSA-AES128-SHA
- ECDHE-RSA-AES128-SHA
- ECDHE-ECDSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-SHA
- ECDHE-ECDSA-AES256-SHA
- AES128-GCM-SHA256
- AES128-SHA
- AES256-GCM-SHA384
- AES256-SHA

oci-compatible-ssl-cipher-suite-v1 (with session persistence)

This cipher suite supports the broadest set of ciphers. It contains ciphers supported by TLS versions 1.1 and 1.2.

- ECDHE-ECDSA-AES128-GCM-SHA256
- ECDHE-RSA-AES128-GCM-SHA256
- ECDHE-ECDSA-AES128-SHA256
- ECDHE-RSA-AES128-SHA256

- ECDHE-ECDSA-AES128-SHA
- ECDHE-RSA-AES128-SHA
- ECDHE-ECDSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-ECDSA-AES256-SHA384
- ECDHE-RSA-AES256-SHA384
- ECDHE-RSA-AES256-SHA
- ECDHE-ECDSA-AES256-SHA
- AES128-GCM-SHA256
- AES128-SHA256
- AES128-SHA
- AES256-GCM-SHA384
- AES256-SHA256
- AES256-SHA
- DHE-RSA-AES256-GCM-SHA384
- DHE-RSA-AES256-SHA256
- DHE-RSA-AES128-GCM-SHA256
- DHE-RSA-AES128-SHA256

oci-wider-compatible-ssl-cipher-suite-v1 (without session persistence)

This cipher suite contains all supported ciphers.

- ECDHE-ECDSA-AES128-GCM-SHA256
- ECDHE-RSA-AES128-GCM-SHA256
- ECDHE-ECDSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-GCM-SHA384
- AES256-GCM-SHA384
- ECDHE-ECDSA-AES128-SHA
- ECDHE-ECDSA-AES256-SHA
- ECDHE-RSA-AES128-SHA
- ECDHE-RSA-AES256-SHA
- AES128-GCM-SHA256
- AES128-SHA
- AES256-SHA
- DES-CBC3-SHA
- PSK-AES256-CBC-SHA
- PSK-AES128-CBC-SHA

oci-wider-compatible-ssl-cipher-suite-v1 (with session persistence)

This cipher suite contains all supported ciphers.

- ECDHE-ECDSA-AES128-GCM-SHA256
- ECDHE-RSA-AES128-GCM-SHA256
- ECDHE-ECDSA-AES128-SHA256
- ECDHE-ECDSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-ECDSA-AES256-SHA384
- ECDHE-RSA-AES256-SHA384
- AES128-SHA256
- AES256-GCM-SHA384
- AES256-SHA256

- DHE-RSA-AES256-GCM-SHA384
- DHE-RSA-AES256-SHA256
- DHE-RSA-AES128-GCM-SHA256
- DHE-RSA-AES128-SHA256
- DH-DSS-AES256-GCM-SHA384
- DHE-DSS-AES256-GCM-SHA384
- DH-RSA-AES256-GCM-SHA384
- DHE-DSS-AES256-SHA256
- DH-RSA-AES256-SHA256
- DH-DSS-AES256-SHA256
- ECDH-RSA-AES256-GCM-SHA384
- ECDH-ECDSA-AES256-GCM-SHA384
- ECDH-RSA-AES256-SHA384
- ECDH-ECDSA-AES256-SHA384
- DH-DSS-AES128-GCM-SHA256
- DHE-DSS-AES128-GCM-SHA256
- DH-RSA-AES128-GCM-SHA256
- DHE-DSS-AES128-SHA256
- DH-RSA-AES128-SHA256
- DH-DSS-AES128-SHA256
- ECDH-RSA-AES128-GCM-SHA256
- ECDH-ECDSA-AES128-GCM-SHA256
- ECDH-RSA-AES128-SHA256
- ECDH-ECDSA-AES128-SHA256
- ECDHE-ECDSA-AES128-SHA
- ECDHE-ECDSA-AES256-SHA
- ECDHE-RSA-AES128-SHA
- ECDHE-RSA-AES256-SHA
- AES128-GCM-SHA256
- AES128-SHA
- AES256-SHA
- DES-CBC3-SHA
- DHE-RSA-AES128-SHA
- DHE-RSA-CAMELLIA256-SHA
- DHE-RSA-CAMELLIA128-SHA
- DHE-RSA-SEED-SHA
- DHE-RSA-AES256-SHA
- DHE-DSS-AES256-SHA
- DH-RSA-AES256-SHA
- DH-DSS-AES256-SHA
- DHE-RSA-CAMELLIA256-SHA
- DHE-DSS-CAMELLIA256-SHA
- DH-RSA-CAMELLIA256-SHA
- DH-DSS-CAMELLIA256-SHA
- ECDH-RSA-AES256-SHA
- ECDH-ECDSA-AES256-SHA
- CAMELLIA256-SHA
- PSK-AES256-CBC-SHA
- DHE-RSA-AES128-SHA
- DHE-DSS-AES128-SHA

- DH-RSA-AES128-SHA
- DH-DSS-AES128-SHA
- DHE-RSA-CAMELLIA128-SHA
- DHE-DSS-CAMELLIA128-SHA
- DH-RSA-CAMELLIA128-SHA
- DH-DSS-CAMELLIA128-SHA
- ECDH-RSA-AES128-SHA
- ECDH-ECDSA-AES128-SHA
- CAMELLIA128-SHA
- PSK-AES128-CBC-SHA

Custom Cipher Suites
On Private Cloud Appliance,

Instead of selecting from the predefined cipher suites, you can create a cipher suite of your own to match the specific requirements of your environment. You build a custom cipher suite by adding individual ciphers associated with the TLS versions used in your configuration. A custom cipher suite must contain at least one cipher. Include only ciphers for the TLS versions that your environment effectively supports.

> **Note:**

- Ensure compatibility between specified SSL protocols and configured ciphers in the cipher suite, otherwise the SSL handshake will fail.
- Ensure compatibility between configured ciphers in the cipher suite and configured certificates. For example: RSA-based ciphers require an RSA certificate, whereas ECDSA-based ciphers require ECDSA certificates.

Supported Ciphers
On Private Cloud Appliance, the Load Balancing service supports specific ciphers. Note that different ciphers are supported when session persistence is enabled on the load balancer.

TLS Version 1.2 Ciphers (without session persistence)

- AES128-GCM-SHA256
- AES256-GCM-SHA384
- ECDHE-ECDSA-AES128-GCM-SHA256
- ECDHE-ECDSA-AES256-GCM-SHA384
- ECDHE-RSA-AES128-GCM-SHA256
- ECDHE-RSA-AES256-GCM-SHA384

TLS Version 1.2 Ciphers (with session persistence)

- AES128-GCM-SHA256
- AES128-SHA256
- AES256-GCM-SHA384
- AES256-SHA256
- DH-DSS-AES128-GCM-SHA256
- DH-DSS-AES128-SHA256
- DH-DSS-AES256-GCM-SHA384
- DH-DSS-AES256-SHA256
- DH-RSA-AES128-GCM-SHA256
- DH-RSA-AES128-SHA256
- DH-RSA-AES256-GCM-SHA384
- DH-RSA-AES256-SHA256
- DHE-DSS-AES128-GCM-SHA256
- DHE-DSS-AES128-SHA256

- DHE-DSS-AES256-GCM-SHA384
- DHE-DSS-AES256-SHA256
- DHE-RSA-AES128-GCM-SHA256
- DHE-RSA-AES128-SHA256
- DHE-RSA-AES256-GCM-SHA384
- DHE-RSA-AES256-SHA256
- ECDH-ECDSA-AES128-GCM-SHA256
- ECDH-ECDSA-AES128-SHA256
- ECDH-ECDSA-AES256-GCM-SHA384
- ECDH-ECDSA-AES256-SHA384
- ECDH-RSA-AES128-GCM-SHA256
- ECDH-RSA-AES128-SHA256
- ECDH-RSA-AES256-GCM-SHA384
- ECDH-RSA-AES256-SHA384
- ECDHE-ECDSA-AES128-GCM-SHA256
- ECDHE-ECDSA-AES128-SHA256
- ECDHE-ECDSA-AES256-GCM-SHA384
- ECDHE-ECDSA-AES256-SHA384
- ECDHE-RSA-AES128-GCM-SHA256
- ECDHE-RSA-AES128-SHA256
- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-SHA384

TLS Version 1.0/1.1 Ciphers Supported in TLS Version 1.2 (without session persistence)

- AES128-SHA
- AES256-SHA
- DES-CBC3-SHA
- ECDHE-ECDSA-AES128-SHA
- ECDHE-ECDSA-AES256-SHA
- ECDHE-RSA-AES128-SHA
- ECDHE-RSA-AES256-SHA
- PSK-AES128-CBC-SHA
- PSK-AES256-CBC-SHA

TLS Version 1.0/1.1 Ciphers Supported in TLS Version 1.2 (with session persistence)

- AES128-SHA
- AES256-SHA
- CAMELLIA128-SHA
- CAMELLIA256-SHADES-CBC3-SHA
- DH-DSS-AES128-SHA
- DH-DSS-AES256-SHA
- DH-DSS-CAMELLIA128-SHA
- DH-DSS-CAMELLIA256-SHA
- DH-DSS-DES-CBC3-SHA
- DH-DSS-SEED-SHA
- DH-RSA-AES128-SHA
- DH-RSA-AES256-SHA
- DH-RSA-CAMELLIA128-SHA
- DH-RSA-CAMELLIA256-SHA

- DH-RSA-DES-CBC3-SHA
- DH-RSA-SEED-SHA
- DHE-DSS-AES128-SHA
- DHE-DSS-AES256-SHA
- DHE-DSS-CAMELLIA128-SHA
- DHE-DSS-CAMELLIA256-SHA
- DHE-DSS-DES-CBC3-SHA
- DHE-DSS-SEED-SHA
- DHE-RSA-AES128-SHA
- DHE-RSA-AES256-SHA
- DHE-RSA-CAMELLIA128-SHA
- DHE-RSA-CAMELLIA256-SHA
- DHE-RSA-DES-CBC3-SHA
- DHE-RSA-SEED-SHA
- ECDH-ECDSA-AES128-SHA
- ECDH-ECDSA-AES256-SHA
- ECDH-ECDSA-DES-CBC3-SHA
- ECDH-ECDSA-RC4-SHA
- ECDH-RSA-AES128-SHA
- ECDH-RSA-AES256-SHA
- ECDH-RSA-DES-CBC3-SHA
- ECDH-RSA-RC4-SHA
- ECDHE-ECDSA-AES128-SHA
- ECDHE-ECDSA-AES256-SHA
- ECDHE-ECDSA-DES-CBC3-SHA
- ECDHE-ECDSA-RC4-SHA
- ECDHE-RSA-AES128-SHA
- ECDHE-RSA-AES256-SHA
- ECDHE-RSA-DES-CBC3-SHA
- ECDHE-RSA-RC4-SHA
- IDEA-CBC-SHA
- KRB5-DES-CBC3-MD5
- KRB5-DES-CBC3-SHA
- KRB5-IDEA-CBC-MD5
- KRB5-IDEA-CBC-SHA
- KRB5-RC4-MD5
- KRB5-RC4-SHA
- PSK-3DES-EDE-CBC-SHA
- PSK-AES128-CBC-SHA
- PSK-AES256-CBC-SHA
- PSK-RC4-SHA
- RC4-MD5
- RC4-SHA
- SEED-SHA

Request Routing

On Private Cloud Appliance, the Load Balancing service enables you to route incoming requests to different backend servers based on certain properties of the request. The goal is to optimize the usage of backend resources.

Request routing can be configured using virtual hostnames path route sets, and rule sets.

When multiple applications are hosted on the backend, several virtual host names can be associated with the load balancer listeners, so that incoming HTTP requests matching a particular host name are routed to the right backend servers.

When backend applications use multiple endpoints or content types, represented by a distinct URI or path, HTTP requests for each of those endpoints can be routed to different backend servers based on path route rules. These path route rules contain a string that's matched against the URI of an incoming request to determine the appropriate backend destination.

Virtual host names and path route rule sets are frequently combined in request routing. Note that virtual host names take precedence over URI matching.

To apply actions to the traffic on a load balancer's listener, you can associate a rule set with the listener. Rules can add or remove HTTP request or response headers, or they can restrict access to the application hosted on the backend based on the source of the request.

Session Persistence

Session persistence is supported on Private Cloud Appliance.

> **Tip:**
>
> Cookie-based session persistence is not provided by default. It's available for application load balancers if a defined tag is applied to enable that specific implementation. For more information, see LBaaS Implementation Options on page 551.

Session persistence is a method to direct all requests originating from a single logical client to a single backend web server. Backend servers that use caching to improve performance, or to enable log-in sessions or shopping carts, can benefit from session persistence.

You enable session persistence when you create a load balancer or when you create a backend set. You can also edit an existing backend set to enable, disable, or change the session persistence configuration.

The Load Balancing service offers two mutually exclusive cookie-based configurations for enabling session persistence: application cookie stickiness and load balancer cookie stickiness.

> **Note:**
>
> **IP address-driven session persistence**
>
> Some products offer session persistence support without cookies. These products depend on the IP address of the incoming request. ISP proxies and company exit gateways can issue many requests from a single IP address. In this case, a single backend server can be subject to high traffic volumes. Your backend fleet can become overwhelmed, one server at a time, even though effective load balancing is possible.
>
> Another weakness of IP address-driven session persistence is that the originating IP address can change. In this case, session persistence can be lost or the request redirected to the wrong backend server.

Application Cookie Stickiness

To configure application cookie session persistence, you only need to specify a cookie name. By default, traffic from a persistent session client is redirected to a different backend server when the original server is unavailable. This fallback behavior cannot be disabled.

The Load Balancing service activates application cookie session persistence (stickiness) when a backend server sends a Set-Cookie response header containing a recognized cookie name. The cookie name must match the name specified in the backend set configuration. If the configuration specifies a match-all pattern ("*") any cookie set by the server activates session persistence. Unless a backend server activates session persistence, the service follows the load balancing policy specified when you created the load balancer.

Requirements:

- The load balancer must operate in HTTP mode to support server side, cookie-driven session persistence.
- The client computer must accept cookies for load balancing session persistence to work.

How It Works:

The Load Balancing service calculates a hash of the configured cookie and other request parameters, and sends that value to the client in a cookie. The value stored in the cookie enables the service to route subsequent client requests to the correct backend server. If your backend servers change any of the defined cookies, the service recomputes the cookie's value and resends it to the client.

> **Note:**
>
> We recommend that you treat cookie data as an opaque entity. Do not use it in your applications.
>
> The backend server can stop application cookie persistence by deleting the session persistence cookie. If you used the match-all pattern, it must delete all cookies. You can delete cookies by sending a Set-Cookie response header with a past expiration date. The Load Balancing service routes subsequent requests using the configured load balancing policy.

Load Balancer Cookie Stickiness

When you configure load balancer cookie stickiness, the load balancer inserts a cookie into the response. The parameters configured within the cookie enable session stickiness. This method is useful when you have applications and web backend services that cannot generate their own cookies.

To configure load balancer cookie session persistence, you specify these parameters and settings:

The cookie name

If you do not specify a cookie name, the default name is `X-Oracle-BMC-LBS-Route`.

> **Note:**
>
> Ensure that the cookie name used at the backend application servers is different from the cookie name used at the load balancer. To minimize the chance of name collision, We recommend that you use a prefix such as `X-Oracle-OCI-`.
>
> If both a backend server and the load balancer insert cookies with the same name, the client or browser behavior can vary depending on the domain value associated with the cookie. If the name and domain values of the Set-cookie header (generated by a backend server) and the Set-cookie header (generated by the load balancer) are the same, the client or browser treats them as one cookie. The client returns only one of the cookie values in subsequent requests. If both Set-cookie names are the same, but the domain names are different, the client or browser treats them as two different cookies.

The domain in which the cookie is valid

The Set-cookie header inserted by the load balancer contains a domain attribute with the specified value. This attribute has no default value. If you do not specify a value, the load balancer does not insert the domain attribute into the Set-cookie header.

> **Note:**
>
> - RFC 6265 - HTTP State Management Mechanism describes client and browser behavior when the domain attribute is present or not present in the `Set-cookie` header.

If the value of the `Domain` attribute is `example.com` in the `Set-cookie` header, the client includes the same cookie in the `Cookie` header when making HTTP requests to `example.com`, `www.example.com`, and `www.abc.example.com`. If the `Domain` attribute is not present, the client returns the cookie only for the domain to which the original request was made.

- Ensure that this attribute specifies the correct domain value. If the `Domain` attribute in the `Set-cookie` header does not include the domain to which the original request was made, the client or browser might reject the cookie. As specified in RFC 6265, the client accepts a cookie with the `Domain` attribute value `example.com` or `www.example.com` sent from `www.example.com`. It does not accept a cookie with the `Domain` attribute `abc.example.com` or `www.abc.example.com` sent from `www.example.com`.

Whether the `Set-cookie` header should contain the `Secure` attribute

The `Secure` attribute directs the client or browser to send the cookie only using a secure protocol.

Note:

If you set this field to true, you cannot associate the corresponding backend set with an HTTP listener.

Whether the `Set-cookie` header should contain the `HttpOnly` attribute

The `HttpOnly` attribute limits the scope of the cookie to HTTP requests. This attribute directs the client or browser to omit the cookie when providing access to cookies through non-HTTP APIs. For example, it restricts the cookie from JavaScript channels.

These parameters cannot be used:

The amount of time the cookie remains valid

The maximum age attribute (`maxAgeInSeconds`) of the Set-cookie header inserted by the load balancer must not be specified. It has no default value either. The client or browser typically retains the cookie until the current session ends, as defined by the client.

The URI path in which the cookie is valid

The path attribute of the Set-cookie header inserted by the load balancer must not be specified. The default value ("`/`") is always used.

Whether to disable fallback for unavailable servers

Refer to the Fallback section below.

Fallback

By default, the Load Balancing service directs traffic from a persistent session client to a different backend server when the original server is unavailable. You cannot configure the backend set to disable this fallback behavior.

Note:

The Load Balancing service considers a server marked *drain* available for existing persisted sessions. New requests that are not part of an existing persisted session are not sent to that server.

Connection Timeout Settings

On Private Cloud Appliance, you can configure load balancer listeners to control the maximum idle time allowed during each TCP connection or HTTP request and response pair.

Load balancers support connection multiplexing. The load balancer can route many incoming requests from multiple clients to the destination backend server through a small number (one or multiple) of backend connections.

After your load balancer connects a client to a backend server, the connection can be closed because of inactivity. You can configure load balancer listeners to control the maximum idle time allowed during each TCP connection or HTTP request and response pair.

The following timeout settings affect the load balancer behavior:

- Keep-alive setting between load balancer and client
- Keep-alive setting between load balancer and backend server
- Idle timeout in seconds

Keep-Alive Settings

The Load Balancing service does not honor keep-alive settings from backend servers. The load balancer closes backend server connections that are idle for more than 300 seconds. We recommend that you do not allow your backend servers to close connections to the load balancer. To prevent possible 502 errors, ensure that your backend servers do not close idle connections in less than 310 seconds.

The Load Balancing service sets the keep-alive value to maintain the connection until it has been idle for 65 seconds. The value for this setting cannot be changed.

Idle Timeout Settings

When you create or edit a TCP or HTTP listener, you can specify the maximum idle time in seconds. This setting applies to the time allowed between two successive receive or two successive send network input/output operations during the HTTP request-response phase. If the configured timeout has elapsed with no packets sent or received, the client's connection is closed. For HTTP and WebSocket connections, a send operation does not reset the timer for receive operations and a receive operation does not reset the timer for send operations.

> **Note:**
>
> This timeout setting does not apply to idle time between a completed response and a subsequent HTTP request.

The default timeout values are: 300 seconds for TCP listeners and 60 seconds for HTTP listeners. The maximum timeout value is 7200 seconds.

Modify the timeout parameter if either the client or the backend server requires more time to transmit data; for example:

- The client sends a database query to the backend server and the database takes over 300 seconds to run. Therefore, the backend server does not transmit any data within 300 seconds.
- The client uploads data using the HTTP protocol. During the upload, the backend does not transmit any data to the client for more than 60 seconds.
- The client downloads data using the HTTP protocol. After the initial request, it stops transmitting data to the backend server for more than 60 seconds.
- The client starts transmitting data after establishing a WebSocket connection, but the backend server does not transmit data for more than 60 seconds.
- The backend server starts transmitting data after establishing a WebSocket connection, but the client does not transmit data for more than 60 seconds.

Managing Load Balancers

On Private Cloud Appliance, load balancing is the method of sharing a workload equally among servers.

Load balancing prevents clients from overwhelming certain servers and helps the service to scale. When a flexible load balancer process on a system is configurable and available to all clients and servers then it's called Load Balancer as a Service (LBaaS).

LBaaS On Private Cloud Appliance provides automated traffic distribution from one entry point to multiple servers reachable from the virtual cloud network (VCN). The service offers a load balancer with a choice of using a public or private IP address, and various load balancing policies.

When you create an LB using the OCI CLI, you can either configure all the resources at the same time or create a minimal LB and supply other configuration details later. In other words, more than the basic LB is needed to assemble the complete LB after creation. These other components are added by editing the LB resources.

The two type of load balancers are:

- Private: A private load balancer is isolated from the network outside the Private Cloud Appliance and security is therefore easier. A private load balancer gets a private IP address assigned from the address block that serves as the entry point for incoming traffic.
- Public: A public load balancer accepts traffic from a network location outside of Private Cloud Appliance. Because of this outside traffic, a public load balancer is deployed in a public subnet of a VCN that has an internet gateway (IGW) configured. The service assigns the load balancer a public IP address that serves as the entry point for incoming traffic. You can associate the public IP address with a friendly DNS name through any DNS provider, but a public IP address must be available in order for public load balancer provisioning to succeed.

This chapter creates load balancers with minimal information, using the Compute Web UI, API, or OCI CLI, and then supplies the details for the other resources. An implied order can be used in the way things are configured in the Compute Web UI (which differs from the LB resource listing) because some items must be configured before use in other places (for example, virtual hostnames are needed to complete listener configuration). After initial load balancer configuration, the order used in this section is:

- Cipher Suites and SSL Certificates
- Backend Sets and Backend Servers
- Virtual Hostnames
- Path Route Sets
- Listeners

This section also includes other parameters and commands that relate to these major configuration steps that aren't LB resources. They include:

- Backend-set health check
- Backend (server) health check
- LB health policy

This isn't an exhaustive list.

In addition, many configuration steps are processed as work requests. Work requests are tasks that can take some time to complete and therefore are tracked as they are processed and displayed as separate line items or OCI CLI output. For example, the act of creating an LB or cipher suite is a work request that records and displays start and finish times, state (succeeded, failed, and so on), and other relevant details. Work requests are part of the Compute Web UI LB resource list.

The status of a work request, such as "Succeeded," isn't the same as completion of the provisioning of the resource. In other words, load balancer creation can "succeed" while provisioning is still in progress.

Creating a Load Balancer

On Private Cloud Appliance, you can create a load balancer (LB).

> **Tip:**
>
> Cookie-based session persistence is not provided by default. It's available for application load balancers if a defined tag is applied to enable that specific implementation. For more information, see LBaaS Implementation Options on page 551.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. Select **Create Load Balancer**.
3. Enter the following information:

   - **Name:** Enter a descriptive name for the LB. The name doesn't need to be unique, and you can change it. Avoid entering confidential information.
   - **Create in Compartment:** Select the compartment in which to create the LB. The LB doesn't have to be in the same compartment as the VCN or backend set. If you aren't sure which compartment to use, create the LB in the same compartment as the VCN.
   - Select **Visibility Type:**

     - **Public Load Balancer**. The Select Public IP menu is shown. Select a public IP from the list. You might need to change the compartment above the menu. If the menu displays None Available or if you do not select a public IP from the list, a public IP is automatically assigned from the configured public IP range. You can use the assigned public IP address as a front end for incoming traffic.
     - **Private Load Balancer**. The LB receives a private IP address from the selected subnet. You can use the assigned private IP address as a front end for internal incoming VCN traffic.

     See Load Balancers on page 548 for more information about private and public load balancers.
   - **Bandwidth:** Select the shape type (either Flexible or Dynamic) and bandwidth for this LB. The bandwidth cannot be changed after the LB is created.

     - **Flexible Shapes**. Enter a minimum bandwidth and a maximum bandwidth in megabits per second (Mbps) in the fields below the type selection. To specify a fixed shape size, set the minimum and maximum bandwidth to the same value.

       The minimum bandwidth is the amount of bandwidth that is always available to provide instant readiness for the workloads.

       The maximum bandwidth is the greatest amount of bandwidth the LB supports during time of peak workload. The maximum bandwidth is set for all listeners such that all listeners share the given bandwidth inside the LB. The maximum bandwidth is a limit and is not guaranteed to be achieved. The maximum value that you can specify is 8000 Mbps.
     - **Dynamic Shapes**. Select the size from the menu below the type selection: Small (100 Mbps), Medium (400 Mbps), or Large (8000 Mbps).

       > **Note:**
       >
       > Oracle recommends using flexible shapes. Dynamic shapes are being deprecated.
   - **Subnet:** Select the names of the VCN and Subnet for the LB. You might need to change the compartment above the menus.
   - **Network Security Group:** (Optional) By default, the LB is not attached to any NSG. Select the box labeled Enable Network Security Group to add this LB to one or more NSGs.

     a. Select an NSG from the drop-down list. You might need to change the compartment to find the NSG that you want.
     b. Select **Add Another NSG** if you want to attach to another NSG.

   c. To remove an NSG from the list, select the trash can to the right of that NSG. To remove the last NSG or all NSGs, clear the **Enable Network Security Groups** box.
- **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.

4. Select **Create Load Balancer**.

   The details LB details page is displayed.

What's next?

On the LB details page, under **Resources**, select resources to create to complete the configuration.

Using the OCI CLI

Use the oci lb load-balancer create command and required parameters to create a private load balancer.

Procedure

1. Gather the information that you need to run the command:
   - Compartment OCID: `oci iam compartment list`
   - At least one subnet OCID: `oci network subnet list`
2. Run the create LB command.

   Syntax:

   The following examples show only the required parameters. Use the `-h` option to get information about optional parameters such as backend sets and listeners.

```
oci lb load-balancer create --compartment-id <compartment_OCID> --
display-name <load-balancer-name> --shape-name 400Mbps --subnet-ids
 file://<subnet_OCIDs>.json
```

```
oci lb load-balancer create --compartment-id compartment_OCID --display-
name load-balancer-name --shape-name [flexible | fixed-shape-name] --
subnet-ids file://subnet_OCIDs.json
```

   Example:

   The following example creates a private LB with a fixed bandwidth of 400 Mbps.

   The `--is-private` option value is `false` by default. If `--is-private` is omitted, a public IP address is assigned from one of the specified subnets if available. If `--is-private true` is specified, a private IP address is assigned from one of the specified subnets. See Load Balancers on page 548 for more information about private and public load balancers.

   Specify "flexible" for `--shape-name` and then specify the minimum and maximum bandwidth in megabits per second (Mbps) in the value of the `--shape-details` option. The maximum value that you can specify is 8000 Mbps. To specify a fixed bandwidth, specify the same value for both minimum and maximum.

```
--shape-details '{"maximumBandwidthInMbps": 400, "minimumBandwidthInMbps":
 400}'
```

   The bandwidth can't be changed after the LB is created.

> **Note:**
>
> We recommend specifying the "flexible" shape name and not a fixed bandwidth shape name such as 400Mbps. Dynamic shapes (fixed bandwidth shape names) are being deprecated.

```
$ oci lb load-balancer create --compartment-id ocid1.compartment.unique_ID \
--display-name Private_LB1 --shape-name 400Mbps \
--subnet-ids '["ocid1.subnet.unique_ID1", "ocid1.subnet.unique_ID2"]'
```

```
{
  "data": {
    "backend-sets": {},
    "certificates": {},
    "compartment-id": "ocid1.compartment.unique_ID",
    "defined-tags": {
      "Oracle-Tags": {
        "CreatedBy": "auser",
        "CreatedOn": "2025-01-28T23:12:58.28Z"
      },
    "display-name": "Private_LB1",
    "freeform-tags": null,
    "hostnames": {},
    "id": "ocid1.loadbalancer.unique_ID",
    "ip-addresses": [
      {
        "ip-address": "IP_address",
        "is-public": false,
        "reserved-ip": null
      }
    ],
    "is-private": true,
    "lifecycle-state": "ACTIVE",
    "listeners": {},
    "network-security-group-ids": null,
    "path-route-sets": {},
    "routing-policies": null,
    "rule-sets": {},
    "shape-details": null,
    "shape-name": "400Mbps",
    "ssl-cipher-suites": {},
    "subnet-ids": [
      "ocid1.subnet.unique_ID1",
      "ocid1.subnet.unique_ID2"
      ],
    "system-tags": null,
    "time-created": "2025-01-28T23:12:58.000001+00:00"
  },
"etag": "00c648d7-b654-4583-b7bf-k5oed55"
}
```

```
$ oci lb load-balancer create --compartment-id ocid1.compartment.unique_ID \
--display-name Private_LB1 --is-private true --shape-name flexible \
--shape-details '{"maximumBandwidthInMbps": 400, "minimumBandwidthInMbps":
 400}' \
--subnet-ids '["ocid1.subnet.unique_ID1", "ocid1.subnet.unique_ID2"]'

{
  "data": {
    "backend-sets": {},
    "certificates": {},
    "compartment-id": "ocid1.compartment.unique_ID",
    "defined-tags": {
      "Oracle-Tags": {
        "CreatedBy": "auser",
        "CreatedOn": "2025-01-28T23:12:58.28Z"
      },
    "display-name": "Private_LB1",
    "freeform-tags": null,
    "hostnames": {},
    "id": "ocid1.loadbalancer.unique_ID",
    "ip-addresses": [
      {
        "ip-address": "IP_address",
```

```
          "is-public": false,
          "reserved-ip": null
        }
      ],
      "is-private": true,
      "lifecycle-state": "ACTIVE",
      "listeners": {},
      "network-security-group-ids": null,
      "path-route-sets": {},
      "routing-policies": null,
      "rule-sets": {},
      "shape-details": {
        "maximumBandwidthInMbps": 400,
        "minimumBandwidthInMbps": 400
      },
      "shape-name": "flexible",
      "ssl-cipher-suites": {},
      "subnet-ids": [
        "ocid1.subnet.unique_ID1",
        "ocid1.subnet.unique_ID2"
        ],
      "system-tags": null,
      "time-created": "2025-01-28T23:12:58.000001+00:00"
    },
  "etag": "00c648d7-b654-4583-b7bf-k5oed55"
  }
```

This output is the same as the output of the `oci lb load-balancer get` command.

Next Steps: If you didn't create all the resources needed for the LB in the `load-balancer create` command, complete the LB configuration by adding resources using their separate commands, such as `listener create`. For a list of commands, use `oci lb -h`.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateLoadBalancer operation to create a load balancer.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing Load Balancer Details

On Private Cloud Appliance, you can view a list of existing load balancers and view their details.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the LB to go to its details page.

   Alternatively, select the Actions menu (⋮), then select the **View details** option.

Using the OCI CLI

Use the oci lb load-balancer get command and required parameters to view load balancer details.

```
oci lb load-balancer get --load-balancer-id <loadbalancer_OCID>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the GetLoadBalancer operation to view the specified load balancer configuration information.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing a Load Balancer

On Private Cloud Appliance, You can change load balancer (LB) name and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. For the load balancer you want to edit, select the Actions menu (⋮), then select **Edit**.
4. Make your changes and select **Update Load Balancer**.

Using the OCI CLI

Use the oci lb load-balancer update command and required parameters to edit load balancer properties.

```
oci lb load-balancer update --load-balancer-id <loadbalancer_OCID> --
display-name <name-of-load-balancer>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

If you didn't add resources such as backend sets or listeners when you created the LB, add them by using their separate command, such as `oci lb listener create`. If you did add resources when you created the LB, update them by using their separate command, such as `oci lb listener update`.

Using the API

Use the UpdateLoadBalancer operation to update a load balancer configuration.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Load Balancer

On Private Cloud Appliance, you can delete a load balancer (LB) and remove it from service.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. For the LB that you want to delete, select the Actions menu (⋮), and select **Terminate**.
4. Confirm the operation.

Using the OCI CLI

Use the oci lb load-balancer delete command and required parameters to remove a load balancer from service.

```
oci lb load-balancer delete --load-balancer-id <loadbalancer_OCID>
Are you sure you want to delete this resource? [y/N]: y
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteLoadBalancer operation to stop a load balancer and remove it from service.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing Cipher Suites

On Private Cloud Appliance, you can use cipher suites with a load balancer (LB) to determine the security, compatibility, and speed of HTTPS traffic.

For more details about the types of cipher suites you an use, see Cipher Suites on page 554.

Creating a Load Balancer SSL Cipher Suite

On Private Cloud Appliance, a lload balancer (LB) uses a cipher suite to secure Transport Layer Security (TLS) or Secure Socket Layer (SSL) network connections. The cipher suite defines a list of security algorithms that the LB uses to negotiate with peers exchanging information with the LB. The cipher suites used affect the security level, performance, and compatibility of data traffic.

A series of predefined cipher suites exist that you can use when you create an SSL configuration. If the predefined cipher suites don't meet requirements, you can create custom cipher suites.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the Load Balancer for which you want to create the load balancer SSL cipher suite.
4. Under **Resources**, select **Cipher Suites**.

   The list of available cipher suites appears. If none are listed, you must create one.
5. To create a load balancer cipher suite, select **Create Cipher Suite**.
6. In the **Load Balancer SSL Cipher Suite** dialog box, give the LB SSL cipher suite a name.

   > **Note:**
   >
   > The name of a user-defined cipher suite can't be the same as any of Oracle's predefined or reserved SSL cipher suite names.
7. Select the cipher suite components to be part of the SSL cipher suite.
8. Select **Create Cipher Suite**.

   To check the result, select the cipher suite name in the Cipher Suites list in the Resources section of the LB details page.

Using the OCI CLI

Use the oci lb ssl-cipher-suite create command and required parameters to create a load balancer SSL cipher suite.

```
oci lb ssl-cipher-suite create --ciphers <ssl_ciphers_complex_type> --load-
balancer-id <load-balancer_OCID> --name <ssl_cipher_suite_name>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateSSLCipherSuite operation to create a load balancer SSL cipher suite.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing a Load Balancer SSL Cipher Suite Details

On Private Cloud Appliance, you can view a list of the SSL cipher suites associated with a load balancer (LB) and how to view the details of a specific cipher suite.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the LB for which you want to view existing cipher suites.
4. Under **Resources**, select **Cipher Suites**.

   The list of cipher suites is displayed.

5. To view the details page for a cipher suite, either select the name of the cipher suite in the list, or select the Actions menu (⋮) and then select **View Details**.

Using the OCI CLI

Use the oci lb ssl-cipher-suite get command and required parameters to view a load balancer SSL cipher suite details.

```
oci lb ssl-cipher-suite get --load-balancer-id <load-balancer_OCID> --
name <cipher-suite-name>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the GetSSLCipherSuite operation to get a load balancer SSL cipher suite details.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing a Load Balancer SSL Cipher Suite

On Private Cloud Appliance, you can edit SSL cipher suites associated with a load balancer (LB) to add or remove ciphers. The name of the cipher suite can't be changed.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the Load Balancer that has the cipher suite you want to edit.
4. Under **Resources**, select **Cipher Suites**.

   The cipher suites are displayed.
5. Use one of the following methods to update a cipher suite:
   - Select the name of the cipher suite. On the cipher suite details page, select **Edit**.
   - Select the Actions menu (⋮) for the cipher suite and select **Edit**.
6. Select **Update ciphers**.

Using the OCI CLI

Use the oci lb ssl-cipher-suite update command and required parameters to update the details of a load balancer SSL cipher suite.

```
oci lb ssl-cipher-suite update --load-balancer-id <load-balancer_OCID> --
name <cipher-suite-name>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateSSLCipherSuire operation to update the details of a load balancer SSL cipher suite.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Load Balancer SSL Cipher Suite

On Private Cloud Appliance, you can delete an SSL cipher suite associated with a load balancer (LB).

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the Load Balancer that has the cipher suite you want to delete.
4. Under **Resources**, select **Cipher Suites**.

   The cipher suites are displayed.

5. Use one of the following methods to delete a cipher suite:
   - Select the name of the cipher suite. On the cipher suite details page, select the Delete button.
   - Select the Actions menu (⋮) for the cipher suite and select **Delete**.

Using the OCI CLI

Use the oci lb ssl-cipher-suite delete command and required parameters to delete a load balancer cipher suite:

```
oci lb ssl-cipher-suite delete --load-balancer-id <load-balancer_OCID> --
name <cipher-suite-name>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteSSLCipherSuire operation to delete a load balancer cipher suite.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Load Balancer SSL Certificates

On Private Cloud Appliance, you can import and manage SSL certificates through the Load Balancing service. The service doesn't generate any certificates.

An SSL certificate can be one issued by a vendor such as VeriSign or GoDaddy, or a self-signed certificate that you generate with a tool like OpenSSL or Let's Encrypt.

> **Note:**
>
> You can use a custom, self-signed SSL certificate. However, for production environments, Oracle recommends that you use a CA-issued SSL certificate, which reduces the risk of a man-in-the-middle attack.

If you configure HTTPS or SSL for a listener, an SSL server certificate must be associated with the load balancer. A certificate enables the load balancer to terminate the connection and decrypt incoming requests before passing them to the backend servers. You can apply the following SSL configurations to your load balancer:

- **SSL termination:** The load balancer handles incoming SSL traffic and passes the unencrypted request to a backend server.
- **Point-to-point SSL:** The load balancer terminates the SSL connection with an incoming traffic client, and then initiates an SSL connection to a backend server.
- **SSL tunneling:** If you configure the load balancer listener for TCP traffic, the load balancer tunnels incoming SSL connections to your application servers.

Load Balancing supports the TLS 1.2 protocol with a default setting of strong cipher strength.

To use standard SSL with a load balancer and its resources, you must supply a certificate. To use mutual TLS (mTLS) with your load balancer, you must add one or more certificate authority bundles (CA bundles) to your system.
A certificate bundle includes the public certificate, the corresponding private key, and any associated Certificate Authority (CA) certificates. We recommend that you upload the certificate bundles before creating the listeners or backend sets you want to associate them with. Only X.509 certificates in PEM format are accepted.

Load balancers commonly use single domain certificates. However, load balancers with listeners that include request routing configuration might require a subject alternative name (SAN) certificate (also called multidomain certificate) or a wildcard certificate. The Load Balancing service supports each of these certificate types.

SSL Traffic Handling Configuration

On Private Cloud Appliance, you can configure a load balancer to handle SSL traffic in different ways at different stages of the secured connection.

Terminating SSL at the Load Balancer

This configuration is known as *frontend SSL*. Your load balancer can accept encrypted traffic from a client. No encryption of traffic exists between the load balancer and the backend servers.

To terminate SSL at the load balancer, you must create a listener at a port such as 443, and then associate an uploaded certificate bundle with the listener.

Implementing Backend SSL

In a *backend SSL* configuration, the load balancer does not accept encrypted traffic from client servers. Traffic between the load balancer and the backend servers is encrypted.

To implement SSL between the load balancer and your backend servers, you must associate an uploaded certificate bundle with the backend set.

If you want to have more than one backend server in the backend set, sign your backend servers with an intermediate CA certificate. The intermediate CA certificate must be included as part of the certificate bundle.

Your backend services must be able to accept and terminate SSL.

Implementing Point-to-Point SSL

In a *point-to-point SSL* configuration, the load balancer accepts SSL-encrypted traffic from clients and encrypts traffic to the backend servers.

To implement point-to-point SSL, you must associate uploaded certificate bundles with both the listener and the backend set.

Peer Certificate Verification

On Private Cloud Appliance, peer certificate verification establishes mutual authentication between a client and load balancer.

When SSL is set up on the load balancer, the listener responds to an incoming client request with a certificate that the client can verify for authenticity. Enabling peer certificate verification adds an extra layer of security: it requires the client (peer) to present a certificate that the listener can validate.

In case peer certificate verification is configured incorrectly, the client is unable to verify the certificate and returns a client SSL handshake failure. The error message varies by client type. You can use the OpenSSL utility to check the depth at which the validation fails:

```
$ openssl verify -verbose -CAfile root-cert.pem intermediate-cert.pem
error 20 at 0 depth lookup: unable to get local issuer certificate
```

To resolve these verification errors, confirm that the client certificate and certificate authority certificate match, and ensure the certificates are included in the correct order.

SSL Key Issues

On Private Cloud Appliance, the following key-related problems are known to occur when configuring SSL for a load balancer.

Submitting Private Keys

If an error is returned when you submit a private key, these are the most common causes:

- You provided an incorrect passphrase.
- Your private key is malformed.
- The system does not recognize the encryption method used for your key.

Key Pair Mismatch

If the private and public key do not match, attempting to upload them results in a mismatch error. Use the following OpenSSL commands to confirm that the public and private key are part of the same pair:

```
openssl x509 -in certificate_name.crt -noout -modulus | openssl sha1
openssl rsa -in private_key_name.key -noout -modulus | openssl sha1
```

The `sha1` hash values returned from these commands must match exactly. If they are different, then the private key is not used to sign the public certificate, and this key cannot be used.

Private Key Consistency

In general, when errors occur that are related to a private key, you can use the OpenSSL utility to check the key's consistency. This command verifies that the key is intact, the passphrase is correct, and the file contains a valid RSA private key:

```
openssl rsa -check -in private_key.pem
```

Private Key Decryption

If the system does not recognize the encryption technology used for a private key, you should decrypt the key and upload the unencrypted version of the key with your certificate bundle. You can use the OpenSSL utility to decrypt a private key:

```
openssl rsa -in private_key.pem -out private_key_decrypted.pem
```

Managing SSL Certificates

On Private Cloud Appliance, you can use secure socket layer (SSL) certificates with your load balancer.

Uploading Certificate Chains

Use the following process for certificate chains on Private Cloud Appliance.

If there are multiple certificates that form a single certification chain – for example when intermediate certificate authority (CA) certificates are used –, then include all relevant certificates in a single PEM file in the correct order before uploading them to the system. The correct order begins with the certificate directly signed by the trusted root certificate authority at the bottom of the list. Any additional certificates are pasted above the signed certificate.

Combine the server certificate (`ssl-certificate.crt`) and the intermediate CA certificate (`intermediate-ca-cert.crt`) files into a single, concatenated PEM file:

```
cat ssl_certificate.crt intermediate-ca-cert.crt >> certbundle.pem
```

Adding a Load Balancer Certificate

On Private Cloud Appliance, you can add a public SSL certificate to use with a load balancer.

Optionally, you can also provide a certificate for a Certificate Authority (CA) or configure a private key.

> **Note:**
>
> You can use a custom, self-signed SSL certificate. However, for production environments, Oracle recommends that you use a CA-issued SSL certificate, which reduces the risk of a man-in-the-middle attack.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the LB for which you want to add the certificate.
4. Under **Resources**, select **Certificates**.
5. Select **Create Certificate**.
6. In the **Load Balancer Create Certificate** dialog box, enter the following information:

- **Name:** Enter a descriptive name for the certificate bundle. The name must be unique and can't be changed. The name can include only alphanumeric characters, dashes, and underscores. The name cannot contain spaces.
- **Public certificate:** Either upload the certificate `.pem` file, or paste the content from the `.pem` file directly into the dialog box using drag.
- **Certificate Authority:** Select the Enable certificate authority box if you are also using a certificate authority (CA) certificate. Either upload the CA certificate `.pem` file, or paste the content from the `.pem` file directly into the dialog box using drag.
- **Private Key:** Select the Enable private key box if you are also using a private key certificate. Either upload the private key `.pem` file, or paste the content from the `.pem` file directly into the dialog box using drag.

7. Select **Create Certificate**.

Using the OCI CLI

Use the oci lb certificate create command and required parameters to create a public SSL certificate to use with a load balancer.

```
oci lb certificate create --certicate-name <certificate-name> --load-
balancer-id <load-balancer_OCID> --certificate-file <path_to_file>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateCertificate operation to create a load balancer certificate.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing a Load Balancer Certificate

On Private Cloud Appliance, you can view a public SSL certificate used with a load balancer.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the LB for which you want to view the certificate.
4. Under **Resources**, select **Certificates**.

   All the details of the configured certificates are displayed.

Using the OCI CLI

Use the oci lb certificate list command and required parameters to a view the public SSL certificate used with a load balancer.

```
oci lb certificate list --load-balancer-id <load-balancer_OCID>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListCertificate operation to a view the public SSL certificate used with a load balancer.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Load Balancer Certificate

On Private Cloud Appliance, you can delete a public SSL certificate associated with an existing load balancer.

To change a certificate, delete the certificate and add a new certificate.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.

2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the LB for which you want to delete the certificate.
4. Under **Resources**, select **Certificates**.

   The configured certificates are displayed.
5. Select the Actions menu (⋮), then select **Delete**.
6. Confirm the deletion.

Using the OCI CLI

Use the oci lb certificate delete command and required parameters to delete a public SSL certificate associated with a load balancer

```
oci lb certificate delete --certificate-name <certificate_name> --load-
balancer-id <load-balancer_OCID>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteCertificate operation to delete the public SSL certificate used with a load balancer.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing Backend Sets

On Private Cloud Appliance, in the context of load balancers, the term backend refers to the components that receive, process, and respond to forwarded client requests.

Backend servers are grouped into backend sets; they receive client requests based on the configured load balancing policy. Health checks ensure that traffic only goes to healthy backend servers.

> **Tip:**
>
> Cookie-based session persistence is not provided by default. It's available for application load balancers if a defined tag is applied to enable that specific implementation. For more information, see LBaaS Implementation Options on page 551.
>
> Some backend configuration settings are specific to load balancers with session persistence. This is called out in the instructions.

Backend Sets

The backend set in a load balancer configuration consists of a list of backend servers, a load balancing policy, and a health check script.

A backend set must be associated with one or more listeners.

> **Caution:**
>
> Changing the backend set's load balancing policy interrupts traffic and may cause active connections to be dropped.

Backend Servers

Backend servers can be set up as individual compute instances or as instance pools. The transport protocol of a backend server is configured as part of the backend set. You can add and remove backend servers without disrupting traffic. When creating a load balancer, you must specify the backend servers to include in each backend set.

When you add backend servers to a backend set, you specify either the instance OCID or an IP address for the server to add. An instance with multiple VNICs attached can have multiple IP addresses pointing to it. If

you identify a backend server by OCID, the Load Balancing service uses the primary VNIC primary private IP address. If you identify the backend servers to add to a backend set by their IP addresses, it is possible to point to the same instance more than once.

The load balancer routes incoming traffic to the backend servers based on the selected load balancing policy. To route traffic to a backend server, the Load Balancing service requires the IP address of the compute instance and the relevant application port. If the backend server resides within the same VCN as the load balancer, we recommend that you specify the compute instance's private IP address. The private IP address will also work if a local peering gateway enables traffic between the load balancer VCN and the backend server's VCN. If the backend server and load balancer reside in different VCNs without peering connection, you must specify the public IP address of the compute instance. You must also ensure that the VCN security rules allow external traffic.

To enable backend traffic, your backend server subnets must have appropriate ingress and egress security rules. When you add backend servers to a backend set, you can specify the applicable network security groups (NSGs). If you prefer to use security lists for your VCN, you can configure them through the Networking service.

> **Note:**
>
> To accommodate high-volume traffic, we recommend that you use stateless security rules for your load balancer subnets. See in the chapter Virtual Networking Overview.

Creating a Load Balancer Backend Set

On Private Cloud Appliance, you can create a backend set for an existing load balancer. The backend set is a group of servers to which traffic is load balanced.

Using the OCI CLI or API, you can create backend servers when you create the backend set. Using the Compute Web UI, you must add backend servers after the backend set is created.

For more information, including how to diagnose misconfigurations, see Health Checks on page 595.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the LB for which you want to create the backend set.
4. Under **Resources**, select **Backend Sets**.

   Any existing backend sets are listed, otherwise the list says **No data available**.
5. Select **Create Backend Set**.
6. Enter the following information:

   - **Name:** Enter a descriptive name for the LB backend set. This must be unique and can't be changed. Avoid entering confidential information.
   - **Traffic Distribution Policy:** Select one of the following policies for the backend set:
     - **Weighted Round Robin:** Traffic is balanced in a "next turn" fashion, with some servers having a preference.
     - **Least Connections:** Traffic is balanced based on the server with the fewest current connections.
     - **IP Hash:** Traffic is balanced based on a hash of several fields in the IP header.
   - **Session Persistence:** (only LB with session persistence) Select the box to enable session persistence.
     - **Disable session persistence:** No method is configured to send all requests from a single client to the same backend server. Each client request is independently load balanced.
     - **Enable application cookie persistence:** The load balancer uses an existing application-generated cookie to keep the client session on the same backend server. The application issues and manages the cookie. You enter the cookie name in the applicable field.
     - **Enable load balancer cookie persistence:** The load balancer inserts and manages its own cookie to ensure a client's requests go to the same backend server, with no application changes required. You enter the

cookie name and configure to which domain and URL path it applies. Security settings determine how the cookie is exposed.

- **SSL:** Associate an SSL certificate with the backend set.
  - **Use SSL:** When you check the Use SSL box, a drop-down list of certificates appears.
  - **Certificates:** Select a certificate from the list.
  - **Verify peer certificate:** Check this box to enable peer certificate verification.
- **Health Checking:** Enter the health checking parameters to use to test the health of backend servers. These values are optional (the parameters have default values) except for protocol.
  - **Protocol:** Select the protocol to use: HTTP or TCP. Select the protocol that matches your application or service.
  - **Port:** Enter the backend server port against which to run health checks.
  - **Interval in Milliseconds:** Specify how often to run health checks, in milliseconds. Enter a number between 1 and 1,800,000.
  - **Timeout in Milliseconds:** Specify the maximum time to wait for a reply to a health check, in milliseconds. Enter a number between 1 and 600,000.
  - **Number of Retries:** Enter the number of times to retry the health check before the server is considered unhealthy.
  - **Status Code:** (HTTP only) Specify the HTTP status code a healthy server must return.
  - **URL Path:** (HTTP only) Specify a URL endpoint against which to run the health check.
  - **Response Body:** (only LB with session persistence) Enter the response body regex, such as .*, that the health checker uses.

7. Select **Create Backend Set**.

   To check the configuration, select the backend set name in the Backend Sets list in the Resources section of the LB details page.

Using the OCI CLI

Use the oci lb backend-set create command and required parameters to create a backend set for an existing load balancer.

```
oci lb backend-set create --health-checker-protocol [HTTP | TCP] --load-
balancer-id <load-balancer_OCID> --name <backend-set-name> --policy <load-
balancer-policy>
```

Example:

1. Get the LB OCID: `oci lb load-balancer list`
2. Construct an argument for the `--backends` option.

   The `--backends` option is a list of backend server definitions in the following JSON format. For brevity, only one list item is shown in the following output:

```
$ oci lb backend-set create --generate-param-json-input backends
[
  {
    "backup": true,
    "drain": true,
    "ipAddress": "string",
    "offline": true,
    "port": 0,
    "weight": 0
  }
]
```

3. Run the `create backend set` command.

   The create backend set command has many options. Use the `-h` option to learn about them.

This example creates a set of backend servers for the backend set. The backends are defined in a file named `backendsSet1`. You can also create backend servers later.

```
oci lb backend-set create --health-checker-protocol TCP \
--load-balancer-id ocid1.loadbalancer.unique_ID --name BackendSet1 \
--policy LEAST_CONNECTIONS --backends file://./backendsSet1 \
--health-checker-port 22 --health-checker-return-code 200

{
  "opc-work-request-id": "ocid1.workrequest.unique_ID"
}
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateBackendSet operation to create a backend set for an existing load balancer.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing a Load Balancer Backend Set

On Private Cloud Appliance, you can view a list of backend sets of a load balancer (LB) and how to view the configuration information and the list of servers in a backend set.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the LB for which you want to view the backend set details.
4. Under **Resources**, select the **Backend Sets**.

   The list of backend sets for this LB is shown.
5. Select the name of the backend set that you're interested in.
6. View the backend set details:

   - Select the **Backend Set Information** tab to see the load balancing policy and the overall health of the servers in the set.
   - Select the B**ackend Set Configuration** tab to see health checker and SSL configuration details.
   - Under **Resources**, view the list of backend servers.

Using the OCI CLI

Use the oci lb backend-set get command and required parameters to view a list of the servers in a backend set of an existing load balancer and view their details.

```
oci lb backend-set get --load-balancer-id <loadbalancer_OCID>
--backend-set-name <backendset_name>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the GetBackendSet operation to get a list of the servers in a backend set of an existing load balancer and view their details.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing a Load Balancer Backend Set

On Private Cloud Appliance, you can change load balancer (LB) backend set properties, such as the health checker protocol that's used.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.

2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the LB for which you want to modify a backend set.
4. Under **Resources**, select **Backend Sets**.
5. For the backend set that you want to modify, select the Actions menu (⋮), then select **Edit**.
6. In the Edit Load Balancer Backend Set dialog box, make your changes.
7. Select **Update Load Balancer Backend Set**.

Using the OCI CLI

Use the oci lb backend-set update command and required parameters to change load balancer (LB) backend set properties, such as the health checker protocol used.

```
oci lb backend-set update --load-balancer-id <loadbalancer_OCID> --backend-
set-name <backendset_name> --backends <list_of_server_definitions> --health-
checker-protocol [HTTP | TCP] --policy <load-balancer-policy>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateBackendSet operation to change load balancer backend set properties.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Load Balancer Backend Set

On Private Cloud Appliance, you can delete a load balancer (LB) backend set and remove it from service.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the load balancer for which you want to delete the backend set.
4. Under **Resources**, select **Backend Sets**.
5. For the LB backend set you want to delete, select the Actions menu (⋮), then select **Terminate**.
6. Confirm the deletion.

Using the OCI CLI

Use the oci lb backend-set delete command and required parameters to delete a load balancer backend set and remove it from service.

```
$ oci lb backend-set delete --load-balancer-id <oloadbalancer_OCID> --
backend-set-name <name> --force
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteBackendSet operation to delete a load balancer backend set and remove it from service.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing Backend Servers

On Private Cloud Appliance, when creating a load balancer, you must specify the backend servers to include in each backend set.

Creating a Load Balancer Backend Server

On Private Cloud Appliance, you can create a backend server to add to a backend set. A backend set is a group of backend servers to which traffic is load balanced.

If you create a backend server in a VCN that's not the same as the load balancer (LB) VCN, then you must set up a Local Peering Gateway to enable communication between the LB and the backend server. See Connecting VCNs through a Local Peering Gateway on page 510.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the load balancer for which you want to create a backend server.
4. Under **Resources**, select **Backend Sets**.
5. In the list of backend sets, select the name of the backend set for which you want to create a backend server.
6. On the backend set details page, select **Create Backend**.
7. Enter the following information:
   - **Computed Instances**
     - **Instance:** If you select Computed Instances, the IP Address area presents a drop-down list of instances. Select one of these instances for a backend server. You can change the compartment above the list.
     - **Port:** The server port to load balance.
     - **Weight:** The load balancing policy weight assigned to the server. Backend servers with a higher weight receive a larger proportion of incoming traffic. For example, a server with weight 3 receives 3 times the number of new connections as a server with weight 1. For more information about load balancing policies, see Load Balancing Policies on page 553.
     - **Security Rules:** To enable load balancer traffic, you must add ingress and egress security rules to the corresponding subnets.

       If you select **Configure Manually**, then when you are finished adding backends, go to the VCN and create or update a security list to add security rules, and ensure that the applicable subnet is using that security list. See Controlling Traffic with Security Lists on page 496.

       If you select Configure Automatically, a table of egress rules is shown and a table of ingress rules is shown. Each table lists the name of the security list, the name of the subnet, and the rule: the CIDR block and port to allow egress or ingress traffic. Select the button to the right of the rule to enable or disable that rule for each backend that you added.
   - **IP Addresses**
     - **IP Address:** If you select IP Addresses, the IP Address area presents a text field where you must enter the IP address of the instance that you want to use as a backend server.
     - **Port:** The server port to load balance.
     - **Weight:** See the previous description under Compute Instances.

       To add another backend, select **Add IP Address**. You can select an instance or you can specify an IP address that's already a member of the backend set if you specify a different port.
8. Select **Submit**.

Using the OCI CLI

Use the oci lb backend create command and required parameters to create a backend server to add to an existing load balancer backend set.

```
oci lb backend create --load-balancer-id <load-balancer_OCID> --backend-set-
name <backend-set-name> --ip-address <backend-svr-ip-addr> --port <port-
number>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateBackend operation to add a backend server to a backend set.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing Load Balancer Backend Server Details

On Private Cloud Appliance, you can view a list of the backend servers in a load balancer (LB) backend set and view the configuration details of a specific backend.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the load balancer for which you want to list backend servers.
4. Under **Resources**, select Backend Sets to see the list of backend sets for this LB.
5. Select the name of a backend set to see the list of backends in that backend set. For each backend in the list, the IP address, port, weight, and other attributes are shown.

Using the OCI CLI

Use the oci lb backend list command and required parameters to list the backend servers in a load balancer backend set.

```
oci lb backend list --backend-set-name <backend-set-name> --load-balancer-
id <load-balancer_OCID>
```

Use the oci lb backend get command and required parameters to get the details of a load balancer backend server.

```
oci lb backend get --backend-name <backend_name> --backend-set-
name <backend_set_name> --load-balancer-id <load_balancer_id> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListBackend operation to view a list of the backend servers in an existing load balancer backend set and view their details.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing a Load Balancer Backend Server

On Private Cloud Appliance, you can change the weight, drain, offline, and backup attributes of a load balancer (LB) backend server.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the load balancer for which you want to edit backend server.
4. Under **Resources**, select **Backend Sets**.
5. Select the name of the backend set that contains the backend server that you want to edit.
6. Select the name of the LB backend server that you want to edit.
7. Select the Actions menu (⋮), then select **Edit**.
8. Make allowable changes in the dialog box.
9. Select **Submit**.

Using the OCI CLI

Use the oci lb backend update command and required parameters to edit a load balancer backend server.

```
oci lb backend update --load-balancer-id <load-balancer_OCID> --backend-set-
name example_backend_set --backend-name 10.0.0.3:8080 --backup false --drain
 false --offline false --weight 3
```

The `--backup`, `--drain`, `--offline`, and `--weight` options are required; They don't have default values. You can also specify `--max-connections`. Use the `-h` option to see descriptions of these options.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateBackend operation to change the properties of a load balancer backend server that's a member of a backend set.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Load Balancer Backend Server

On Private Cloud Appliance, you can delete a load balancer (LB) backend server from a backend set and remove it from the service.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the load balancer for which you want to delete the backend server.
4. Under **Resources**, select **Backend Sets**.
5. Select the name of the backend set that contains the backend server you want to delete.
6. Select the Actions menu (⋮), then select **Terminate**.
7. Confirm the deletion.

Using the OCI CLI

Use the oci lb backend delete command and required parameters to remove a backend server from a given load balancer and backend set.

```
oci lb backend delete --backend-name <backend_server_name> --backend-set-
name <backendset_name> --load-balancer-id <load-balancer_OCID>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteBackend operation to remove a backend server from a given load balancer and backend set.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing Virtual Hostnames

On Private Cloud Appliance, you can use virtual hostnames with a load balancer for one or more listeners.

Creating a Load Balancer Virtual Hostname

On Private Cloud Appliance, a virtual hostname is associated with a load balancer (LB) and used by one or more listeners. Hostnames associated with a listener correspond to the backend set of that listener. The backend set routes traffic to specific backends which host different applications.

Virtual hostnames simplify the construction of the hostnames associated with listeners and backend servers because virtual hostnames can use wild card asterisks (*) at the start or end of the hostname. Listeners detect a hostname pattern that matches the virtual hostname patterns created.

> **Note:**
>
> The asterisk (*) doesn't have to be used in a virtual hostname. However, when used, the asterisk can only be added at the beginning or ending of a virtual hostname. Traffic sent to ***`app.example.com`*** is load balanced by a listener and backend server set when configured as `app.example.com` (exact), `*example.com` (wild card at start), or `app.example*` (wild card at end).

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the LB for which you want to create the virtual hostname.
4. Under **Resources**, select **Hostnames**.
5. Select **Create Hostname**.
6. Enter the following information:
   - **Name:** A name for the LB virtual hostname.
   - **Hostname:** The virtual hostname.
7. Select **Create Hostname**.

Using the OCI CLI

Use the oci lb hostname create command and required parameters to create a virtual hostname associated with a load balancer.

1. Get the LB OCID: `oci lb load-balancer list`
2. Run the create virtual hostname command.

   Syntax:

   ```
   oci lb hostname create --load-balancer-id <load-balancer_OCID> --
   hostname <virtual-hostname> --name virtual-hostname-friendly-name
   ```

   Example:

   ```
   $ oci lb hostname create --load-balancer-id ocid1.loadbalancer.unique_ID
    --hostname *example.com --name my_virtual_hostname

   {
     "opc-work-request-id": "ocid1.workrequest.oc1.pca.unique_ID"
   }
   ```

   To verify that the hostname was created, see Viewing a Load Balancer Virtual Hostname on page 585.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

This task isn't available in the API.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing a Load Balancer Virtual Hostname

On Private Cloud Appliance, you can view the virtual hostnames associated with a load balancer (LB).

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the Load Balancer for which you want to view the virtual hostnames.

4. Under **Resources**, select **Hostnames**.

Using the OCI CLI

Use the oci lb hostname list command and required parameters to list all hostname resources associated with the specified load balancer..

```
oci lb hostname list --load-balancer-id <load-balancer_OCID>
```

Use the oci lb hostname get command and required parameters to get the specified hostname resource configuration information.

```
oci lb hostname get --load-balancer-id <load-balancer_OCID>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

This task isn't available in the API.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing a Load Balancer Virtual Hostname

On Private Cloud Appliance, you can change the virtual hostname associated with a load balancer (LB).

Virtual hostnames can use wild card asterisks (*) at the start or end of the hostname. Listeners detect a hostname pattern that matches the virtual hostname patterns created.

> **Note:**
>
> The asterisk (*) doesn't have to be used in a virtual hostname. However, when used, the asterisk can only be added at the beginning or ending of a virtual hostname. Traffic sent to ***app.example.com*** is load balanced by a listener and backend server set when configured as ***app.example.com*** (exact), ****example.com*** (wild card at start), or ***app.example**** (wild card at end).

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the Load Balancer for which you want to edit the virtual hostnames.
4. Under **Resources**, select **Hostnames**.
5. In the Actions menu (⋮), select **Edit**.
6. Edit the virtual hostname.

   To change the name of the virtual hostname, delete the virtual hostname and create a new one.
7. Select **Save Changes**.

Using the OCI CLI

Use the oci lb hostname update command and required parameters to change the virtual hostname associated with a load balancer.

```
oci lb hostname update --hostname <virtual-hostname> --load-balancer-
id <load-balancer_OCID> --name <virtual-hostname-friendly-name>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

This task is not available in the API.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Load Balancer Virtual Hostname

On Private Cloud Appliance, you can delete a virtual hostname associated with a load balancer (LB).

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the load balancer for which you want to delete the virtual hostname.
4. Under **Resources**, select **Hostnames**.
5. Select the Actions menu (⋮), then select **Delete**.
6. Confirm the deletion.

Using the OCI CLI

Use the oci lb hostname delete command and required parameters to delete the virtual hostname associated with a load balancer.

```
oci lb hostname delete --load-balancer-id <load-balancer_OCID> --
name <virtual-hostname-friendly-name>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

This task is not available in the API.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing Path Route Sets

On Private Cloud Appliance, you can apply a set of path routes to a load balancer (LB) to determine the appropriate destination backend set for incoming URIs.

Some applications have multiple endpoints or content types, each distinguished by a unique URI path such as `/admin/`, `/data/`, `/video/`, or `/cgi/`. Each rule in a path route set names a backend set, a partial URI to match, and the pattern match type.

Path route rules route traffic to the correct backend set without requiring multiple listeners or LBs.

A path route set includes all path route rules that define the data routing for a particular listener. You can have at most one path route set for each listener. You can specify at most 20 path route rules for each path route set.

Path route rules apply only to HTTP, HTTP/2, and HTTPS requests. Path route rules do not apply to TCP requests.

Path route rule URL strings have the following restrictions:

- You can't use asterisks in path route strings.
- You can't use regular expressions.
- Path route string matching is case-insensitive. For example, both "data" and "DATA" match.

Browsers often add an ending slash to the path in a request. You might want to configure a rule with a URL string that includes the trailing slash and a second rule with a URL string that does not include the trailing slash (for example,`/admin` and `/admin/`).

A path route rule pattern match type is one of the following:

- EXACT_MATCH: The path string must match the incoming URI path exactly.
- FORCE_LONGEST_PREFIX_MATCH: The path string must match longest ("best") match of the beginning portion of the incoming URI path.
- PREFIX_MATCH: The path string must match the beginning portion of the incoming URI path.

- SUFFIX_MATCH: The path string must match the ending portion of the incoming URI path.

Creating Path Route Sets

On Private Cloud Appliance, you can create a path route set to route URIs to load balancer (LB) backend sets. To create a path route set, a backend set must already exist.

See Managing Path Route Sets on page 587 for information about URI patterns, pattern match types, and path route set limits.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. Select the name of LB for which you want to create the path route set.
3. Under **Resources**, select **Path Route Sets**.
4. Select **Create Path Route Set**.
5. Enter the following information:
   - **Name:** Enter a descriptive name for the Path Route Set. Avoid entering confidential information.
   - **Path Route Rules:** Enter the following information:
     - **Match Style:** Select Exact Match, Force Longest Prefix Match, Prefix Match, or Suffix Match. The match style must match the URL string entered next.
     - **URL String:** Enter the pattern to match.
     - **Backend Set:** Select the name of the backend set from the drop-down list.

   To create another rule, select the New Rule button.
6. Select **Create Path Route Set**.

Using the OCI CLI

Use the oci lb path-route-set create command and required parameters to create a path route set for a load balancer resource to distinguish by unique URL paths such as `/admin/`, `/data/`, `/video/`, or `/cgi/`.

```
oci lb path-route-set create --name <name-of-path-route-set> --path-
routes [COMPLEX-TYPE] --load-balancer-id <load-balancer_OCID>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreatePathRouteSet operation to add a path route set to a load balancer.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing Path Route Sets

On Private Cloud Appliance, you can view path route set details for a load balancer resource.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of load balancer for which to view path route set details.
4. Select Path Route Sets under the Resources for the load balancer.
5. Select the Actions menu (⋮), then select **View Details**.

Using the OCI CLI

Use the oci lb path-route-set get command and required parameters to view path route set details for a load balancer resource.

```
oci lb path-route-set get --load-balancer-id <load-balancer_OCID> --path-
route-set-name <name-of-path-route-set>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the GetPathRouteSet operation to get the specified path route set's configuration information.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing Path Route Sets

On Private Cloud Appliance, you can edit a path route set to update the details.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of load balancer for which to update the path route set details.
4. Under **Resources**, select **Path Route Sets**.
5. Select the Actions menu (⋮), then select **Edit**.
6. Select **Save Changes**.

Using the OCI CLI

Use the oci lb path-route-set update command and required parameters to update path route set details for a load balancer resource.

```
oci lb path-route-set update --load-balancer-id <load-balancer_OCID> --path-
route-set-name <name-of-path-route-set>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdatePathRouteSet operation to change the specified path route set's configuration information.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting Path Route Sets

On Private Cloud Appliance, you can delete a path route set for a load balancer resource.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of load balancer for which to delete the path route set.
4. Under **Resources**, select **Path Route Sets**.
5. Select the Actions menu (⋮), then select **Delete**.
6. Select **Confirm**.

Using the OCI CLI

Use the oci lb path-route-set delete command and required parameters to delete a path route set resource.

```
oci lb path-route-set delete --load-balancer-id <load-balancer_OCID> --path-
route-set-name <name-of-path-route-set>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeletePathRouteSet operation to delete a path route set resource.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing Load Balancer Rule Sets

On Private Cloud Appliance, you associate rule sets with listeners. The rules define actions to be applied to traffic passing through the listener.

A rule set is a named set of rules associated with a load balancer and applied to one or more listeners on that load balancer. To apply a rule set to a listener, you first create the rule set that contains the rules. Rules are objects that represent actions applied to traffic at a load balancer listener. The rule set becomes a part of the load balancer's configuration. You can specify the rule set to use when you create or update a listener for the load balancer.

You can apply the same rule set to several listeners on the same load balancer. Rule sets aren't shared between load balancers. To use the same set of rules on another load balancer, you must create a new, identical rule set under that load balancer.

You can include the following types of rules in a rule set:

- Access control rules, which restrict access to application resources based on the source of the request.
- Request header rules, which add or remove HTTP request headers.
- Response header rules, which add or remove HTTP response headers.

Creating a Load Balancer Rule Set

On Private Cloud Appliance, you create a rule set first, so you can associate it with a listener for a load balancer (LB).

Not all OCI rule set actions are supported on Private Cloud Appliance.

- Allow rules can only specify source IP address.
- HTTP header rules can only add or remove request and response headers, not alter them.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. Select the name of LB for which you want to create the rule set.
3. Under **Resources**, select **Rule Sets**.
4. Select **Create Rule Set**.
5. Enter the following information:
   - **Name:** Enter a descriptive name for the Rule Set. Avoid entering confidential information.
   - **Rules:** Select a type from the list and enter the following information:
     - **Add HTTP Request Header:** Enter the name and value of the header to add.
     - **Remove HTTP Request Header:** Enter the name of the header to remove.
     - **Add HTTP Response Header:** Enter the name and value of the header to add.
     - **Remove HTTP Response Header:** Enter the name of the header to remove.
     - **Allow:** Enter the IP address from where the HTTP requests are forwarded to the backend. The listener responds to requests from other clients with 403: Forbidden.

     To create another rule, select the Add Rule button.
6. Select **Create Rule Set**.

Using the OCI CLI

Use the oci lb rule-set create command and required parameters to create a rule set for a load balancer.

```
oci lb rule-set create --name <name-of-rule-set> --load-balancer-id <load-
balancer_OCID> --items [COMPLEX-TYPE]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateRuleSet operation to create a rule set for a load balancer.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing a Load Balancer Rule Set

On Private Cloud Appliance, you can view the rule sets associated with a load balancer.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of LB for which you want to view the rule sets.
4. Under **Resources**, select **Rule Sets**.
5. To view the details of a rule set, select the Actions menu (⋮), then select **View Details**.

Using the OCI CLI

Use the oci lb rule-set list command and required parameters to display all rule sets for a load balancer.

```
oci lb rule-set list --load-balancer-id <load-balancer_OCID>
```

Use the oci lb rule-set get command and required parameters to display the details of a particular rule set associated with a load balancer.

```
oci lb rule-set get --load-balancer-id <load-balancer_OCID> --rule-set-
name <name-of-rule-set>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListRuleSets operation to display all rule sets for a load balancer.

Use the GetRuleSet operation to display the details of a particular rule set associated with a load balancer.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing a Load Balancer Rule Set

On Private Cloud Appliance, you can update an existing rule set for a load balancer.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of LB for which you want to update the rule set.
4. Under **Resources**, select **Rule Sets**.
5. Select the Actions menu (⋮), then select **Edit**.
6. Select **Save Changes**.

Using the OCI CLI

Use the oci lb rule-set update command and required parameters to update a rule set for a load balancer.

```
oci lb rule-set update --rule-set-name <name-of-rule-set> --load-balancer-
id <load-balancer_OCID> --items [COMPLEX-TYPE]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateRuleSet operation to update a rule set for a load balancer.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Load Balancer Rule Set

On Private Cloud Appliance, you can remove an existing rule set from a load balancer.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of LB for which you want to delete the rule set.
4. Under **Resources**, select **Rule Sets**.
5. Select the Actions menu (⋮), then select **Delete**.
6. Select **Confirm**.

Using the OCI CLI

Use the oci lb rule-set delete command and required parameters to delete a rule set for a load balancer.

```
oci lb rule-set delete --rule-set-name <name-of-rule-set> --load-balancer-
id <load-balancer_OCID>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteRuleSet operation to delete a rule set for a load balancer.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing Listeners

On Private Cloud Appliance, you can use listeners to check for incoming traffic on the load balancer IP address.

Creating a Load Balancer Listener

On Private Cloud Appliance, you can configure a listener for a load balancer (LB). The listener checks for incoming traffic on the LB IP address. Configure at least one listener per traffic type: HTTP, HTTP/2, HTTPS, and TCP.

Prerequisites

1. Ensure that your VCN's security rules allow the listener to accept traffic.
2. Know the traffic protocols that the LB accepts.
   - Using the Compute Web UI: On the LB details page, under Resources, select **Create Listener**. The supported traffic protocols are shown on the **Protocol** drop-down list.
   - Using the OCI CLI: `oci lb protocol list -c compartment_OCID`
3. If the LB accepts HTTPS traffic, and you plan to select HTTPS, create a certificate and cipher suite for use in SSL configuration. See Adding a Load Balancer Certificate on page 575 and Creating a Load Balancer SSL Cipher Suite on page 571.
4. At least one backend set must exist for this LB. See Managing Backend Sets on page 577.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the Load Balancer for which you want to create the listener.

   Any existing listeners are listed under the Load Balancer Information, otherwise the list says **No data available**.
4. Select **Create Listener**.

5. Enter the following information. Name, protocol, port, and backend set are required. Other parameters, such as hostnames, path route sets, rule sets, SSL certificate, and cipher suites can be added later.
   - **Name:** Enter a descriptive name for the Listener. The name must be unique and cannot be changed.
   - **Protocol:** Select the protocol to listen for (HTTP, HTTP/2, HTTPS, TCP) from the drop-down list.

     If you select HTTP/2, HTTPS, or TCP, an SSL section appears at the bottom of the dialog, after Idle Timeout in Seconds.
   - **Port:** A default port value, depending on the protocol you selected, is preselected. Use the up or down arrows to change the port value, or enter a value between 1 and 65,535.
   - **Backend Set:** Select a backend set from the list.
   - **Hostnames:** Select a hostname from the list.
   - **Path Route Set:** Select a path route set from the list.
   - **Idle Timeout in Seconds:** Use the up or down arrows to change the idle timeout value, or enter a value greater than or equal to 1.
   - **Rule Sets:** Select Add Rule Set and select a rule set from the list. Repeat to add more rule sets.
   - **SSL:** This item appears if you select HTTP/2, HTTPS, or TCP for protocol.
     - **Use SSL:** For HTTP/2 and HTTPS protocols, this box is selected and can't be cleared. For TCP protocol, this box isn't checked. If you check this box, the following parameters appear, just as they do if you selected HTTP/2 or HTTPS protocol.

       Selecting this box enables SSL handling for this listener. The following settings are required to associate an SSL certificate bundle with the listener to enable SSL handling.
     - **Certificates:** Select a certificate from the list.
     - **Verify peer certificate:** Select this box to enable peer certificate verification.
     - **TLS Version:** Select at least one TLS version.
     - **Cipher Suite:** Select a cipher suite from the list.
6. Select **Create Listener**.

   To display the details of the listener, you must view the details for the entire load balancer.

Using the OCI CLI

Use the oci lb listener create command and required parameters to create a listener for an existing load balancer. You must list the traffic protocols that the load balancer accepts before you create the listener.

Procedure

1. Get the LB OCID: `oci lb load-balancer list`
2. Run the create listener command.

   Syntax:

```
oci lb listener create --default-backend-set-name backend-set-name \
--load-balancer-id load-balancer_OCID \
--name listener-name --port listener-port \
--protocol listener-protocol
```

   Option values:

   • *backend-set-name* - The name of the associated backend set.

   • *load-balancer_OCID* - The OCID of the load balancer on which to add a listener.

   • *listener_name* - A user-friendly name for the listener. It must be unique and it can't be changed.

   • *listener-port* - The communication port number for the listener.

   • *listener-protocol* - The protocol on which the listener accepts connection requests.

   Example:

This example shows only required parameters. Other parameters, such as hostnames, path route sets, SSL certificate, and cipher suites can be added later. Use the `-h` option for more information.

```
$ oci lb listener create --default-backend-set-name PublicLB1-Backend-Set
 --load-balancer-id ocid1.loadbalancer.unique_ID --name LB1-Listener --
port 80 --protocol HTTP

{
 "opc-work-request-id": "ocid1.workrequest.oc1.pca.loadbalancer.unique_ID"
}
```

To view the listener details, use the `load-balancer get` command. In the following example, most of the command output is omitted to show only the listener details:

```
$ oci lb load-balancer get --load-balancer-id ocid1.loadbalancer.unique_ID
{
  "data": {
    ...
    "listeners": {
      "LB1_Listener": {
        "connection-configuration": {
          "backend-tcp-proxy-protocol-version": null,
          "idle-timeout": 60
          },
        "default-backend-set-name": "PublicLB1-Backend-Set",
        "hostname-names": null,
        "name": "LB1_Listener",
        "path-route-set-name": null,
        "port": 80,
        "protocol": "HTTP",
        "routing-policy-name": null,
        "rule-set-names": null,
        "ssl-configuration": null
        }
      },
    ...
    },
  "etag": "9326dbb5-d842-4975-9cfb-ced7717e92d6"
}
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateListener operation to create a listener for an existing load balancer.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing a Load Balancer Listener

On Private Cloud Appliance, you can change load balancer (LB) listener properties, such as the listener communication port.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the LB for which you want to modify a backend set.
4. Under **Resources**, select **Listeners**.
5. Select the name of the load balancer listener that you want to edit.
6. For the LB listener that you want to edit, select the Actions menu (⋮), then select **Edit**.

7. In the **Edit Listener** dialog box, you can change anything except the name of the listener.
8. Select **Save Changes**.

Using the OCI CLI

Use the oci lb listener update command and required parameters to change load balancer listener properties, such as the listener communication port used load.

```
oci lb listener update --default-backend-set-name <default-backendset-name>
 --listener-name <listener-name> --load-balancer-id <loadbalancer_OCID>\
--port <port-integer> --protocol <protocol-text>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateListener operation to change load balancer listener properties.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Load Balancer Listener

On Private Cloud Appliance, you can delete a load balancer (LB) Listener.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the LB for which you want to delete a listener.
4. Under **Resources**, select **Listeners**.
5. For the listener that you want to delete, select the Actions menu (⋮), then select **Delete**.
6. Confirm the deletion.

Using the OCI CLI

Use the oci lb listener delete command and required parameters to delete a load balancer listener and remove it from service.

```
oci lb listener delete --listener-name <listener-name-text> --load-balancer-
id ocid1.loadbalancer.....….….….uniqueID
Are you sure you want to delete this resource? [y/N]: y
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteListener operation to delete load balancer listener.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Health Checks

Load balancer health checks are tests to confirm the availability of backend servers.

These tests occur in the form of a request or a connection attempt depending on the protocol. The health check policy includes a time interval you specify, to ensure that the backend servers are continuously monitored. If a server fails a health check, the load balancer takes the server temporarily out of rotation. If the server later passes a subsequent health check, then the LB returns that backend server to the balancing rotation.

The health check policy is configured when you create a backend set. You can configure TCP-level or HTTP-level health checks for your backend servers. For backend sets configured with SSL the health checks also use SSL encryption.

- TCP-level health checks attempt to make a TCP connection with the backend servers and validate the response based on the connection status.
- HTTP-level health checks send requests to the backend servers at a specific URI and validate the response based on the status code or entity data (body) returned.

Configure your health check protocol to match your application or service. If you run an HTTP service, then configure an HTTP-level health check. If you run a TCP-level health check against an HTTP service, then you might not get an accurate response. The TCP handshake can succeed and indicate that the service is up even when the HTTP service is incorrectly configured or having other issues. Although the health check returns no errors, you might experience transaction failures.

For example:

- The backend HTTP service has issues when communicating with the health check URL and the health check URL returns 5*nn* messages. An HTTP health check catches the message from the health check URL and marks the service as down. In this case, a TCP health check handshake succeeds and marks the service as healthy, even though the HTTP service might not be usable.
- The backend HTTP service responds with 4*nn* messages because of authorization issues or no configured content. A TCP health check does not catch these errors.

The service provides application-specific health check capabilities to help you increase availability and reduce your application maintenance window.

Health status indicators are used to report the general health of a load balancer and its backend servers/sets. The possible statuses are: `ok`, `warning`, `critical`, `unknown`. Health status is updated every three minutes. No finer granularity is available. Historical health data is not provided.

Interpreting Load Balancer Health Issues

At the highest level, load balancer health reflects the status of its components. The health status indicators provide information you might need to drill down into and investigate further. Below are several common issues that the health status indicators can help you detect and correct.

Health check misconfigured

All the backend servers for one or more of the affected listeners report as unhealthy. If your investigation finds that the backend servers do not have problems, then a backend set probably includes a misconfigured health check.

Listener misconfigured

All the backend server health status indicators report OK, but the load balancer does not pass traffic on a listener. The listener might be configured to listen on the wrong port, use the wrong protocol, or use the wrong policy. If your investigation shows that the listener is not at fault, check the security rule configuration.

Security rule misconfigured

Health status indicators help you diagnose these cases of misconfigured security rules:

- All health status indicators report OK, but traffic does not flow (as with misconfigured listeners). If the listener is not at fault, check the security rule configuration.
- All health status indicators report as unhealthy. You have checked your health check configuration and your services run properly on your backend servers. In this case, your security rules might not include the IP range for the source of the health check requests. The source IP for health check requests belongs to a compute instance managed by the Load Balancing service.

> **Note:**
>
> Traffic may also be blocked because of misconfigured route tables in the compute instances.

Backend server unhealthy

A backend server might be unhealthy or the health check might be misconfigured. To see the corresponding error code, check the status field in the backend server's details through the Compute Web UI or OCI CLI.

Common Side Effects of Load Balancer Health Check Misconfiguration

Misconfiguration scenarios are known to occur regularly. This page helps with troubleshooting.

Wrong Port

In this scenario, all backend servers are reported as unhealthy. If you confirmed that there are no issues with the backend servers, you might have made a mistake setting the port. Traffic must be allowed, and the backend must be listening on that port.

Wrong Path

In this scenario, all backend servers are reported as unhealthy. If you confirmed that there are no issues with the backend servers, you might have made a mistake setting the path for the HTTP health check. It needs to match an actual application on the backend.

You can use the curl utility to run a test from a system within the same network. For example: `curl -i http://backend_ip_address/health.`

You receive the configured status code in the response:

```
"msg":"invalid statusCode","statusCode":404,"expected":"200"
```

Wrong Protocol

In this scenario, all backend servers are reported as unhealthy. If you confirmed that there are no issues with the backend servers, for an HTTP health check you might have made a mistake setting the status code. It must match the actual status code being returned from the backend. A typical mismatch is when a backend returns a 302 status code while a 200 status code is expected. This is often caused by the backend directing you to a login page or another location on the server. You can either make the backend return the expected code or use 302 in your health check configuration.

Error message: `msg:invalid statusCode, statusCode:nnn,expected:200` (where `nnn` represents the actual status code returned).

Wrong Regex Pattern

Applies only to load balancers with session persistence enabled. In this scenario, all backend servers are reported as unhealthy. If you confirmed that there are no issues with the backend servers, you might have made a mistake setting a regex pattern that's not consistent with the body, or the backend isn't returning the expected body. In this scenario, you can either change the backend to match the pattern or correct the pattern to match the backend.

Error message: `response match resulte: failed`.

Misconfigured Security Rules

In this scenario, all or some backend servers report as unhealthy. If you confirmed that there are no issues with the backend servers, then you might have improperly configured either the network security groups, security lists, or local firewalls (such as firewalld, iptables, or SELinux).

In this scenario, you can use either the curl or netcat utilities to run a test from a system within the same subnet and network security group (NSG) as your load balancer instance HTTP. For example: `curl -i http://backend_ip_address/health TCP or nc -zvw3 backend_ip_address 443`.

Local firewalls can be verified by using the command: firewall-cmd --list-all --zone=public. If the expected rules are missing from the firewall configuration, then add the required service. For example, to add HTTP port 80:

```
firewall-cmd --zone=public --add-service=http
```

```
firewall-cmd --zone=public --permanent --add-service=http
```

Viewing Health Status and Health Check Configuration

On Private Cloud Appliance, you can view the overall health of a load balancer (LB) and backend set of servers and the specific health checker configuration values set for the backend set.

Overall Health

- Critical
- Warning
- Incomplete
- Pending
- OK

Health Checker Configuration Parameters

- Protocol - HTTP or TCP
- Port - The backend server port against which to run the health check
- Interval In Milliseconds - Time between health checks
- Timeout In Milliseconds - Maximum time to wait for the health check response
- Number of Retries - The number of retries to attempt before a backend server is considered "unhealthy"
- Status Code (HTTP only) - The code a healthy backend server should return
- URL Path (HTTP only, Optional) - The path against which to run the health check

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the LB for which you want to view health status and configuration.

   On the LB details page, the Overall Health of the LB is shown in the second column of the Load Balancer **Information** tab.
4. On the LB details page, under **Resources**, select **Backend Sets**.
5. Select the name of the backend set for which you want to view health status and configuration.

   On the backend set details page, the Overall Health of the servers in the backend set is shown in the first column of the Backend Set **Information** tab.
6. On the backend set details page, select the **Backend Set Configuration** tab.

   All health check configuration parameters listed at the beginning of this section are shown in the **Health Checker Configuration** column.
7. On the backend set details page, under Resources, select **Backends**.

   The overall health of each backend server is shown in the **Overall Health** column.

Using the OCI CLI

Use the oci lb backend-set-health get command and required parameters to view the parameters used by the backend set of an existing load balancer to check the health of the backend servers in the set.

```
oci lb backend-set-health get --backend-set-name <backend-set-name-text> --
load-balancer-id <load-balancer_OCID>
```

Procedure

1. Gather the information you need to run the command:
   - Compartment OCID: `oci iam compartment list`
   - LB OCID: `oci lb load-balancer list`
   - Backend set name: `oci lb backend-set list`

2. List the health statuses for all load balancers in the specified compartment.

```
$ oci lb load-balancer-health list --compartment-id compartment_OCID
```

3. Show the health status for the specified load balancer.

```
$ oci lb load-balancer-health get --load-balancer-id loadbalancer_OCID
```

4. Show the health status for the specified backend set.

```
$ oci lb backend-set-health get --load-balancer-id loadbalancer_OCID \
--backend-set-name backendset_name
```

5. Show the health status of the specified backend server.

```
$ oci lb backend-health get --load-balancer-id loadbalancer_OCID \
--backend-set-name backendset_name --backend-name backend_name
```

   The *backend_name* is the IP address and port of the backend server to retrieve the health status for, such as 10.0.0.3:8080.

6. Show the health check policy information for the specified load balancer and backend set.

```
$ oci lb health-checker get --load-balancer-id loadbalancer_OCID \
--backend-set-name backendset_name
```

   Health check configuration parameters listed at the beginning of this section are shown.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the GetBackendSetHealth operation to view the parameters used by the backend set of an existing load balancer to check the health of the backend servers in the set.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing Backend Set Health Check Parameters

On Private Cloud Appliance, you can change load balancer backend set properties, such as the health check interval.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the load balancer for which you want to list the backend set health check parameters.
4. Under **Resources**, select **Backend Sets**.
5. To edit the health check parameters of a backend set, you can:
   a. Select the Actions menu (⋮), then select **Edit**.
   b. Select the backend set to view its details, then select **Edit**.
6. Select **Save**.

Using the OCI CLI

Use the oci lb backend-set update command and required parameters to change load balancer backend set properties, such as the health check interval.

```
oci lb backend-set update --backend-set-name [<name-of-backend-set>]--
backends [<complex-type>] --health-checker-protocol [protocol-text] --load-
balancer-id [loadbalancer_OCID] --policy [<policy-text>] --force
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateBackendSet operation to change load balancer backend set properties, such as the health check interval.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing Load Balancer Health Policy

On Private Cloud Appliance, you can view the policy used by the load balancer to check backend set health.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the load balancer for which you want to list backend sets.
4. Under **Resources**, select **Backend Sets**.
5. To view the health check parameters of a backend set, you can:
   a. Select the Actions menu (⋮), then select **View Details**.
   b. Select the backend set.
6. On the Backend-Set Details page, under the backend-set name for which to you want to view the health policy, select **Backend Set Configuration**.

Using the OCI CLI

Use the oci lb health-checker get command and required parameters to view the policy used by the load balancer to check backend set health.

```
oci lb health-checker get --backend-set-name [<backend-set-name-text>] --
load-balancer-id <load-balancer_OCID>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

This task is not available in the API.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing Load Balancer Health Check Properties

On Private Cloud Appliance, you can change load balancer (LB) backend set health check configuration.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the LB for which you want to edit backend set health check parameters.
4. Under **Resources**, select **Backend Sets**.
5. For the backend set that you want to modify, select the Actions menu (⋮), and select **Edit**.
6. Make changes.
7. Select **Update Backend Set**.

Using the OCI CLI

Use the oci lb health-checker update command and required parameters to change load balancer (LB) and backend server set health check properties, such as the health check interval.

Procedure

1. Gather the information you need to run the command:
   - LB OCID: `oci lb load-balancer list`

- Backend set name: `oci lb backend-set list`

2. Update the health check policy for the specified load balancer and backend set.

   Syntax:

   ```
   ci lb health-checker update --load-balancer-id <loadbalancer_OCID> \
   --backend-set-name <backendset_name> --interval-in-millis <integer> \
   --port <integer> --protocol [HTTPS | TCP] --retries <integer> \
   --return-code <integer> --timeout-in-millis <integer>
   ```

   Example:

   ```
   $ oci lb health-checker update --load-balancer-id
    ocid1.loadbalancer.uniqueID --backend-set-name BackendSet1 --interval-in-
   millis 10000 --port 8080 --protocol HTTPS --retries 3 --return-code 200 --
   timeout-in-millis 3000 --url-path /healthcheck
   ```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

This task is not available in the API.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Network Load Balancers

On Private Cloud Appliance, you can configure the Network Load Balancing (NLB) feature to automatically distribute network traffic.

Network Load Balancer (NLB) provides automated traffic distribution from one entry point to multiple servers reachable from the virtual cloud network (VCN). NLBs, like LBs, offer a choice of using a public or private IP address and various load balancing policies.

NLB Essentials

When working with network load balancers (NLBs), understanding the building blocks and related key terminology is important.

Backend Server

An NLB has a set of backend servers to route incoming traffic to destinations while balancing the load placed on each individual server in the set.

A backend server is an application server responsible for generating content in reply to the incoming client traffic. You typically identify application servers with a unique combination of overlay (private) IPv4 address and port number. For example, `10.10.10.1:8080` and `10.10.10.2:8080`.

> **Note:**
>
> The backend server can't function as both a client and a backend simultaneously as is unable to be a traffic source to the NLB virtual IP (VIP).

Backend Set

The backend set is a logical entity that includes:

- A list of backend servers (compute instances) that accept traffic from the NLB.
- A network-level load balancing policy to distribute traffic to the backend servers in a consistent manner.
- A health check policy to decide if the backend servers are functioning and can accept traffic from the NLBs.

Network Load Balancing Policy

A network load balancing policy tells the NLB how to distribute incoming traffic to backend servers in the backend set.

A NLB uses an IP hash mapped from an *n-tuple*, which is an ordered and finite list of elements. These elements are typically drawn from the fields in a packet header or from items from the grid of a relational database. The *n* indicates how many fields the tuple contains.

Allowed NLB policies for the IP hash include:

- A hash on 5 fields in the TCP/IP header (called "5-Tuple Hash")
- A hash on 3 fields in the TCP/IP header (called "3-Tuple Hash")
- A hash on 2 fields in the TCP/IP header (called "2-Tuple Hash"

> **Note:**
>
> NLB policies are different than LBaaS policies. NLB policies don't handle IP addresses and port numbers directly. Hashing is a way to manipulate these and other packet header fields to provide a number that the NLB can use without processing the entire packet header. The more tuples that are used for the load balancing policy, the more processing required for each load balanced packet, but the wider the range of values that can be used by the NLB to distribute traffic.

Health Check

A health check is a test to confirm the availability of backend servers. A health check can be a request or a connection try. Based on a time interval you specify, the NLB applies the health check policy to continuously monitor backend servers. If a server fails the health check, the NLB takes the server temporarily out of rotation. If the server later passes the health check, the NLB returns it to the rotation.

You configure a health check policy when you create a backend set. TCP-level health checks try to make a TCP connection with the backend servers and validate the response based on the connection status

The health check results update the health status of the NLB. The health status is an indicator that reports the general health of NLBs and their components.

Listener

A logical entity that checks for incoming traffic on the NLB IP address. You configure a listener's port number and various optional settings. When creating a listener, the VCN security rules must let the listener accept traffic.

Work Request

An object that reports on the current state of an NLB request. Some NLB configuration tasks can take much longer to complete than others. Work requests let NLBs handle configuration steps asynchronously and form a queue. This lets the user issue more configuration commands without waiting for the completion of a previous configuration directive.

NLB Frontend Configuration

On Private Cloud Appliance, in the context of network load balancers (NLB), the term *frontend* refers to the components that a client can see and send requests to. The entry point of client requests is the outward-facing floating IP address of the NLB.

Three main configuration tasks are needed for the NLB frontend:

- Type (public or private)
- Policy (the number of tuples used for the hash value)
- Listeners (Port number and other parameters that the NLB uses when waiting for traffic)

See Managing Network Load Balancers on page 604.

NLB Types

On Private Cloud Appliance, there are two types of network load balancers (NLBs): *public* and *private*.

Private NLBs use a private IP address to serve as the entry point for incoming traffic. The load balancing service requires only one subnet to host both the primary and secondary load balancers. However, the NLB is only accessible from within the VCN that contains the host subnet, or as further restricted by security rules.

Public NLBs use a public IP address and accept traffic from a network location outside of Private Cloud Appliance. However, the public NLB must be deployed in a public subnet of a VCN that has an internet gateway (IGW) correctly configured.

NLB Policies

On Private Cloud Appliance, a network load balancer (NLB) policy acts on an *n-tuple* mapped to an IP hash instead of directly on the packet header. A tuple is an ordered and finite list of elements. For an NLB, these elements to be hashed are drawn from the fields in a packet header. The *n* indicates how many fields the tuple contains.

Hashing generates a smaller number of bits than used in the concatenated fields before the hashing algorithm is applied. Packets that have the same hash value are treated the same way by the NLB. The more tuples that are used for the load balancing policy, the more processing required for each load balanced packet, but the wider the range of values that can be used by the NLB to distribute traffic.

For compatibility with OCI, these NLB policies can be selected:

- 5-Tuple Hash: a default policy based on a hash on 5 fields in the TCP/IP header (source IP, source port, destination IP, destination port, protocol)
- 3-Tuple Hash: a default policy based on a hash on 3 fields in the TCP/IP header (source IP, destination IP, protocol)
- 2-Tuple Hash: a default policy based on a hash on 2 fields in the TCP/IP header (source IP, destination IP)

However, due to implementation differences, any selected NLB policy in Private Cloud Appliance is always mapped internally to an IP hash based only on the source IP address of an incoming request. The NLB routes requests from the same client to the same backend server as long as that server is available. This policy honors server weight settings when establishing the initial connection.

NLB Listeners

On Private Cloud Appliance, a listener is a logical entity at the ingress side of the network load balancer (NLB) setup. This key component detects incoming traffic on the NLB IP address. It listens for requests from clients using a particular protocol and port. Requests are then routed to the appropriate backend servers based on the rules defined in the NLB configuration. You must configure a listener.

When you create a listener, ensure that the VCN security rules let the listener accept traffic.

> **Note:**
>
> To accommodate high-volume traffic, we recommend that you use *stateless* security rules for NLB subnets. For more information, see Virtual Firewall on page 485.

See Managing NLB Listeners on page 614.

NLB Backend Configuration

On Private Cloud Appliance, in the context of network load balancers (NLBs), the term *backend* refers to the components that receive, process, and respond to forwarded client requests. Backend servers are grouped into backend sets; they receive client requests based on the configured load balancing policy. Health checks ensure that traffic only goes to healthy backend servers.

NLB Backend Sets

On Private Cloud Appliance, the backend set in a network load balancer (NLB) configuration consists of a list of backend servers, an NLB policy, and a health check script. Backend sets help to group backend servers together and make them easier to configure and manage form an NLB perspective.

A backend set must be associated with one or more listeners.

See Managing NLB Backend Sets on page 607.

NLB Backend Servers

On Private Cloud Appliance, when creating a network load balancer (NLB), you must specify the backend servers to include in each backend set.

Backend servers can be set up as individual compute instances or as instance pools. You can add and remove backend servers without disrupting traffic.

TCP is the transport protocol of a backend server and is configured as part of the backend set.

When you add backend servers to a backend set, you specify either the instance OCID or an IP address for the server to add. An instance with multiple VNICs attached can have multiple IP addresses pointing to it. If you identify a backend server by OCID, the NLB uses the primary VNIC primary private IP address. If you identify the backend servers to add to a backend set by their IP addresses, it is possible to point to the same instance more than once.

The NLB routes incoming traffic to the backend servers based on the configured load balancing policy. To route traffic to a backend server, the NLB requires the IP address of the compute instance and the relevant application port. If the backend server resides within the same VCN as the NLB, we recommend that you specify the compute instance's private IP address. The private IP address also works if a local peering gateway enables traffic between the NLB VCN and the backend server VCN. If the backend server and NLB reside in different VCNs without a peering connection, you must specify the public IP address of the compute instance. You must also ensure that the VCN security rules allow external traffic.

To enable backend traffic, your backend server subnets must have appropriate ingress and egress security rules. When you add backend servers to a backend set, you can specify the applicable network security groups (NSGs). If you prefer to use security lists for your VCN, you can configure them through the Networking service.

See Managing NLB Backend Servers on page 610.

Managing Network Load Balancers

On Private Cloud Appliance, you can create, view, edit and delete Network Load Balancers (NLBs).

For conceptual information about NLBs, see Network Load Balancers on page 601.

When you create an NLB, you can either configure all the resources at the same time or create a minimal NLB and supply other configuration details later. More than the basic NLB is needed to establish the NLB service. You add more components by editing the NLB resources.

Many configuration steps are processed as work requests. Work requests are tasks that can take some time to complete and therefore are tracked as they're processed. For example, the act of creating an NLB or cipher suite is a work request that records and displays start and finish times, state (succeeded, failed, and so on), and other relevant details. Work requests are part of the NLB resource list.

The status of a work request, such as succeeded, isn't the same as completion of the provisioning of the resource. NLB creation can succeed while provisioning is still in progress.

Creating a Network Load Balancer

On Private Cloud Appliance, you can create a private Network Load Balancer (NLB) using the Compute Web UI, OCI CLI, or API.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. Select Create **Load Balancer**.
3. Enter the following information:
   - **Name:** Enter a descriptive name for the NLB. Avoid entering confidential information.
   - **Create in Compartment:** Select the compartment in which to create the NLB.
   - **Visibility Type:** Select either **Public** or **Private Load Balancer**.
     - If you select **Private Load Balancer**, the NLB receives a private IP address from the selected subnet.
     - If you select **Public Load Balancer**, the **Select Public IP** menu is shown. Select a public IP from the list. You might need to change the compartment above the menu. If the menu displays **None Available** or if

you don't select a public IP from the list, a public IP is automatically assigned from the configured public IP range.

For more information about private and public load balancers see NLB Types on page 603.

- **Subnet:** Select the name of the VCN and Subnet for the NLB. You might need to change the compartment above the menus.
- **Network Security Group:** (Optional) By default, the NLB is not attached to any NSG. Select the box labeled **Enable Network Security Group** to add this NLB to one or more NSGs.
  a. Select an NSG from the drop-down list. You might need to change the compartment to find the NSG you want.
  b. Select the Add Another NSG button to attach to another NSG.
  c. To remove an NSG from the list, select the trash can to the right of that NSG. To remove the last NSG or all NSGs, clear the Enable Network Security Groups box.
- **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.

4. Select **Create Network Load Balancer**. The details page of the new NLB is displayed.

What's Next?

On the NLB details page, under Resources, select resources to create to complete the configuration.

Using the OCI CLI

Use the oci nlb network-load-balancer create command and required parameters to create a new private NLB in the specified compartment.

```
oci nlb network-load-balancer create --compartment-id <compartment_OCID>
 --display-name <name-of-network-load-balancer> --is-private true --subnet-
id <subnet_OCID>
```

Procedure

1. Gather the information that you need to run the command:
   - Compartment OCID (`oci iam compartment list`)
   - At least one subnet OCID (`oci network subnet list`)
2. Enter the create NLB command.

   Syntax:

   The following shows only the required parameters. Use `oci nlb network-load-balancer create -h` to get information about optional parameters such as backend sets and listeners.

   ```
   oci nlb network-load-balancer create --compartment-id compartment_OCID \
   --display-name name-of-network-load-balancer \
   --subnet-id subnet_OCID
   ```

   Example:

   The following example creates a private NLB (the `--is-private` option value is `true` by default), and a private IP address is assigned from the specified subnet. See Load Balancers on page 548 for more information about private and public load balancers.

   ```
   $ oci nlb network-load-balancer create --compartment-id
    ocid1.compartment.unique_ID \
     --display-name Private_NLB1 --subnet-id ocid1.subnet.unique_ID

   {
     "data": {
       "backend-sets": {},
       "compartment-id": "ocid1.tenancy.unique_ID",
   ```

```
    "defined-tags": {
      "Oracle-Tags": {
        "CreatedBy": "auser",
        "CreatedOn": "2025-01-28T23:12:58.28Z"
      },
    "display-name": "Private_NLB1",
    "freeform-tags": null,
    "hostnames": {},
    "id": "ocid1.networkloadbalancer.unique_ID",
    "ip-addresses": [
    {
      "ip-address": 10.10.1.16,
      "is-public"; false,
      "reserved-ip": null
    }
    ],
    "is-preserve-source-destination": false,
    "is-private": true,
    "lifecycle-details": null,
    "lifecycle-state": "ACTIVE",
    "listeners": {},
    "network-security-group-ids": null,
    "nlb-ip-version": "IPV4",
    "subnet-id":"ocid1.subnet.unique_ID",
    "system-tags": null,
    "time-created": "2025-01-28T23:12:58.000001+00:00",
    "time-updated": null
  },
"etag": "00c648d7-b654-4583-dbdb-k5oed55"
}
```

This output is the same as the output of the `oci nlb network-load-balancer get` command.

What's Next?

Complete the NLB configuration by adding resources using their separate commands, such as `oci nlb listener create`. For a list of commands, see `oci nlb -h`.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateNetworkLoadBalancer operation to create a new private NLB in the specified compartment.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing NLB Details

On Private Cloud Appliance, you can view a list of existing network load balancers (NLBs) and view their details.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. At the top of the page, select the compartment that contains the NLB.
3. Select the name of the NLB to go to its details page.

   Alternatively, for the NLB for which you want to see the details, select the Actions menu (⋮) and select the View details option.

Using the OCI CLI

Use the following commands to view NLBs:

- Use oci nlb network-load-balancer list to see a list of NLBs.

```
oci nlb network-load-balancer list [OPTIONS]
```

- Use oci nlb network-load-balancer get to see the details of a particular NLB.

```
oci nlb network-load-balancer get --network-load-balancer-
id networkloadbalancer_OCID  [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the following operations to view NLBs:

- Use ListNetworkLoadBalancers to list all NLBs.
- Use GetNetworkLoadBalancer to get the details of a particular NLB.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing NLB Backend Sets

On Private Cloud Appliance, you can use backend sets to create logical entities consisting of a network load balancing (NLB) policy, health check policy, and a list of backend servers for a Network Load Balancer resource.

Creating an NLB Backend Set

On Private Cloud Appliance, you can create a backend set for an existing network load balancer (NLB). The backend set is a group of servers to which network traffic is load balanced. You can create backend servers *after* you create the backend set, or at the same time. This topic creates only the backend set.

> **Note:**
>
> The Private Cloud Appliance implementation of network load balancers is different from OCI.
>
> - The NLB policy is always mapped to an IP hash based on the source IP address of the incoming request. For OCI compatibility, you must select one of the available hashes (5-, 3-, or 2-tuple).
> - The source IP of the packet is not preserved. The `--is-preserve-source` parameter for backend sets is always disabled. Therefore, the `--is-preserve-source-destination` parameter cannot be enabled for the NLB.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the Network Load Balancer for which you want to create the network load balancer backend set.
4. Under **Resources**, select **Backend Sets**.

   Any existing backend sets are listed, otherwise the list says **No data available**
5. Select **Create Backend Set**.
6. Enter the following information:

   - **Name:** Specify a friendly name for the backend set. It must be unique within the network load balancer, and can't be changed. Valid backend set names include only alphanumeric characters, dashes, and underscores. Backend set names can't contain spaces. Avoid entering confidential information.
   - **Load Balancing Policy:** The *IP Hash* policy uses an incoming request's source IP address as a hashing key to route "non-sticky" traffic to the same backend server. The load balancer routes requests from the same client to the same backend server as long as that server is available. This policy honors server weight settings when establishing the initial connection.

Internally, any policy you select is mapped to IP Hash. For OCI compatibility, select one of these policies: following load balancing policies: 5-tuple hash, 3-tuple hash, 2-tuple hash.

- **Source Header Preservation:** The default value *False* can't be changed.
- **IP Protocol Version:** The network load balancer listener and backend set must use the same IP protocol version. Accepted values are: IPV4.
- **Health Check:** Specify the parameters to confirm the health of backend servers in the set:
  - **Protocol:** Enter the protocol: TCP or HTTP. HTTP is valid for NLB health checks. When using TCP as the protocol, you can optionally provide the request data and the response data.
  - **Port:** Specify the backend server port against which to run the health check. You can enter the value '0' to have the health check use the backend server's traffic port.
  - **Interval in MS:** Specify how often to run the health check in milliseconds. The default value is 10000 (10 seconds).
  - **Timeout in MS:** Specify the maximum time in milliseconds to wait for a reply to a health check. A health check is successful only if a reply returns within this timeout period. The default is 3000 (3 seconds).
  - **Number of Retries:** Specify the number of retries to try before a backend server is considered "unhealthy." This number also applies when recovering a server to the "healthy" state. The default is 3.
  - **Status Code:** Specify the status code a healthy backend server must return.
  - **URL Path (URI):** Specify a URL endpoint against which to run the health check.
  - **Response Body Regex:** Provide a regular expression for parsing the response body from the backend server.

7. Select **Create Backend Set**.

   To display the details of the new backend set, view the backend set.

Using the OCI CLI

Use the oci nlb backend-set create command and required parameters to add a backend set to a network load balancer.

```
oci nlb backend-set create [OPTIONS]
```

Procedure

1. Get the NLB OCID (`oci nlb network-load-balancer list`)
2. Run the create backend set command.

   Syntax:

```
oci nlb backend-set create --health-checker health-checker-parameters
 --name backend-set-name --network-load-balancer-id network-load-
balancer_OCID \
--policy network-load-balancer-policy
```

   Where:

   • ***health-checker-parameters*** is the set of parameters associated with the health checker for this backend set.

   - **Protocol:** Enter the protocol: TCP. Configure your health check protocol to match your application or service. When using TCP as the protocol, you can optionally provide the request data and the response data.
   - **Port:** Specify the backend server port against which to run the health check. You can enter the value '0' to have the health check use the backend server's traffic port.
   - **Interval in MS:** Specify how frequently to run the health check in milliseconds. The default value is 10000 (10 seconds).
   - **Timeout in MS:** Specify the maximum time in milliseconds to wait for a reply to a health check. A health check is successful only if a reply returns within this timeout period. The default is 3000 (3 seconds).
   - **Number of Retries:** Specify the number of retries to attempt before a backend server is considered "unhealthy." This number also applies when recovering a server to the "healthy" state. The default is 3.
   - **Status Code:** Specify the status code a healthy backend server must return.

- **URL Path (URI):** Specify a URL endpoint against which to run the health check.
- ***backend-set-name*** is the name specific to the backend set.
- ***network-load-balancer_OCID*** is the OCID of the NLB.
- ***network-load-balancer-policy*** is the policy associated with the load balancer.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateBackendSet operation to add a backend set to a network load balancer.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing an NLB Backend Set

On Private Cloud Appliance, you can view a list of the backends in an existing network load balancer backend set and view their details.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the network load balancer (NLB) to which you want to list existing backends.
4. Under **Resources**, select the name of the existing Backend Set to view its details, such as IP address, port, and weight.

Using the OCI CLI

Use the oci nlb backend-set list command and required parameters to list all backend sets associated with a network load balancer.

```
oci nlb backend-set list [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the GetBackendSet operation to list details for a backend set.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing an NLB Backend Set

On Private Cloud Appliance, you can change some properties of a network load balancer (NLB) backend that's a member of a backend set.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the network load balancer for which you want to edit backend properties.
4. Under **Resources**, select **Backend Sets**.
5. Select the name of the Backend Set with the backend that you want to edit.
6. Select the name of the NLB backend that you want to edit.
7. In the Actions menu (⋮), select **Edit**.
8. Make allowable changes in the dialog box.
9. Select **Update Network Load Balancer Backend**.

Using the OCI CLI

Use the oci nlb backend-set update command and required parameters to update a backend set.

```
oci nlb backend-set update [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateBackendSet operation to update a backend set.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting an NLB Backend Set

On Private Cloud Appliance, you can delete a Backend Set from a network load balancer.

Deleting a backend set removes its backend servers from the network load balancer.

Before you can delete a backend set, you must remove it from any active listeners.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. At the top of the page, select the compartment that contains the network load balancer.
3. Select the name of the network load balancer for which you want to delete the backend server.
4. Under **Resources**, select **Backend Sets**.
5. For the backend set you want to delete, select the Actions menu (⋮), and select **Delete**.
6. Confirm the operation when prompted.

Using the OCI CLI

Use the oci nlb backend-set delete command and required parameters to delete the specified backend set from a network load balancer.

```
oci nlb backend-set delete [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteBackendSet operation to delete the specified backend set from a network load balancer.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing NLB Backend Servers

On Private Cloud Appliance, manage the backend servers that receive incoming traffic based on the policies you specified for the backend set that contains it for the network load balancer (NLB).

When you create an NLB, you must specify the backend servers (compute instances) to include in each backend set. The NLB routes incoming traffic to these backend servers based on the policies you specified for the backend set.

For more conceptual information, see NLB Backend Servers on page 604.

Creating an NLB Backend Server

On Private Cloud Appliance, you can create backend servers *after* you create the backend set, or at the same time. This topic creates only the backend server.

> **Note:**
>
> The Private Cloud Appliance implementation of network load balancers is different from OCI. Backend servers cannot be set as backup units, cannot be

> drained, and cannot be marked as offline. These optional backend create and update parameters are not supported: `--is-backup`, `--is-drain`, `--is-offline`.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the Network Load Balancer for which you want to create the network load balancer backend server.
4. Under **Resources**, select **Backend sets**.

   Any existing backend servers are listed, otherwise the list displays **No data available**.
5. Select the hyperlink of the BackendSet to add the backend.
6. Select Create **Backend**.
7. Select the method for adding the backend server. Possible values are:
   - **Computed Instances** Backend servers are added by instance.
   - **IP Addresses** Backend servers are added by instance IP address.
8. For **Computed Instances**, enter the following information:
   - **Instance:** Enter the name of the backend server.
   - **Port:** Enter 22 (TCP).
   - **Name:** Leave blank to take the default (ip_address:port#).
   - **Weight:** Enter a weight in the range 1 to 100.
   - **Security Rules:** Select the security rules of the backend. Possible values are:
     - **Configure Manually** You manually configure security rules for the backend.
     - **Configure Automatically** The system automatically configures security rules for the backend.
9. For **IP Addresses**, enter the following information:
   - **IP Address:** Enter the IP address of the backend server.
   - **Port:** Enter the port number.
   - **Name:** Enter the name of the backend.
   - **Weight:** Enter a weight in the range 1 to 100.
10. Select **Submit**.

Using the OCI CLI

Use the oci nlb backend create command and required parameters to add a backend server to a backend set.

```
oci nlb backend create [OPTIONS]
```

Procedure

1. Gather the information you need to run the command:
   - Compartment OCID (`oci iam compartment list --compartment-id-in-subtree true`)
   - NLB OCID (`oci nlb load-balancer list --compartment-id` ***`compartment_OCID`***)
2. Run the backend create command.

   Use the NLB OCID and backend set name to create the backend set for the LB in the specified compartment.

> **Note:**
>
> For information about optional parameters, run the command with the `--help` option.

Syntax (entered on a single line):

```
oci nlb backend create --backend-set-name [backend-set-name] --ip-address
 [text]\
  --network-load-balancer-id [network-load-balancer_OCID] \
  --port [port-number-integer]
```

Where:

• ***backend-set-name*** is the name specific to the backend set where the backend is to be added.

• ***ip-address*** is the IP address for the backend to be added.

• ***network-load-balancer_OCID*** is the OCID of the load balancer associated with the backend set and servers.

• ***port-number-integer*** is the port number associated with the backend.

Example:

```
oci nlb backend create --backend-set-name PrivTCP_NLB1BESet \
  --ip-address 10.10.1.13 \
  --network-load-balancer-id ocid1.networkloadbalancer..….….uniqueID \
  --port 22

{
  "opc-work-request-id": "ocid1.workrequest.....….….….uniqueID "
}
```

> **Note:**
>
> The command returns a work request ID. To see the backend set results, you must list backends associated with the specified NLB and verify that the backend created is listed. Use the ***oci nlb backend list*** command to view the backend details.

```
oci nlb backend list --network-load-balancer-id
 ocid1.networkloadbalancer.....….….….uniqueID

{
  "data": {
    "items": [
    {
      "ip-address": "10.10.1.13",
      "is-backup": false,
      "is-drain": false,
      "is-offline": false,
      "name": "10.10.1.13:22",
      "port": 22,
      "target-id": "ocid",
      "weight": 1
      },
      {
      "ip-address": "10.10.1.14",
      "is-backup": false,
      "is-drain": false,
      "is-offline": false,
      "name": "10.10.1.14:22",
      "port": 22,
      "target-id": "ocid",
      "weight": 3
      },
    ]
  }
```

```
}
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateBackend operation to add a backend server to a backend set.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing an NLB Backend Server

On Private Cloud Appliance, you can view a list of the backend servers in an existing network load balancer backend set and view their details.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the network load balancer (NLB) to which you want to list existing backend servers.
4. Under **Resources**, select **Backend sets**.
5. Select the name of the backend set.
6. Under **Resources**, select **Backends**.

Using the OCI CLI

Use the oci nlb backend list command and required parameters to list the backend servers for a particular network load balancer and backend set.

```
oci nlb backend list [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the GetBackend operation to list the backend servers for a particular network load balancer and backend set.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing an NLB Backend Server

On Private Cloud Appliance, you can change some properties of a network load balancer (NLB) backend server.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the network load balancer for which you want to edit backend properties.
4. Under **Resources**, select **Backend Sets**.
5. Select the name of the backend set with the backend server that you want to edit.
6. For the backend server you want to edit, select the Actions menu (⋮), then select **Edit**.
7. Make allowable changes in the dialog box.
8. Select **Update Network Load Balancer Backend**.

Using the OCI CLI

Use the oci nlb backend update command and required parameters to update the configuration of a backend server within the specified backend set.

```
SYNTX [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateBackend operation to update the configuration of a backend server within the specified backend set.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting an NLB Backend Server

On Private Cloud Appliance, you can delete a network load balancer (NLB) Backend server from a Backend Set.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the network load balancer for which you want to delete the backend server.
4. Under **Resources**, select **Backend Sets**.
5. Select the name of the backend set that contains the backend server you want to delete.
6. For the backend server you want to delete, select the Actions menu (⋮), then select **Delete**.
7. Confirm the operation when prompted.

Using the OCI CLI

Use the oci nlb backend delete command and required parameters to remove a backend server from a particular network load balancer and backend set.

```
oci nlb backend delete [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteBackend operation to remove a backend server from a particular network load balancer and backend set.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing NLB Listeners

On Private Cloud Appliance, use listeners to check for incoming traffic on the network load balancer IP address.

Creating am NLB Listener

On Private Cloud Appliance, you can create a listener for an existing network load balancer (NLB).

A listener waits for traffic to arrive for an IP address and distributes the traffic to the backend set servers of the NLB. Configure at least one listener for each traffic type. Ensure that your VCN security rules allow the listener to accept traffic.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the name of the Network Load Balancer for which you want to create the listener.
4. On the network load balancer's details page, under **Resources**, select **Listeners**.
5. Select **Create Listener**.
6. Enter the following information:

   - **Name:** Enter a descriptive name for the Listener. Avoid entering confidential information.
   - **Protocol:** Select TCP, UDP, or TCP_AND_UDP from the drop-down list.
   - **Port:** Enter the port you want to use. For TCP, the default port value 22 is preselected. Use the up or down arrows to change the port value, or enter a value between 1 and 65,535.
   - **Backend Set:** Select the backend set for the listener from the drop-down list. If the value is **None Available**, then you haven't yet created any NLB backend sets and must do so before this parameter can be configured.

- **IP Version:** The default IP Version 4 is preselected.

7. Select **Create Listener**.

   To display the details of the listener, you view the details for the network load balancer. See Viewing NLB Details on page 606.

Using the OCI CLI

Use the oci nlb listener create command and required parameters to add a listener to a network load balancer.

```
oci nlb listener create --default-backend-set-name backend-set-name \
--network-load-balancer-id network-load-balancer_OCID \
--name listener-name --port listener-port \
--protocol listener-protocol  [OPTIONS]
```

Procedure

1. Gather the information you need to run the command:
   - Compartment OCID (`oci iam compartment list`)
   - NLB OCID (`oci nlb network-load-balancer list`)
   - NLB accepted protocol list (`oci nlb protocol list`)
2. Run the create listener command.

   Syntax:

   ```
   oci nlb listener create --default-backend-set-name backend-set-name \
   --network-load-balancer-id network-load-balancer_OCID \
   --name listener-name --port listener-port \
   --protocol listener-protocol
   ```

   Where:

   - ***backend-set-name*** is the name of the associated backend set.
   - ***network-load-balancer_OCID*** is the OCID of the load balancer on which to add a listener.
   - ***listener-name*** is a user-friendly name for the listener. It must be unique and it cannot be changed.
   - ***listener-port*** is the communication port integer for the listener.
   - ***listener-protocol*** is the protocol on which the listener accepts connection requests: TCP, UDP, or TCP_AND_UDP.

   For information about optional parameters, use `oci nlb listener create -h`.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateListener operation to add a listener to a network load balancer.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing an NLB Listener

On Private Cloud Appliance, you can change some network load balancer (NLB) listener properties.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. At the top of the page, select the compartment that contains the load balancer.
3. Select the Network Load Balancer for which you want to edit listeners.
4. Under **Resources**, select **Listeners**.
5. Select the name of the NLB listener that you want to edit.

6. For the NLB listener you want to edit, select the Actions menu (⋮), then select **Edit**.
7. Make allowable changes to the listener in the dialog box.
8. Select **Update Network Load Balancer Listener**.

Using the OCI CLI

Use the oci nlb listener update command and required parameters to update a listener for a particular network load balancer.

```
oci nlb listener update --default-backend-set-name default-backendset-name
 \
--listener-name listener-name --network-load-balancer-
id networkloadbalancer_OCID\
--port port-integer --protocol protocol-text [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateListener operation to update a listener for a particular network load balancer.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting an NLB Listener

On Private Cloud Appliance, you can delete a network load balancer (NLB) Listener and remove it from service.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. At the top of the page, select the compartment that contains the NLB.
3. Select the name of the network load balancer (NLB) or which you want to list existing listeners.
4. Under **Resources**, select **Listeners**.
5. Select the name of the NLB listener you want to delete.
6. On the row with the NLB listener you want to delete, select the Actions menu (⋮), then select **Delete**.
7. Confirm the operation when prompted.

Using the OCI CLI

Use the oci nlb listener delete command and required parameters to delete the NLB listener.

```
oci nlb listener delete [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteListener operation to delete the NLB listener.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Checking NLB Health

On Private Cloud Appliance, use health status indicators to check the health of your network load balancers (NLBs) and their resources.

You can see health status indicators and summaries in the NLBs, backend sets, and backend servers.

On Private Cloud Appliance, Network Load Balancer (NLB) health checks are tests to confirm the availability of backend servers. These tests occur in the form of an attempt to make a TCP connection with the backend servers and validate the response based on the connection status.

The health status of the specified backend set server is reported by the primary and standby NLB.

The health check policy includes a time interval you specify, to ensure that the backend servers are continuously monitored. If a server fails a health check, the NLB takes the server temporarily out of rotation. If the server later passes the health check, the NLB returns it to the rotation.

The health check policy is configured when you create a backend set.

The TCP handshake can succeed and indicate that the service is up even when the HTTP service is incorrectly configured or having other issues. Although the health check returns no errors, you might experience transaction failures.

The service provides application-specific health check capabilities to help you increase availability and reduce your application maintenance window.

Health status indicators are used to report the general health of an NLB and its backend servers and sets. The possible statuses are:

- `ok` – No attention required. The resource is functioning as expected.
- `warning` – Some reporting entities require attention. The resource is not functioning at peak efficiency or the resource is incomplete and requires further work.
- `critical` – Some or all reporting entities require immediate attention. The resource isn't functioning or unexpected failure is imminent.
- `unknown` – Health status cannot be determined. The resource isn't responding or is in transition and might resolve to another status over time.

Health status is updated every three minutes. No finer granularity is available. Historical health data isn't provided.

Viewing NLB Health Checker Summaries

On Private Cloud Appliance, you can view health summaries for all the network load balancers in a compartment.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. At the top of the page, select the compartment that contains the NLB.
3. Select the name of the NLB for which you want to view health checker status.

Using the OCI CLI

Use the oci nlb network-load-balancer-health list command and required parameters to retrieve the health status for all NLBs.

```
oci nlb network-load-balancer-health list [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListNetworkLoadBalancerHealths operation to retrieve the health status for all NLBs.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing NLB Health Checker Status Details

On Private Cloud Appliance, you can view the health status details of a network load balancer (NLB).

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. At the top of the page, select the compartment that contains the NLB.
3. Select the name of the NLB for which you want to view existing load balancer backend set health parameters.
4. Under **Resources**, select **Backend Sets**.
5. Select the name of the Backend Set to view its health check details.

Using the OCI CLI

Use the oci nlb health-checker get command and required parameters to see the health status details for the specified network load balancer.

```
oci nlb network-load-balancer-health get [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the GetNetworkLoadBalancerHealth operation to get the health status details for the specified network load balancer.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing an NLB Health Checker Policy

On Private Cloud Appliance, you can view the health checker policy parameters used by a network load balancer (NLB) to check backend set health.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. At the top of the page, select the compartment that contains the NLB.
3. Select the name of the NLB for which you want to view heath check policy.
4. Under **Resources**, select **Backend Sets**.
5. Select the name of the backend set.
6. Under the backend-set name for which to you want to view the health check policy, select **Backend Set Configuration**.

Using the OCI CLI

Use the oci nlb health-checker get command and required parameters to see the health check policy information for a particular network load balancer and backend set.

```
oci nlb health-checker get --backend-set-name backend-set-name-text \
 --network-load-balancer-id network-load-balancer_OCID [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the GetHealthChecker operation to get the health check policy information for a particular network load balancer and backend set.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing NLB Health Check Parameters

On Private Cloud Appliance, you can update the health check policy, such as the health check interval, for a network load balancer and backend set.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. At the top of the page, select the compartment that contains the NLB.
3. Select the name of the NLB for which you want to edit backend set health check parameters.
4. Under **Resources**, select **Backend Sets**.
5. Select the name of the backend set to view the details page.
6. Select **Edit**.
7. Make changes.
8. Select **Save**.

Using the OCI CLI

Use the oci nlb health-checker update command and required parameters to update the health check policy for a particular network load balancer and backend set.

```
oci nlb health-checker update [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateHealthChecker operation to update the health check policy for a particular network load balancer and backend set.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing the Health of an NLB Backend Set

On Private Cloud Appliance, you can view the network load balancer (NLB) backend set health status.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. At the top of the page, select the compartment that contains the NLB.
3. Select the name of the NLB for which you want to view backend set health status.
4. Under **Resources**, select **Backend Sets**.
5. Select the Backend Set to view the backend set health status.

Using the OCI CLI

Use the oci nlb backend-set-health get command and required parameters to see the health status for the specified backend set.

```
oci nlb backend-set-health get [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the GetBackendSetHealth operation to get the health status for the specified backend set.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing the Health of an NLB Backend Server

On Private Cloud Appliance, you can view the network load balancer (NLB) backend server health status.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. At the top of the page, select the compartment that contains the NLB.
3. Select the name of the NLB for which you want to view backend set health status.
4. Under **Resources**, select **Backend Sets**.
5. Select the Backend Set to view the backend health stat.

Using the OCI CLI

Use the oci nlb backend-health get command and required parameters to see the health status for the specified backend set.

```
oci nlb backend-set-health get [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the GetBackendHealth operation to get the health status for the specified backend set.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing NLB Work Request Errors

On Private Cloud Appliance, you can view NLB work requests to see if and NLB work request encountered errors.

Many of the configuration steps used to create and configure network load balancer (NLB) operation don't take effect immediately. In these cases, the request initiates an asynchronous work flow known as a work request to carry out the operation.

Because of the asynchronous nature of work request fulfillment, it's not always obvious that a configuration step has failed with an error. The failed step is often not revealed until the next step.

You can view NLB work request status indicators to find out if a work request has failed with an error. You can check the progress of each operation, whether it resulted in a failed state, which step it failed on, and the reason for the failure.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Networking**, then select **Network Load Balancers**.
2. At the top of the page, select the compartment that contains the NLB.
3. Select the name of the NLB for which you want to view work requests.
4. Under **Resources**, select **Work Requests**.
5. For each work request, you can view the following information:
   - Type of work request
   - State of the work request (Succeeded, Failed, and so on)
   - Start and finish timestamp
6. Select the name of a work request to see the following work request details:
   - General information about the work request, such as the type
   - OCID of the work request
   - Error detail of a failed work request (nothing is displayed for a nonfailure status)
   - Start and finish timestamp

Using the OCI CLI

Use the oci nlb work-request-error list command and required parameters to see a paginated list of errors for a particular work request.

```
oci nlb work-request-error list [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListWorkRequestErrors operation to list errors for a particular work request.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Images for Private Cloud Appliance Instances

On Private Cloud Appliance, an image is a template of a virtual hard drive. The image provides the OS and other software for a compute instance. You specify an image to use when you create a compute instance.

These are the types of images you can use to create a compute instance:

- **Platform images:** Private Cloud Appliance includes Oracle Linux and Oracle Solaris images. These included images are available in every compartment of every tenancy. These images don't need to be downloaded or imported to access them to create an instance. See Private Cloud Appliance Platform Images on page 621.
- **Custom images:** You can create a custom image of a compute instance's boot disk and use it to create other compute instances. Instances you create from your image include the customizations, configuration, and software installed when you created the image. See Managing Custom Images on page 622.
- **Bring your own image (BYOI):** You can bring your own versions of images to the cloud as long as the underlying hardware supports it. See Bring Your Own Image (BYOI) on page 629.

For a list of Oracle tested OSs, see Guest Operating System Matrix on page 53.

Listing Images and Viewing Details

On Private Cloud Appliance, in both the Compute Web UI and OCI CLI, Oracle provided images are listed first, followed by custom images.

The list of Oracle provided images includes the three most recently published versions of each major distribution. Any older versions that were previously listed are still available by specifying the image OCID.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Images**.
2. At the top of the page, select the compartment that contains the images you want to see.
3. To see details of an image, select the name of the image in the list.

Using the OCI CLI

Use the oci compute image list command and required parameters to list the images in the specified compartment.

```
oci compute image list [OPTIONS]
```

Use the oci compute image get command and required parameters to get the image details for just one image.

```
oci compute image get [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListImages and GetImage operations to list all images or get one image.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Private Cloud Appliance Platform Images

You can use the platform images that are included with Private Cloud Appliance to create instances. You can select from Oracle Linux and Oracle Solaris images.

> **Note:**
>
> The platform images provided with Private Cloud Appliance have been specifically configured for use On Private Cloud Appliance only. The following restrictions apply:
>
> - Private Cloud Appliance images can only be used to create instances On Private Cloud Appliance.
> - Oracle Cloud Infrastructure Platform images can't be used to create instances On Private Cloud Appliance.

> **Note:**
>
> Only the three most recently published versions of each major distribution are listed in the Compute Web UI and by the `compute image list` command. After an upgrade, older versions might no longer be listed, but they're still accessible. To use an older image version, use the OCI CLI and specify the OCID of the image. One way to get the OCID of an older image is from the output of the `compute instance get` command of an instance that's using the image.

To see the Private Cloud Appliance platform images, see Listing Images and Viewing Details on page 621.

To create an instance using an image provided with Private Cloud Appliance, see:

- Tutorial: Launching Your First Instance on page 640
- Creating an Instance on page 649

Initial User Account

After you create an instance from a platform image, you initially connect to the instance using `ssh` with the user account `opc`. The `opc` user has `sudo` privileges.

The SSH connection is authenticated using your SSH key pair that's used during instance creation. For more information, see Connecting to an Instance.

Remote Access

Access to the instance is permitted only over SSH v2 protocol. All other remote access services are disabled.

Image OS Administration

After an image is used to create an instance, the instance OS administration is managed according to the OS type and version. For OSs provided in the platform images, refer to the Oracle Linux and Oracle Solaris documentation available on Oracle Help Center's Operating Environments page.

Managing Custom Images

On Private Cloud Appliance, you can create, edit, move, import, export, and delete custom images.

For a list of Oracle-tested OSs for Private Cloud Appliance, see Guest Operating System Matrix on page 53.

Creating a Custom Image from an Instance

On Private Cloud Appliance, you can create a custom image of a compute instance's boot disk and use that custom image to create other compute instances. Instances that you create from this image include the customizations, configuration, and software that were installed on the boot disk when you created the image.

Custom images don't include the data from any attached block volumes.

After the new custom image reaches the Available state, you can use it to create new instances. See Creating an Instance on page 649.

Limitations and Considerations

- Certain IP addresses are reserved for Private Cloud Appliance use and can not be used in your address numbering scheme.
- When you create an image of an instance, the instance must be in the stopped state. After the custom image is created, you can restart the instance.
- You can't create extra custom images of a compute instance while the compute instance is engaged in the image creation process. You can, however, create images of different compute instances at the same time.
- Custom images are available to all users authorized for the compartment in which the image was created.
- Custom images inherit the compatible shapes that are set by default from the base image.

Avoid entering confidential information in names.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. At the top of the page, select the compartment that contains the instance you plan to use.
3. Select the name of the instance that you want to use as the basis for the custom image.
4. Ensure that the instance is in the Stopped state:
   a. On the details page for the instance, select **Controls** (upper right corner), and then select **Stop**.
   b. Wait for the instance status to change to Stopped. The status is displayed above the icon of the object.
5. Select **Controls** and then select **Create Custom Image**.
6. In the **Create Image From Instance** dialog box, enter the following information:
   - **Name:** Replace the name with the name you want for the image.
   - **Create in Compartment:** Select the compartment where the image will be stored.
7. Select **Create Custom Image**.

   The status of the instance changes to Creating Image.
8. Monitor the image creation progress.

   The time required to create the custom image depends on the size of the instance's boot volume.

   To monitor the progress, on the navigation menu, select **Compute** and then select **Custom Images**. Select the correct compartment, and select the image name in the list. On the details page for the image, under **Resources** select **Work Requests**.
9. (Optional) Restart the instance.

   When the instance status changes from Creating Image to Stopped, you can restart the instance.

Using the OCI CLI

Use the oci compute image create command and required parameters to create an image from an instance.

```
oci compute image create [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateImage operation to create an image from an instance.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Updating an Image Name

On Private Cloud Appliance, you change a custom image's display name.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Images**.
2. At the top of the page, select the compartment that contains the image you want to edit.
3. Use one of the following methods to open the **Update Image** dialog box.
   - For the image that you want to update, select the Actions menu (⋮), and select **Edit**.
   - Select the name of the image that you want to update. On the image details page, select **Controls** (upper right corner), then select **Edit Details**.
4. In the **Update Image** dialog box, change the image name.

   The image name doesn't need to be unique.

Avoid entering confidential information

5. Select **Save Changes**.

Using the OCI CLI

Use the oci compute image update command and required parameters to change an image name.

```
oci compute image update [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateImage operation to update the specified image.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Moving an Image to a Different Compartment

On Private Cloud Appliance, you can move an image to another compartment using the OCI CLI or API.

Custom images are available to all users authorized for the compartment in which the image was created. Ensure that you move the image to a compartment that has the appropriate authorized users.

Using the Compute Web UI

This task can't be performed using the Compute Web UI.

Using the OCI CLI

Use the oci compute image change-compartment command and required parameters to move an image to another compartment.

```
oci compute image change-compartment [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ChangeImageCompartment operation to move an image to another compartment.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Uploading an Image to an Object Storage Bucket

On Private Cloud Appliance, you can store images in an object storage bucket.

Advantages of storing an image in an Object Storage bucket are that you can implement object versioning or preauthenticated requests as described in Managing Object Versioning on page 807 and Using Preauthenticated Requests on page 811.

1. Create an Object Storage bucket as described in Creating a Bucket on page 795.
2. Upload an image from a local file system to the bucket. See Uploading an Object on page 800.
3. Import the image from the Object Storage bucket to a compartment so that the image is available to select when you create an instance. See Importing an Image from an Object Storage Bucket on page 624.

Importing an Image from an Object Storage Bucket

On Private Cloud Appliance, you can import an image into a compartment from an Object Storage bucket.

> **Note:**
>
> Alternatively, you can import an image from a URL as described in Importing an Image from a URL on page 626.

Before You Begin

Ensure you have read access to the Object Storage bucket, and that the bucket contains the image that you want to use. See Managing Object Storage Buckets on page 794.

If the bucket doesn't contain the image that you want to use, perform the procedure described in Uploading an Image to an Object Storage Bucket on page 624.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Custom Images**.
2. Select **Import Image**.
3. In the **Import Image** dialog box, enter the following information:
   - **Name:** Enter a descriptive name for the image.

     Avoid entering confidential information.
   - **Create in Compartment:** Select the compartment for the image.
   - **Source Type:** Select the **Import from an Object Storage Bucket** option.
   - **Bucket:** Select a bucket. You might need to change the compartment to locate the bucket.
   - **Object Name:** Select the name of an image object from the list.
   - **Image Type:** Select one of the following options based on the type of image you're importing.
     - **VMDK:** Virtual machine disk file format (`.vmdk`), used for virtual machine disk images.
     - **QCOW2:** For disk image files (`.qcow2`) used by QEMU copy on write.
     - **OCI:** For Oracle Cloud Infrastructure files with a QCOW2 image and OCI metadata (`.oci`).
   - **Launch Mode:** Paravirtualized is the default and can't be changed.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
4. Select **Import Image**.

   The imported image appears in the **Custom Images** list for the compartment, with a state of Importing. To track the progress of the operation, view the associated work request.

   When the import completes successfully, the image state changes to available, and the image can be used to create instances as described in Creating an Instance on page 649.

Using the OCI CLI

Use the oci compute image import from-object command and required parameters to import an image from an Object Storage bucket.

```
oci compute image import from-object --compartment-id compartment_OCID
 --bucket-name bucket_name --name bucket_image_object_name --
namespace namespace [OPTIONS]
```

> **Important:**
>
> If you are importing a Microsoft Windows image, specify the `--operating-system` option and include the case-insensitive string "Windows" in the value to ensure optimal performance of the instance.
>
> If you specify the `--operating-system` option and this is *not* a Microsoft Windows image, ensure the value doesn't contain the case-insensitive string "Windows".

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

This operation can't be performed using the API.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Importing an Image from a URL

On Private Cloud Appliance, you can import an image into a compartment by specifying the URL of the image file.

Alternatively, you can import an image from an Object Storage bucket as described in Importing an Image from an Object Storage Bucket on page 624.

Before You Begin

Get the URL that you need for this procedure. Ensure that the URL is accessible from your tenancy.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Custom Images**.
2. On the **Custom Images** page, select **Import Image**.
3. In the **Import Image** dialog box, enter the following information:
   - **Name:** Enter a descriptive name for the image.

     Avoid entering confidential information.
   - **Create in Compartment:** Select the compartment where the image will be placed.
   - **Source Type:** Select the Import from an Object Storage URL option.
   - **Object Storage URL:** Enter the URL of the image. The URL does not need to be an Object Storage URL. It can be any URL that provides access to the image.
   - **Image Type:** Select one of the following options based on the type of image you are importing.
     - **VMDK:** Virtual machine disk file format (`.vmdk`), used for virtual machine disk images.
     - **QCOW2:** For disk image files (`.qcow2`) used by QEMU copy on write.
     - **OCI:** For Oracle Cloud Infrastructure files with a QCOW2 image and OCI metadata (`.oci`).
   - **Launch Mode:** Paravirtualized is the default and can't be changed.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
4. Select **Import Image**.

   The imported image appears in the **Custom Images** list for the compartment, with a state of Importing. To track the progress of the operation, view the associated work request.

   When the import completes successfully, the image state changes to Available, and the image can be used to create instances as described in Creating an Instance on page 649.

Using the OCI CLI

Use the oci compute image import from-object-uri command and required parameters to import an image from a URL.

```
oci compute image import from-object-uri --compartment-id compartment_OCID
 --uri URL_for_image[OPTIONS]
```

> **Important:**
>
> If you are importing a Microsoft Windows image, specify the `--operating-system` option and include the case-insensitive string "Windows" in the value to ensure optimal performance of the instance.
>
> If you specify the `--operating-system` option and this is *not* a Microsoft Windows image, ensure the value doesn't contain the case-insensitive string "Windows".

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

This operation can't be performed using the API.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Exporting an Image to an Object Storage Bucket

On Private Cloud Appliance, you can export an image to an Object Storage bucket.

Exported images are a copy of the boot volume and metadata when the image was created.

> **Note:**
>
> As an alternative, you can export an image to an Object Storage URL as described in Exporting an Image to an Object Storage URL on page 628.

Prerequisites

- Ensure that a bucket is available. See Listing Buckets on page 795 and Creating a Bucket on page 795.
- You need write access to the to the export location.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Custom Images**.
2. At the top of the page, select the compartment that contains the image you want to export.
3. Select the name of the custom image that you want to export.
4. On the image details page, select **Controls** (upper right corner), and then select **Export Image**.
5. In the **Export Custom Image** dialog box, enter the following information:
   - **Export Destination:** Select the **Export to an Object Storage Bucket** option.
   - **Bucket:** Select a bucket. If needed, use the menu to select a different compartment.
   - **Object Name:** Enter a name for the exported image.
   - **Export Format:** Select one of the following options based on the type of image you're exporting.
     - **VMDK:** Virtual machine disk file format (`.vmdk`), used for virtual machine disk images.
     - **QCOW2:** For disk image files (`.qcow2`) used by QEMU copy on write.
     - **OCI:** For Oracle Cloud Infrastructure files with a QCOW2 image and OCI metadata (`.oci`).
6. Select **Create Export**.

   The image state changes to Exporting. Exporting a custom image copies the data to the Object Storage location that you specified. To track the progress of the operation, view the associated work request.

   You can still create instances while the image is exporting, but you can't delete the image until the export has finished.

   When the export is complete, the image state changes to Available. If the image state changes to Available, but you don't see the exported image in the Object Storage location you specified, the export failed, and you need to go through the steps again to export the image.

Using the OCI CLI

Use the oci compute image export to-object command and required parameters to export the specified image to an Object Storage bucket.

```
oci compute image export to-object --bucket-name bucketname --image-
id image_OCID --namespace namespace --name exported_image_object_name --
export-format VMDK [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ExportImage operation to export the specified image to an Object Storage bucket.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Exporting an Image to an Object Storage URL

On Private Cloud Appliance, you can export an image to an Object Storage URL.

Exported images are a copy of the boot volume and metadata when the image was created.

> **Note:**
>
> As an alternative, you can export an image to a URL as described in Exporting an Image to an Object Storage Bucket on page 627.

Prerequisite

- You need write access to the to the export location.

Using the Compute Web UI

1. In the cnvCommonPhrases navigation menu, select **Compute**, then select **Custom Images**.
2. At the top of the page, select the compartment that contains the image you want to export.
3. Select the name of the custom image that you want to export.
4. On the image details page, select **Controls** (upper right corner), and then select **Export Image**.
5. In the **Export Custom Image** dialog box, enter the following information:
   - **Export Destination:** Select the **Export to an Object Storage URL** option.
   - **Object Storage URL:** Enter the URL where you want to export the image. The URL must be an Object Storage URL.
   - **Export Format:** Select one of the following options based on the type of image you're exporting.
     - **VMDK:** Virtual machine disk file format (`.vmdk`), used for virtual machine disk images.
     - **QCOW2:** For disk image files (`.qcow2`) used by QEMU copy on write.
     - **OCI:** For Oracle Cloud Infrastructure files with a QCOW2 image and OCI metadata (`.oci`).
6. Select **Export Image**.

   The image state changes to Exporting. Exporting a custom image copies the data to the Object Storage location that you specified. To track the progress of the operation, view the associated work request.

   You can still create instances while the image is exporting, but you can't delete the image until the export has finished.

   When the export is complete, the image state changes to Available. If the image state changes to Available, but you don't see the exported image in the Object Storage location you specified, the export failed, and you need to go through the steps again to export the image.

Using the OCI CLI

Use the oci compute image export to-object-uri command and required parameters to export the specified image to the Object Storage service using the Object Storage URL to identify the location to export to.

```
oci compute image export to-object-uri --image-id <image_OCID> --
uri <url_to_export_to> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ExportImage operation to export the specified image to the Object Storage service using the Object Storage URL to identify the location to export to.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Sharing Custom Images Across Tenancies

On Private Cloud Appliance, you can use image import and export to share custom images across tenancies so that you don't need to recreate the image manually in each tenancy. You create the image in one of the tenancies, and then export the image, making it available for import in additional tenancies.

These are the high-level tasks:

1. Export the image to an Object Storage bucket. See Exporting an Image to an Object Storage Bucket on page 627.
2. Create a preauthenticated request with read-only access for the image in the bucket. See Using Preauthenticated Requests on page 811.
3. In the destination tenancy, import the image. Use the preauthenticated request URL as the Object Storage URL. See Importing an Image from a URL on page 626.

Deleting an Image

On Private Cloud Appliance, you can delete custom images.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Images**.
2. At the top of the page, select the compartment that contains the image you want to delete.
3. For the custom image that you want to delete, select Actions menu (⋮), then select **Delete image**.

   The image is deleted.

Using the OCI CLI

Use the oci compute image delete command and required parameters to delete a custom image.

```
oci compute image delete [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteImage operation to delete a custom image.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Bring Your Own Image (BYOI)

On Private Cloud Appliance, the Bring Your Own Image (BYOI) feature enables you to bring your own versions of OSs to the Private Cloud Appliance service as long as the underlying hardware supports it.

The BYOI feature provides these benefits:

- Enables cloud migration projects.
- Supports both old and new OSs.
- Encourages experimentation.
- Increases flexibility.

> **Important:**
>
> You must comply with all licensing requirements when you upload and start instances based on OS images that you supply.

A critical part of any lift-and-shift cloud migration project is the migration of on-premises virtual machines (VMs) to the cloud. You can import your on-premises virtualized root volumes into Private Cloud Appliance using the custom image import feature, and then create compute instances using those images.

For a list of Oracle-tested OSs for Private Cloud Appliance, see Guest Operating System Matrix on page 53.

For information about creating custom images, see Managing Custom Images.

For information about importing images, see Importing Custom Linux Images on page 630 and Importing Custom Microsoft Windows Images on page 631.

For information about creating instances, see Working with Instances on page 649.

Importing Custom Linux Images

You can bring your own Linux image to Private Cloud Appliance as long as the image meets specific requirements.

Linux Source Image Requirements

Custom images must meet the following requirements:

- The maximum image size is 400 GB.
- The image must be set up for BIOS boot.
- Only one disk is supported, and it must be the boot drive with a valid master boot record (MBR) and boot loader. You can migrate additional data volumes after you import the image's boot volume.
- The boot process must not require more data volumes to be present for a successful boot.
- The boot loader must use LVM or a UUID to locate the boot volume.
- The disk image can't be encrypted.
- The disk image must be a VMDK or QCOW2 file. These images can be converted to `.oci` type images.
  - Create the image file by cloning the source volume, not by creating a snapshot.
  - VMDK files must be either the "single growable" (monolithicSparse) type or the "stream optimized" (streamOptimized) type, both of which consist of a single VMDK file. All other VMDK formats, such as those that use multiple files, split volumes, or contain snapshots, are not supported.
- The network interface must use DHCP to discover the network settings. When you import a custom image, existing network interfaces are not re-created. Any existing network interfaces are replaced with a single NIC after the import process is complete. You can attach more VNICs after you create the imported instance.
- The network configuration must not hard code the MAC address for the network interface.
- Oracle recommends that you enable certificate-based SSH, however this recommendation is optional.

Preparing Linux VMs for Import

Before you import a custom Linux image, you must prepare the image to ensure that instances created from the image can boot correctly and that network connections will work.

Perform these steps.

1. Ensure that the source image meets the requirements.

   See Importing Custom Linux Images on page 630.
2. Create a backup of the root volume.
3. If the VM has remotely attached storage, such as NFS or block volumes, configure any services that rely on this storage to start manually. Remotely attached storage isn't available the first time an imported instance boots.
4. Ensure that all network interfaces use DHCP, and that the MAC address and IP addresses aren't hardcoded. See your system documentation for steps to perform network configuration for your system.
5. Stop the VM.
6. Clone the stopped VM as a VMDK or QCOW2 file, and then export the image from your virtualization environment.

   Refer to the tools documentation for your virtualization environment.

Importing a Linux Image

After you prepare a Linux image for import, follow these steps to import the image:

1. Upload the image file to an Object Storage bucket.

Ensure that you select a bucket where you have read and write access. See Exporting an Image to an Object Storage Bucket on page 627.

2. Import the image from the bucket to your tenancy.

   See Importing an Image from an Object Storage Bucket on page 624.
3. Complete the post import tasks.

   See Post Import Tasks for Linux Images on page 631.

Post Import Tasks for Linux Images

After you import a custom Linux image, perform these steps.

1. Use the imported image to create an instance.

   For the image source, select Custom Images, and then select the image that you imported. See Creating an Instance on page 649.
2. If the instance requires any remotely attached storage, such as block volumes, create and attach the storage.

   See Creating and Attaching Block Volumes on page 716.
3. Create and attach any required secondary VNICs.

   See Configuring VNICs on page 518.
4. Test that all applications are working as expected.
5. Reconfigure any services that were set to start manually.

Importing Custom Microsoft Windows Images

When you bring your own Microsoft Window image to Private Cloud Appliance, it must meet specific requirements.

Microsoft Windows Source Image Requirements

- The maximum image size is 400 GB.
- The image must be set up for a BIOS boot.
- Only one disk is supported, and it must be the boot drive with a valid master boot record (MBR) and boot loader. You can migrate additional data volumes after you import the image's boot volume.
- The minimum boot volume size is 256 GB.
- The boot process must not require other data volumes to be present for a successful boot.
- The disk image can't be encrypted.
- The disk image must be a VMDK or QCOW2 file. Create the image file by cloning the source volume, not by creating a snapshot. VMDK files must be either the "single growable" (monolithicSparse) type or the "stream optimized" (streamOptimized) type, both of which consist of a single VMDK file. All other VMDK formats, such as those that use multiple files, split volumes, or contain snapshots, are not supported.
- The network interface must use DHCP to discover the network settings. When you import a custom image, existing network interfaces are not re-created. Any existing network interfaces are replaced with a single NIC after the import process is complete. You can attach additional VNICs after you launch the imported instance.
- The network configuration must not hard code the MAC address for the network interface.

Preparing Microsoft Windows Systems for Import

The configuration described in this section is required so that Compute instances that are created from the Microsoft Windows system image can boot correctly and network connections will work.

> **Important:**
>
> The system drive configuration where the Microsoft Windows source instance (virtual machine or physical system) is installed will be imported to the image. All partitions on the drive will follow through the imported image. Any other drives will not be imported, and you must re-create them on the instance after the instance is created from the image. Then manually move the data on the noninstance drives to storage on the instance.

You can perform this configuration on the running source system or after you have launched the Compute instance.

- **Preparing the Source System Prior to Creating the Image**. This is the recommended method.
- **Preparing the Compute Instance After Instance Launch**. If you have concerns about modifying the live source system, you can use this method. If you use this method, your Compute instance is not initially viable. After you launch your Compute instance, connect to the VNC console and use the VNC window to make the changes described in **Preparing the Source System Prior to Creating the Image**.

Preparing the Source System Prior to Creating the Image

1. Review the requirements.

   See Importing Custom Microsoft Windows Images on page 631.
2. Clone the existing instance, using the procedures specific to the current platform. Then perform all changes on the clone to prevent disrupting the production instance.
3. Follow your organization's security guidelines to ensure that the Microsoft Windows system is secured. This can include, but is not limited to the following tasks:
   - Install the latest security updates for the OS and installed applications.
   - Enable the firewall, and configure it so that you only enable the rules that are needed.
   - Disable unnecessary privileged accounts.
   - Use strong passwords for all accounts.
4. Configure Remote Desktop Protocol (RDP) access.
   a. Enable Remote Desktop connections to the image. See To connect to a Windows instance from a Remote Desktop Client.
   b. Modify the Microsoft Windows Firewall inbound port rule to allow RDP access for both Private and Public network location types. When you import the image, the Microsoft Windows Network Location Awareness service will identify the network connection as a Public network type.
5. Determine whether the current Microsoft Windows license type is a volume license by running the following command in PowerShell:

   ```
   Get-CimInstance -ClassName SoftwareLicensingProduct | where
    {$_.PartialProductKey} | select ProductKeyChannel
   ```

   If the license is not a volume license, after you import the image, you will update the license type.
6. If you plan to use the custom image to create more than one instance, create a generalized image of the boot disk. A generalized image is cleaned of computer-specific information, such as unique identifiers. When you create instances from a generalized image, the unique identifiers are regenerated. This prevents two instances that are created from the same image from colliding on the same identifiers.
7. Create a backup of the root volume.
8. If the Windows instance has remotely attached storage, such as NFS or block volumes, configure any services that rely on this storage to start manually. Remotely attached storage is not available the first time an instance that was created from a custom image boots on Oracle Cloud Infrastructure.
9. Ensure that all network interfaces use DHCP, and that the MAC address and IP addresses are not hardcoded. See your system documentation for steps to perform network configuration for your system.
10. Download the Oracle Windows VirtIO drivers:
    a. Sign in to the Oracle Software Delivery Cloud site.
    b. In the All Categories drop-down, select Release.
    c. Type Oracle Linux 7.9 in the search box and select Search.
    d. Select REL: Oracle Linux 7.9.0.0.0.
    e. At the top right of the page, to the right of your cart, select Continue.
    f. In the Platforms/Languages list, select x86 64 bit. Select Continue.
    g. Accept the license agreement and then select Continue.

h. Select the checkbox next to Oracle VirtIO Drivers Version for Microsoft Windows 1.1.x. Clear the other check boxes.
i. Select Download and then follow the prompts.

11. Install the Oracle VirtIO drivers for Windows:
    a. Follow the prompts in the installation workflow. On the Installation Type page, select Custom.
    b. Reboot the Windows instance.
12. Perform the Creating and Exporting an Image on page 633 procedure.

Creating and Exporting an Image

1. Stop the system.
2. Clone the stopped system as a VMDK or QCOW2 file. Refer to the tools documentation for your OS.
3. Export the image from your physical system or virtualization environment.
4. Perform the Importing a Microsoft Windows Image on page 633 procedure to import the image into Oracle Cloud Infrastructure.

Preparing the Compute Instance After Instance Launch

1. Perform as many of the **Preparing the Source System Prior to Creating the Image** steps as you are comfortable performing.
2. Perform the **Creating and Exporting an Image** procedure above.

   After importing the image, do *not* perform the Post Import Tasks for Microsoft Windows Images on page 633 procedure.
3. Use the imported image to launch an instance.

   For the image source, select Custom Images, and then select the image that you imported. See Creating an Instance on page 649.
4. Connect to the console as described in Connecting to a Compute Instance on page 674.
5. Perform the Preparing Microsoft Windows Systems for Import on page 631 procedure.
6. Perform the Post Import Tasks for Microsoft Windows Images on page 633 procedure.

Importing a Microsoft Windows Image

After you prepare a Microsoft Windows image for import, follow these steps to import the image:

1. Upload the image file to an Object Storage bucket.

   Ensure that you select a bucket where you have read and write access. See Exporting an Image to an Object Storage Bucket on page 627.
2. Import the image from the bucket to your tenancy.

   See Importing an Image from an Object Storage Bucket on page 624 and Importing an Image from a URL on page 626. Use the OCI CLI procedure and specify the `--operating-system` option. Ensure the value of the `--operating-system` option includes the case-insensitive string "Windows".
3. Complete the post-import tasks.

   See Post Import Tasks for Microsoft Windows Images on page 633.

Post Import Tasks for Microsoft Windows Images

After you import a custom Microsoft Windows image, perform these steps.

1. Use the imported image to create an instance.

   For the image source, select Custom Images, and then select the image that you imported. See Creating an Instance on page 649.
2. Enable Remote Desktop Protocol (RDP) access to the Compute instance.

   See To connect to a Windows instance from a Remote Desktop Client.

3. Connect to the instance using RDP.

   See To connect to a Windows instance from a Remote Desktop Client.
4. If the instance requires any remotely attached storage, such as block volumes, create and attach the storage.

   See Creating and Attaching Block Volumes on page 716.
5. Create and attach any required secondary VNICs.

   See Configuring VNICs on page 518.
6. Test that all applications are working as expected.
7. Reconfigure any services that were set to start manually.
8. Configure your instance to use the Network Time Protocol (NTP).

To avoid performing this configuration every time you create an instance using this custom image, consider creating a new image from the fully configured instance. See Creating a Custom Image from an Instance on page 622.

Compute Instances

You can provision and manage compute instances On Private Cloud Appliance.

A compute instance is a virtual machine (VM), which is an independent computing environment that runs on top of physical hardware. The virtualization makes it possible to run multiple compute instances that are isolated from each other.

When you create a compute instance, you can select the most appropriate type of compute instance for your applications based on characteristics such as the number of CPUs, amount of memory, and network resources. See Tutorial: Launching Your First Instance on page 640 and Working with Instances on page 649.

After you create a compute instance, you can access it securely from your computer, restart it, attach and detach volumes, and delete it when you're done with it.

With an instance configuration, you can create a single instance or pool of instances quickly. You can create an instance configuration from an existing instance to replicate that instance more quickly. See Working with Instance Configurations on page 666.

You can attach instances to a pool or detach instances from a pool manually, or you can configure autoscaling to automatically grow or shrink the pool on a predefined schedule. See Instance Pools on page 682.

You can create a pool of compute instances (nodes) in a Private Cloud Appliance Kubernetes Engine (OKE) cluster. See Kubernetes Engine (OKE) On Private Cloud Appliance on page 821.

You can connect to a compute instance. See Connecting to a Compute Instance on page 674.

You can back up an instance and restore the instance from backup. See Backing Up and Restoring an Instance on page 677.

Components for Creating Instances

These components are required to create a compute instance:

Compartment

A collection of related resources that are only accessible by certain groups that have been given permission by an administrator in your organization. Compute instances are created in compartments. All compartments exist in a tenancy, which is the root compartment.

Virtual Cloud Network (VCN)

A virtual version of a traditional network—including subnets, route tables, and gateways—on which your compute instance runs. At least one cloud network must be set up before you create compute instances.

SSH Key Pair

If the image that's used to create the instance is configured to require Secure Shell (SSH) for authentication, then you need an SSH key pair before creating the instance. This requirement applies to instances created from Private Cloud Appliance platform images and by most UNIX type images. If the image is configured to use passwords instead, you need the password instead of the key pair.

Image

A template of a virtual hard drive that determines the OS and other software for a compute instance. You can also create compute instances using these images:

- Private Cloud Appliance platform images
- Custom images created from other instances
- Import your own image

For more information about images, see Images.

Shape

A template that determines the number of CPUs, amount of memory, and other resources allocated to a newly created compute instance. See Compute Shapes.

Boot Volumes

When you launch a compute instance based on an Private Cloud Appliance platform image or custom image, a new boot volume for the compute instance is created in the same compartment. That boot volume is associated with that compute instance until you delete the compute instance.

When you delete the compute instance, you can preserve the boot volume and its data. This feature gives you more control and management options for your compute instance boot volumes, and enables:

- **Instance scaling:** When you delete your compute instance, you can keep the associated boot volume and use it to launch a new compute instance using a different compute instance type or shape. This flexibility enables you to easily scale up or down the number of cores for a compute instance.
- **Troubleshooting and repair:** If you think a boot volume issue is causing a compute instance problem, you can stop the compute instance and detach the boot volume. Then you can attach it to another compute instance as a data volume to troubleshoot it. After resolving the issue, you can then reattach it to the original compute instance or use it to launch a new compute instance.

Boot volume Encryption

Boot volumes are encrypted by default, the same as other block storage volumes.

> **Important:**
>
> In most cases, encryption isn't supported for compute instances launched from custom images imported for "bring your own image" (BYOI) scenarios.

For more information about Private Cloud Appliance boot volumes, see Managing Boot Volumes on page 731

For information about backing up boot volumes, see Backing Up Block Volumes on page 744.

Storage for Instances

You can expand the storage that's available for your compute instances with the following services:

- **Block Volume:** Lets you dynamically provision and manage block volumes that you can attach to one or more compute instances. See Block Volume Storage on page 712.
- **File Storage:** A durable, scalable, secure, enterprise-grade network file system that you can connect to from any compute instance in your virtual cloud network (VCN). See File Storage on page 754.
- **Object Storage:** An internet-scale, high-performance storage platform that lets you store a large amount of unstructured data of any content type. This storage not tied to any specific compute instance. See Object Storage on page 793.

Simplifying Compute Instance Management

You can simplify the management of your compute instances using these features:

- **Instance Configurations:** Are templates that define the settings to use when creating compute instances.
- **Instance Pools:** are a group of compute instances that are created from the same compute instance configuration and managed as a group.

Calling Services from an Instance

A Private Cloud Appliance instance can be configured to enable applications running on the instance to call services and manage resources similar to the way users call services to manage resources.

An instance that can perform actions on service resources is called an *instance principal*.

You can authorize an instance to make API calls in Private Cloud Appliance services. After you set up the required resources and policies, an application running on an instance can call Private Cloud Appliance public services, removing the need to configure user credentials or a configuration file.

See Configuring Instances for Calling Services on page 663.

Metadata Key Restrictions

Metadata keys have the following restrictions, with the noted exceptions:

- Metadata can have a maximum of 128 keys.
- Key names can have a maximum of 255 characters.
- Most key values can have a maximum of 255 characters.

The value of the `ssh_authorized_keys` metadata key can be more than 255 characters. This value must be a valid public key in OpenSSH format. Use a newline character to separate multiple keys.

The value of the `user_data` metadata key can be a maximum of 16KB. This value is data that cloud-init can use to run custom scripts or provide custom cloud-init configuration. For Linux instances with cloud-init configured, the `user_data` value is a Base64-encoded string of cloud-init user data. For more information, see cloud-init data formats.

Compute Shapes

A shape is a template that determines the number of OCPUs, amount of memory, and number of VNICs that are allocated to a Private Cloud Appliance compute instance. You specify a shape when you create an instance.

Private Cloud Appliance supports the following types of shapes:

- **Standard shapes:** Each shape has a fixed number of OCPUs and memory that are allocated to an instance when the instance is created.
- **Flexible shapes:** You choose the number of OCPUs and amount of memory that are allocated to an instance.
- **GPU shapes:** Each shape has a fixed number of OCPUs, memory, and GPUs that are allocated to an instance when the instance is created.

Standard Shapes

Standard shapes are designed for general-purpose workloads and suitable for a wide range of applications and use cases. Standard shapes provide a balance of cores, memory, and network resources. All standard shapes use block storage for the boot device.

The following table lists the standard shapes.

Shape	OCPUs	Memory (GB)	Maximum VNICs	Maximum Bandwidth (Gbps)
VM.PCAStandard1.1	1	16	2	24.6
VM.PCAStandard1.2	2	32	2	24.6
VM.PCAStandard1.4	4	64	4	24.6
VM.PCAStandard1.8	8	128	8	24.6
VM.PCAStandard1.16	16	256	16	24.6
VM.PCAStandard1.24	24	384	24	24.6
VM.PCAStandard1.32	32	512	24	32.0
VM.PCAStandard1.48	48	768	24	48.0
VM.PCAStandard1.60	60	960	24	100.0

Flexible Shapes

A flexible shape lets you customize the number of OCPUs and the amount of memory for an instance. When you update an instance, you can change these properties of the flexible shape, and you can change from a fixed shape to a flexible shape. This flexibility lets you create instances that meet your workload requirements, while optimizing performance and using resources efficiently.

Private Cloud Appliance platform images support the flex shape.

The VM.PCAStandard.E5.Flex shape can be selected only for Private Cloud Appliance X10 systems.

Shape	OCPUs	Memory (GB)	Maximum VNICs	Maximum Bandwidth (Gbps)
VM.PCAStandard1.Flex	1-32	64 GB maximum per OCPU 512 GB maximum per instance	1 OCPU: 2 VNICs 2 to 24 OCPUs: 1 VNIC per OCPU 25 to 32 OCPUs: 24 VNICs	1 to 24 OCPUs: 24.6 Gbps 25 to 32 OCPUs: 1 Gbps per OCPU
VM.PCAStandard.E5.Flex	1-96	64 GB maximum per OCPU 960 GB maximum per instance	1 OCPU: 2 VNICs 2 to 24 OCPUs: 1 VNIC per OCPU 25 to 96 OCPUs: 24 VNICs	1 to 24 OCPUs: 24.6 Gbps 25 to 40 OCPUs: 1 Gbps per OCPU 41 to 96 OCPUs: 40.0 Gbps

GPU Shapes

GPU shapes are designed for enterprise GPU-accelerated workloads, such as AI and graphics intensive applications. These shapes are available on systems with an optional GPU expansion rack, when an Oracle Linux 8 or Oracle Linux 9 platform image is selected to create the instance. Instances created with a GPU shape have direct (passthrough) access to 1-4 physical GPUs. The ratio between GPUs, OCPUs and memory is fixed.

The following table lists the GPU shapes:

Shape	GPUs	OCPUs	Memory (GB)	Maximum VNICs	Maximum Bandwidth (Gbps)
VM.GPU.L40S.1	1	27	VM: 200 GPU: 48	24	400
VM.GPU.L40S.2	2	54	VM: 400 GPU: 96	24	400
VM.GPU.L40S.3	3	81	VM: 600 GPU: 144	24	400
VM.GPU.L40S.4	4	108	VM: 800 GPU: 192	24	400

Physical Resource Allocation

When users in Private Cloud Appliance deploy compute instances, these consume physical resources provided by the hardware layer. The hypervisor manages the allocation of those physical resources based on algorithms that enable the best performance possible for a given configuration.

The compute nodes in Private Cloud Appliance have a Non-Uniform Memory Access (NUMA) architecture, meaning each CPU has access not only to its own local memory but also the memory of the other CPUs. Each CPU socket and its associated local memory banks are called a *NUMA node*. Local memory access, within the same NUMA node, always provides higher bandwidth and lower latency.

In general, the memory sharing design of NUMA helps with the scaling of multiprocessor workloads, but it may also adversely affect virtual machine performance if its resources are distributed across multiple NUMA nodes. Hypervisor policies ensure that a virtual machine's CPU and memory reside on the same NUMA node whenever possible. Using specific CPU pinning techniques, each virtual machine is pinned to one or multiple CPU cores so that memory is accessed locally on the NUMA node where the virtual machine is running. Cross-node memory transports are avoided or kept at a minimum, so the users of compute instances benefit from optimal performance across the entire system.

If a virtual machine can fit into a single NUMA node on a hypervisor host then this NUMA node is used for pinning, referred to as *strict pinning*. If a virtual machine cannot fit into single NUMA node on a hypervisor host, but can fit into multiple NUMA nodes, then multiple NUMA nodes are used for pinning, referred to as *loose pinning*.

> **Caution:**
>
> The CPU pinning applied through the hypervisor to optimize CPU and memory allocation to compute instances is not configurable by the appliance administrator, tenancy administrator or instance owner.

Private Cloud Appliance detects the NUMA topology of compute nodes during their provisioning (or any upgrade path from one release to another) and stores this information for use during the deployment of compute instances. The NUMA details of each compute instance are stored in the instance configuration. The NUMA settings are preserved

when compute instances are migrated to another host compute node, but they might be overridden and dynamically adjusted if the target compute node is unable to accommodate that particular configuration.

Compute instances running on a Private Cloud Appliance that is not yet NUMA-aware, can take advantage of the optimization policies as soon as the system has been patched or upgraded. There is no action required from the administrator to align instances with NUMA topology; the existing instance configurations are made compatible as part of the process.

Compute Instance Lifecycle

On Private Cloud Appliance, a compute instance can be in a variety of different lifecycle states.

This list describes the different lifecycle states for compute instances.

- **Launching:** Occurs when you create a compute instance. The instance is displayed in the Compute Web UI in a provisioning state. Expect provisioning to take several minutes before the state updates to running. After the instance is running, wait another few minutes for the OS to boot before you attempt to connect.
- **Connecting:** You connect to a running Linux or Oracle Solaris instance using a Secure Shell (SSH) connection. Most Linux and UNIX-like OSs include an SSH client by default.
- **Backing up the boot volume:** You can back up the boot volume using the Block Volume backup feature using one of these methods:
  - **Manual backups:** You manually perform create, get, list, rename, and delete backup commands.
  - **Automatic backups:** You create a backup policy and a backup policy assignment that specifies the time and frequency of the volume backups. The system automatically performs the commands that back up the volume.
- **Stopping:** You can stop an instance using the Compute Web UI, OCI CLI, and API, or by using the commands available in the OS when you are logged in to the instance.

  If the applications that run on the instance take more than 15 minutes to shut down, they could be improperly stopped. To avoid this situation, shut down the instance using the commands available in the OS before you stop the instance.
- **Starting or restarting:** You can start or restart an instance as needed using the Compute Web UI, OCI CLI, and API.
- **Rebooting:** You can reboot an instance as needed using the Compute Web UI, OCI CLI, and API. By default, a reboot gracefully restarts the instance by sending a shutdown command to the OS. After waiting 15 minutes for the OS to shut down, the instance is powered off and then powered back on.
- **Terminating:** You can permanently terminate (delete) instances that you no longer need. Any attached VNICs and volumes are automatically detached when the instance terminates. Eventually, the instance's public and private IP addresses are released and become available for other instances.

  By default, the instance's boot volume is preserved when you terminate the instance. You can attach the boot volume to a different instance as a data volume, or use it to launch a new instance. If you no longer need the boot volume, you can permanently delete it.

For more information, see Working with Instances on page 649.

Extending Compute Resources

On Private Cloud Appliance, you can extend storage resources and add more network interfaces.

Expanding Volumes

You can expand the size of block volumes and boot volumes. You can't decrease the size.

You have several options to increase the size of your volumes:

- Expand an existing volume in place with online resizing.
- Restore from a volume backup to a larger volume.
- Clone an existing volume to a new, larger volume.
- Expand an existing volume in place with offline resizing.

> **Caution:**
>
> Before you resize a boot or block volume, create a backup of the volume. See Backing Up Block Volumes on page 744.

After a volume has been resized, the first backup of the resized volume is a full backup.

Offline Resizing of Block Volumes

With offline resizing, you detach the volume from a compute instance before you expand the volume size. After the volume is resized and reattached, you need to extend the partition, but you don't need to rescan the disk.

Whenever you detach and reattach volumes, there are complexities and risks for both Linux-based and Microsoft Windows-based compute instances. For more information.

Rescanning the Disk for a Block Volume or Boot Volume

The Block Volume service lets you expand the size of block volumes and boot volumes while they're online and attached to compute instances.

After the volume is provisioned, you need to run commands to rescan the disk so that the OS recognizes the expanded volume size. You run different rescan commands depending on the OS of the attached compute instance.

For more information, see Resizing Volumes on page 735.

Adding Another Network Interface

You can add additional VNICs to a compute instance. Each additional VNIC can be in a subnet in the same VCN as the primary VNIC, or in a different subnet that's either in the same VCN or a different one.

You might add a VNIC to connect a compute instance to subnets in multiple VCNs. For example, you might set up your own firewall to protect traffic between VCNs, so the compute instance needs to connect to subnets in different VCNs.

Secondary VNICs are supported for these types of compute instances:

- **Linux and Oracle Solaris**
- **Microsoft Windows**

Here are more details about additional VNICs:

- There's a limit to how many VNICs can be attached to a compute instance, and it varies by shape. See Compute Shapes.
- VNICs can be added only after the compute instance is launched.
- VNICs must always be attached to a compute instance and can't be moved. The process of creating an additional VNIC automatically attaches it to the compute instance. The process of detaching a secondary VNIC automatically deletes it.
- VNICs are automatically detached and deleted when you delete the compute instance.
- The compute instance's bandwidth is fixed regardless of the number of VNICs attached. You can't specify a bandwidth limit for a particular VNIC on a compute instance.
- Attaching multiple VNICs from the same subnet CIDR block to a compute instance can introduce asymmetric routing, especially on instances using a variant of Linux. If you need this type of configuration, assign multiple private IP addresses to one VNIC, or using policy-based routing.

For more information, see Configuring VNICs on page 518.

Tutorial: Launching Your First Instance

In this tutorial, you'll learn the basic features of Private Cloud Appliance by performing some guided steps to create and connect to an instance. After your instance is up and running, this tutorial steps you through creating and attaching a block volume to your instance.

This tutorial also includes optional instructions for deleting all the resources you create.

Prerequisites

To successfully perform this tutorial, you must have the following prerequisites in place.

- You must be able to sign in to the Compute Web UI using your Oracle Cloud Console (OCI) account.
- You must be able to sign in to Oracle Cloud Console using your Oracle Cloud Console (OCI) account. Or have access to a compartment where you can create resources.
- An SSH key pair that's used for user authentication in the instance. If you want to create a key pair for this tutorial, see Managing Key Pairs on Linux Instances.

Tutorial Tasks

Task 1: Create a Compartment on page 641

Task 2: Create a Virtual Cloud Network (VCN) on page 642

Task 3: Create a Subnet on page 642

Task 4: Create an Internet Gateway and Configure Route Rules on page 643

Task 5: Launch an Instance on page 644

Task 6: Get the Instance IP Address on page 645

Task 7: Connect to Your Instance on page 645

Task 8: Add a Block Volume on page 646

Task 9: Attach the Block Volume to an Instance on page 647

Task 10: (Optional) Clean Up Resources on page 647

Task 1: Create a Compartment

On Private Cloud Appliance, compartments help you organize and control access to your resources. A compartment is a collection of resources (such as cloud networks, compute instances, and block volumes) that can be accessed only by those groups that have been given permission by an administrator in your organization.

In this tutorial, you use one compartment for all resources. However, when you're ready to create a production environment, you can separate resources into different compartments. For example, you might place all instances in one compartment and all networking resources in another compartment.

If you want to use an existing compartment rather than create a new compartment, you can do so. Skip this task and use your compartment for the tasks that follow.

For more information about compartments, Managing Compartments.

> **Note:**
>
> On Private Cloud Appliance, IAM resources such as compartments are managed in Oracle Cloud Console (OCI), and synchronized to Private Cloud Appliance.

1. Sign in to the Oracle Cloud Console in OCI using your federated identity credentials.

   For more information, see Sign In to the Console.
2. In the Oracle Cloud Console navigation menu, select **Identity & Security**. Under **Identity**, select **Compartments**.
3. On the Compartments page, select **Create Compartment**.
4. Enter the following information:

   - **Name:** Enter Sandbox.

     Avoid entering confidential information.
   - **Description:** (Required) Enter a description for the compartment.
   - **Create in Compartment:** Select the compartment in which to create this new compartment.

5. Select **Create Compartment**.

   The new compartment is displayed in the list.

   It might take 10 minutes for the new compartment to show up on the Private Cloud Appliance.

> **Note:**
>
> The remaining tutorial tasks use the Compute Web UI instead of the Oracle Cloud Console, except for 10-4: Delete the Compartment on page 649.

Perform the next task:

Task 2: Create a Virtual Cloud Network (VCN) on page 642

Task 2: Create a Virtual Cloud Network (VCN)

Before you can launch an instance, you need a virtual cloud network (VCN) and a subnet.

A VCN is a software-defined equivalent of a traditional network, with firewall rules and various types of communication gateways.

In a production environment, a VCN that you can use for the instance might already exist, and you could use it instead of creating a new VCN. However, in this tutorial, you create a new VCN to learn how to do it.

> **Important:**
>
> This tutorial creates a simple cloud network to make it easy to launch an instance for learning purposes. When you create your production instances, ensure that you create appropriate security lists and route table rules to restrict network traffic to your instances.

For more information, see Managing VCNs and Subnets on page 478.

Avoid entering confidential information in names and tags.

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. Select **Create Virtual Cloud Network**.
3. In the **Create Virtual Cloud Network** dialog box, enter the following information:
   - **Name:** Enter a descriptive name for the cloud network.
   - **Create in Compartment:** Select the Sandbox compartment.
   - **CIDR Block:** Enter a valid CIDR block for the VCN. For example 10.0.0.0/16.
   - **Use DNS hostnames in this VCN:** Indicate whether you want to use DNS host names in the VCN.
   - **DNS Label:** If you selected to use DNS, enter a DNS label or leave the field blank to let the system generate a DNS name for you.
   - **Tagging:** Leave blank. This tutorial doesn't use tags.
4. Select **Create Virtual Cloud Network**.

Perform the next task:

Task 3: Create a Subnet on page 642

Task 3: Create a Subnet

A subnet is a subdivision of your VCN. The subnet directs traffic according to a route table.

For this tutorial, you'll access the instance over the internet using its public IP address, so your route table directs traffic to an internet gateway. The subnet also uses a security list to control traffic in and out of the instance.

Avoid entering confidential information in names and tags.

1. Return to the **Virtual Cloud Networks** page.

   A quick way to do this is to select the name of page in the breadcrumb that's at the top of the page.

If the VCN details page isn't in your breadcrumb, in the navigation menu, select **Networking**, **Virtual Cloud Networks**, then select the name of your VCN.

2. Scroll down to the **Resources** panel, select **Subnets**, and select **Create Subnet**.
3. In the **Create Subnet** dialog box, enter the following information:
   - **Name:** Enter a descriptive name for your subnet.
   - **Create in Compartment:** Select the Sandbox compartment.
   - **CIDR Block:** Enter a valid CIDR block for the subnet. The value must be within the VCN CIDR block. For example, 10.0.0.0/24.
   - **Route Table:** For this tutorial, select the default route table.
   - **Subnet Access:** For this tutorial, select Public Subnet to allow public IP addresses for instances in the subnet.
   - **Use DNS Hostnames in this Subnet:** For this tutorial, leave this unselected.
   - **DHCP Options:** Leave this unselected.
   - **Security Lists:** Select Add Security List and select the default security list.
   - **Tagging:** Leave blank. This tutorial does not use tags.
4. Select **Create Subnet**.

Perform the next task:

Task 4: Create an Internet Gateway and Configure Route Rules on page 643

Task 4: Create an Internet Gateway and Configure Route Rules

An internet gateway is an optional virtual router you can add to your VCN to enable access to your data center network.

The gateway supports connections initiated from within the VCN (egress) and connections initiated from the internet (ingress).

Security list rules control the types of traffic allowed in and out of resources in that subnet. Ensure to only permit the required types of internet traffic.

Each public subnet that needs to use the internet gateway must have a route table rule that specifies the gateway as the target.

Avoid entering confidential information in names and tags.

1. Navigate to your VCN details page.
2. In the **Resources** panel, select **Internet Gateways**.
3. Select **Create Internet Gateway**.
4. Enter the required information:
   - **Name:** Enter a descriptive name for your internet gateway.
   - **Create in Compartment:** Select the Sandbox compartment.
   - **Enabled:** Select whether you want this internet gateway to be enabled upon creation.
   - **Tagging:** Leave blank. This tutorial doesn't use tags.
5. Select **Create**.
6. Under **Resources**, select **Route Tables**.
7. Select the name of the default route table.
8. Scroll down to the **Resources** panel and select **Add Route Rules**.
9. On the **Create Route Table Rule** dialog box, enter the required information:
   - **Target Type:** From the menu, select Internet Gateway.
   - **CIDR Block:** Enter: `0.0.0.0/0` (which means that all nonintra-VCN traffic that isn't already covered by other rules in the route table will go to the target specified in this rule).
   - **Internet Gateway:** From the menu, select the name of the Internet Gateway that you created.
   - **Description:** An optional description of the rule.
10. Select **Create Route Table Rule**.

Perform the next task:

Task 5: Launch an Instance on page 644

Task 5: Launch an Instance

In this task, launch an instance with an image and a shape.

A compute instance is a virtual machine (VM), which is an independent computing environment that runs on top of physical hardware. The virtualization makes it possible to run multiple compute instances that are isolated from each other.

A shape describes the instance resources such as the number of CPUs, amount of memory, and network resources. In a production environment, you would select a shape that best suits workload and application requirements for the instance.

For more information about instances and instance components, see these topics:

- Images
- Compute Shapes
- Working with Instances on page 649

Avoid entering confidential information in names and tags.

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. On the **Instances** page, select **Create Instance**.
3. In the **Launch Instance** dialog box, enter the following information:
   - **Name:** Enter a descriptive name for your compute instance.
   - **Create in Compartment:** Select the Sandbox compartment.
   - **Fault Domain:** Leave the default set to “Automatically select the best fault domain.”
   - **Source Image:**
     - **Source Type:** Select Platform Image.
     - **List of images:** Select an Oracle Linux image.
   - **Shape:** Select a flexible shape (with *Flex* in its name), and configure the number of OCPUs and memory. For this temporary instance, set the OCPU value to 1, and memory to 16. For descriptions of available shapes, see Compute Shapes.
   - **Boot Volume:** Leave the check box empty so that the default boot volume size is created.
   - **Subnet:**
     - **VCN:** Select the VCN you created.
     - **Subnet:** Select the subnet you created.
   - **Public IP Address:** Ensure the check box is checked so that a public IP address is assigned to the instance.
   - **Private IP Address:** Leave the field blank.
   - **Hostname:** You can leave this field blank or enter a hostname.
   - **SSH Keys:** Do one of the following to provide your public SSH key:
     - Select inside the Drag and Drop box to open a file browser and select the file.
     - Drag the file from your file browser listing and drop the file on the Drag and Drop box.
     - Select Paste the public keys, copy your public SSH key text, and paste the text into the field.
   - **Initialization Script:** Leave this area as is.
   - **Network Security Group:** Leave the checkbox clear.
   - **Instance Options:** Don't select **Leave the Legacy Instance Metadata Service Endpoints Disabled**
   - **Availability configuration:** Leave this area as is.
   - **Tagging:** Leave blank. This tutorial doesn't use tags.
4. Select **Create Instance**.
5. Monitor the state of the instance.

   The state is displayed under the icon of the object.

Your instance begins in the PROVISIONING state. After the instance is in the RUNNING state, you can connect to it.

Perform the next task:

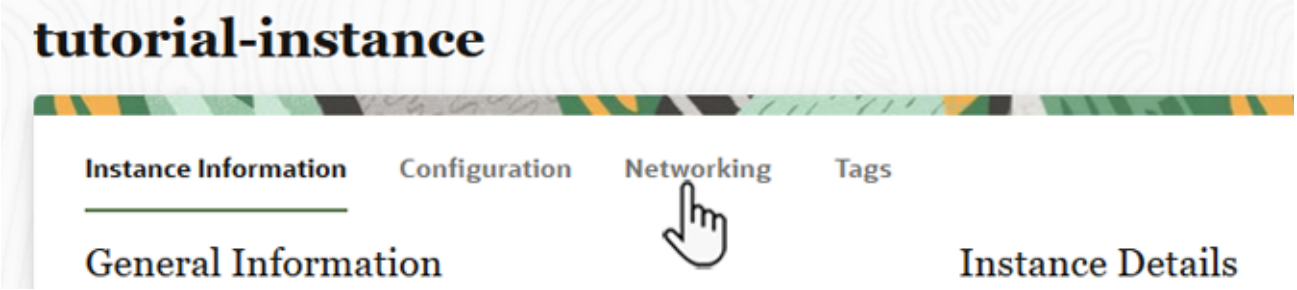
Task 6: Get the Instance IP Address on page 645

Task 6: Get the Instance IP Address

You connect to the instance using SSH with the instance IP address.

1. After creating the instance, you're likely on the instance details page. If not, navigate to the details page: In the Compute Web UI navigation menu, select **Compute**, then select **Instances**. Then select the name of your instance.
2. Select the **Networking** tab.

   The tabs are displayed at the top of the details panel:



3. Under **Instance Access** on the **Networking** tab, note the Public IP Address.

Perform the next task:

Task 7: Connect to Your Instance on page 645

Task 7: Connect to Your Instance

Connect to your instance using SSH.

Usually, you connect to a running instance using a Secure Shell (SSH) connection. Some instances support authenticating your connection with a password. This tutorial assumes you used one of the images provided on Private Cloud Appliance, which creates an instance that authenticates your SSH connection with an SSH key pair.

For the system that you will be connecting from, most Linux and other UNIX operating systems include an SSH client by default.

Microsoft Windows 10 and Microsoft Windows Server 2019 systems should include the OpenSSH client, which you'll need if you created your instance using the SSH keys generated by Oracle Cloud Infrastructure. For other Microsoft Windows versions, you can download a free SSH client called PuTTY.

Before You Begin

- Get the public IP address of your instance, as described in Task 6: Get the Instance IP Address on page 645.
- Get the path to your private key file.
- Get the valid user name.

  The user name is configured in the image used to launch the instance. If you launched an instance using one of the platform images that is provided on the Private Cloud Appliance, the default user is `opc`. See Initial User Account on page 622.

Using the ssh Command

1. Open a terminal window.
2. Use the `ssh` command to connect to your instance.

Syntax:

```
ssh –i private_key_pathname username@public-ip-address
```

- ***private_key_pathname*** is the full path name of the file that contains the private key associated with the instance you want to access.
- ***username*** is the default username for the instance. For this tutorial, `opc` is the username.
- ***public-ip-address*** is your instance IP address.

Example:

```
$ ssh -i /home/flast/.ssh/my_private_key opc@192.0.2.1
```

3. If asked whether you want to continue connecting, type `yes`.

   You are now logged in to your instance.

Perform the next task:

Task 8: Add a Block Volume on page 646

Using PuTTY

This connection method is commonly performed from Microsoft Windows systems.

Use this procedure if the instance uses a key pair that you created using the PuTTY Key Generator. See Creating an SSH Key Pair Using PuTTY Key Generator.

1. Open PuTTY.
2. In the **Category** pane (on the left), select **Session** and enter the following:
   - **Host Name (or IP address):** ***username*****@*****public-ip-address***
     - ***username*** is the default username for the instance. For this tutorial, the username is `opc`.
     - ***public-ip-address*** is your instance IP address.
   - **Port:** 22
   - **Connection type:** SSH
3. In the **Category** pane, expand Window, and then select **Translation**.
4. In the **Remote character set** drop-down list, select UTF-8. The default locale setting on Linux-based instances is UTF-8, and this setting configures PuTTY to use the same locale.
5. In the **Category** pane, expand **Connection**, expand **SSH**, and then select **Auth**.
6. Select **Browse**, and then select your `.ppk` private key file.
7. Select **Open** to start the session.

   If this is your first time connecting to the instance, you might see a message that the server's host key isn't cached in the registry. Select **Yes** or **Accept** to continue the connection.

   > **Tip:**
   >
   > If the connection fails, you might need to update your PuTTY proxy configuration.

Perform the next task:

Task 8: Add a Block Volume on page 646

Task 8: Add a Block Volume

Add additional storage by adding a block volume to your instance.

After a block volume is created, you attach the volume to one or more instances. You can use the volume like a regular hard drive.

Avoid entering confidential information in names and tags.

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Block Volumes**.
2. Select **Create Block Volume**.
3. In the **Create Block Volume** dialog box, enter the following information:
   - **Name:** Enter a descriptive name for your block volume.
   - **Create in Compartment:** Select the Sandbox compartment.
   - **Size:** Leave the default size (1024 GB).
   - **Backup Policy:** Don't select a backup policy.
   - **Tags:** Leave blank. This tutorial doesn't use tags.
4. Select **Create Block Volume**.
5. Monitor the state of the new block volume.

   The state is displayed above the icon for the object. You can also scroll down to the Resources section and check the Work Request.

   Initially, the block volume is in the provisioning state. When the volume changes to the available state, you can attach it to your instance.

Perform the next task:

Task 9: Attach the Block Volume to an Instance on page 647

Task 9: Attach the Block Volume to an Instance

Attach the block volume to an instance.

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. Ensure that the **Sandbox** compartment is selected at the top of the page.
3. In the **Instances** list, select the name of your instance to view its details.
4. Scroll down to the **Resources** panel, and select **Attached Block Volumes**.
5. Select **Attach Block Volume**.
6. In the **Attach Block Volume** dialog box, enter the following information:
   - **Select from Compartment:** Select the Sandbox compartment.
   - **Block Volume:** Select the block volume you created.
   - **Access:** Select Read/Write.
7. Select **Attach to Instance**.

   The attachment process takes about a minute. The volume is ready when the Attachment State for the volume is attached.

   If your block volume isn't displayed, reload the web page.

When a block volume is initially attached to an instance, the instance sees the volume as a new disk. To make the volume available to the instance OS, you need to give the volume a file system and mount it to the OS.

To learn about the block volume and how to make it available to the instance OS, refer to these sections outside of this tutorial:

- Find Your Volume in the Instance on page 721
- Configuring Volumes to Automatically Mount (Linux Instances) on page 724

Perform the next task:

Task 10: (Optional) Clean Up Resources on page 647

Task 10: (Optional) Clean Up Resources

After you've finished with the resources you created for this tutorial, you can delete and release the resources you no longer plan to use.

Perform the following tasks in the order listed:

10-1: Detach and Delete the Block Volume

After you detach a block volume, you can delete it as long as the volume isn't attached to any other instances.

> **Caution:**
>
> You can't undo a termination. Any data on a volume is permanently deleted when the volume is deleted.

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. Select the **Sandbox** compartment.
3. Select the name of your instance.
4. In the **Resources** panel, select **Attached Block Volumes**.
5. Find your volume, select the Actions menu (⋮), then select **Detach**. Confirm the detachment.

   You might need to refresh the web page to see that the block volume is no longer attached.
6. In the Compute Web UI navigation menu, select **Block Storage**, then select **Block Volumes**.
7. Select the **Sandbox** compartment.
8. Find your volume, select the Actions menu (⋮), and then select **Terminate**. Confirm the termination.

Perform the next task:

10-2: Terminate the Instance on page 648

10-2: Terminate the Instance

You can permanently terminate (delete) instances that you no longer need. Any attached VNICs and boot volumes are automatically detached when the instance terminates. Eventually, the instance's public and private IP addresses are released and become available for other instances.

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. Select the **Sandbox** compartment.
3. Find the instance you created, select the Actions menu (⋮), then select **Terminate**.
4. In the **Confirm Instances termination** dialog box, move the "**Permanently delete the attached boot volume**" selector to the right, and select **Confirm**.

   Moving the selector to the right results in the boot volume being permanently deleted, which is appropriate for this tutorial.

   In production, you can leave the selector in the left position to preserve the boot volume for use with another instance. This is convenient when you want to reuse a configured OS or data on the boot volume.

Perform the next task:

10-3: Delete the Subnet, Internet Gateway, and VCN on page 648

10-3: Delete the Subnet, Internet Gateway, and VCN

You can delete networking resources as long as the resource doesn't have any dependencies.

- **Subnets**: Must be empty (for example, no instance VNICs or route rules).
- **VCNs**: Must be empty and have no related resources or attached gateways (for example, no internet gateway, dynamic routing gateway, and so on).

1. In the Compute Web UI navigation menu, select **Networking**, then select **Virtual Cloud Networks**.
2. Select the **Sandbox** compartment.
3. Select the name of your VCN.
4. Under **Resources**, select **Route Tables**.
5. Select the name of the route table.
6. For the route rule you created, select the Actions menu (⋮), select **Delete**, and confirm the deletion.

   The route rule is deleted.

7. In the breadcrumb path at the top of the page, select the name of your VCN.

   The VCN details page is displayed.
8. Under **Resources**, select **Internet Gateways**.
9. For the internet gateway that you created, select the Actions menu (⋮), then select **Delete**, and confirm the deletion.

   The internet gateway is deleted.
10. Under **Resources**, select **Subnets**.
11. For the subnet you created, select the Actions menu (⋮), then select **Delete**, and confirm the deletion.

    The subnet is deleted.
12. At the top of the **VCN details** page, select **Terminate** and confirm the termination.

    The VCN is deleted.

Perform the next task:

10-4: Delete the Compartment on page 649

10-4: Delete the Compartment

You must remove all resources from a compartment before you can delete it, otherwise, the delete action fails and the compartment returns to an active state.

> **Note:**
>
> On Private Cloud Appliance, IAM resources such as compartments are managed in Oracle Cloud Console (OCI), and synchronized to Private Cloud Appliance.

1. Sign in to the Oracle Cloud Console in OCI using your federated identity credentials.

   For more information, see Sign In to the Console.
2. In the Oracle Cloud Console navigation menu, select **Identity & Security**, then under **Identity**, select **Compartments**.
3. For the Sandbox compartment, select the Actions menu (⋮), and then select **Delete**.
4. Confirm the deletion.

This concludes the tutorial.

Working with Instances

On Private Cloud Appliance, you can create compute instances as needed to meet your compute and application requirements. After you create an instance, you can access it securely from your computer, restart it, attach and detach volumes, and delete it.

Creating an Instance

On Private Cloud Appliance, you can create an instance using the Compute Web UI, OCI CLI, and API.

Before You Begin

See Tutorial: Launching Your First Instance on page 640 for information about input you need to create an instance.

The following list describes the minimum information that you must provide to create an instance:

- A name for the instance
- The compartment where you want to create the instance
- An image or boot volume

- A shape – If you're using a platform image, specify a standard/flexible shape for generic workloads and applications. For GPU-accelerated workloads, select a dedicated GPU shape. For descriptions of available shapes, see Compute Shapes.
- A subnet
- A public SSH key

  To log in to the instance, users need either an SSH key or a password, depending on how the image was built. If the instance requires SSH keys for authentication, you must provide the public key when you create the instance. You can't provide the public SSH key after the instance is created.

To create an instance using the OCI CLI, you need the same information as previously listed for the Compute Web UI except that you don't need an instance name. If you don't provide a name for the instance, the default name will be `instance`***`YYYYMMDDhhmmss`***, where ***`YYYYMMDDhhmmss`*** is the creation date and time.

To modify launch options, use the OCI CLI. You can't change launch options or boot volume VPUs per GB after the instance is created.

Instance metadata options enable you to attach important information to instances such as an SSH key, information for cloud-init to use, and labels and proxies for nodes in a Kubernetes Engine cluster node pool. There are metadata key restrictions. See Metadata Key Restrictions on page 636. To configure metadata, create the instance using the OCI CLI `--metadata` or `--extended-metadata` option.

An alternative way to create an instance is to create an instance configuration and use that configuration to launch an instance, as described in Working with Instance Configurations on page 666. When you use an instance configuration to create an instance, you can specify `blockVolumes` and `secondaryVnics`, as shown in the OCI CLI procedure in Creating an Instance Configuration by Entering Configuration Values on page 667.

After you create an instance, you can optionally attach additional VNICs and assign private and public IP addresses. See Creating and Attaching a Secondary VNIC on page 520, Managing Private IP Addresses on page 523, and Managing Public IP Addresses on page 528.

> **Note:**
>
> If you specify a boot volume size that's larger than the default, you need to extend the volume to take advantage of the larger size. See Resizing Volumes on page 735.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. Create or get the following resources and information:
   - An image or boot volume and the compartment where the image or boot volume is located. See Listing Images and Viewing Details on page 621 and Listing Boot Volumes on page 731.
   - A virtual cloud network (VCN) and subnet and the compartment where the VCN and subnet are located. See Creating a VCN on page 479.
   - A public Secure Shell (SSH) key if users will connect to the instance using SSH
2. On the Compute Web UI **Dashboard**, do one of the following to open the **Create Instance** dialog box:
   - Select **Create Instance** in the top-left corner.
   - In the **Compute** block, select **Instances**. At the top of the instances list on the right side, select **Create Instance**.
   - In the **Compute** block, select **Custom Images**. For the image that you want to use to create the instance, select the Actions menu (⋮), then select the **Create Instance From Image**. You might have to change the compartment at the top of the image list to see the image you want.

     When you use the **Create Instance From Image** option, the image name is already entered in the **Create Instance** dialog box and you can't change it. You don't need to enter any of the information described in "Source Image" in the following step.
3. In the **Create Instance** dialog box, enter the following information:

- **Name**: Enter a name for the instance. Instance names have the following characteristics:
  - Can be changed after the instance is created.
  - Doesn't need to be unique.
  - Can contain only alphanumeric characters and the hyphen (-) character.
  - Can be a maximum of 63 characters.
- **Create in Compartment:** Select the compartment where you want to create the instance.
- **Fault Domain:** (Optional) Select a fault domain. By default, the system automatically selects the best fault domain for the instance when the instance is created. If you specify a fault domain, and the requested fault domain can't accommodate the instance, instance launch fails. The fault domain can be changed after the instance is created.
- **Source Image:** Select an image or boot volume.

  a. Select the **Source Type: Platform Image**, **Custom Image**, or **Boot Volume**.

  b. If you selected **Custom Image** or **Boot Volume**, select the compartment where the image or boot volume that you want to use is located.

  c. Select an image or boot volume from the list.

  If you selected **Platform Image**, you see a tabular list with columns **Operating System**, **OS Version**, and **Image Build** (the date the image was built). You can use the drop-down menu arrow to the right of the OS Version to select a different version. For example, for the Oracle Linux operating system, you can use the drop-down menu to select 9, 8, or 7.9.

  If you selected **Custom Image**, you see a tabular list with columns **Name**, **Operating System**, and **OS Version**. You can use the arrows in the column headings to sort the list. You can filter the list by using the **Operating System** drop-down menu above the list of images.

  If you selected **Boot Volume**, you see a tabular list with columns **Name**, **Size (GB)**, and **Created** (the date the boot volume was created). You can use the arrows in the column headings to sort the list. In the **Boot Volume** section (after the Shape section), you can customize the boot volume size.

  If the list is too long to fit in one view, use the arrow to view another page of the list.

  To use a platform image that was previously available but is no longer listed, use the OCI CLI to create the instance and specify the OCID of the image.

  The source image can't be changed after the instance is created.
- **Shape**: If you're using a platform image, select the appropriate shape for the intended use:
  - Generic workloads and applications: Select a flexible shape and configure the number of OCPUs and memory.

    For the OCPU and memory values, select inside each value field to see the minimum and maximum allowed values. The OCPU and memory configuration can be changed after the instance is created.
  - If the system includes an optional GPU expansion and you plan to use the instance for GPU-accelerated workloads, select a dedicated GPU shape. You can select 1-4 GPUs, and the instance is allocated 27 OCPUs and 200GB memory per GPU.

  For a description of the supported shapes, see Compute Shapes.
- **Boot Volume:** (Optional) Check the box to specify a custom boot volume size or volume performance setting.
  - **Boot volume size (GB)**: The default boot volume size for the selected image is shown. To specify a larger size, enter an integer of gigabytes up to 16384 GB (16 TB) or use the increment and decrement arrows. You can't enter a value smaller than the default.

    If you specify a custom boot volume size, you need to extend the partition to take advantage of the larger size. Oracle Linux platform images include the `oci-utils` package. Use the `oci-growfs` command from that package to extend the root partition and then grow the file system. For other OSs or for custom images, follow the instructions for that OS.

- **Boot volume performance (VPUs)**: Use the increment and decrement arrows to toggle between balanced performance (10 VPUs/GB) and high performance (20 VPUs/GB). For more information see Block Volume Performance Options on page 713.

  If you specify the high performance option and a high performance pool exists but the specified image doesn't exist in the high performance pool (the high performance pool was created after the image was imported), the specified image is copied from the capacity pool to the high performance pool. This operation can take 20-30 minutes, depending on the image size, network configuration, and load.

  This copy operation is a one-time operation for each image. Future requests to create an instance specifying this image and the high performance pool won't incur this image copy delay.

  You receive an error message from the image copy if the image is larger than 200 gigabytes. Platform images aren't larger than 200 gigabytes.

- **Subnet:** Select a subnet.

  a. Select a VCN from the list. You might need to change the compartment to the compartment where the VCN is located.
  b. Select a subnet.

- **Public IP Address:** To connect to the instance using SSH, check the **Assign Public IP** box to have a public IP address assigned to the instance. This box is checked by default if you specified a public subnet. If you don't check this box, or if you clear this box, and then want to assign a public IP address later, see Assigning an Ephemeral Public IP Address to an Instance on page 530 for instructions.
- **Private IP Address:** (Optional) Specify an available private IP address from the subnet CIDR. By default, a private IP address is automatically assigned.
- **Hostname:** (Optional) Enter a hostname if you're using DNS within the cloud network. The hostname must be unique across all VNICs in the subnet.

  By default, the instance name is used for the hostname. The hostname can also be configured in the OS after the instance is created.

  If this is a UNIX instance, see Creating a Mount Target on page 763 and Mounting File Systems on UNIX-based Instances on page 774 for more information about setting the host name correctly for mounting file systems.
- **SSH Keys:** To connect to the instance using SSH, provide a public SSH key.

  > **Note:**
  >
  > You can't provide this SSH key after the instance is created.

- **Initialization Script:** (Optional) Provide an initialization script. This is a file of data to be used for custom instance initialization.
- **Network Security Group:** (Optional) By default, the new instance isn't attached to any NSG. Check the box labeled Enable Network Security Group to add the primary VNIC for this instance to one or more NSGs.

  a. Select an NSG from the drop-down list. You might need to change the compartment to find the NSG you want.
  b. Select Add Another NSG if you want to attach to another NSG.
  c. To remove an NSG from the list, select the trash can to the right of that NSG. To remove the last NSG or all NSGs, clear the **Enable Network Security Groups** box.

  To update NSG attachments for this instance later, see Updating a VNIC on page 522.

  See Controlling Traffic with Network Security Groups on page 501 for information about NSGs.
- **Instance Options**: Check the box to disable Legacy Instance Metadata Service Endpoints. By default, legacy (`/v1`) Instance Metadata Service (IMDS) routes are enabled. If you have upgraded your applications to use `/v2` endpoints, check this box to disable `/v1` endpoints. For more information about the Instance Metadata Service, see Retrieving Instance Metadata from Within the Instance on page 657. For more information about upgrading your applications, see Upgrading to IMDS Version 2 Endpoints on page 657.
- **Availability configuration:** (Optional) Specify how to handle this instance during compute node maintenance"

- **Let** Private Cloud Appliance **choose the best migration option**

  This option is selected by default to allow the system to choose the best option to handle this instance during compute node maintenance. The best option typically is live migration to a healthy compute node. You can't change this setting. If this instance should not be live migrated, for example live migration isn't supported for instances in a Microsoft Windows cluster, then set the `PCA_no_lm` free-form tag to True to prevent live migration for this instance.
- **Restore instance lifecycle state after infrastructure maintenance**

  This option is selected by default to specify that running instances should be automatically restarted after a maintenance operation such as live migration. If this box isn't checked, the instance is recovered in the stopped state.

- **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.

4. Select **Create Instance**.

   On success, the instance details page is displayed. On the **Configuration** tab, the **Shape Configuration** column shows the shape, the number of OCPUs, the network bandwidth, and the total memory. On the **Networking** tab, the **VNIC** column shows the VCN and subnet, and the **Instance Access** column shows the primary private IP address and any assigned public IP address.

   To check the status of the instance launch, scroll to the **Resources** section and select **Work Request(s)**.

   If instance launch fails because of resource constraints, try remedies such as the following:

   - Specify a different fault domain, or don't specify any fault domain and allow the system to choose it.
   - Specify a less resource-intensive shape.
   - Stop an instance that you no longer need.
   - Terminate an instance that you no longer need.

   If the status of the work request is Failed, and no reason is listed for the failure, the cause of the failure might be temporary. Wait a short time and then retry the instance create.

Using the OCI CLI

Use the oci compute instance launch command and required parameters to create an instance.

```
oci compute instance launch --availability-domain availability_domain --
compartment-id compartment_OCID --shape shape --subnet-id subnet_OCID --
source-details file://image_info.json [OPTIONS]
```

For a complete list of required and optional parameters, use the following command:

```
oci compute instance launch -h
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Procedure

1. Create or get the following resources and information:

   - The name of the availability domain that you want to use: `oci iam availability-domain list`
   - The OCID of the compartment where you want to create the instance: `oci iam compartment list`
   - The name of the shape for this instance. Select the appropriate shape for the intended use:

     - Generic workloads and applications – select a flexible shape (with *Flex* in its name) – for example:

       ```
       VM.PCAStandard.E5.Flex
       ```

You must also specify the shape configuration, as shown in the following example. You must provide a value for `ocpus`. The `memoryInGBs` property is optional; the default value in GBs is 16 times the number of `ocpus`.

```
--shape-config '{"ocpus": 32, "memoryInGBs": 512}'
```

- GPU-accelerated workloads:

```
VM.GPU.L40S.1 | VM.GPU.L40S.2 | VM.GPU.L40S.4
```

GPU shapes are not flexible. There are no custom configuration parameters.

See Compute Shapes.

The shape configuration can be changed after the instance is created.

- The OCID of the subnet where the VNIC that is attached to this instance will be created: `oci compute vnic-attachment list`
- If you provide a value for the `--hostname-label` option, see the description of Hostname in the Console tab.
- Gather one of the following values to specify the image source – either an image or a boot volume.
  - The OCID of the image used to boot the instance:

```
oci compute image list
```

> **Note:**
>
> Don't select an image that has "`-OKE-`" in its display name. The "`-OKE-`" images can only be used withKubernetes Engine (OKE).

  - The OCID of the boot volume used to boot the instance:

```
oci compute boot-volume-attachment list
```

- A public Secure Shell (SSH) key to connect to the instance using SSH.

> **Note:**
>
> You can't provide this SSH key after the instance is created.

For a complete list of required and optional parameters, use the following command:

```
$ oci compute instance launch -h
```

For information about `--display-name` and `--hostname-label` values, see descriptions in the Console tab on this page.

2. Construct an argument for the `--source-details` option.

The `--source-details` argument can be a JSON file or a command-line string. Use the following command to show the correct format of the JSON properties and values:

```
oci compute instance launch --generate-param-json-input source-details
```

For information about `bootVolumeSizeInGBs`, see "Boot volume size" in the Console tab.

For information about `bootVolumeVpusPerGB`, see "High Performance" in the Console tab.

> **Note:**
>
> When you later `list` or `get` this instance, the value of `bootVolumeVpusPerGB` is `null` because this boot volume property isn't stored in the instance object after the instance is created. To check the

value after instance launch, use the `bv boot-volume list` or `get` command and check the value of `vpus-per-gb`.

3. (Optional) Construct an argument for the `--launch-options` option.

   Only the `firmware` property can be changed. The default value is BIOS. You can alternatively specify UEFI_64. If you don't provide a correct value for `firmware`, the instance might not launch. You can't update the value of the `firmware` property with the `instance update` command.

   The following shows the default values:

   ```
   {
     "bootVolumeType": "PARAVIRTUALIZED",
     "firmware": "BIOS",
     "isConsistentVolumeNamingEnabled": false,
     "is-pv-encryption-in-transit-enabled": false,
     "networkType": "PARAVIRTUALIZED",
     "remoteDataVolumeType": "PARAVIRTUALIZED"
   }
   ```

   To change the value of the `firmware` property, specify the following option:

   ```
   --launch-options file://launch_options.json
   ```

   Where the following is the content of the `launch_options.json` file:

   ```
   {
     "bootVolumeType": "PARAVIRTUALIZED",
     "firmware": "UEFI_64",
     "isConsistentVolumeNamingEnabled": false,
     "is-pv-encryption-in-transit-enabled": false,
     "networkType": "PARAVIRTUALIZED",
     "remoteDataVolumeType": "PARAVIRTUALIZED"
   }
   ```

4. (Optional) Construct an argument for the `--metadata` or `--extended-metadata` option.

   Custom user data can be attached to the instance using the `--metadata` and `--extended-metadata` options. Metadata key/value pairs are string/string maps in JSON format. Extended metadata can be nested JSON objects. Extended metadata can be nested JSON objects.

   The combined size of the metadata and extended metadata can be a maximum of 32,000 bytes.

   SSH keys can alternatively be provided in the file argument of the `--ssh-authorized-keys-file` option. User data can alternatively be provided in the file argument of the `--user-data-file` option. Use the `-h` option for more information.

   In the example in the next step, the authorized keys file contains one or more public SSH keys in the format required by the SSH `authorized_keys` file. Use a newline character to separate multiple keys. SSH public keys can be provided as the value of the `ssh_authorized_keys` key in the `--metadata` option, or in the file argument of the `--ssh-authorized-keys-file` option. Use `-h` for more information.

5. Run the instance launch command.

   Syntax:

   ```
   oci compute instance launch --availability-domain availability_domain_name
    \
   --compartment-id compartment_OCID --shape shape --subnet-id subnet_OCID \
   --source-details file://image_info.json
   ```

   Example:

If you're using a public subnet, a public IP address is assigned by default, or you can set the `--assign-public-ip` option value to `true`. If you need to assign a public IP address later, see Assigning an Ephemeral Public IP Address to an Instance on page 530 for instructions.

If you have upgraded your applications to use `/v2` Instance Metadata Service (IMDS) endpoints, use the `--instance-options` option to set `areLegacyImdsEndpointsDisabled` to `true`. By default, legacy (`/v1`) Instance Metadata Service routes are enabled. For more information about the Instance Metadata Service, see Retrieving Instance Metadata from Within the Instance on page 657.

```
$ oci compute instance launch --availability-domain AD-1 --
compartment-id <compartment_OCID> --display-name ops1 --shape
 VM.PCAStandard.E5.Flex --subnet-id <subnet_OCID> --source-details
 '{"bootVolumeSizeInGBs":100,"bootVolumeVpusPerGB":20,"imageId":"<image_OCID>","sourc
 --assign-public-ip true --ssh-authorized-keys-file ./.ssh/<ssh_key_file>
 --instance-options '{"areLegacyImdsEndpointsDisabled": true}'
{
  "data": {
    "agent-config": null,
    "availability-config": {
      "is-live-migration-preferred": null,
      "recovery-action": "RESTORE_INSTANCE"
    },
    "availability-domain": "AD-1",
    "capacity-reservation-id": null,
    "compartment-id": "ocid1.compartment.unique_ID",
    "dedicated-vm-host-id": null,
    "defined-tags": {},
    "display-name": "ops1",
    "extended-metadata": null,
    "fault-domain": "FAULT-DOMAIN-1",
    "freeform-tags": {},
    "id": "ocid1.instance.unique_ID",
    "image-id": "ocid1.image.unique_ID",
    "instance-options": {
      "are-legacy-imds-endpoints-disabled": true
    },
    "ipxe-script": null,
    "launch-mode": "PARAVIRTUALIZED",
    "launch-options": {
      "boot-volume-type": "PARAVIRTUALIZED",
      "firmware": "BIOS",
      "is-consistent-volume-naming-enabled": false,
      "is-pv-encryption-in-transit-enabled": false,
      "network-type": "PARAVIRTUALIZED",
      "remote-data-volume-type": "PARAVIRTUALIZED"
    },
    "lifecycle-state": "PROVISIONING",
    "metadata": {
      "ssh_authorized_keys": <public_ssh_key>"
    },
    "platform-config": null,
    "preemptible-instance-config": null,
    "region": "region_name",
    "shape": "VM.PCAStandard.E5.Flex",
    "shape-config": {
      "baseline-ocpu-utilization": null,
      "gpu-description": null,
      "gpus": null,
      "local-disk-description": null,
      "local-disks": null,
      "local-disks-total-size-in-gbs": null,
      "max-vnic-attachments": 16,
```

```
      "memory-in-gbs": 256.0,
      "networking-bandwidth-in-gbps": 24.6,
      "ocpus": 16.0,
      "processor-description": null
    },
    "source-details": {
      "boot-volume-size-in-gbs": 100,
      "bootVolumeVpusPerGB": 20,
      "image-id": "ocid1.image.unique_ID",
      "kms-key-id": null,
      "source-type": "image"
    },
    "system-tags": null,
    "time-created": "2021-09-22T20:20:04.715304+00:00",
    "time-maintenance-reboot-due": null
  },
  "etag": "92180faa-3660-446c-9559-c12a6e6111f9",
  "opc-work-request-id": "ocid1.workrequest.unique_ID"
}
```

Use the `work-requests work-request get` command to monitor the status of the instance launch:

```
$ oci work-requests work-request get --work-request-id
 ocid1.workrequest.unique_ID
```

If the status of the work request is Failed, and no reason is given for the failure, the cause of the failure might be temporary. If no reason is given for the failure, wait a short time and then retry the instance create.

Using the API

Use the LaunchInstance operation to create an instance.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Retrieving Instance Metadata from Within the Instance

On Private Cloud Appliance, the Instance Metadata Service (IMDS) serves information about a running instance to users who are logged in to that instance. IMDS also provides information to cloud-init that you can use for various system initialization tasks.

> **Note:**
>
> To access IMDS metadata, use an instance image that's provided by Oracle.

The IMDS metadata includes instance information such as the following:

- The SSH public key that enables users to log in to the instance
- Instance attached VNICs, VNIC IDs
- Instance CIDR blocks

In general, the IMDS instance metadata includes the following information:

- The same information that you see on the details page of an instance in the Compute Web UI and in the output of the instance `get` OCI CLI.
- Custom information that you add to an instance by using the `--metadata`, `--extended-metadata`, `--ssh-authorized-keys-file`, and `--user-data-file` options of the instance `launch` command. This metadata can't be updated after instance launch. For a user logged into the instance, the instance metadata is read-only.

Upgrading to IMDS Version 2 Endpoints

The Instance Metadata Service is available in two versions: version 1 and version 2.

> **Important:**
>
> To increase the security of metadata requests, upgrade all applications to use the IMDS version 2 endpoints, if supported by the image. Then disable use of IMDS version 1 endpoints.

IMDS version 2 endpoints (IMDSv2) are supported on the Oracle Linux images listed in Guest Operating System Matrix on page 53. Other platform images and most other images don't support IMDSv2.

For each instance, perform the following steps to upgrade to IMDSv2:

1. Identify applications that are making IMDSv1 requests.

   For example, `cloud-init` makes requests to `/v#` instance endpoints.
2. Migrate the identified applications to support IMDSv2 endpoints.

   When you use `/v2` endpoints, you must include the "Authorization: Bearer Oracle" header. See the examples in Retrieving IMDS Instance Metadata on page 658.
3. Disable IMDSv1 endpoints.

   Perform one of the following steps, as described in Creating an Instance on page 649.

   - On the details page of an instance, under Instance Details, check the value of Legacy Instance Metadata Service Endpoints. If the value of Legacy Instance Metadata Service Endpoints is Enabled, select Edit on the Controls menu, and check the box for Legacy Instance Metadata Service Endpoints Disabled.
   - In the output from `instance list` or `instance get`, under `instance-options`, check the value of `are-legacy-imds-endpoints-disabled`. If the value of `are-legacy-imds-endpoints-disabled` is `null` or `false`, use the `instance update` command to specify the following option:

     ```
     --instance-options '{"areLegacyImdsEndpointsDisabled": true}'
     ```

   Future requests to legacy (`v1`) endpoints will be rejected with a 404 not found error.

Retrieving IMDS Instance Metadata

To retrieve the IMDS instance metadata, follow these steps:

1. Log in to the instance.
2. Use a cURL command to retrieve the metadata information from the HTTP endpoint.

   Information is provided through an HTTP endpoint that listens on 169.254.169.254. If an instance has multiple VNICs, you must send the request using the primary VNIC.

   Use the `instance` command to retrieve the instance metadata. Use the `vnics` command to retrieve the VNIC data.

   If you are using `/v2` endpoints, as shown in the following examples, then you must include the "Authorization: Bearer Oracle" header.

   Example: Instance Metadata

   ```
   $ curl -H "Authorization: Bearer Oracle" -L http://169.254.169.254/opc/v2/
   instance/
   {
       "availabilityDomain": "AD-1",
       "faultDomain": "FAULT-DOMAIN-1",
       "compartmentId": "ocid1.compartment.unique_ID",
       "displayName": "dev1",
       "hostname": "hostname",
       "id": "ocid1.instance.unique_ID",
       "image": "ocid1.image.unique_ID",
       "metadata": {
           "ssh_authorized_keys": "public_SSH_key"
       },
   ```

```
    "region": "PCA",
    "canonicalRegionName": "PCA",
    "ociAdName": "PCA",
    "regionInfo": null,
    "shape": "VM.PCAStandard.E5.Flex",
    "state": "RUNNING",
    "timeCreated": 1634943279000,
    "agentConfig": null
}
```

To retrieve a single value, specify the key name as shown in the following example.

Example: VNIC Metadata

```
$ curl -H "Authorization: Bearer Oracle" -L http://169.254.169.254/opc/v2/
vnics/
[
    {
        "vnicId": "ocid1.vnic.unique_ID",
        "privateIp": "privateIp",
        "vlanTag": 0,
        "macAddr": "00:13:97:9f:16:32",
        "virtualRouterIp": "virtualRouterIp",
        "subnetCidrBlock": "subnetCidrBlock"
    }
]
```

You can view all the data for one of multiple VNICs by specifying the array index for that VNIC data, or you can retrieve a single value for that specified VNIC:

```
$ curl -H "Authorization: Bearer Oracle" -L http://169.254.169.254/opc/v2/
vnics/0/privateIp
privateIp
```

Updating an Instance

On Private Cloud Appliance, you can change instance parameters.

In addition to updating the properties of an instance, you might want to attach additional block volumes or secondary VNICs. See Creating and Attaching Block Volumes on page 716 and Creating and Attaching a Secondary VNIC on page 520. You can also specify block volumes and secondary VNICs when you use an instance configuration to create an instance.

If you didn't add a public IP address when you created the instance, and you want to assign a public IP address now, see Assigning an Ephemeral Public IP Address to an Instance on page 530 for instructions.

You can update the display name, fault domain, shape, instance options, availability configuration, and tags of an instance. Using the OCI CLI, you can also update the instance metadata. See the descriptions of these properties in Creating an Instance on page 649.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

When you update an instance using the Compute Web UI, you can change the following instance parameters:

- **Instance display name**

  Avoid entering confidential information
- **Fault domain**

  When you change the fault domain of a stopped instance, the new fault domain is set in the instance properties. When you change the fault domain of a running instance, the instance is stopped, moved, and started on a new

compute node in the new fault domain. See Stopping, Starting, and Resetting an Instance on page 661 for how to prepare for an instance to stop. Stopping and starting an instance can take up to five minutes.

If you specify a fault domain, and the requested fault domain can't accommodate the instance, instance restart fails; the instance remains stopped. The new fault domain specification remains in the instance properties.

- **Shape**

  When you change the shape of a stopped instance, the new shape and shape configuration are set in the instance properties. When you change the shape of a running instance, the instance is stopped, reconfigured, and restarted. See Stopping, Starting, and Resetting an Instance on page 661 for how to prepare for an instance to stop. Stopping and starting an instance can take up to five minutes.

  If you're using a flexible shape, you must specify the number of OCPUs you want and you can specify the total amount of memory you want. See Compute Shapes.

  If the specified shape and shape configuration can't be accommodated in the fault domain, instance restart fails; the instance remains stopped. The new shape and shape configuration remain in the instance properties.
- **Instance Options**

  If you have upgraded your applications to use `/v2` Instance Metadata Service (IMDS) endpoints, check this box to disable `/v1` endpoints. For more information about the Instance Metadata Service, see Retrieving Instance Metadata from Within the Instance on page 657.
- **Availability configuration**

  This sets whether an instance that has been stopped by the Compute service is restarted when resources become available (the default) or remains stopped.
- **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.

1. On the Compute Web UI **Dashboard**, in the **Compute** block, select **Instances**.
2. At the top of the page, select the compartment that contains the instance.
3. For the instance that you want to update, select the Actions menu (⋮), then select **Edit**.
4. In the **Edit *`instance_name`*** dialog box, make the changes.
5. Select **Save Changes**.

   If you changed the fault domain, shape, or shape configuration of a running instance, you must confirm that you understand that the instance must be rebooted. For these changes, the instance is stopped, reconfigured, and restarted.

Using the OCI CLI

Use the oci compute instance update command and required parameters to update the specified instance.

```
oci compute instance update --instance-
id <instance_OCID> <options_with_values_to_update> [OPTIONS]
```

For descriptions of instance properties that you can change using the OCI CLI, enter the following command and scroll to Optional Parameters:

```
$ oci compute instance update -h
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Procedure

1. Get the OCID of the instance that you want to update:

   ```
   oci compute instance list
   ```

If you want to change the fault domain of the instance, get the OCID of the fault domain:

```
$ oci iam fault-domain list --compartment-id <compartment_OCID> --
availability-domain AD-1
```

If you want to change the shape of the instance, get the name of the shape:

```
$ oci compute shape list --compartment-id <compartment_OCID> --image-
id <image_OCID>
```

2. Run the instance update command.

   Syntax:

```
oci compute instance update --instance-
id <instance_OCID> <options_with_values_to_update>
```

Using the API

Use theUpdateInstance operation to update the specified instance.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Stopping, Starting, and Resetting an Instance

On Private Cloud Appliance, you can control the state of an instance using the Compute Web UI, OCI CLI, and API.

The following note applies to all the stop and reset actions.

> **Important:**
>
> For soft stop and soft reset, any application that's running on the instance that takes more than 15 minutes to shut down could be improperly stopped, resulting in data corruption. For stop and reset, any application that's running on the instance is immediately stopped, possibly resulting in data corruption. To avoid stopping the instance while applications are running, manually shut down the instance using the commands available in the instance OS.
>
> After the instance is shut down from the OS, then stop, soft stop, reset, or soft reset the instance.

You can perform the following actions on an instance:

- **STOP**. Power off the instance. Compute resources are released and the instance is disconnected and unassigned from the compute node. See the important note at the beginning of this section.

  Stopping an instance can take up to five minutes.

  An instance that's Stopped can't be migrated to a different compute node.
- **START**. Power on the instance. The Compute service attempts to restart the instance in the same fault domain that it was in when it was stopped, or in the specified fault domain if the fault domain was updated while the instance was stopped. If the instance start operation fails, the instance remains stopped.

  If the start operation fails because of resource constraints, you could specify a different fault domain for the instance (see Updating an Instance on page 659), change the configuration of the instance, or stop, reconfigure, or delete other instances.

  If the start operation fails, and no reason is provided for the failure, the cause of the failure might be temporary. If no reason is provided for the failure, wait a short time and then retry the instance start.

  Starting an instance can take up to five minutes.
- **RESET**. Power off the instance and then power it back on. See the descriptions of STOP and START and the important note at the beginning of this section.

- **SOFTSTOP**. Gracefully shut down the instance by sending a shutdown command to the OS. After waiting 15 minutes for the OS to shut down, the instance is powered off. If the applications that run on the instance take more than 15 minutes to shut down, they could be improperly stopped, resulting in data corruption. To avoid this, manually shut down the instance using the commands available in the OS before you softstop the instance. See the description of STOP and the important note at the beginning of this section.
- **SOFTRESET**. Gracefully reboot the instance by sending a shutdown command to the OS. After waiting 15 minutes for the OS to shut down, the instance is powered off and then powered back on. See the important note at the beginning of this section.

Using the Compute Web UI

1. On the Compute Web UI **Dashboard**, in the **Compute** block, select **Instances**.
2. At the top of the page, select the compartment that contains the instance you want to manage.
3. For the instance that you want to manage, select Actions menu (⋮), then select **Start**, **Stop**, or **Reset**.

   Alternatively, in the instance list, select the name of the instance to display the details page for that instance. Select the **Controls**, then select the **Start**, **Stop**, or **Reset** option.
4. If applicable, enable force stop or force reboot options.

   By default, selecting **Stop** selects Soft Stop and selecting **Reset** selects Soft Reset. To stop or reset the instance immediately, select the **Force** option on the confirmation dialog box.

   - On the **Stop the instance named <instance_name>** dialog box, enable **Force stop the instance by immediately powering off**.
   - On the**Reboot the instance named <instance_name>** dialog box, enable **Force reboot the instance by immediately powering off, then powering back on**.
5. Select **Start Instance**, **Stop Instance**, or **Reboot Instance** on the confirmation dialog box.

   On the instance details page, under **Resources**, select **Work Request(s)** to check the status of the instance stop, start, or reboot.

Using the OCI CLI

Use the oci compute instance action command and required parameters to stop, start, or reset the specified instance.

```
oci compute instance action --instance-id <instance_OCID> --action {START |
 STOP | RESET | SOFTSTOP | SOFTRESET} [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the InstanceAction operation to stop, start, or reset the specified instance.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting an Instance

On Private Cloud Appliance, you can delete instances using the Compute Web UI, OCI CLI, and API.

By default, the boot volume of the instance is preserved when you delete (terminate) the instance. You can attach the boot volume to a different instance as a data volume, or use it to create a new instance.

You have the option to permanently delete the attached boot volume when you delete the instance. If you preserve the attached boot volume and then later decide to permanently delete the volume, see Deleting a Boot Volume on page 734.

Using the Compute Web UI

1. On the Compute Web UI **Dashboard**, in the **Compute** block, select **Instances**.
2. At the top of the page, select the compartment that contains the instance you want to delete.
3. Select the name of the instance you want to delete.

4. On the instance details page, select **Controls** (upper right corner), then select **Terminate**.
5. On confirmation dialog box, do one of the following actions:
   - To keep the boot volume: Leave the slider in the default position.
   - To permanently delete the boot volume: Move the slider to the right.
6. Select **Confirm**.
7. Under **Resources**, select **Work Request(s)** to check the status of the termination.

   After the TerminateInstance operation is 100% complete and in state Succeeded, the instance remains visible in the instance list in state Terminated for at least 24 hours, up to 24.5 hours. No further action is needed to terminate the instance.

Using the OCI CLI

Use the oci compute instance terminate command and required parameters to delete the specified instance.

```
oci compute instance terminate --instance-id <instance_OCID> [OPTIONS]
```

To permanently delete the attached boot volume when the instance is terminated, specify `--preserve-boot-volume false` on the command line.

Use the `work-requests work-request get` command to check the status of the instance terminate. After the `TerminateInstance` operation is `percent-complete` `100.0` and `status` `SUCCEEDED`, the instance remains visible in instance `list` or `get` in `lifecycle-state` `TERMINATED` for at least 24 hours, up to 24.5 hours. No further action is needed to terminate the instance.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the TerminateInstance operation to delete the specified instance.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Configuring Instances for Calling Services

A Private Cloud Appliance compute instance can be configured to enable applications running on the instance to call services and manage resources similar to the way Private Cloud Appliance users call services to manage resources.

The IAM service feature that enables instances to be authorized actors (or principals) to perform actions on service resources is called an *instance principal*.

Perform the following steps to set up and use an instance as a principal:

- Configure the instance firewall to enable the instance to access service endpoints. See Configuring Instance Firewalls to Allow Calling Services on page 663.
- Ensure the instance is included in a dynamic group (configured in your tenancy) that grants permissions to access the required resources. See Managing Dynamic Groups.

  The instance must be created in or moved to a compartment that's named in a matching rule of the dynamic group, or the instance must have a resource tag assigned that's named in a matching rule. See Writing Matching Rules to Define Dynamic Groups.

Configuring Instance Firewalls to Allow Calling Services

This section describes how to modify the instance firewall configuration and how to create a `systemd` service to restore the changes if the system reboots.

Modify the Firewall Configuration

1. As a privileged user, modify the instance firewall configuration to enable the instance to access service endpoints such as `iaas` and `identity`.

2. Use the `iptables` command to add the following `BareMetalInstanceServices` rules to the instance firewall:

```
iptables -I BareMetalInstanceServices 14 -p tcp -d 169.254.169.254
 --dport 443 -j ACCEPT
iptables -I BareMetalInstanceServices 14 -p tcp -d 169.254.240.254
 --dport 443 -j ACCEPT
```

   The first entry is required for all endpoints. The second entry is required to contact the Object Storage endpoint.

Make the Configuration Changes Persistent

Use the following procedure to make these firewall configuration changes persist across instance reboots.

1. Save the updated IP tables configuration.

```
iptables-save > /etc/sysconfig/iptables.rules
```

2. Create a script to automatically restore the current (modified) firewall configuration on reboot.

   In this example, the script is named `/sbin/restore-iptables.sh`. The following is the content of the file `/sbin/restore-iptables.sh`:

```
#!/bin/sh
/sbin/iptables-restore < /etc/sysconfig/iptables.rules
```

3. Set the executable bit on the script.

```
chmod +x /sbin/restore-iptables.sh
```

4. Create a `systemd oneshot` service to execute the `/sbin/restore-iptables.sh` script at boot time.

   In this example, the service is named `/etc/systemd/system/restore-iptables.service`. The following is the content of the file `/etc/systemd/system/restore-iptables.service`:

```
[Unit]
Description=Restore IP Tables
After=cloud-final.service

[Service]
ExecStart=/sbin/restore-iptables.sh
User=root
Group=root
Type=oneshot

[Install]
WantedBy=multi-user.target
```

5. Reload the `systemd` manager configuration, and enable the service to run at boot time.

```
systemctl daemon-reload
systemctl enable restore-iptables
```

Configuring Instance Certificates to Allow Calling Services

On Private Cloud Appliance, by default, endpoints (such as `iaas` and `identity`) offer a certificate that's signed by a CA that's specific to that Private Cloud Appliance. By default, OSs don't trust certificates that are signed by a CA that's specific to this Private Cloud Appliance. If the OS doesn't trust the certificates that are offered, attempts to use the OCI SDK or OCI CLI fail with a `CERTIFICATE_VERIFY_FAILED` error.

Implement one of the solutions described in this topic to successfully use the OCI SDK or OCI CLI on the instance.

> **Important:**
>
> Any user who can SSH to the instance automatically inherits the privileges granted to the instance.

Option 1: Bring Your Own Certificate (BYOC)

On Private Cloud Appliance, you can provide your own Certificate Authority (CA) certificates which enables you to use your CA trust chain. To use your own certificates, open a Support Request, and request to use your own certificates. See Creating a Support Request. To access support, sign in to the Oracle Cloud Console as described in Sign In to the OCI Console.

On a Linux OS, the following command lists CAs that are trusted by default:

```
trust list --filter=ca-anchors
```

Option 2: Specify in the SDK Code the CA Bundle to Use

This method copies the Private Cloud Appliance-specific CA bundle to the instance, but doesn't verify the server's certificate (`--insecure`). To ensure security, verify the content of the retrieved bundle (`external_ca.crt`).

1. Retrieve the certificate from the `iaas` endpoint of the Private Cloud Appliance.

   ```
   curl --insecure -sS -o external_ca.crt --noproxy "*" https://
   iaas.ccc_name.domain_name/cachain
   ```

   This command could be in a script that is passed to the instance at launch time by using either the `--user-data-file` option or the `--metadata` option with a `user_data` field. The script will be run by cloud-init inside the instance during init, saving the effort of manually retrieving this certificate file on many instances.
2. Verify the content of the CA bundle saved in the `external_ca.crt` file.
3. Specify the CA bundle in the Python SDK code.

   ```
   signer = oci.auth.signers.InstancePrincipalsSecurityTokenSigner(
       federation_client_cert_bundle_verify="/home/opc/external_ca.crt"
   )
   identity_client = oci.identity.IdentityClient(config={}, signer=signer)
   identity_client.base_client.session.verify = "/home/opc/external_ca.crt"
   ```

Option 3: Globally Trust the Private Cloud Appliance CA Bundle

This method is the same as the preceding method with the following difference: Instead of specifying the CA bundle in the SDK code, this method adds the CA bundle to the trust chain.

> **Important:**
>
> When the CA bundle is added to the trust chain, every application on this compute instance will trust certificates signed with the CA specified in this bundle. Consider whether this is an acceptable security risk.

1. Retrieve the certificate from the `iaas` endpoint of the Private Cloud Appliance.

   ```
   curl --insecure -sS -o external_ca.crt --noproxy "*" https://
   iaas.ccc_name.domain_name/cachain
   ```
2. Verify the content of the CA bundle saved in the `external_ca.crt` file.
3. Update the global CA trust chain.

   ```
   cp external_ca.crt /etc/pki/ca-trust/source/anchors/
   update-ca-trust extract
   ```

Steps 1 and 3 in this method could be in a script that's passed to the instance at launch time by using either the `--user-data-file` option or the `--metadata` option with a `user_data` field. The script runs by cloud-init inside the instance during init, saving the effort of performing these steps manually on many instances.

Working with Instance Configurations

On Private Cloud Appliance, an instance configuration contains settings that are used to create compute instances. Instance configurations enable you to consistently create instances with the same configuration without reentering configuration values. You can use an instance configuration to create a single instance or to create an instance pool.

Creating an Instance Configuration

On Private Cloud Appliance, you can create an instance configuration from an existing instance (a template instance) or by entering the individual configuration settings.

Creating an Instance Configuration from an Instance

On Private Cloud Appliance you can create an instance configuration by using the configuration information from an existing compute instance (a template instance).

Note the following when you use a template instance to create an instance configuration:

- The new instance configuration doesn't include any information from the boot volume of the template instance. For example, installed applications, binaries, and files that are on the template instance aren't included in the instance configuration.

  Use the following procedure to create an instance configuration that includes the custom setup from the template instance:

  1. Create a custom image from the template instance. See Creating a Custom Image from an Instance on page 622.
  2. Use the custom image to create a new instance. See Creating an Instance on page 649.
  3. Use the instance that you created in the preceding step as the template to create the instance configuration.
- Instances with replicated boot volumes can't be used as template instances for an instance configuration. For example, instances that are included in a disaster recovery configuration can't be used as template instances for an instance configuration.
- The instance configuration doesn't include the contents of any block volumes that are attached to the template instance.
- Instances that are created from this new instance configuration are placed in the same compartment as the template instance.

To include block volumes or secondary VNICs in the configuration, or to specify the compartment where new instances will be created, create the instance configuration as described in Creating an Instance Configuration by Entering Configuration Values on page 667.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. At the top of the page, select the compartment that contains the instance that you want to use to create the new instance configuration.
3. Select the name of the instance that you want to use to create the new instance configuration.
4. On the instance details page, select **Controls** (upper right corner), then select **Create Instance Configuration**.
5. In the **Create Instance Configuration** dialog box, enter the following information:
   - **Name:** Enter a name for the instance configuration.
   - **Compartment:** Select the compartment where this instance configuration will be created.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
6. Select **Create Instance Configuration**.

Using the OCI CLI

Use the oci compute-management instance-configuration create-from-instance command and required parameters to create an instance configuration by using the configuration information from an existing compute instance.

```
oci compute-management instance-configuration create-from-instance --
compartment-id <compartment_OCID> --instance-id <instance_OCID> --display-
name <IC_name>
```

The specified compartment is where this instance configuration is created.

The specified display name is the name of the instance configuration. If you don't provide a value for the `--display-name` option, the default name of the instance configuration is `instanceconfiguration`***`YYYYMMDDhhmmss`***, where ***`YYYYMMDDhhmmss`*** is the creation date and time.

The output of this command is the same as the output of the `instance-configuration get` command.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateInstanceConfiguration operation to create an instance configuration using an instance.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Creating an Instance Configuration by Entering Configuration Values

On Private Cloud Appliance, you can create an instance configuration by entering values for individual instance configuration settings.

> **Note:**
>
> If you specify a boot volume size that's larger than the default, you need to extend the volume to take advantage of the larger size. See Resizing Volumes on page 735.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instance Configurations**.
2. Select **Create Instance Configuration**.
3. In the **Create Instance Configuration** dialog box, enter the following information:
   - **Name:** Enter a name for the instance configuration.
   - **Create in compartment:** Select the compartment where you want this instance configuration to be created.
   - **Compartment to create instances in**: Select the compartment where you want the instances to be created.
   - **Fault Domain**: (Optional) You can select a fault domain. By default, the system automatically selects the best fault domain for instances. If you specify a fault domain, and the requested fault domain can't accommodate the instance, instance launch fails. See more information about fault domains in Creating an Instance on page 649.
   - **Source Image:** Select an image or boot volume.
     a. Select the **Source Type:**
        - **Platform Image**: Select to create an instance based on one of the Oracle Linux and Oracle Solaris images that are available in every compartment. See Private Cloud Appliance Platform Images on page 621.
        - **Custom Image**: Selectable if a custom image was created. You might need to select the compartment where the custom image is located. See Managing Custom Images on page 622.
        - **Boot Volume**: Select to create the instance based on an existing boot volume. You might need to select the compartment where the boot volume is located.
     b. Select an image or boot volume from the list.

If you selected **Platform Image**, you see a tabular list with columns **Operating System**, **OS Version**, and **Image Build** (the date the image was built). You can use the drop-down menu arrow to the right of the OS version to select a different version. For example, for the Oracle Linux OS, you can use the drop-down menu to select 9, 8, or 7.9.

If you selected **Custom Image**, you see a tabular list with columns **Name**, **Operating System**, and **OS Version**. You can use the arrows in the column headings to sort the list. You can filter the list by using the **Operating System** drop-down menu above the list of images.

If you selected **Boot Volume**, you see a tabular list with columns **Name**, **Size (GB)**, and **Created** (the date the boot volume was created). You can use the arrows in the column headings to sort the list. In the **Boot Volume** section (after the **Shape** section), you can customize the boot volume size.

If the list is too long to fit in one view, use the arrow buttons to view another page of the list.

To use a platform image that was previously available but is no longer listed, use the Compute Web UI to create the instance and specify the OCID of the image.

- **Shape:** If you're using a platform image, select a flexible shape – with *Flex* in its name, for example VM.PCAStandard.E5.Flex –, and configure the number of **OCPUs** and **memory**.

  For the **OCPU** and **memory** values, select inside each value field to see the minimum and maximum allowed values. The OCPU and memory configuration can be changed after the instance is created.

  For a description of the supported shape, see Compute Shapes.
- **Boot Volume:** (Optional) Check the box to specify a custom boot volume size or volume performance setting.
  - **Boot volume size (GB)**: The default boot volume size for the selected image is shown. To specify a larger size, enter an integer number of gigabytes up to 16384 (16 TB) or use the increment and decrement arrows. You can't enter a value smaller than the default size.

    If you specify a custom boot volume size, you need to extend the partition to take advantage of the larger size. Oracle Linux platform images include the `oci-utils` package. Use the `oci-growfs` command from that package to extend the root partition and then grow the file system. For other OSs or for custom images, follow the instructions for that OS.
  - **Boot volume performance (VPUs)**: Use the increment and decrement arrows to toggle between balanced performance (10 VPUs/GB) and high performance (20 VPUs/GB).
- **Subnet:** Select a subnet.
  1. Select a VCN from the list. You might need to change the compartment to the compartment where the VCN is located.
  2. Select a subnet.
- **Public IP Address:** To use SSH to connect to instances created with this instance configuration, check the **Assign Public IP** box to have a public IP address assigned to the instances. This box is checked by default if you specified a public subnet. If you don't check this box, or if you clear this box, and then want to assign a public IP address later, see Assigning an Ephemeral Public IP Address to an Instance on page 530 for instructions.
- **Secondary VNICs:** (Optional) Check the **Create Additional VNIC** box to create secondary VNICs for instances created with this instance configuration. For descriptions of the requested information requested, see Creating and Attaching a Secondary VNIC on page 520.
- **Private IP Address:** (Optional) Specify an available private IP address from the subnet CIDR. By default, a private IP address is automatically assigned.

  > **Note:**
  >
  > Because the private IP address must be unique for each instance, don't specify a private IP address if you're going to use this instance configuration to create an instance pool.
- **DNS Record:** (Optional) Check the **Assign a private DNS record** box to assign a DNS record to instances created with this instance configuration.

- **Hostname:** (Optional) Enter a hostname if you're using DNS within the cloud network. The hostname must be unique across all VNICs in the subnet.

  > **Note:**
  >
  > Don't specify a host name if you are going to use this instance configuration to create an instance pool.

  By default, the instance name is used for the hostname. The hostname can also be configured in the OS after the instance is created.

  If this is a UNIX instance, see Creating a Mount Target on page 763 and Mounting File Systems on UNIX-based Instances on page 774 for more information about setting the host name correctly for mounting file systems.
- SSH Keys: To connect to the instance using SSH, provide a public SSH key.

  > **Note:**
  >
  > You can't provide this SSH key after the instance is created.
- **Network Security Group:** (Optional) By default, instances aren't attached to any NSG. Check the **Enable Network Security Group** box to add the primary VNIC for this instance to one or more NSGs.

  a. Select an NSG from the drop-down list. You might need to change the compartment to find the NSG you want.
  b. Select **Add Network Security Group** to attach to another NSG.
  c. To remove an NSG from the list, select the trash can to the right of that NSG. To remove the last NSG or all NSGs, clear the **Enable Network Security Groups** box.

  See Controlling Traffic with Network Security Groups on page 501 for information about NSGs.
- **Instance Options**: Check the box to disable Legacy Instance Metadata Service Endpoints. By default, legacy (`/v1`) Instance Metadata Service (IMDS) routes are enabled. If you have upgraded your applications to use `/v2` endpoints, check this box to disable `/v1` endpoints. For more information about the Instance Metadata Service, see Retrieving Instance Metadata from Within the Instance on page 657. For more information about upgrading your applications, see Upgrading to IMDS Version 2 Endpoints on page 657.
- **Availability configuration:** (Optional) By default, the system automatically selects the best instance availability option during a maintenance operation such as live migration. Check the **Restore instance lifecycle state after infrastructure maintenance** box to specify that running instances should be automatically restarted after a maintenance event. If this box isn't checked, the instance is recovered in the stopped state.
- **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.

4. Select **Create Instance Configuration**.

Using the OCI CLI

Use the oci compute-management instance-configuration create command and required parameters to create an instance configuration by entering configuration values.

```
oci compute-management instance-configuration create -
c <compartment_OCID> --display-name <IC_name> --instance-details
 file://<custom_config_file>.json
```

The specified compartment is where this instance configuration is created. This compartment could be different from the compartment specified in the instance details JSON file, which is where the instances will be created.

The specified display name is the name of the instance configuration. If you don't provide a value for the `--display-name` option, the default name of the instance configuration is `instanceconfiguration` ***`YYYYMMDDhhmmss`*** , where ***`YYYYMMDDhhmmss`*** is the creation date and time.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Procedure

1. Get the following information:
   - The OCID of the compartment where you want to create this instance configuration.
   - The OCID of the compartment where you want instances that use this instance configuration to be created.
   - The name of the availability domain for instances that use this instance configuration.
   - The OCID of the image or boot volume for instances that use this instance configuration.
   - The name of the shape for instances that use this instance configuration.
   - The OCID of the subnet for instances that use this instance configuration.
2. Create the configuration file that supplies input to the configuration create command.

   The configuration file is a JSON file of property/value pairs.

   - Use the following command to generate the correct syntax of the configuration file and names of properties:

     ```
     $ oci compute-management instance-configuration create --generate-param-
     json-input instance-details > instance_details.json
     ```

     You don't need all the data that's output by this command. Copy only the information you need, being careful to keep each property in its correct context.

     If you omit the fault domain specification, the system automatically selects the best fault domain. If you specify only a single fault domain, all instances are placed in only that fault domain.

     If a fault domain that you specify doesn't have enough resources, instances might fail to launch:

     - When you create a single instance (Using an Instance Configuration to Create an Instance on page 674), and you specify a fault domain in the instance configuration, only that specified fault domain is used to create the instance. Resource constraints might cause the instance launch to fail.
     - When you create instances in a pool, fault domains specified in the placement configuration override fault domains specified in the instance configuration. See Creating an Instance Pool on page 683 for more information.

     You can specify secondary VNICs and subnets. If you specify a hostname label for a secondary VNIC, the specified hostname label must be unique across all VNICs in the subnet. If you provide a value for the `hostnameLabel` property, you must also set the value of `assignPrivateDnsRecord` to `true`.

     - If the specified hostname label is already in use in the subnet, instance launch (Using an Instance Configuration to Create an Instance on page 674) will fail with the error "Hostname *`hostname`* already in use for the subnet."
     - The `hostnameLabel` property is ignored when you use the instance configuration to create a pool of instances. By default, the instance name is used for the hostname.

     If you omit the `assignPublicIp` property, a public IP address is assigned by default if you specify a public subnet. If you set this property to `false` and then decide to assign a public IP address later, see Assigning an Ephemeral Public IP Address to an Instance on page 530 for instructions.

     If users will use `ssh` to connect to the instance, specify the SSH public key as the value of the `ssh_authorized_keys` property in the `metadata` block. You can't add the SSH public key after the instance is created.

     The `displayName` property is used for the instance name when you use the `launch-compute-instance` command as described in Using an Instance Configuration to Create an Instance on page 674. If you don't provide a value for the `displayName` property, the default name of instances is `instance` ***`YYYYMMDDhhmmss`*** , where ***`YYYYMMDDhhmmss`*** is the creation date and time.

     The `displayName` property is ignored when you create instances in a pool as described in Creating an Instance Pool on page 683.

- The following command shows which properties are required to create an instance:

  ```
  $ oci compute instance launch -h
  ```

  Scroll to the Required Parameters section. Optional parameters are described below the required parameters.

The names of the properties in the configuration file are similar to, but different from, the names of the `instance launch` options. Also, some properties are organized into groups of properties, such as `createVnicDetails`, `shapeConfig`, and `sourceDetails`, as shown in the following example configuration file:

```
{
  "instanceType": "compute",
  "launchDetails": {
    "availabilityDomain": "AD-1",
    "compartmentId": "compartment_OCID",
    "createVnicDetails": {
      "assignPublicIp": true,
      "freeformTags": {
        "ConfigType": "Configuration for an XYZ instance."
      },
      "subnetId": "subnet_OCID"
    },
    "displayName": "instance_name",
    "instanceOptions": {
      "areLegacyImdsEndpointsDisabled": true
    },
    "metadata": {
      "ssh_authorized_keys": "public_SSH_key"
    },
    "shape": "shape_name",
    "shapeConfig": {
      "memoryInGBs": 512,
      "ocpus": 32
    },
    "sourceDetails": {
      "bootVolumeSizeInGBs": 100,
      "bootVolumeVpusPerGB": 20,
      "imageId": "image_OCID",
      "sourceType": "image"
    }
  }
}
```

Use `instanceOptions` if you need to disable IMDSv1 endpoints for this instance. See Retrieving Instance Metadata from Within the Instance on page 657.

Specify a flexible shape – with *Flex* in its name, for example `VM.PCAStandard.E5.Flex` –, and also specify the shape configuration. You must provide a value for `ocpus`. The `memoryInGBs` property is optional; the default value in gigabytes is 16 times the number of `ocpus`.

For information about `bootVolumeSizeInGBs`, see "Boot volume size" in Creating an Instance Configuration from an Instance on page 666 CLI tab.

For information about `bootVolumeVpusPerGB`, see "High Performance" in Creating an Instance Configuration from an Instance on page 666 CLI tab. When instances are launched, the value of `bootVolumeVpusPerGB` is `null` because this boot volume property isn't stored in the instance object after the instance is launched. To check the value, use the `get boot volume` command and see the value of `vpus-per-gb`.

To change the value of the `firmware` property, provide a value for the `launchOptions` property. The default value is BIOS. You can alternatively specify UEFI_64. Other properties in `launchOptions` can't be changed.

```
"launchOptions": {
  "bootVolumeType": "PARAVIRTUALIZED",
  "firmware": "UEFI_64",
  "isConsistentVolumeNamingEnabled": false,
  "isPvEncryptionInTransitEnabled": false,
  "networkType": "PARAVIRTUALIZED",
  "remoteDataVolumeType": "PARAVIRTUALIZED"
}
```

3. Run the instance configuration create command.

   Syntax:

   ```
   oci compute-management instance-configuration create -
   c <compartment_OCID> --display-name <IC_name> --instance-details
    file://<custom_config_file>.json
   ```

   The specified compartment is where this instance configuration is created. This compartment could be different from the compartment specified in the instance details JSON file, which is where the instances will be created.

   The specified display name is the name of the instance configuration. If you don't provide a value for the `--display-name` option, the default name of the instance configuration is `instanceconfiguration` ***`YYYYMMDDhhmmss`*** , where ***`YYYYMMDDhhmmss`*** is the creation date and time.

   The output of this command is the same as the output of the `instance-configuration get` command.

Using the API

Use the CreateInstanceConfiguration operation to create an instance configuration entering configuration values.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Updating an Instance Configuration

On Private Cloud Appliance, you can change the instance configuration name, compartment, and tags.

> **Note:**
>
> To change other configuration details such as the subnet, or image, create a new instance configuration. See Creating an Instance Configuration on page 666.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instance Configurations**.
2. At the top of the page, select the compartment that contains the instance configuration that you want to update.
3. For the instance configuration that you want to update, select the Actions menu (⋮), and select **Edit**.
4. In the **Update Instance Configuration** dialog box, make the changes.
5. Select **Update Instance Configuration**.

Using the OCI CLI

Use the oci compute-management instance-configuration update command and required parameters to change the instance configuration name or tags.

```
oci compute-management instance-configuration update --instance-
configuration-id <compartment_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateInstanceConfiguration operation to change the instance configuration name or tags.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Moving an Instance Configuration to a Different Compartment

On Private Cloud Appliance, you can move an instance configuration to a different compartment within the same tenancy.

When you move an instance configuration to a different compartment, instances and instance pools created by using this instance configuration aren't moved.

New instances and instance pools that are created using this instance configuration are created in the compartment specified in the instance configuration, not in the compartment to which the instance configuration has been moved.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci compute-management instance-configuration change-compartment command and required parameters move an instance configuration to a different compartment.

```
oci compute-management instance-configuration change-compartment --
compartment-id destination_compartment_OCID --instance-configuration-
id instance_configuration_OCID
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

This task is not available in the API.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting an Instance Configuration

On Private Cloud Appliance, you can delete an instance configuration as long as the instance configuration isn't being used by an instance pools.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instance Configurations**.
2. At the top of the page, select the compartment that contains the instance configuration.
3. Select the name of the instance configuration that you want to delete.
4. On the instance configuration details page, select **Delete**.
5. Select **Confirm**.

Using the OCI CLI

Use the oci compute-management instance-configuration delete command and required parameters to delete an instance configuration.

```
oci compute-management instance-configuration delete [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteInstanceConfiguration operation to delete an instance configuration.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Using an Instance Configuration to Create an Instance

On Private Cloud Appliance, you can use an instance configuration to launch a compute instance.

This method of launching a compute instance is an alternative to the method described in Creating an Instance on page 649.

The name of the instance is determined as follows:

- If the instance configuration specifies a value for the `displayName` property, the name of the instance will be `displayName`. If you use the same instance configuration with multiple `launch-compute-instance` commands, all instances will have the same name. Instance names aren't required to be unique.
- If the instance configuration doesn't specify a value for the `displayName` property, the default name of the instance will be `instance`***`YYYYMMDDhhmmss`***, where ***`YYYYMMDDhhmmss`*** is the creation date and time.

Avoid entering confidential information in names.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci compute-management instance-configuration launch-compute-instance command and required parameters to change the instance configuration name or tags.

```
oci compute-management instance-configuration launch-compute-instance
 [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the GetInstanceConfiguration operation to change the instance configuration name or tags.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Connecting to a Compute Instance

On Private Cloud Appliance, you can connect to a running instance by using a Secure Shell (SSH) or Remote Desktop connection the same way you connect to instances in Oracle Cloud Console (OCI).

For more information, see these sections:

- To create an instance console connection, see Creating an Instance Console Connection on page 675 on Private Cloud Appliance.
- For connection prerequisites, connection instructions, and troubleshooting suggestions, see Connecting to an Instance.
- For information about SSH key pairs, see Managing Key Pairs on Linux Instances.

Creating an Instance Console Connection

On Private Cloud Appliance, before you can connect to an instance VNC console or serial console, you need to create an instance console connection.

> **Note:**
>
> Instance console connections are limited to one client at a time. If the client attempts to connect but fails to connect within five minutes, the connection is closed and a different client can connect. During the five-minute timeout, any attempt to connect a different client fails.

The instance console connection resource provides the command that you need to create the secure tunnel. The command is a little different depending on whether your local system is UNIX or Windows and whether you want to connect to the VNC console or the serial console on the instance.

Console Connection Prerequisites

Ensure that you have the following resources on the system that you plan to use to connect to the instance console.

- The console connection needs access to port 1443. Previously, port 443 was used for VM console access. For more information, see Port Matrix
- SSH key pair

  If you don't already have an SSH key pair, you can use the `ssh-keygen` utility on UNIX systems or PuTTY `puttygen.exe` on Windows systems. Specify a key size of 2048 bits (this value should be the default). Give the key a name. You don't need to provide a passphrase; using a passphrase makes it more difficult to automate connecting.
- Command-line shell and SSH client

  On Windows systems, use one of the following:

  - Windows PowerShell

    If you use PowerShell to connect to the VNC server on the instance, `plink.exe` is required. `plink.exe` is the command link connection tool included with PuTTY. You can install PuTTY or install `plink.exe` separately.
  - Git for Windows

    Git for Windows includes OpenSSH.
  - Windows Subsystem for Linux (WSL)

    WSL includes OpenSSH.
- VNC viewer to connect to the VNC console
- Ensure that you belong to a group that has the following permissions.

```
Allow group group_name to manage instance-console-connection in tenancy
Allow group group_name to read instance in tenancy
```

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. At the top of the page, select the compartment that contains the instance.
3. Select the name of the instance where you want to create a console connection.
4. On the instance details page, under **Resources**, select **Console Connection**.
5. If a console connection doesn't already exist, select **Create Console Connection**.
6. Provide the public key portion of your SSH key.

   In the **Create Console Connection** dialog box, do one of the following to enter your public SSH key:

   - Select the key file(s).
     - Select inside the Drag and Drop box to open a file browser and select the file.

    - Drag the file from your file browser listing and drop the file on the Drag and Drop box.
  - Paste the public key(s). Copy your public SSH key text, and paste the text into the field.
7. Select **Create Console Connection** in the dialog.

   When the console connection is created and is available, the state changes to **Active**.

What's Next

Continue to Connecting to the VNC Console or Making a Local Connection to the Serial Console.

Using the OCI CLI

Use the oci compute instance-console-connection create command and required parameters to create a new serial console connection to the specified instance. Once the serial console connection is created and is available, you connect to the serial console using an SSH client.

```
oci compute instance-console-connection create --instance-id <instance_OCID>
 --ssh-public-key-file public_SSH_key_path [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Procedure

1. Get the following information:
   - The OCID of the instance where you want to create the console connection: `oci compute instance list`
   - Your SSH public key file.
2. Determine whether a console connection already exists for this instance.

   ```
   oci compute instance-console-connection list -c compartment_OCID --
   instance-id instance_OCID
   ```
3. Run the create console connection command.

   ```
   oci compute instance-console-connection create --instance-id
    ocid1.instance.unique_ID --ssh-public-key-file public_SSH_key_path
   {
     "data": {
       "compartment-id": "ocid1.compartment.unique_ID",
       "connection-string": "ssh -i private_SSH_key_path -t -p
    443 user_name@proxy_host tty@instance_OCID",
       "defined-tags": {},
       "fingerprint": "SHA256:unique_ID",
       "freeform-tags": {},
       "id": "ocid1.instanceconnectionconsole.unique_ID",
       "instance-id": "ocid1.instance.unique_ID",
       "lifecycle-state": "ACTIVE",
       "service-host-key-fingerprint": null,
       "vnc-connection-string": "ssh -i public_SSH_key_path -p 443
    -L local_vnc_port:localhost:remote_vnc_port user_name@proxy_host
    vnc@ocid1.instance.unique_ID"
     },
     "etag": "afc7eb68-5f1a-40cc-8dc3-8a1cae237230"
   }
   ```

   The value of `connection-string` is the SSH connection string for the instance serial console connection. The value of `vnc-connection-string` is the SSH connection string for the instance VNC console connection.

4. When you're finished using this instance console connection, use the following command to delete the connection.

```
$ oci compute instance-console-connection delete --instance-console-
connection-id instance_console_connection_OCID
```

What's Next

Continue to Connecting to the VNC Console or Making a Local Connection to the Serial Console.

Using the API

This task can't be performed using the API.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Backing Up and Restoring an Instance

On Private Cloud Appliance, supports backing up and restoring instances. The instance backup is created in an Object Storage bucket. From there, you can copy it to another server in your data center for safekeeping. When needed, you can import the backup into any Private Cloud Appliance Object Storage bucket, and use it to create instances.

Use Cases

- Back up instances and any attached block volumes.
- Store the backups on another server for safekeeping.
- Restore a faulty instance and any attached block volumes.
- Use the backup to create matching instances.
- Use the backup and restore feature to migrate instances to another tenancy, or to another Private Cloud Appliance.

The restored instance has some of the same characteristics as the source instance. For example:

- The type and version of the OS.
- The OS configuration matches the OS configuration of the source instance. This includes things like OS user accounts, installed applications, and so on.
- The type of storage, high-performance or balanced-performance, matches the source instance.
- All the software on the block volumes is available after you attached the block volumes to the restored instance.

Some aspects of the restored instance differ from the source instance, such as:

- The restored instance and associated components, like boot and block volumes have unique OCIDs that don't match the source instance.
- The source instance user account SSH keys aren't included in the restored instance.
- While creating the restored instance, you can configure the instance in a different compartment, with a different name, shape, subnet, and all the other attributes that you configure during the launch.

Task Map - Backing Up an Instance

No.	Task	Links
1.	Ensure that you have an Object Storage bucket in the same tenancy where the instance is located.	• Listing Buckets on page 795 • Creating a Bucket on page 795
2.	Create the instance backup.	Creating an Instance Backup on page 678
3.	Transfer the backup object from Object Storage to another system in your data center.	Transferring an Instance Backup to Another System on page 680

Task Map – Restoring an Instance From a Backup

The following tasks assume that you are restoring an instance from a backup that's on another system in your data center. If the backup is already on the Private Cloud Appliance where you plan to restore the instance, start with task number 3.

No.	Task	Links
1.	Ensure that you have an Object Storage bucket in the same tenancy where you plan to restore the instance.	• Listing Buckets on page 795 • Creating a Bucket on page 795
2.	Upload the backup to the bucket.	Transferring an Instance Backup from Another System to Private Cloud Appliance on page 680
3.	Identify the backup OCID.	Listing Instance Backups on page 679
4.	Import the backup from the bucket into the Private Cloud Appliance.	Importing an Instance Backup on page 681
5.	Finish restoring the instance by creating an instance using the imported instance backup.	Finishing the Instance Restore on page 681

Creating an Instance Backup

On Private Cloud Appliance, you can use the Compute Web UI and API to back up an instance.

The instance can be running or stopped. The boot volume and any block volumes must be attached.

> **Caution:**
>
> During the backup process, don't perform any volume attach or detach operations on the instance.

The duration of the backup varies based on the amount of data on the instance boot and block volumes. A small instance that only has a 50 GB boot volume takes only a few minutes to complete. If the instance volumes are as large as 32 TB, the backup can take up to 6 hours to complete.

Prerequisites

- You must have an Object Storage bucket in the tenancy where the instance is located. See Creating a Bucket on page 795.
- Quiesce instance activities such as running applications so that the backup is created at a known state.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. At the top of the page, select the compartment that contains the instance.
3. Perform **one** of the following actions:
   - Select the instance Actions menu (⋮), and select **Export**.
   - Select the name of the instance you plan to back up. Then select **Controls** (upper right) and select **Export**.
4. In the dialog box, select these items:
   - If needed, change the compartment to the compartment where the bucket is located by selecting **(Change)**.
   - Select the bucket.
   - Select **Create Export**.
5. To see the status of an instance backup, under **Resources**, select **Instance Exports**.

Using the OCI CLI

This task can't be performed using the OCI CLI.

Using the API

The `export` API creates an instance backup to an Object Storage bucket in the specified compartment.

API Endpoint

```
https://<mgmt_node_VIP>:30003/20160918/instances/<instance_OCID>/actions/
export
```

where:

- <*mgmt_node_VIP*> is the management node VIP host name or IP address.
- <*instance_OCID*> is the instance ID.

Pass these key-value pairs:

```
{
 "bucketName": "<bucket_name>",
 "destinationType": "objectStorageTuple",
 "namespaceName": "<namespace_name>",
 "compartmentId": "<bucket_compartment_OCID>"
}
```

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Listing Instance Backups

On Private Cloud Appliance, you can list instance backups using the Compute Web UI and API.

When you list instance backups, you are able to identify these components of the backup:

- Backup OCID
- Boot volume OCID
- Instance OCID
- Image OCID

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select any of the following links.
   - **Instance Exports**: Displays the list of instance backups.
   - **Instance Imports**: Displays the list of imported instance backups that can be used to create new instances.

Using the OCI CLI

This task can't be performed using the OCI CLI.

Using the API

There are two API endpoints for viewing instance backups:

- *List All Instance Backups in a Bucket.*

  API endpoint

  ```
  https://<mgmt_node_VIP>:30003/20160918/instances/instanceBackups?
  compartmentId=<bucket_compartment_OCID>
  ```

  where:

  - <*mgmt_node_VIP*> is the management node VIP host name or IP address.

  - <*bucket_compartment_OCID*> is the compartment ID where the Object Storage bucket is located.
- **Get Instance Backup Details**

  API endpoint

  ```
  https://<mgmt_node_VIP>:30003/20160918/
  instanceBackups/<instance_backup_id>
  ```

  where:

  - <*mgmt_node_VIP*> is the management node VIP host name or IP address.
  - <*instance_backup_id*> is the backup ID, which you can get from the backup export output or from listing backups with the API.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Transferring an Instance Backup

On Private Cloud Appliance, you can transfer instance backups to and from another system in your data center.

Transferring an Instance Backup to Another System

You can use this procedure to transfer the instance backup to another system in your data center for safekeeping.

Instance backups are large files. Ensure that you have enough space on your system to store the backup. You can use the `oci os object list` command to display the size of the instance backup. See Viewing Objects in a Bucket on page 799.

Using the OCI CLI

1. Gather the information that you need to run the command.
   - Namespace name (see Obtaining the Object Storage Namespace on page 794)
   - Bucket name (`oci os bucket list`), see Listing Buckets on page 795
   - Object name (`oci os object list`), see Viewing Objects in a Bucket on page 799
2. Run the `object get` command.

   Syntax (entered on a single line):

   ```
   oci os object get
   --namespace-name <object_storage_namespace>
   --bucket-name <bucket_name>
   --name <object_name>
   --file <file_location>
   ```

   <*file_location*> is the destination path for the file being downloaded, such as `C:\workspace\backups\ocid1.instance.uniqueID` or `/home/downloads/backups/ocid1.instance.uniqueID`.

   Example:

   ```
   oci os object get  \
   --namespace-name mytenant  \
   --bucket-name my-backup-bucket  \
   --name ocid1.instance.........uniqueID  \
   --file /home/downloads/backups/ocid1.instance.........uniqueID
   Downloading object  [########################-----------]   68%  00:00:05
   ```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Transferring an Instance Backup from Another System to Private Cloud Appliance

Using the OCI CLI

1. Gather the information that you need to run the command.

- Namespace name. See Obtaining the Object Storage Namespace on page 794.
- Bucket name (`oci os bucket list`). See Listing Buckets on page 795.

2. Upload the instance backup to an Object Storage bucket in the target Private Cloud Appliance.

   Use the `oci os object put` command.

   Syntax (entered on a single line):

   ```
   oci os object put  \
   --namespace-name <namespace_name>  \
   --bucket-name <bucket_name>  \
   --file <instance_backup_pathname>
   ```

   The value of <*instance_backup_pathname*> is the path name of the object being uploaded, such as `C:\workspace\backups\ocid1.instance.uniqueID` or `/home/downloads/backups/ocid1.instance.uniqueID`.

   Example:

   ```
   oci os object put  \
   --namespace-name mytenant  \
   --bucket-name target-bucket  \
   --file ./ocid1.instance.….….….uniqueID

   Upload ID: f000bf64-9a96-4008-b1cc-f6b2595b04b1
   Split file into 35 parts for upload.
   Uploading object  [####################################]  100%
   {
     "etag": "ef7bdd67a72536e29da97f3414f4118e",
     "last-modified": "2022-07-06T17:50:00",
     "opc-multipart-md5": "htOEPyjWDFA4Bs2urJJPRQ==-35"
   ```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Restoring an Instance from an Instance Backup

On Private Cloud Appliance, you restore an instance by importing the instance backup from an Object Storage bucket. Next, create the instance using the boot volume from the backup as the image source. Then attach any block volumes that were included in the backup.

Importing an Instance Backup

On Private Cloud Appliance, importing an instance backup copies the backup from an Object Storage backup to a location that's internal.

For a particular instance backup, you can only have one imported copy. If you need to import the same instance backup again, you must first delete the original instance backup. See Deleting an Instance Backup on page 682.

Prerequisites

- You must have an instance backup in an Object Storage bucket.

  If needed, transfer the backup to Private Cloud Appliance. See Transferring an Instance Backup from Another System to Private Cloud Appliance on page 680.

Finishing the Instance Restore

On Private Cloud Appliance, you can create new instances from instance backups.

Perform the following steps to create as many instances as you like from the instance backup.

1. Create an instance by following the instructions in Creating an Instance on page 649). While doing so, perform these actions:

   - For the source image, specify a backup boot volume instead of an image. Then select an imported instance.
   - If the instance requires access using SSH, ensure that you include an SSH public key.

2. (Optional) Attach any block volumes that were included in the instance backup.

   See Attaching a Volume on page 719.

Deleting an Instance Backup

On Private Cloud Appliance, you can use the `DELETE` API to delete exported and imported instance backups.

The API sets the exported backup lifecycle state to **TERMINATED** and performs one of these actions based on the type of backup:

- **Imported instance backups:** Deletes the instance backup.
- **Exported instance backups:** Deletes the instance backup (source for importing) from object storage bucket

Terminated instance backups are eventually cleaned up by a background task.

Using the API

`DELETE` API – Deletes instance backups.

API endpoint

```
https://<mgmt_node_VIP>:30003/20160918/
instancesBackups<instance_backup_OCID>
```

where:

- <*mgmt_node_VIP*> is the management node VIP host name or IP address.
- <*instance_backup_OCID*> is the instance backup ID.

Instance Pools

On Private Cloud Appliance, instance pools simplify the management of compute instances. An instance pool defines a set of compute instances that's managed as a group. Managing instances as a group enables you to efficiently provision instances and manage the state of instances.

Creating an instance pool requires an instance configuration and a placement configuration. If you update the instance configuration or placement configuration, existing instances aren't affected. New instances that are added to the pool use the new instance and placement configurations.

> **Caution:**
>
> When you delete an instance pool, the resources that were created by the pool are permanently deleted, including associated instances, attached boot volumes, and block volumes.

You can define an autoscaling configuration and policies that scale the instance pool in (remove instances) or out (add instances) or stop, start, or reboot the instances according to a schedule. Autoscaling a pool enables you to use resources even more effectively by stopping or removing instances when demand is lower and starting or adding instances when demand is higher. You can use instance pool autoscaling along with network load balancing by defining the backend set of servers to be an autoscaled instance pool.

When working with compute instance configurations and compute instance pools, keep the following points in mind:

- You can't delete a compute instance configuration if it's associated with at least one compute instance pool.
- You can use the same compute instance configuration for multiple compute instance pools. However, a compute instance pool can have only one compute instance configuration associated with it.
- If you modify the compute instance configuration for a compute instance pool, existing compute instances that are part of that pool don't change. Any new compute instances that are created after you modify the compute instance configuration will use the new compute instance configuration. New compute instances will not be created unless you have increased the size of the compute instance pool or terminate existing compute instances.

- If you decrease the size of a compute instance pool, the oldest compute instances are deleted first.

Creating an Instance Pool

On Private Cloud Appliance, you can create an instance pool of instances that are within the same region.

Performing operations such as reset or delete on the pool object performs that operation on all instances that are members of the pool. Performing these operations on an individual instance that's a member of the pool doesn't affect any other member instances.

Creating an instance pool requires an instance configuration and a placement configuration. Instances that are added to the pool in a pool update can be created with different instance and placement configurations.

For instances in a pool, the value of the `displayName` property in the instance configuration is ignored. Instances in a pool are named `inst-`***`aaaaa`***`-`***`pool_name`***, where ***`aaaaa`*** is five random alphanumeric characters.

Placement Configuration

In addition to an instance configuration, pool creation requires a placement configuration. Values specified in a placement configuration override values specified in the instance configuration.

A placement configuration can specify fault domains, primary subnet, and secondary VNIC subnets.

Fault Domains

If you don't specify a fault domain in either the instance configuration or the placement configuration, the system automatically selects the best fault domains for the pool instances. If you specify only a single fault domain, all instances will be placed in only that fault domain. If you specify more than one fault domain, pool instances are placed in those fault domains evenly, providing better High Availability for the pool. If one fault domain can't accommodate additional instances, instance creation stops. The system will not place more instances in one fault domain than in another fault domain.

If some instances can't launch because of resource constraints, those instances remain in the Provisioning state and the pool remains in the Scaling state. After `size` instances are launched, the pool can transition to the Running state. While the pool is in the Scaling state, pool instances that are in the Running state are available to use.

The following are examples of actions you can take if a pool instance fails to launch because of resource constraints:

- Update the pool and reduce the "Number of instances" or `size` value.
- Update the pool and change the Fault Domain specification in the Compute Web UI or in a new instance or placement configuration.
- Update the pool to specify a new instance configuration that creates instances that require fewer resources.
- Stop an instance that isn't a member of a pool in the same fault domain where the pool instance is failing to launch because of resource constraints.
- Delete an instance that isn't a member of a pool in the same fault domain where the pool instance is failing to launch because of resource constraints.

Prerequisite

Before you can create an instance pool, you need an instance configuration. An instance configuration is a template that defines the settings to use when creating instances. See Working with Instance Configurations on page 666.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instance Configurations**.
2. At the top of the page, select the compartment that contains the instance configuration that you want to use to create the pool.
3. Select the instance configuration that you want to use for the instances in this pool.
4. Under **Resources**, select **Attached Instance Pools**.
5. At the top of the page, select other compartments to list pools in other compartments.

Select **Create Instance Pool**.

6. In the **Attach Instance Pool to *`instance_configuration_name`*** dialog box, enter the following information:

   - **Name**: Enter a name for the instance pool. The name doesn't need to be unique. This name is used in the names of the created instances. If you don't provide a name for the pool, the default name of the instance pool is `instancepool`***`YYYYMMDDhhmmss`***, where ***`YYYYMMDDhhmmss`*** is the creation date and time.
   - **Create in Compartment**: Select a compartment for this instance pool definition. Note that the instances in the pool will be created in the compartment that's specified in the instance configuration.
   - **Number of instances**: Specify the number of instances to create in this instance pool.
   - **Pool Placement**: Select the Fault Domains, VCN, and Subnet for instances in this instance pool. You can select a different compartment from which to choose the VCN and Subnet. See the descriptions of Placement Configuration and Fault Domains at the beginning of this section.
   - **Load Balancers**: Select the **Attach Load Balancers** box to specify load balancing for this pool. For information about load balancing, see Load Balancer as a Service on page 550. Provide the following information:

     - Select the load balancer to attach to this pool.
     - Select the backend set to which to add these pool instances.
     - Enter the port number on the instances to which the load balancer must direct traffic.
     - Select the VNIC to use when adding the instance to the backend set. The private IP address is used.

     To attach another load balancer, select **Add Load Balancer**. To attach a load balancer after the instance pool is created, see Managing Instance Pool Load Balancer Attachments on page 698.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.

7. Select **Create Instance Pool**.

   The details page of the new pool is displayed. The requested instances are listed in the **Attached Instances** table in the **Resources** section as they're created. The new instances are named `inst-`***`aaaaa`***`-`***`pool_name`***, where ***`aaaaa`*** is five random alphanumeric characters. If you change the name of the pool and then add new instances to the pool, the new instances have the new name.

   Select **Work Requests** in the **Resources** box to check the status of the instance pool create.

Using the Compute Web UI

Use the oci compute-management instance-pool create command and required parameters to create an instance pool.

```
oci compute-management instance-pool create --compartment-
id <compartment_OCID> --instance-configuration-
id <instance-configuration_OCID> --placement-
configurations <placement_configurations.json> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Procedure

1. Get the following information:

   - The OCID of the compartment where you want to create the instance pool definition: `oci iam compartment list`

     Note that the instances in the pool are created in the compartment specified in the instance configuration.
   - The OCID of the instance configuration that you want to use: `oci compute-management instance-configuration list`
   - The size of the instance pool. This is the number of compute instances in the instance pool.
   - If you want load balancing for this pool, get the following information:

- OCID of the load balancer to attach to this pool and name of the backend set to which to add these pool instances: `oci lb load-balancer list`
- Port value to use when creating the backend set.
- VNIC to associate with the load balancer. The value can be `PrimaryVnic` or the display name of one of the secondary VNICs on the instance configuration associated with the instance pool.

2. Construct an argument for the `--placement-configurations` option.

   See the descriptions of Placement Configuration and Fault Domains at the beginning of this section.

   Use the following command to show the content of the placement configurations argument:

   ```
   $ oci compute-management instance-pool create --generate-param-json-input
    placement-configurations
   ```

3. If you want load balancing for this pool, construct an argument for the `--load-balancers` option.

   Use the following command to show the content of the load balancers argument:

   ```
   $ oci compute-management instance-pool create --generate-param-json-input
    load-balancers
   ```

   To attach a load balancer after the instance pool is created, see Managing Instance Pool Load Balancer Attachments on page 698.

4. Run the instance pool create command.

   Syntax:

   ```
   oci compute-management instance-pool create -c compartment_OCID
    --instance-configuration-id instance_configuration_OCID --
   placement-configurations file://placement_configuration.json --
   size number_of_instances
   ```

   Example:

   ```
   $ oci compute-management instance-pool create --compartment-id
    ocid1.compartment.unique_ID --display-name support-pool --instance-
   configuration-id ocid1.instanceConfiguration.unique_ID --placement-
   configurations file://./placement_configurations.json --load-balancers
    file://./load_balancers.json --size 10
   ```

   The value of the `--display-name` option is the name of the pool. The pool name isn't required to be unique. If you don't provide a value for the `--display-name` option, the default name of the instance pool is `instancepool`***`YYYYMMDDhhmmss`***, where ***`YYYYMMDDhhmmss`*** is the creation date and time.

   The pool name is used in the names of the instances. Instances in a pool are named `inst-`***`aaaaa-pool_name`***, where ***`aaaaa`*** is five random alphanumeric characters. If you change the name of the pool and then add new instances to the pool, the new instances will have the new name.

   The output of this command is the same as the output of the `instance-pool get` command. The list of instances in the pool isn't shown.

   To list the instances that belong to this pool, use the following command:

   ```
   $ oci compute-management instance-pool list-instances -
   c <compartment_OCID> --instance-pool-id <instance_pool_OCID>
   ```

   The output for each instance is abbreviated compared with the output from the `instance get` command.

The following command shows the same abbreviated output for only the specified instance:

```
$ oci compute-management instance-pool-instance get --instance-id
 ocid1.instance.unique_ID \
--instance-pool-id ocid1.instancePool.unique_ID
```

Using the API

Use the CreateInstancePool operation to create an instance pool.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Using Schedule-Based Autoscaling

On Private Cloud Appliance, autoscaling instance pools enables you to effectively manage instance resource use.

An instance pool can have an autoscaling configuration and policies that scale the instance pool in the following ways according to a schedule:

- Scale out: add instances
- Scale in: remove instances
- Lifecycle or power action: stop, start, or reboot instances

When an instance pool scales out or scales in, instances are created or terminated as described in Updating an Instance Pool on page 695.

Policies define the schedule for autoscaling and the specific actions to take. An autoscaling configuration can have up to 50 schedule-based autoscaling policies, each with a different schedule and target pool size or lifecycle action. An instance pool can have only one autoscaling configuration.

If you manually change the pool size or lifecycle state as described in Updating an Instance Pool on page 695 and Stopping and Starting Instances in an Instance Pool on page 699, autoscaling resets the pool size or lifecycle state to the value that is set in the policy the next time the scheduled autoscaling policy runs.

> **Note:**
>
> To use autoscaling, ensure that you have installed OCI CLI 3.15.1 or newer and Oracle Cloud Infrastructure Python SDK 2.80.0 or newer.

Multiple Schedule Management

On Private Cloud Appliance, when you create and enable an autoscaling configuration, the Autoscaling service evaluates the schedule rules in the policies in the configuration.

If multiple policies in the same configuration run at the same time, only one lifecycle state policy and one pool size policy run. The lifecycle state policy runs first.

If multiple lifecycle state policies in the same configuration run at the same time, the policy with the highest priority action runs. The following list shows actions in priority order from highest to lowest priority:

1. Force Reboot
2. Reboot
3. Start
4. Force Stop

If multiple pool size policies in the same configuration run at the same time, the policy that specifies the largest pool size runs.

Creating an Autoscaling Configuration

On Private Cloud Appliance, an autoscaling configuration contains policies that schedule adding or removing instances in a specified pool, or stopping, starting, or rebooting all the instances in the pool.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Autoscaling Configurations**.
2. Select **Create Autoscaling Configuration**.
3. In the **Create Autoscaling Configuration** dialog box, enter the following information:
   - **Name**: Enter a name for the autoscaling configuration.
   - **Create in Compartment**: Select the compartment where you want to create the autoscaling configuration.
   - **Instance Pool**: Select the instance pool that you want to scale with this autoscaling configuration.
   - **Autoscaling Policies**: For each policy, provide the following information:
     - **Action To Perform**: Select either Change Lifecycle State or Scale Pool Size.
     - **Policy Name**: Enter a name for the policy.
     - **Lifecycle Action**: If you selected Change Lifecycle State for Action To Perform, then select one of the following states to which to transition all instances of the pool when this policy is executed: Start, Stop, Soft Reset, Reset.
     - **Instance Pool Limit**: If you selected Scale Pool Size for Action To Perform, then enter a value for the pool size.
     - **Enable Schedule**: By default, the Schedule Enabled box is selected to enable the policy to execute at the next scheduled time. Clear the box to disable this policy.
     - **Policy Schedule**: Enter values for Minute, Hour, Day Of Month, Month, Day Of Week, and Year. Provide all schedule times in UTC. For more information, see Creating a Schedule-Based Autoscaling Policy on page 689.

     To add another policy, select **Add Policy**. You can also add policies after the autoscaling configuration is created, as described in Creating a Schedule-Based Autoscaling Policy on page 689.

     To delete a policy, select the trash can icon for that policy.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
   - Select **Submit**.

     The details page for the new autoscaling configuration is displayed.

     On the details page, ignore the Cooldown Period value. Cooldown period doesn't apply to schedule-based autoscaling configurations.

     The new autoscaling configuration is enabled by default. To disable the configuration, see Updating an Autoscaling Configuration on page 693.

Using the Compute Web UI

Use the oci autoscaling configuration create command and required parameters to create an autoscaling configuration.

```
oci autoscaling configuration create --compartment-id compartment_OCID --
from-json file://input_file.json [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Procedure

1. Get the following information:
   - The OCID of the compartment where you want to create this autoscaling configuration: `oci iam compartment list`
   - The OCID of the instance pool that will be managed by this autoscaling configuration: `oci compute-management instance-pool list`
2. Construct a file that contains all the input for the command.

Use the following command to show the content and format of the command input:

```
$ oci autoscaling configuration create \
--generate-full-command-json-input > autoscalingCfgCreate.json
```

The `resource` property is required and is the OCID of the instance pool that will be managed by this autoscaling configuration. The `type` of this resource must be `instancePool`.

At least one policy is required to create an autoscaling configuration. To add policies after the autoscaling configuration is created, see Creating a Schedule-Based Autoscaling Policy on page 689.

The optional display name is 1-255 characters, doesn't need to be unique, and can be updated. If you don't provide a value for `-displayName`, the default name of the autoscaling configuration is `autoscalingConfiguration`***`YYYYMMDDhhmmss`***, where `instanceconfiguration`***`YYYYMMDDhhmmss`*** is the creation date and time.

The autoscaling configuration is enabled by default. To disable the configuration, set `isEnabled` to `false`.

> **Note:**
>
> Don't specify values for `coolDownInSeconds` or `capacity` `min` or `max`. These properties do not apply to schedule-based autoscaling configurations.

The default values for `cool-down-in-seconds` and `capacity` `min` and `max` appear in the created autoscaling configuration but aren't used for schedule-based autoscaling.

The following is an example autoscaling configuration create input file with one policy:

```
{
  "compartmentId": "ocid1.compartment.unique_ID",
  "displayName": "salesPoolCfg",
  "policies":
      {
        "displayName": "reboot policy",
        "executionSchedule":
          {
            "expression": "0 0 2 ? * 1#1 *",
            "timezone": "UTC",
            "type": "cron"
          },
        "policyType": "scheduled",
        "resourceAction": {
          "actionType": "power",
          "action": "SOFTRESET"
        }
      },
  "resource":
    {
      "id": "ocid1.instancePool.unique_ID",
      "type": "instancePool"
    }
}
```

3. Run the command to create the autoscaling configuration.

   Syntax:

   ```
   oci autoscaling configuration create --compartment-id compartment_OCID \
   --from-json file://input_file.json
   ```

Example:

```
$ oci autoscaling configuration create --c ocid1.compartment.unique_ID \
--from-json file://./salesPoolCfg.json
{
  "data": {
    "compartment-id": "ocid1.compartment.unique_ID",
    "cool-down-in-seconds": 300,
    "defined-tags": {},
    "display-name": "salesPoolCfg",
    "freeform-tags": {},
    "id": "ocid1.autoScalingConfiguration.unique_ID",
    "is-enabled": true,
    "max-resource-count": null,
    "min-resource-count": null,
    "policies":
      {
        "capacity": null,
        "displayName": "reboot policy",
        "executionSchedule":
          {
            "expression": "0 0 2 ? * 1#1 *",
            "timezone": "UTC",
            "type": "cron"
          },
        },
        "id": "unique_ID",
        "is-enabled": true,
        "policy-type": "scheduled",
        "resourceAction": {
          "actionType": "power",
          "action": "SOFTRESET"
        },
        "time-created": "2023-01-25T21:28:56.131801+00:00"
      },
    "resource": {
      "id": "ocid1.instancePool.unique_ID",
      "type": "instancePool"
    },
    "time-created": "2023-01-25T21:28:56.140747+00:00"
  },
  "etag": "7c70532a-1d41-4861-a40f-bf840136a9c5"
}
```

Use the `work-requests work-request get` command to check the status of the autoscaling configuration creation.

Using the API

Use the AutoScalingConfiguration operation to create an autoscaling configuration.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Creating a Schedule-Based Autoscaling Policy

On Private Cloud Appliance, you can create schedule-based autoscaling policies.

An autoscaling policy is part of an autoscaling configuration. Each policy of a schedule-based autoscaling configuration has a schedule and either a target pool size or a lifecycle action.

The procedures in this section describe how to create policies separate from creating the autoscaling configuration.

Designing Policies

This section provides some tips for designing and troubleshooting policies.

Create two separate policies to scale a pool in and out or to change the state of the pool between stopped and running.

- Scale example: One policy specifies a larger size for the pool at the beginning of a high demand period, and a second policy specifies a smaller pool size at the end of the high demand period.
- State example: One policy stops all instances in the pool at the beginning of a regular compute node maintenance period, and a second policy starts the pool at the end of the maintenance period.

Design the policy schedule as follows:

- Use `cron` expressions. Autoscaling uses a cron implementation similar to the Quartz cron implementation. All fields require a value. If fields conflict, such as day of month and day of week, use a specific value for one and a question mark for the other.
- Provide all schedule times in UTC.
- Use an online `cron` expression generator such as Quartz cron expression generator to verify your schedule expressions.
- Ensure that policy schedules don't conflict. See the descriptions in Multiple Schedule Management on page 686 of which policies run when schedules conflict.

Take the following steps if a policy fails to run, or appears to fail to run:

- Check that the autoscaling configuration and autoscaling policy are both enabled.
- Check the schedule expression. Is the policy set to run when you meant for it to run? Remember, all expression times must be provided in UTC.
- Was the policy set to start instances that were already running, or stop instances that were already stopped?

  In addition to a policy conflict, a power action might have been performed on the pool separate from any autoscaling policy. That separate power action could prevent the policy action from succeeding. The policy power action is not retried.
- Was the policy set to scale out, but not enough resources were available?

  The scale policy sets the pool size, and the pool continues to try to reach that size as resources become available.
- Is the operation specified by the policy still executing or waiting to run?
  - Check whether the pool is in the Scaling, Starting, Stopping, or Rebooting state, which indicates that the policy operation is still running.
  - If a state change operation tries to run while a state change operation is already running on the same pool, the second operation fails to run.
  - A limited number of pools can be changing state concurrently. If too many other pools are already changing state, then the pool must wait to begin changing state. The time to change state is longer when more instances are involved because instances are started, stopped, or rebooted serially.
  - A limited number of pools can be changing size concurrently. If other pools are already changing size, your pool might need to wait to begin scaling. The time to scale is longer when more instances are involved because instances are deleted or created serially. Delete and creating instances are both background operations and take some time to begin after the pool size has been updated by the policy.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Autoscaling Configurations**.
2. At the top of the page, select the compartment that contains the configuration to which you want to add a policy.
3. Select the name of the autoscaling configuration to which you want to add a policy.
4. Under **Resources**, select **Autoscaling Policies**, then select **Create Scheduled Policy**.
5. In the **Create policy** dialog box, enter the following information:
   - **Name:** Enter a name for the new autoscaling policy.
   - **Action to perform:** Select Scale pool size or Change lifecycle state of all instances.
     - **Scale pool size:** Enter the number of instances that the pool scales to at the scheduled time.

- **Change lifecycle state of all instances:** Select the state that all instances in the pool transition to at the scheduled time.
- **Enable Schedule:** By default, the **Schedule Enabled** box is selected to enable the policy to run at the next scheduled time. Clear the box to disable this policy.

- **Execution schedule:** Define the schedule for implementing this autoscaling policy. See Designing Policies on page 690.

6. Select **Submit**.

Using the Compute Web UI

Use the oci autoscaling policy create command and required parameters to create an autoscaling policy for the specified autoscaling configuration.

```
oci autoscaling policy create  --auto-scaling-configuration-
id autoscaling_configuration_OCID  --from-json
 file://policy_definitions.json --policy-type scheduled [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Procedure

1. Get the OCID of the autoscaling configuration where you want to add this autoscaling policy: `oci autoscaling configuration list`
2. Construct a file that contains the policy definitions.

   Use the following command to show the content and format of the file:

   ```
   $ oci autoscaling policy create \
   --generate-full-command-json-input > autoscalingPolicyCreate.json
   ```

   > **Note:**
   >
   > Don't specify values for `capacity` `min` or `max`. These properties don't apply to schedule-based autoscaling configurations.

   The default values for `capacity` `min` and `max` appear in the created autoscaling policy but aren't used for schedule-based autoscaling.

   The display name is 1-255 characters, doesn't need to be unique, and can be updated.

   Use the references in Creating a Schedule-Based Autoscaling Policy on page 689 for help with setting the policy execution schedule. The `timezone` must be `UTC` and the `type` must be `cron`.

   The policy is enabled by default.

   The policy type must be `scheduled`.

   The resource action type must be `power`, and the action must be one of `STOP`, `START`, `SOFTRESET`, `RESET`.

   The following is an example autoscaling policy create input file:

   ```
   {
     [
       {
         "capacity": {
           "initial": 10
         },
         "displayName": "size 10",
         "executionSchedule":
           {
             "expression": "0 0 10 ? 1 2#2 *",
             "timezone": "UTC",
             "type": "cron"
           },
   ```

```
      "isEnabled": true,
      "policyType": "scheduled"
    },
      "capacity": {
        "initial": 30
      },
      "displayName": "size 30",
      "executionSchedule":
        {
          "expression": "0 0 7 ? 11 5#1 *",
          "timezone": "UTC",
          "type": "cron"
        },
      "isEnabled": true,
      "policyType": "scheduled"
    },
    {
      "displayName": "stop policy",
      "executionSchedule":
        {
          "expression": "0 0 7 ? JAN,APR,JUL,OCT 4#3 *",
          "timezone": "UTC",
          "type": "cron"
        },
      "isEnabled": true,
      "policyType": "scheduled",
      "resourceAction": {
        "actionType": "power",
        "action": "STOP"
      }
    },
    {
      "displayName": "start policy",
      "executionSchedule":
        {
          "expression": "0 0 13 ? JAN,APR,JUL,OCT 4#3 *",
          "timezone": "UTC",
          "type": "cron"
        },
      "isEnabled": true,
      "policyType": "scheduled",
      "resourceAction": {
        "actionType": "power",
        "action": "START"
      }
    }
  ]
}
```

3. Run the command to create new policies for the specified autoscaling configuration.

   Example:

   ```
   $ oci autoscaling policy create \
   --auto-scaling-configuration-id ocid1.autoscalingConfiguration.unique_ID \
   --from-json file://./salesPoolPolicies.json --policy-type scheduled
   ```

   Use the `work-requests work-request get` command to check the status of the autoscaling policy create command.

Using the API

See the AutoScalingPolicy Reference for API operations that define the criteria that trigger autoscaling actions and the actions to take.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Updating an Autoscaling Configuration

On Private Cloud Appliance, you can change the display name of the autoscaling configuration, the tags, and whether the autoscaling configuration is enabled.

To add policies to an autoscaling configuration or update existing policies, see Creating a Schedule-Based Autoscaling Policy on page 689 and Updating a Schedule-Based Autoscaling Policy on page 693.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Autoscaling Configurations**.
2. At the top of the page, select the compartment that contains the autoscaling configuration that you want to update.
3. For the autoscaling configuration that you want to update, select the Actions menu (⋮) and select **Edit**.
4. In the **Update Autoscaling Configuration** dialog box, make the changes. Ignore the Cooldown Period value.
5. Select **Submit**.

Using the Compute Web UI

Use the oci autoscaling configuration update command and required parameters to update certain fields on the specified autoscaling configuration, such as the name, and whether the autoscaling configuration is enabled.

```
oci autoscaling configuration update --auto-scaling-configuration-
id autoscalingConfiguration_OCID  [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

See the AutoScaling API Reference for API operations that enable you to dynamically scale compute resources to meet application requirements.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Updating a Schedule-Based Autoscaling Policy

On Private Cloud Appliance, you can update a autoscaling policy display name, either the target pool size or the lifecycle action, and the execution schedule. You can also enable or disable the policy.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Autoscaling Configurations**.
2. At the top of the page, select the compartment that contains the configuration for which you want to update a policy.
3. Select the name of the autoscaling configuration for which you want to update a policy.
4. Under **Resources**, select **Autoscaling Policies**.
5. For the policy that you want to update, select the Actions menu (⋮) and then select **Edit**.
6. In the **Update Policy** dialog box, change the policy information.
7. Select **Submit**.

Using the Compute Web UI

Use the oci autoscaling policy update command and required parameters to update an autoscaling policy in the specified autoscaling configuration.

```
oci autoscaling policy update  --auto-scaling-configuration-
id <autoscaling_config_OCID> --auto-scaling-policy-id <policy_OCID> --
policy-type <policy_type> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

See the AutoScalingPolicy Reference for API operations that define the criteria that trigger autoscaling actions and the actions to take.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting an Autoscaling Configuration

On Private Cloud Appliance, you can delete an autoscaling configuration.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Autoscaling Configurations**.
2. At the top of the page, select the compartment that contains the configuration that you want to delete.
3. For the autoscaling configuration that you want to delete, select the Actions menu (⋮), then select **Delete**.

Using the Compute Web UI

Use the oci autoscaling configuration delete command and required parameters to delete an autoscaling configuration.

```
SYNTX [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

See the AutoScaling API Reference for API operations that enable you to dynamically scale compute resources to meet application requirements.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting an Autoscaling Policy

On Private Cloud Appliance, you can delete an autoscaling policy.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Autoscaling Configurations**.
2. At the top of the page, select the compartment that contains the autoscaling configuration from which you want to delete a policy.
3. Select the name of the autoscaling configuration for which you want to delete a policy.
4. Under **Resources**, select **Autoscaling Policies**.
5. For the policy that you want to delete, select the Actions menu (⋮) and then select **Delete**.

Using the Compute Web UI

Use the oci autoscaling policy delete command and required parameters to delete an autoscaling policy for the specified autoscaling configuration.

```
oci autoscaling policy delete --auto-scaling-configuration-
id <autoscaling_configuration_OCID>  --auto-scaling-policy-
id <autoscaling_policy_id> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

See the AutoScalingPolicy Reference for API operations that define the criteria that trigger autoscaling actions and the actions to take.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Updating an Instance Pool

On Private Cloud Appliance, when you update an instance pool, you can change the name of the pool, the size of the pool, the instance configuration that's used to create new instances, the fault domains, VCN, and subnet.

To attach or detach an instance, see Attaching an Instance to an Instance Pool on page 696 and Detaching an Instance from an Instance Pool on page 697.

To attach load balancers or detach load balancer attachments, see Managing Instance Pool Load Balancer Attachments on page 698.

Configuration changes don't affect existing instances; configuration changes only affect new instances. New instances will be provisioned using the new instance configuration and placement configuration.

If you increase the size of the pool, new instances are provisioned. The new instances are created evenly across the fault domains specified by the instance configuration or the placement configuration.

If you decrease the size of the pool, instances are deleted evenly across the fault domains specified by the instance configuration or the placement configuration. In each fault domain, instances are deleted in creation date order, oldest first.

You can't select which instances to delete when you decrease the size of a pool. If you delete an individual instance that's a member of a pool, as described in Deleting an Instance on page 662, a new instance is automatically provisioned to keep the pool at the specified pool size.

If you increase the size of the pool, and some new instances can't be provisioned because of resource constraints, those instances remain in Provisioning state and the pool remains in Scaling state until all instances are provisioned. See the suggested remedies in Creating an Instance Pool on page 683.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instance Pools**.
2. At the top of the page, select the compartment that contains the instance pool that you want to update.
3. For the instance pool that you want to update, select the Actions menu (⋮), then select **Edit**.
4. In the **Update Instance Pool** dialog box, make the changes.
5. Select **Update Instance Pool**.

Using the Compute Web UI

Use the oci compute-management instance-pool update command and required parameters to update the specified instance pool.

```
oci compute-management instance-pool update --instance-pool-
id instance_pool_OCID options_with_values_to_update[OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateInstancePool operation to update the specified instance pool.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Attaching an Instance to an Instance Pool

On Private Cloud Appliance, when you attach an instance to an instance pool, the pool size increases.

> **Important:**
>
> If an autoscaling configuration is associated with the instance pool, then ensure that the autoscaling policy defines a target pool size that's large enough for the expanded pool. The next time the scheduled autoscaling policy runs, the target pool size is reset to the value that's set in the policy; if the policy size is smaller than the current size, instances are deleted.

If load balancers are attached to the pool, then the instance is also added to the load balancers.

Ensure the following conditions before you attach an instance to an instance pool:

- Both the pool and the instance to be attached are running.
- The instance isn't attached to another pool.
- The instance is in the same fault domain as the pool.
- The primary VNIC of the instance is in the same VCN and subnet as the pool.
- If secondary VNICs are defined, then the secondary VNIC of the instance is in the same VCN and subnet as the secondary VNICs used by other instances in the pool.

To attach an instance that's in a fault domain that's not included in the pool instance configuration, or is using a VCN and subnet that aren't specified by the pool instance configuration, first update the instance configuration, then attach the instance. To update the instance configuration, create and attach a new instance configuration as shown in Updating an Instance Pool on page 695.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instance Pools**.
2. At the top of the page, select the compartment that contains the instance pool.
3. Select the name of the instance pool to which you want to attach an instance.
4. On the instance pool details page, under **Resources**, select **Attached Instances**.
5. Select **Attach Instance** on the **Attached Instances** table.
6. In the **Input type** field of the **Attach Instance** dialog box, select either **Instance name** or **Instance OCID**.
   - If you select **Instance name**, a list of instances is displayed. The list of instances is labeled **Potentially attachable instances** because instances must meet certain criteria to be eligible to be attached. For example, the instance must be in the same VCN and subnet that's specified by the instance pool configuration, and the instance must not already be attached to this pool or any other pool. See the list of criteria at the top of the **Attach Instance** dialog box.

     If an instance that you think should be attachable isn't shown, try using **Instance OCID**.
   - If you select **Instance OCID**, a text field labeled **Instance OCID** is displayed where you can paste the OCID of the instance that you want to attach.
7. Select **Attach**.

   Even if the criteria listed in the dialog are met, the instance could fail to attach for some other reason. Under **Resources**, select **Work Requests**, then select the applicable work request in the list to troubleshoot any problems.

Using the Compute Web UI

Use the oci compute-management instance-pool-instance attach command and required parameters to attach an instance to an instance pool.

1. Get the information you need to run the command.
   - OCID of the instance pool that you want to update: `oci compute-management instance-pool list`

- OCID of the instance that you want to attach: `oci compute instance list`

2. Run the instance pool attach instance command.

```
oci compute-management instance-pool-instance attach --instance-pool-
id <instance_pool_OCID> --instance-id <instance_ocid>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

This operation can't be performed using the API.

Detaching an Instance from an Instance Pool

On Private Cloud Appliance, when you detach an instance from a pool, you can choose whether to delete the instance or to retain the instance separate from the pool.

Using the Compute Web UI, you can choose whether to replace the detached instance by creating a new instance in the pool. If you don't replace the detached instance, then the pool size is decremented.

If load balancers are attached to the pool, then the instance is removed from the load balancers.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instance Pools**.
2. At the top of the page, select the compartment that contains the instance pool.
3. Select the name of the instance pool from which you want to detach an instance.
4. On the **Instance Pool** details page, under **Resources**, select **Attached Instances**.
5. For the instance that you want to detach, select the Actions menu (⋮), then select **Detach**.
6. (Optional) To delete the instance and its boot volume, select the button under "Permanently terminate (delete) this instance and its attached boot volume."
7. Select **Confirm**.

   The pool size is decreased.

Using the Compute Web UI

Use the oci compute-management instance-pool detach-lb command and required parameters to detach an instance from an instance pool.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Procedure

1. Get the information you need to run the command.
   - OCID of the instance pool that you want to update: `oci compute-management instance-pool list`
   - OCID of the instance that you want to detach: `oci compute-management instance-pool list-instances`
2. Run the instance pool detach instance command.

   Syntax:

```
oci compute-management instance-pool-instance detach \
--instance-pool-id instance_pool_OCID --instance-id instance_OCID \
--is-auto-terminate [true|false] --is-decrement-size [true|false]
```

   Provide the following options if you don't want the default behavior:

`--is-auto-terminate`

If `true`, permanently terminate (delete) the instance and its attached boot volume when the instance is detached from the instance pool. The default value is `false`.

`--is-decrement-size`

If `true`, decrement the pool `size` when the instance is detached from the instance pool. This is the default.

If `false`, provision a new, replacement instance using the pool's instance configuration after the existing instance is detached from the instance pool. The pool size remains the same as it was before you performed this detach operation.

Example:

In the following example, the specified instance is detached from the pool and terminated, and a new instance is provisioned for the pool.

```
$ oci compute-management instance-pool-instance detach \
--instance-pool-id ocid1.instancePool.unique_ID \
--instance-id ocid1.instance.unique_ID \
--is-auto-terminate true --is-decrement-size false
```

The output of this command is the same as the output of the `instance-pool get` command.

Using the API

This operation can't be performed using the API.

Managing Instance Pool Load Balancer Attachments

On Private Cloud Appliance, you can attach a load balancer to an instance pool or detach a load balancer attachment from an instance pool.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instance Pools**.
2. At the top of the page, select the compartment that contains the instance pool for which you want to manage load balancer attachments.
3. Select the name of the pool that you want to manage.
4. Under **Resources**, select **Load Balancers**.

- To attach a load balancer, select **Attach Load Balancer**.

  On the **Attach Load Balancer** dialog box, specify the load balancer, backend set, port number, and VNIC as described in Creating an Instance Pool on page 683, and select **Attach Load Balancer**.

- To remove a load balancer attachment, select the Actions menu (⋮) for the load balancer attachment that you want to remove, and select **Detach**.

  The load balancer attachment remains visible in the load balancers list in the Detached state for at least 24 hours, up to 24.5 hours. No further action is needed to detach the load balancer attachment.

Using the Compute Web UI

Use the oci compute-management instance-pool attach-lb command and required parameters to attach a load balancer to the instance pool.

```
oci compute-management instance-pool attach-lb --backend-set-
name <backend-set-name> --instance-pool-id <instance-pool_OCID> --load-
balancer-id <load-balancer_OCID> --port <port_for_backend_set> --vnic-
selection <VNIC_display_name> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Procedure

- **To attach a load balancer to an instance pool:**
  1. Get the following information:
     - OCID of the instance pool to which you want to attach a load balancer: `oci compute-management instance-pool list`
     - OCID of the load balancer and name of the backend set: `oci lb load-balancer list`
     - Port value to use when creating the backend set.
     - VNIC to associate with the load balancer. The value can be `PrimaryVnic` or the display name of one of the secondary VNICs on the instance configuration that's associated with the instance pool.
  2. Run the instance pool attach load balancer command.

     Example:

     ```
     $ oci compute-management instance-pool attach-lb --instance-
     pool-id ocid1.instancePool.unique_ID --load-balancer-id
      ocid1.loadbalancer.unique_ID --backend-set-name BES1 --port 80 --vnic-
     selection PrimaryVnic
     ```
- **Remove or detach a load balancer from an instance pool:**
  1. Get the following information:
     - OCID of the instance pool
     - OCID of the load balancer
     - Backend set name
  2. Run the instance pool detach load balancer command.

     Example:

     ```
     $ oci compute-management instance-pool detach-lb --instance-
     pool-id ocid1.instancePool.unique_ID --load-balancer-id
      ocid1.loadbalancer.unique_ID --backend-set-name BES1
     ```

     When you `get` or `list` the instance pool, the load balancer attachment remains visible in state `DETACHED` for at least 24 hours, up to 24.5 hours. No further action is needed to detach the load balancer attachment.

Using the API

Use the following operations to attach or detach a load balancer to an instance pool.

- AttachLoadBalancer
- DetachLoadBalancer

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Stopping and Starting Instances in an Instance Pool

On Private Cloud Appliance, performing operations such as reset or stop on the pool object performs that operation on all instances that are members of the pool. Performing these operations on an individual instance that's a member of the pool doesn't affect any other member instances.

When instances are stopped, the instances are disconnected and unassigned from their compute nodes. When instances are started, the Compute service restarts the instances in the same fault domain that they were in when they were stopped.

Instances continue to count toward the pool size while they're stopped, and configuration of stopped pool instances is preserved. Configuration changes such as fault domain changes don't apply to pool instances that are restarted or reset.

When you perform an operation on an instance pool, all the instances in the pool are stopped, started, or rebooted. See Stopping, Starting, and Resetting an Instance on page 661 for how to prepare for an instance to stop. Stopping and starting an instance can take up to five minutes.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instance Pools**.
2. At the top of the page, select the compartment that contains the instance pool that you want to manage.
3. For the instance pool that you want to manage, select the Actions menu (⋮), and select either **Start**, **Stop**, or **Reboot**.

   By default, selecting Stop selects Soft Stop and selecting Reboot selects Soft Reboot. To stop or reboot all instances in a pool immediately, select the **Force** option on the confirmation dialog.
4. Confirm the action.

   Under **Resources**, select **Work Request(s)** to check the status of the instance pool stop, start, or reboot. Select **Attached Instances** to view the status of the instances.

Using the Compute Web UI

Use the oci compute-management instance-pool commands and required parameters to manage the state of instanced in an instance pool.

```
oci compute-management instance-pool {start | stop | reset | softreset}
 [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use these operations to manage the state of instanced in an instance pool:

- StartInstancePool
- StopInstancePool
- ResetInstancePool

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting an Instance Pool

On Private Cloud Appliance, when you delete an instance pool, the resources that were created by the pool are permanently deleted, including associated instances, attached boot volumes, and block volumes.

> **Caution:**
>
> When you delete an instance pool, the resources that were created by the pool are permanently deleted, including associated instances, attached boot volumes, and block volumes.

After the instance pool deletion, instances that had been attached to this pool remain visible in the instance list in a terminated state for at least 24 hours, up to 24.5 hours.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instance Pools**.

2. At the top of the page, select the compartment that contains the instance pool that you want to delete.
3. For the instance pool that you want to delete, select the Actions menu (⋮), then select **Delete**.
4. Select **Confirm**.

   All the instances in the pool are deleted. Deleted instances aren't attached and therefore aren't listed in **Attached Instances** under **Resources**.

   To check the status of the instance pool delete, Under **Resources**, select **Work Requests**.

Using the Compute Web UI

Use the oci compute-management instance-pool terminate command and required parameters to delete an instance pool.

```
oci compute-management instance-pool terminate --instance-pool-id <instance-pool_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the TerminateInstancePool operation to delete an instance pool.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Container Instances

On Private Cloud Appliance, a Container Instance is a serverless compute service that enables you to quickly and easily run containers without managing any servers. Container Instances run your containers on minimal compute instances that are optimized for container workloads.

A container instance is a minimal instance created from an image that runs only what is needed for containers. A container instance has no boot volume. A single user-specified container is started in the container instance.

Resources are created as needed. A container instance creates a 15 gigabyte ephemeral writable Data Volume. Persistent storage isn't supported for container instances.

Container Instance Shapes

On Private Cloud Appliance, the Container Instance shapes are flexible shapes: You can specify the number of OCPUs and amount of memory when you select the shape.

Both Container Instances shapes have the same specifications, as shown in the following table. The difference between the two shapes is the following:

- The CI.Standard.x86.Generic shape can run on either Intel or AMD processors.
- The CI.Standard.E5.Flex shape runs on the AMD EPYC 9J14 96-Core processor.

The maximum number of VNICs is 1 because each container instance has just one container.

Shape	OCPUs	Memory (GB)	Maximum VNICs	Maximum Bandwidth (Gbps)
CI.Standard.x86.Generic CI.Standard.E5.Flex	1–64	Default: 3 GB per OCPU 64 GB maximum per OCPU 1024 GB maximum per container instance	1 VNIC	1 to 40 OCPUs: 1 Gbps per OCPU 41 to 64 OCPUs: 40.0 Gbps

Working with Container Instances

Learn how to create a container instance On Private Cloud Appliance, view information about container instances and their containers, update the name and tags of a container instance, and delete a container instance. You can stop, start, and restart a container instance and move a container instance to a new compartment. Creating a container instance creates a container in the instance.

Creating a Container Instance

A Private Cloud Appliance container instance includes a single container.

The following is the minimum information that you must provide to create a container instance:

- The compartment where you want to create the container instance
- The shape (see Container Instance Shapes on page 701)
- The VCN and subnet

Container instances leverage 15 GB of ephemeral storage that's shared between the container instance and the container. This ephemeral storage is created automatically when the container instance is created. You can't manually create storage and attach it to the container instance.

Using the Compute Web UI

1. On the Compute Web UI **Dashboard**, select **Containers / Container Instances**.
2. Select **Create Container Instance** above the Container Instances list.
3. In the **Create Container Instance** dialog box, enter the following information:
   - **Name**: Enter a name for the container instance. Instance names have the following characteristics:
     - Can be changed after the container instance is created.
     - Don't need to be unique.
     - Can contain only alphanumeric characters and the hyphen (-) character.
     - Can be a maximum of 63 characters.
   - **General Information**
     - **Create in Compartment**: Select the compartment where you want to create the container instance.
     - **Fault Domain**: (Optional) Select a fault domain. By default, the system automatically selects the best fault domain for the container instance when the instance is created. If you specify a fault domain, and the requested fault domain can't accommodate the container instance, instance create fails. The fault domain can be changed after the container instance is created.
   - **Shape**: Select either the CI.Standard.x86.Generic shape or the CI.Standard.E5.Flex shape.

     Use the sliders to specify the number of OCPUs you want, and the total amount of memory you want in gigabytes.
   - **VNIC**
     - **Name**: Enter the name of the VNIC. By default, the container instance name is used.

- **Networking**
  - **Private IP**: (Optional) Enter the private IP address to use.
  - **Hostname**: (Optional) Enter a hostname if you are using DNS within the cloud network. The hostname must be unique across all VNICs in the subnet. By default, the container instance name is used for the hostname.
  - **Assign Public IP**: The box is selected by default. Clear the box if you don't want a public IP address to be assigned to this container instance.
- **Subnet**
  a. **VCN**: Select a VCN from the list. You might need to change the compartment to the compartment where the VCN is located.
  b. **Subnet**: Select a subnet from the list.
- **Skip Source Destination Check**: (Optional) Skip source destination check.

- **Container Image URL**: Enter the container image URI. The container image must come from a registry that offers a certificate from a well know certificate authority (CA). Registries that offer self-signed certificates or certificates signed by a private CA are not supported at this time. The container image manifest must be V2 schema 2.
- **Environmental Variables**: You can set the environmental variables used by the container. Container images support environment variables to customize their execution. For example, the official NGINX image supports NGINX_HOST and NGINX_PORT environment variables, so you can set the value to customize the execution to the following variables:
  - NGINX_HOST=foobar.com
  - NGINX_PORT=80
- **Command Arguments**: You can specify the actual command or entrypoint the container runs upon starting, allowing you to start the main application, pass specific flags, or run custom scripts. Setting this value overrides or customizes the default command in the container image.
- **Memory Limit**: You can set the maximum amount of RAM allocated to the shape configuration for the instance.
- **vCPU Limit**: You can set the maximum number of virtual CPUs the shape configuration for the instance can utilize. This bounds the computational resources available.
- **Security**: You can specify the security setting of the container.
  - Use the User ID and Group ID fields to set the user ID (ID) and group ID (GID) to run the entrypoint process of the container.
    - The value for User ID and Group ID must be an integer between 0 and 65535. The default value is 0.
    - The User ID and Group ID values that you specify override values that are set in the container image. When the User ID value is not specified, the entrypoint process of the container runs as root user.
    - You must set the User ID before you set the Group ID.
  - **Configure Linux capabilities**: You can configure Linux capabilities for your container. By default, the container launches with several capabilities that you can choose to drop.

    Both the Add capabilities field and Drop capabilities field support the ALL value, which allows or drops all capabilities. The value ALL refers to the Linux capabilities that are enabled by default.
    - If you leave both Add capabilities and Drop capabilities blank, all default capabilities are available for your container.
    - If you enter ALL in the Add capabilities field, all default capabilities are available except those that you list in the Drop capabilities field, and ignores ALL in the Drop capabilities field.
    - If you enter ALL in the Drop capabilities field, the container contains only the default capabilities that you list in the Add capabilities field.
    - For other cases, drops all the capabilities from the default set listed in the Drop capabilities field, then adds the capabilities listed in the Add capabilities field, and finally returns the result as the capability for your container.
    - The creation of the container instance fails if you provide any capabilities that are not in the list below.

Capability	Description
CAP_AUDIT_WRITE	Writes records to kernel auditing log.
CAP_CHOWN	Makes changes to file UIDs and GIDs.
CAP_DAC_OVERRIDE	Discretionary access control (DAC) that bypasses file read, write, and execute permission checks.
CAP_FOWNER	Bypasses permission checks on operations that normally require the file system UID of the process to match the UID of the file, excludes the operations that are covered by CAP_DAC_OVERRIDE and CAP_DAC_READ_SEARCH.
CAP_FSETID	• Does not clear set-user-ID and set-group-ID mode bits when a file is modified. • Sets the set-group-ID bit for a file whose GID does not match the file. system or any of the supplementary GIDs of the calling process.

Capability	Description
CAP_KILL	Bypasses permission checks for sending signals, which includes use of the ioctl(2) KDSIGACCEPT operation.
CAP_MKNOD	Creates special files using mknod(2).
CAP_NET_BIND_SERVICE	Binds a socket to internet domain privileged ports (port numbers less than 1024).
CAP_NET_RAW	• Uses RAW and PACKET sockets. • Binds to any address for transparent proxying.
CAP_SETFCAP	Sets file capabilities.
CAP_SETGID	• Makes manipulations of process GIDs and supplementary GID list. • Forges GID when passing socket credentials via UNIX domain sockets. • Writes a group ID mapping in a user namespace.

Capability	Description
CAP_SETPCAP	If the file capabilities are not supported, grants or removes any capability in the caller's permitted capability set to or from any other process.
CAP_SETUID	• Makes manipulations of process UIDs. • Forges UID when passing socket credentials via UNIX domain sockets. • Writes a user ID mapping in a user namespace.
CAP_SYS_CHROOT	Uses chroot(2) to change to a different root directory.

- **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.

4. Select **Create Container Instance**.

   The instance details page is displayed. On the Container Instance Information tab, the Shape Configuration section shows the shape, the number of OCPUs, the network bandwidth, and the total memory. The Primary VNIC column shows the VCN and subnet, the primary private IP address and any assigned public IP address, and the MAC address.

   To check the status of the container instance create, scroll to the Resources section and select Work Request(s).

   Select Containers in the Resources section. The container is named `computecontainer-`*`unique_ID`*.

Using the OCI CLI

Use the oci container-instances container-instance create command and required parameters to create a container instance.

```
oci container-instances container-instance create --availability-domain
 AD-1 --compartment-id compartment_OCID --containers '[{"imageUrl":
 "image_url"}]' --shape shape --shape-config '{"ocpus": 2}' --vnics
 file://network_cofig.json [OPTIONS]
```

Example:

```
oci container-instances container-instance create --availability-
domain AD-1 --compartment-id ocid1.compartment.unique_ID --containers
 file://container_config.json --shape CI.Standard.E5.Flex --shape-
config '{"ocpus": 2,"memoryInGBs": 2}' --vnics '[{"subnetId":
 "ocid1.subnet.unique_ID"}]'
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Procedure

1. Get the following information:
   - The OCID of the compartment where you want to create the container instance: `oci iam compartment list`
   - The name of the shape to use. Use the following command to list the available shapes and their characteristics. Use the OCID of the compartment where you want to create the instance.

     ```
     $ oci container-instances container-instance list-shapes --compartment-
     id compartment_OCID
     ```

     You must also specify the shape configuration, as shown in the following example. You must provide a value for `ocpus`, and you can specify the `memoryInGBs`.

     ```
     --shape-config '{"ocpus": 2, "memoryInGBs": 2}'
     ```
   - The URL of the image to use to launch the container instance. Specify a container image from an external registry of your choice. The image must be signed by a well known Certificate Authority, and the image manifest must be an up-to-date version.
2. Construct an argument for the `--vnics` option.

   This is a list of the networks available to the container on this container instance. For each VNIC, provide the name of the VNIC, the OCID of the subnet that you want to use, the private IP address, and whether a public IP address should be assigned.

   Use the following command to show the syntax for this argument:

   ```
   $ oci container-instances container-instance create --generate-param-json-
   input vnics
   ```
3. Run the create container instance command.

   Syntax:

   Only the required parameters are shown.

   ```
   oci container-instances container-instance create --availability-domain
    AD-1 \
   --compartment-id compartment_OCID --containers '[{"imageUrl":
    "image_url"}]' \
   --shape shape --shape-config '{"ocpus": 2}' \
   --vnics file://network_cofig.json
   ```

   Example:

   If you set a display name for the container instance, see the Compute Web UI procedure for characteristics.

For a complete list of required and optional parameters, use the following command:

```
$ oci compute instance launch -h
```

```
$ oci container-instances container-instance create --availability-domain
 AD-1 \
--compartment-id ocid1.compartment.unique_ID --containers
 file://container_config.json \
--shape CI.Standard.E5.Flex --shape-config '{"ocpus": 2,"memoryInGBs": 2}'
 \
--vnics '[{"subnetId": "ocid1.subnet.unique_ID"}]'
```

View the work request to monitor the status of the container instance create. The work request OCID is returned in the container instance create output.

```
$ oci container-instances work-request get --work-request-id
 ocid1.workrequest.unique_ID
```

If the status of the container instance create is Failed, list the errors reported in the work request:

```
$ oci container-instances work-request list-errors --work-request-id
 ocid1.workrequest.unique_ID
```

Using the API

Use the CreateContainerInstance operation to create a container instance.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing a Container Instance

Learn how to list all container instances in a compartment and how to view more information about a specified container instance On Private Cloud Appliance.

Using the Compute Web UI

1. On the Compute Web UI **Dashboard**, select **Containers / Container Instances**.
2. If the container instance that you want to view isn't listed, use the Compartment drop-down menu above the container instances list to select the correct compartment.
3. To view more information about the container instance, select the name of the container instance in the list.

   The container instance details page is displayed.
4. On the container instance details page, scroll to the **Resources** section, and select **Containers** to view information about the container instance container.

Using the OCI CLI

Use the oci container-instances container-instance list command and required parameters to list all container instances in a compartment.

```
oci container-instances container-instance list --container-instance-
id <container_instance_OCID> [OPTIONS]
```

Use the oci container-instances container-instance get command and required parameters to get information about a specified container instances.

```
oci container-instances container-instance list --container-instance-
id <container_instance_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the GetContainerInstance operation to get information about a specified container instances.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Updating a Container Instance

On Private Cloud Appliance you can change the name and tags of a specified container instance.

Using the Compute Web UI

1. On the Compute Web UI **Dashboard**, select **Containers / Container Instances**.
2. If the container instance that you want to view isn't listed, use the Compartment drop-down menu above the container instances list to select the correct compartment.
3. In the container instances list, for the container instance that you want to update, select the Actions menu (⋮) and select **Edit**.
4. When you're finished making your changes, select **Save Changes**.

Using the OCI CLI

Use the oci container-instances container-instance update command and required parameters to update a container instance.

```
oci container-instances container-instance update --container-instance-
id <container-instance_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateContainerInstance operation to update a container instance.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Moving a Container Instance

On Private Cloud Appliance, you can move a container instance to a different compartment.

To move a container instance, you must use the OCI CLI or API.

Using the Compute Web UI

This task can't be performed using the Compute Web UI.

Using the OCI CLI

Use the oci container-instances container-instance change-compartment command and required parameters to move a container instance to a different compartment.

```
oci container-instances container-instance change-compartment -compartment-
id <container-instance_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ChangeContainerInstanceCompartment operation to move a container instance to a different compartment.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Stopping, Starting, and Restarting a Container Instance

On Private Cloud Appliance you can stop, start, and restart a container instance.

Using the Compute Web UI

1. On the Compute Web UI **Dashboard**, select **Containers / Container Instances**.
2. If the container instance that you want to view isn't listed, use the Compartment drop-down menu above the container instances list to select the correct compartment.
3. For the container instance that you want to start, stop, or restart, select the Actions menu (⋮), and select **Start**, **Stop**, or **Restart**.

   Alternatively, in the container instance list, select the name of the container instance to display the details page for that container instance. Select **Controls**, then select **Start**, **Stop**, or **Restart**.
4. Confirm the action.

   To check the status of the action, under **Resources**, select **Work Request(s)**.

Using the OCI CLI

Use the following commands to stop, start, and restart a container instance:

- Use the oci container-instances container-instance stop command and required parameters to stop a container instance.

```
oci container-instances container-instance stop --container-instance-
id <container-instance_OCID> [OPTIONS]
```

- Use the oci container-instances container-instance start command and required parameters to start a container instance.

```
oci container-instances container-instance start --container-instance-
id <container-instance_OCID> [OPTIONS]
```

- Use the oci container-instances container-instance restart command and required parameters to restart a container instance.

```
oci container-instances container-instance restart --container-instance-
id <container-instance_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the following API operations to stop, start, and restart a container instance:

- Use the StopContainerInstance operation to stop a container instance.
- Use the StartContainerInstance operation to start a container instance.
- Use the RestartContainerInstance operation to restart a container instance.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Container Instance

On Private Cloud Appliance, you can delete a container instance.

You can permanently delete container instances that you no longer need. Any attached VNICs are automatically detached when the container instance is deleted. Eventually, the container instance's public and private IP addresses are released and become available for other container instances.

> **Important:**
>
> When the containers are deleted by the service, any data on the ephemeral storage is lost.

Using the Compute Web UI

1. On the Compute Web UI **Dashboard**, select **Containers / Container Instances**.
2. If the container instance that you want to view isn't listed, use the Compartment drop-down menu above the container instances list to select the correct compartment.
3. For the container instance that you want to delete, select the Actions menu (⋮), and select **Delete**.

   Alternatively, in the container instance list, select the name of the container instance to display the details page. Select **Controls**, then select **Delete**.
4. Confirm the deletion.

   To check the status of the container instance deletion, under **Resources**, select **Work Request(s)**. Deleted container instances temporarily remain in the list of container instances with the state **Deleted**.

Using the OCI CLI

Use the oci container-instances container-instance delete command and required parameters to delete a container instance.

```
oci container-instances container-instance delete --container-instance-
id <container-instance_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteContainerInstance operation to delete a container instance.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Working with Containers in Container Instances

On Private Cloud Appliance, you can view information about containers in container instances, retrieve log data from a container instance container, and update the name and tags of a container instance container.

Viewing Container Instance Containers

On Private Cloud Appliance, you can view containers.

Using the Compute Web UI

1. On the Compute Web UI **Dashboard**, select **Containers / Container Instances**.
2. If the container instance that you want to view isn't listed, use the Compartment drop-down menu above the container instances list to select the correct compartment.
3. In the container instances list, select the name of the container instance for which you want to view the container.
4. Under **Resources**, select **Containers**.
5. Select the name of the container in the list to show more information about the container, including log data.

Using the OCI CLI

Use the oci container-instances container list command and required parameters to list containers.

```
oci container-instances container list --compartment-id <compartment_OCID>
 [OPTIONS]
```

Use the oci container-instances container get command and required parameters to get the details for a specified container.

```
oci container-instances container get --container-id <container_OCID>
 [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the GetContainer operation to get the details for a specified container.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Updating a Container Instance Container

On Private Cloud Appliance, you can change the name and tags of a specified container instance container.

Using the Compute Web UI

1. On the Compute Web UI **Dashboard**, select **Containers / Container Instances**.
2. If the container instance that you want to view isn't listed, use the Compartment drop-down menu above the container instances list to select the correct compartment.
3. In the container instances list, select the name of the container instance for which you want to update the container.
4. Under **Resources**, select **Containers**.
5. For the container in the list, select the Actions menu (⋮) and select **Edit**.

Using the OCI CLI

Use the oci container-instances container update command and required parameters to update a container.

```
oci container-instances container update --container-
id <container_OCID>[OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateContainer operation to update a container.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Block Volume Storage

On Private Cloud Appliance, Block Volumes provide high-performance network storage capacity that supports a broad range of I/O intensive workloads.

Block Volume Overview

A block volume is a detachable block storage device that enables you to dynamically expand the storage capacity of your compute instances, provide durable and persistent data storage that can be migrated across compute instances, and host large databases.

The Block Volume service enables you to group multiple volumes in a volume group. Volume groups simplify the process to create backups and clones.

You can create, attach, connect, and move volumes, and change volume performance to meet your storage, performance, and application requirements.

After you attach and connect a volume to a compute instance, you can use the volume like a regular hard drive. You can also disconnect a volume and attach it to another compute instance without the loss of data.

The default size for block volumes is 1024 GB (1TB). Block volume size range is 50 GB to 32768 GB (32 TB).

Types of Block Volumes

- **Boot volume**: A detachable boot volume device that contains the image that's used to boot a Compute instance.
- **Block volume**: A detachable block storage device that lets you to dynamically expand the storage capacity of an instance.

You can create, attach, connect, and move volumes, and change volume performance to meet your storage, performance, and application requirements.

After you attach and connect a volume to a compute instance, you can use the volume like a regular hard drive. You can also disconnect a volume and attach it to another compute instance without the loss of data.

When a volume is created, the volume is thin (sparse) provisioned: The volume consumes only the space that has been written to the volume. When the volume is attached to an instance, the volume is thick (non-sparse) provisioned: The volume reserves exactly enough space to completely fill the volume. This behavior avoids out-of-space errors. When the volume is detached, the volume is again thin provisioned if it is not still attached to another instance.

> **Note:**
>
> The disk space consumed by a LUN is larger than the thick provisioned volume size, because overhead and metadata also occupy space within the storage pool.

Required Components

These components are required to create a volume and attach it to a compute instance:

- **Compute Instance**: A virtual machine (VM) running in the Private Cloud Appliance.
- **Volume attachment**: A paravirtualized attachment that's available for compute instances.
- **Volume**: A block volume or boot volume.

Volume Access Types

When you attach a block volume, you can specify one of the following access types:

- **Read/write:** This is the default option for volume attachments. With this option, an instance can read and write data to the volume.
- **Read/write, shareable:** With this option, you can attach a volume to more than one instance at a time and those instances can read and write data to the volume. To prevent data corruption from uncontrolled read/write operations with multiple instance volume attachments you must install and configure a cluster-aware solution before you can use the volume.
- **Read-only:** With this option, an instance can only read data on the volume. It can't update data on the volume. Specify this option to safeguard data against accidental or malicious modifications.

The access type for boot volumes is always read/write.

Block Volume Performance Options

Elastic Performance

When you create block storage, you can optionally enable High performance. By default, the volume is created with balanced performance. The volume performance setting can't be changed after the volume is created.

- **High performance**: The High performance option is recommended for workloads with higher I/O requirements.

- **Balanced performance**: The Balanced performance option is the default for new and existing block and boot volumes. This performance option provides a good balance between performance and cost savings for most workloads, including workloads that perform random I/O such as boot volumes.

Performance Limitations and Considerations

The following performance results are for unformatted data volumes.

- Throughput performance on compute instances depends on the network bandwidth that's available to the compute instance, and further limited by that bandwidth for the volume.
- If Microsoft Defender Advanced Threat Protection (Microsoft Defender ATP) is enabled in an instance, it has a significant negative impact on disk I/O performance. However, consider the security implications of disabling Microsoft Defender ATP.
- Block volume performance is per volume, so when a block volume is attached to multiple compute instances, the performance is shared across all the attached compute instances.

Block Volume Scenarios

Adding Storage Capacity

A common use of a block volume is to add storage capacity to an instance. After you create an instance and set up a cloud network, you can create a block storage volume. Then, you attach the volume to a compute instance using a volume attachment. The volume can then be mounted and used by your compute instance.

Moving a Volume to Another Compute Instance

A block volume can be detached from a compute instance and moved to a different compute instance without the loss of data. This data persistence lets you to migrate data between compute instances and ensures that data is safely stored, even when it's not connected to a compute instance. Any data remains intact until you delete the volume.

To move the volume to another compute instance, unmount the drive from the initial compute instance, detach the block volume, then attach the volume to another compute instance. From there, you connect and mount the drive from that instance's guest OS to have access to the data.

Scaling a Compute Instance

When you delete a compute instance, you can keep the associated boot volume and use it to create a new compute instance with a different compute instance type or shape. This capability lets you to easily scale up or scale down the number of cores for a compute instance.

Volume Backups and Clones

Backup and Restore

Backup and restore operations are supported on boot volumes, data volumes, and volume groups. All backups are full backups, not incremental.

To back up a boot volume, block volume, or volume group, use one of the following methods:

- **Clone**: Cloning volumes enables you to create a copy without performing the backup and restore operations. For more information about clones, see the "Clones" section and "Differences Between Volume Clones and Backups" table below. For general information about cloning boot volumes, see Cloning a Boot Volume on page 734.
- **Manual backup**: These backups are performed one time as soon as you create the backup. These backups are retained indefinitely, or until a policy-based (scheduled) backup is created for this volume.
- **Scheduled backups**: These backups are performed periodically according to a schedule defined in a backup policy. The policy schedule specifies the frequency (period) and time of the backups, and how long to retain the backups. You can create your own backup policies, or you can use Oracle provided backup policies.

Oracle Provided Backup Policies

Oracle defined backup policies cannot be used to back up a volume group. These policies can be used to back up individual block volumes and boot volumes.

Oracle defined policies cannot be changed. If you need a different backup time or retention time, for example, create a user defined backup policy as described in Managing Backup Policies on page 747.

The following backup policies are provided by Oracle and available in every compartment. A resource can have only one backup policy assigned, but a backup policy can have multiple schedules. The Oracle Bronze policy has one schedule, the Silver policy has two schedules, and the Gold policy has three schedules. All schedule times are your regional data center time. All backup types are full backups.

Bronze

1. Monthly backups that run at 00:00 on the first day of the month and are retained for twelve months.

Silver

1. Weekly backups that run at 00:00 every Monday and are retained for four weeks.
2. Monthly backups that run at 00:00 on the first day of the month and are retained for twelve months.

Gold

1. Daily backups that run at 00:00 and are retained for seven days.
2. Weekly backups that run at 00:00 every Monday and are retained for four weeks.
3. Monthly backups that run at 00:00 on the first day of the month and are retained for twelve months.

The Block Volume service will not run more than one scheduled backup of a particular resource in one day. When schedules conflict, for example, daily, weekly, and monthly backups are scheduled to run at the same time, the backup with the longest period runs.

Volume Group Back Up and Restore

You can perform most of the same backup operations and tasks with volume groups that you can perform with individual block volumes.

Volume group backups enable you to manage the backup settings for several volumes in one place. This feature simplifies the process to create time-consistent backups of running enterprise applications that span multiple storage volumes across multiple compute instances.

You can restore a volume group backup to a volume group, or you can restore individual volumes in the volume group from volume backups.

Clones

You can create a clone from a volume using the Block Volume service. Cloning enables you to make a copy of an existing block volume without needing to go through the backup and restore process.

The clone operation occurs immediately, and you can attach and use the cloned volume as a regular volume when the state changes to Available. At this point, the volume data is being copied in the background, and can take up to thirty minutes depending on the size of the volume.

There's a single point-in-time reference for a source volume while it's being cloned. If the source volume is attached when a clone is created, you need to wait for the first clone operation to complete from the source volume before creating additional clones. If the source volume is detached, you can create up to 10 clones from the same source volume simultaneously.

You can only create a clone for a volume within the same tenant. You can create a clone of a volume in a different compartment from the source volume compartment if you have the required access permissions.

Differences Between Volume Clones and Backups

Consider the following criteria when you decide whether to create a backup or a clone of a volume.

Comparison	Volume Backup	Volume Clone
Description	Creates a point-in-time backup of data on a volume. You can restore multiple new volumes from the backup later in the future.	Creates an immediately usable copy of a block volume without having to go through the backup and restore process.

Comparison	Volume Backup	Volume Clone
Use Case	Retain a backup of the data in a volume, so that you can duplicate an environment later or preserve the data for future use. Meet compliance and regulatory requirements, because the data in a backup remains unchanged over time. Support business continuity requirements. Reduce the risk of outages or data mutation over time.	Creates an immediately usable copy of a block volume without having to go through the backup and restore process.
Storage Location	Block Volume	Block Volume
Retention Policy	Policy-based backups expire, manual backups don't expire.	No expiration
Volume Groups	Supported. You can back up a volume group.	Supported. You can clone a volume group.

Creating and Attaching Block Volumes

You can create and attach a block volume to an instance to expand the available storage on the instance. The topics in this section describe how to administer the Block Volume Storage service for Private Cloud Appliance.

To create and attach block volumes to an instance, you perform a series of tasks.

Task Flow

No.	Task	Links
1.	Create a block volume.	Creating a Block Volume on page 716
2.	Attach the block volume to one or more instances.	Attaching a Volume on page 719 or Attaching a Volume to Multiple Instances on page 720
3.	Identify the added block volume and perform administrative tasks.	Find Your Volume in the Instance on page 721
4.	Configure the volume to automatically mount when the instance is rebooted.	Find Your Volume in the Instance on page 721

Creating a Block Volume

On Private Cloud Appliance, block volumes are detachable block storage devices that you can use to dynamically expand the storage capacity of an instance.

During the block volume creation, you can optionally use free-form tags to change the following block volume parameter:

Volume Block size

The default value is 8192 bytes. To change the value, specify a value in bytes for the `PCA_blocksize` free-form tag. Supported values are a power of 2 between 512 bytes and 1 megabyte, specified as a string and fully expanded. We recommend setting the value to at least 8192 bytes.

The block size can't be changed after the volume has been created.

See Using Resource Tags to Extend Functionality (IAM on the Appliance) on page 460.

(Optional) To create a block volume with nondefault values for the following attributes, first configure the required tags as described in Using Resource Tags to Extend Functionality (IAM on the Appliance) on page 460. Then use the following information when you create the block volume.

- **Synchronous Write Bias**

  You can configure the write cache flash devices for a share or LUN ("Logzilla"). The value can be changed later by updating the block volume.

  To use a nondefault value, during block volume creation, assign this defined tag:

  - **Tag namespace:** OraclePCA
  - **Tag name:** logBias
  - **Value** (select one): LATENCY (default) or THROUGHPUT
- **Secondary Cache**

  You can configure the use of the read cache flash devices for a share or LUN ("Readzilla"). The value can be changed later by updating the block volume.

  To use a nondefault value, during block volume creation, assign this defined tag:

  - **Tag namespace:** OraclePCA
  - **Tag name:** secondaryCache
  - **Value** (select one): ALL (default), METADATA, or NONE

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Block Volumes**.
2. Select **Create Block Volume**.
3. Provide the following volume information:
   - **Name:** Provide a name or description for the volume. Avoid entering confidential information.
   - **Compartment:** Select the compartment in which to create the block volume.
   - **Size (in GBs):** The default size of 1024 GB is shown. To change the size, enter a value from 50 to 32768 (50 GB to 32 TB).
   - **High Performance Volume:** (Optional) By default, the volume uses balanced performance. To create a block volume that uses the high performance feature, select the Enable High Performance button. You can only select high performance volumes if such volumes are available. For more information, see Block Volume Performance Options on page 713.

     This selection can't be changed after the volume is created.
   - **Backup Policy:** (Optional) Select a backup policy from the drop-down list. You might need to change the compartment.

     Oracle defined policies are listed, and any user defined policies. For information about Oracle defined policies (`bronze`, `silver`, and `gold`) see Oracle Provided Backup Policies on page 748.

     Backup policies can be assigned or changed after the volume is created. A volume can only have only one volume backup policy assigned at a time. For information about creating, editing, and assigning backup policies, see Managing Backup Policies on page 747. You can also back up this volume manually as described in Creating a Manual Boot or Block Volume Backup on page 745.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.

     To use nondefault values for the block size, write bias, or secondary cache, specify the appropriate tags to set the values for these attributes. See Adding Tags at Resource Creation on page 458.
4. Select **Create Block Volume**.

The volume is ready to attach to an instance after its icon lists the volume in the Available state. See Attaching a Volume on page 719.

Using the OCI CLI

Use the oci bv volume create command and required parameters to create a volume.

```
oci bv volume create --availability-domain <availability_domain_name> --
compartment-id <compartment_OCID>  [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

To use nondefault values for the block size, write bias, or secondary cache, specify the appropriate tags to set the values for these attributes. See Adding Tags at Resource Creation on page 458.

Procedure

1. Gather the information that you need to run the command:
   - Availability domain name (`oci iam availability-domain list`)
   - Compartment OCID (`oci iam compartment list`)
2. Run the volume create command.

   This procedure doesn't show all available parameters for this command. For information about additional parameters, run the command with the `--help` option.

   Example:

   This example specifies VPUs, log bias, secondary cache, and volume block size.

   VPUs per Gigabyte Option

   The value of the `--vpus-per-gb` option is the number of volume performance units (VPUs) that will be applied to this volume per GB. The default value for `vpus-per-gb` is 10, for balanced volume performance. For higher performance, you can specify 20 VPUs/GB. You can only select high performance volumes if such volumes are available. For more information, see Block Volume Performance Options on page 713.

   VPUs per GB can't be changed after the volume is created.

   You can't auto-tune volumes.

   Example:

```
$ oci bv volume create --availability-domain AD-1 --compartment-
id compartment_OCID --display-name myblockvolume --size-
in-gbs 50 --vpus-per-gb 20 --defined-tags '{"OraclePCA":
{"logBias":"THROUGHPUT","secondaryCache":"METADATA"}}'
--freeform-tags '{"PCA_blocksize": "65536"}'
{
  "data": {
    "auto-tuned-vpus-per-gb": null,
    "autotune-policies": null,
    "availability-domain": "AD-1",
    "block-volume-replicas": null,
    "compartment-id": "ocid1.compartment.
                        unique_ID
                      ",
    "defined-tags": {
      "OraclePCA": {
        "logBias": "THROUGHPUT",
        "secondaryCache": "METADATA"
      }
    },
    "display-name": "myblockvolume",
    "freeform-tags": {},
```

```
          "PCA_blocksize": "65536"
    },
    "id": "ocid1.volume.
                        unique_ID
                      ",
    "is-auto-tune-enabled": null,
    "is-hydrated": null,
    "kms-key-id": null,
    "lifecycle-state": "PROVISIONING",
    "size-in-gbs": 50,
    "size-in-mbs": 51200,
    "source-details": null,
    "system-tags": null,
    "time-created": "2022-12-08T21:05:36.647925+00:00",
    "volume-group-id": null,
    "vpus-per-gb": 20
  },
  "etag": "08d0abc9-60c6-4fc7-b6fe-85d0af1c0308",
  "opc-work-request-id": "ocid1.workrequest.
                        unique_ID
                      "
}
```

A `vpus-per-gb` value of 10 indicates that this is a balanced performance volume. A `vpus-per-gb` value of 20 indicates that this is a high performance volume.

When the volume is in the AVAILABLE state, you can attach the volume to an instance. See Attaching a Volume on page 719.

Using the API

Use the CreateVolume operation to create a block volume.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Attaching a Volume

You attach a volume to a Private Cloud Appliance instance to expand the available storage on the instance.

You can attach volumes to more than one instance at a time, see Attaching a Volume to Multiple Instances on page 720.

You can also attach a boot volume that has been detached from its instance to a different instance as a data volume. This operation is convenient for troubleshooting a boot volume and for performing administrative operations while the boot volume is detached from its instance.

> **Important:**
>
> Only attach Linux volumes to Linux instances and Microsoft Windows volumes to Microsoft Windows instances.

> **Important:**
>
> If you are reattaching a volume that was detached, the volume might be associated with a different device name, and the instance operating system might not recognize the volume.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. Select the compartment where the instance resides.
3. In the Instances list, select the instance that you want to attach a volume to.
4. Under **Resources**, select **Attached Block Volumes**.

5. In the **Attached Block Volumes** panel, select **Attach Block Volume**.
6. Enter the following information:
   a. Select the compartment where the block volume resides.
   b. Select a Block Volume.
   c. Select one of the following access modes:
      - **Read/Write:** (Default) Configures the volume attachment with read/write capabilities. The volume cannot be shared with other instances. This option enables attachment to a single instance only.
      - **Read/Write - Shareable:** Configures the volume attachment as read/write, shareable with other instances. This option enables read/write attachment to multiple instances.
      - **Read-Only - Shareable:** Configures the volume attachment as read-only, enabling attachment to multiple instances.
7. Select **Attach to Instance**.

Using the OCI CLI

Use the **oci compute boot-volume-attachment attach** command and required parameters to attach a boot volume to an instance.

```
oci compute volume-attachment attach --instance-id instance_OCID \
--volume-id volume_OCID --type paravirtualizedS]
```

> **Note:**
>
> The `--is-shareable` option is required to attach a shareable volume. The default value of this option is `false`.
>
> ```
> --is-shareable true
> ```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

This task is not available in the API.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Attaching a Volume to Multiple Instances

The Private Cloud Appliance Block Volume service provides the capability to attach a block volume to multiple compute instances. With this feature, you can share block volumes across instances in read/write or read-only mode.

Attaching block volumes as read/write and shareable enables you to deploy and manage cluster-aware solutions.

> **Important:**
>
> If you're attaching a volume that was detached, the volume might be associated with a different device name and the instance OS might not recognize the volume.

Limits and Considerations

- The Block Volume service doesn't provide coordination for concurrent write operations to block volumes attached to multiple instances, so if you configure the block volume as read/write and shareable, you must deploy a cluster aware system or solution on top of the shared storage.
- After you attach a block volume to an instance as read-only, it can only be attached to other instances as read-only. To attach the block volume to an instance as read/write, you need to detach the block volume from all instances and then you can reattach the block volume to instances as read/write.

- If the block volume is already attached to an instance as read/write nonshareable you can't attach it to another instance until you detach it from the first instance. You can then reattach it to both the first and second instances as read/write shareable.
- You can't delete a block volume until it has been detached from all instances it was attached to.
- You can attach a block volume as read/write shareable or read-only shareable up to a maximum of eight instances.
- Block volumes attached as read-only are configured as shareable by default.
- Performance characteristics are per volume, so when a block volume is attached to multiple instances the performance is shared across all the attached instances.

Configuring Multiple Instance Volume Attachments with Read/Write Access

The Block Volume service doesn't provide coordination for concurrent write operations to volumes attached to multiple instances. To prevent data corruption from uncontrolled read/write operations, you must install and configure a cluster aware system or solution such as Oracle Cluster File System version 2 (OCFS2) on top of the shared storage before you can use the volume.

Summary of the required steps:

1. Attach the block volume to an instance as Read/Write and Shareable using the Compute Web UI, OCI CLI, or API.

   See Attaching a Volume on page 719.
2. Set up your OCFS2/O2CB cluster nodes.
3. Create your OCFS2 file system and mount point.

Configuring Multiple Instance Volume Attachments with Read-Only Mode

Once you attach a block volume to an instance as read-only, it can only be attached to other instances as read-only. To attach the block volume to an instance as read/write, you need to detach the block volume from all instances and then you can reattach the block volume to instances as read/write.

1. Attach the block volume to an instance as read-only using the Compute Web UI, OCI CLI, or API.

   See Attaching a Volume on page 719.
2. Attach the block volume to additional instances as read-only using the Compute Web UI, OCI CLI, or API.

   See Attaching a Volume on page 719.

Find Your Volume in the Instance

In Private Cloud Appliance, when a block volume is initially attached to an instance, the instance sees the volume as a new disk, for example: as device `/dev/sdb`. This procedure describes how to list the disk devices in an instance so that you can find the volume and administer it in the OS.

For UNIX images, to mount these volumes when an instance boots, you need to add the volume to the `/etc/fstab` file. See Configuring Volumes to Automatically Mount (Linux Instances) on page 724.

Optionally, you can perform various administrative tasks to configure the storage to suit your storage requirements.

The utilities you use to perform the administrative tasks vary depending on the type of OS in the instance. For more administrative information, refer to the documentation for the version of the OS that's on the instance. These documentation libraries provide access to helpful information:

- Oracle Operating Systems Documentation: https://docs.oracle.com/en/operating-systems/index.html
- Oracle Virtualization Documentation: https://docs.oracle.com/en/virtualization/index.html

Identifying the Boot Volume and the Attached Block Volume Devices in the Instance Using Linux Commands

1. Log on to your instance. See Connecting to an Instance.
2. List the disk devices.

> **Important:**
>
> On UNIX OSs, the order in which volumes are attached is nondeterministic, so it can change with each reboot. If you refer to a volume using the device name, such as `/dev/sdb`, and you have more than one nonroot volume, there's no guarantee that the volume you intend to mount for a specific device name will be the volume mounted. When configuring the OS to recognize the block volume (for example, adding the volume to the `/etc/fstab` file), use the volume's SCSI ID as described in this procedure.

```
sudo ls /dev/sd*
/dev/sda  /dev/sda1  /dev/sda2  /dev/sdb
```

In this example, two devices are listed, `/dev/sda` and `/dev/sdb`.

3. Use the `fdisk -l` command to view configuration information about the devices.

   In this example, `/dev/sda` is the boot volume and `/dev/sdb` is the attached block volume.

```
sudo fdisk -l

Disk /dev/sda: 53.7 GB, 53687091200 bytes, 104857600 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 8192 bytes / 8192 bytes
Disk label type: dos
Disk identifier: 0x000af694

   Device Boot Start End Blocks Id System
/dev/sda1 * 2048 2099199 1048576 83 Linux
/dev/sda2 2099200 61442047 29671424 8e Linux LVM

Disk /dev/mapper/ol-root: 27.2 GB, 27229421568 bytes, 53182464 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 8192 bytes / 8192 bytes


Disk /dev/mapper/ol-swap: 3145 MB, 3145728000 bytes, 6144000 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 8192 bytes / 8192 bytes


Disk /dev/sdb: 1099.5 GB, 1099511627776 bytes, 2147483648 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 8192 bytes / 8192 bytes
```

This example output provides this information about `/dev/sda` and `/dev/sdb`:

- The size of `/dev/sda` is 53.7 GB (boot volume).
- `/dev/sda` has two partitions: `/dev/sda1` and `/dev/sda2`.
- The size of `/dev/sdb` is 1099.5 GB (the attached block volume), and doesn't have any partitions.

4. Identify the devices that have file systems and are mounted in the OS.

```
sudo df -T
Filesystem          Type     1K-blocks    Used Available Use% Mounted on
devtmpfs            devtmpfs  16318164       0  16318164   0% /dev
tmpfs               tmpfs     16332596       0  16332596   0% /dev/shm
```

```
tmpfs                tmpfs     16332596    8744  16323852   1% /run
tmpfs                tmpfs     16332596       0  16332596   0% /sys/fs/
cgroup
/dev/mapper/ol-root xfs       26578248 2907292  23670956  11% /
/dev/sda1            xfs        1038336  292512    745824  29% /boot
tmpfs                tmpfs      3266520       0   3266520   0% /run/user/0
```

In this example:

- `/dev/sda1` has an xfs file system and it's mounted on `/boot` (the boot volume).
- `/dev/sdb` isn't listed because this block volume was just attached and hasn't had a file system created and isn't mountable yet.

5. Find the SCSI ID for the newly attached volume.

```
sudo ls -l /dev/disk/by-id
total 0
lrwxrwxrwx. 1 root root 10 Dec  6 18:26 dm-name-ol-root -> ../../dm-0
lrwxrwxrwx. 1 root root 10 Dec  6 18:26 dm-name-ol-swap -> ../../dm-1
lrwxrwxrwx. 1 root root 10 Dec  6 18:26 dm-uuid-
LVM-83pr2aUrW2ZdCbWgsN4ZRFqvsXGGNZ8JO6il7j1YTWpywZeewYCiA6ywDmIeho1G -
> ../../dm-0
lrwxrwxrwx. 1 root root 10 Dec  6 18:26 dm-uuid-
LVM-83pr2aUrW2ZdCbWgsN4ZRFqvsXGGNZ8JsaUihE3RWozk5u4p5nOwG9sFcj34AU3F -
> ../../dm-1
lrwxrwxrwx. 1 root root 10 Dec  6 18:26 lvm-pv-uuid-Dh9ydC-Rj90-chhj-tkwq-
ZI0Z-mfop-Wtg5bh -> ../../sda2
lrwxrwxrwx. 1 root root  9 Dec  6 18:26
 scsi-3600144f096933b92000061ae9bfc0025 -> ../../sda
lrwxrwxrwx. 1 root root 10 Dec  6 18:26
 scsi-3600144f096933b92000061ae9bfc0025-part1 -> ../../sda1
lrwxrwxrwx. 1 root root 10 Dec  6 18:26
 scsi-3600144f096933b92000061ae9bfc0025-part2 -> ../../sda2
lrwxrwxrwx. 1 root root  9 Dec  8 15:17
 scsi-3600144f096933b92000061b1129e0037 -> ../../sdb
lrwxrwxrwx. 1 root root  9 Dec  6 18:26
 wwn-0x600144f096933b92000061ae9bfc0025 -> ../../sda
lrwxrwxrwx. 1 root root 10 Dec  6 18:26
 wwn-0x600144f096933b92000061ae9bfc0025-part1 -> ../../sda1
lrwxrwxrwx. 1 root root 10 Dec  6 18:26
 wwn-0x600144f096933b92000061ae9bfc0025-part2 -> ../../sda2
lrwxrwxrwx. 1 root root  9 Dec  8 15:17
 wwn-0x600144f096933b92000061b1129e0037 -> ../../sdb
```

In this example, the following line shows the SCSI ID assigned to `sdb`:

```
lrwxrwxrwx. 1 root root 9 Dec 8 15:17 scsi-3600144f096933b92000061b1129e0037
-> ../../sdb
```

where `scsi-3600144f096933b92000061b1129e0037` is the SCSI ID.

The SCSI ID is a persistent device name for `/dev/sdb` and is used when performing administrative operations on the device, such as partitioning, creating a file system, and mounting.

For more information about mounting a block volume file system to an instance, see Configuring Volumes to Automatically Mount (Linux Instances) on page 724.

6. Perform administrative tasks to configure the block volume to suit your storage requirements.

   The specific tasks you perform depend on the type of OS that runs the instance and how you want the storage configured. Refer to your OS documentation for details.

Configuring Volumes to Automatically Mount (Linux Instances)

On Private Cloud Appliance, for Linux instances, if you want to automatically mount volumes during an instance boot, you need add the volumes to the `/etc/fstab` file.

Before You Begin

Get the SCSI ID for the block volume you plan to mount. See Find Your Volume in the Instance on page 721.

On Linux operating systems, specify the volume SCSI ID in the `/etc/fstab` file instead of the device name (for example, `/dev/sdb`). This is an example of a Volume SCSI ID:

```
/dev/disk/by-id/scsi-3600144f096933b92000061b1129e0037
```

Adding Volumes to the `/etc/fstab` File

1. Prepare the newly attached block volume for mounting.

   Use the disk administration utilities included with instance OS to perform tasks such as the following:

   - Partition the volume
   - Create file systems on the volume or partitions

   Consult the documentation for your instance OS for details.

   This is an example of creating an ext4 file system for a block volume attached to a Linux instance:

   ```
   mkfs.ext4 /dev/disk/by-id/scsi-3600144f096933b92000061b1129e0037
   mke2fs 1.42.9 (28-Dec-2013)
   /dev/disk/by-id/scsi-3600144f096933b92000061b1129e0037 is entire device,
    not just one partition!
   Proceed anyway? (y,n) y
   Filesystem label=
   OS type: Linux
   Block size=4096 (log=2)
   Fragment size=4096 (log=2)
   Stride=2 blocks, Stripe width=2 blocks
   67108864 inodes, 268435456 blocks
   13421772 blocks (5.00%) reserved for the super user
   First data block=0
   Maximum filesystem blocks=2415919104
   8192 block groups
   32768 blocks per group, 32768 fragments per group
   8192 inodes per group
   Superblock backups stored on blocks:
           32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632,
    2654208,
           4096000, 7962624, 11239424, 20480000, 23887872, 71663616,
    78675968,
           102400000, 214990848

   Allocating group tables: done
   Writing inode tables: done
   Creating journal (32768 blocks): done
   Writing superblocks and filesystem accounting information: done
   ```

2. Create a mount point for each file system you plan to mount.

   ```
   mkdir /mnt/volume1
   ```

3. Add the volume to the `/etc/fstab` file.

For this example, the following new line is added to the `/etc/fstab` file:

```
/dev/disk/by-id/scsi-3600144f096933b92000061b1129e0037 /mnt/volume1 ext4
 _netdev,nofail 0 0
```

Following are descriptions of these field values:

- **Device:** Specified using the SCSI ID:

  ```
  /dev/disk/by-id/scsi-3600144f096933b92000061b1129e003
  ```

- **Mount point:** The mount point created in the previous step: `/mnt/volume1`
- **Type:** The type of file system: `ext4` in this example.
- **Options:**
  - `_netdev` – Configures the mount process to initiate before the volumes are mounted.
  - `nofail` – If the device does not exist, no errors are reported. This is a good option to use when an instance is used to create a custom image. Future instances created with that image will not include the block volume and might fail to boot without this option.
- **Dump:** The value `0` means do not use the obsolete `dump` utility.
- **fsck:** The value `0` means do not run `fsck`.

4. Use the following command to mount the volumes that are in the `/etc/fstab` file:

   ```
   sudo mount -a
   ```

5. Verify that the file system is mounted:

   ```
   mount | grep /mnt
   /dev/sdb on /mnt/volume1 type ext4
    (rw,relatime,seclabel,stripe=2,data=ordered,_netdev)
   ```

Managing Block Volumes

You can list, edit, move, clone, detach, and delete block volumes On Private Cloud Appliance.

Listing Block Volumes and Block Volume Details

On Private Cloud Appliance, you can list all block volumes in a specific compartment, and detailed information about a single volume.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Block Volumes**.
2. At the top of the page, select the compartment that contains the volume you want to see.
3. To view block volume details, select the name of the block volume.

   The details are displayed.

Detail Item	Description
Block volume icon	Displays the status of the block volume.
Block volume name	The name of the block volume.
Block Volume Information and Tags	Tabs that you can select to display: • General Information • Tags that have been applied to this object
Created	The day and time that the volume was created.

Detail Item	Description
Compartment	The compartment that the volume belongs to.
OCID	The volume's Oracle Cloud ID.
Backup Policy	The backup policy assigned to the volume.
Size	The size of the volume.
High Performance Enabled	Whether the volume is configured as a high performance volume, and the volume performance units (VPUs) per GB.

Using the OCI CLI

Use these commands to list volumes:

- List all volumes in a compartment:

  Use the oci bv boot-volume list command.

  ```
  oci bv volume list --compartment <compartment_OICD>  [OPTIONS]
  ```

- List the details for a single volume:

  oci bv boot-volume get command.

  ```
  oci bv volume get --compartment <compartment_OICD>  [OPTIONS]
  ```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use theListVolumes operation to list all volumes in a compartment.

Use the GetVolume operation to list the details for a single volume.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Listing Block Volume Attachments

On Private Cloud Appliance, you can list all Block Volume volume attachments in a specific compartment, and detailed information for a single volume attachment.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. At the top of the page, select the compartment that contains the instance that has the attached block volume.
3. Select the instance name to display the details.
4. Under **Resources**, select **Attached Block Volumes**.

   Block volumes that are attached to this instance are listed in the table.
5. To see details for a block volume, select the block volume name.

Using the OCI CLI

Use these commands to list volume attachments:

- Use xx to list all volume attachments in a compartment:

  ```
  oci compute volume-attachment list --compartment-id <compartment_OCID>
   [OPTIONS]
  ```

- Use xx to list the details of a single volume attachment:

```
oci compute volume-attachment list --volume-id <volume_OCID>  [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use these APIs:

- List all volume attachments in a compartment: ListVolumeAttachments
- List the details for a single volume attachment: GetVolumeAttachment

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Updating a Block Volume

On Private Cloud Appliance, you can change settings for block volumes while the volumes are online, without any downtime. You can change the display name, increase the size, and change tags.

To increase the volume size, see Resizing Volumes on page 735.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Block Volumes**.
2. At the top of the page, select the compartment that contains the volume that you want to edit.
3. For the volume that you want to edit, select the Actions menu (⋮), then select **Edit**.
4. In the Edit Block Volume dialog box, make changes.
   - **Name:** The volume display name. The name does not have to be unique.
   - **Size (in GB):** You can increase the size in 1 GB increments up to 32 TB. You can't decrease the size. Before you change the size, see Resizing Volumes on page 735 to understand the implications of this change.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.

     To use nondefault values for the block size, write bias, or secondary cache, specify the appropriate tags to set the values for these attributes. See Adding Tags at Resource Creation on page 458.
5. Select **Save Changes**.

Using the OCI CLI

Use the oci bv volume update command and required parameters to update a volume.

```
oci bv volume update --volume-id <volume_OCID> --display-name <new-display-
name>  [OPTIONS]
```

To use nondefault values for the block size, write bias, or secondary cache, specify the appropriate tags to set the values for these attributes. See Adding Tags at Resource Creation on page 458.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateVolume operation to update a volume.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Moving a Volume to a Different Compartment

On Private Cloud Appliance, you can move Block Volume resources such as block volumes, boot volumes, clones, volume backups, volume groups, and volume group backups from one compartment to another.

When you move a resource to a new compartment, associated resources might not be moved. For example, when you move a block volume, any backups of that volume are not moved.

> **Important:**
>
> Any access policies that exist on the new compartment apply immediately. Before you move resources to a different compartment, ensure that the resource users have sufficient access permissions on the compartment the resource is being moved to.

- You can't move a block volume or boot volume from a security zone to a standard compartment.
- You can't move a volume from a standard compartment to a compartment that is in a security zone if the volume violates any security zone policies.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci bv boot-volume change-compartment:

```
oci bv volume change-compartment --volume-id <volume_OCID> --compartment-
id <destination_compartment_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ChangeVolumeCompartment operation to move a volume to another compartment.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Cloning a Block Volume

On Private Cloud Appliance, you can create a clone from a volume using the Block Volume service. Cloning enables you to make a copy of an existing block volume without needing to go through the backup and restore process.

A cloned volume is a point-in-time direct disk-to-disk deep copy of the source volume. All the data that's in the source volume is copied to the clone volume. Any subsequent changes to the data on the source volume aren't copied to the clone.

By default, the clone is the same size as the source volume unless you specify a larger volume size when you create the clone.

The volume data is being copied in the background, and can take up to thirty minutes depending on the size of the volume. You can attach and use the cloned volume as a regular volume when the state changes to Available.

For more information about cloning volumes, see Volume Backups and Clones on page 714.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Block Volumes**.
2. At the top of the page, select the compartment that contains the block volume you want to clone.
3. For the volume you want to clone, select the Actions menu (⋮), then select **Create Clone**.
4. In the dialog box, enter the following information:
   - **Name:** A name or description for the volume. Avoid entering confidential information.

- **Compartment:** Select the compartment in which to clone the block volume.
- **Size (in GBs):** You can keep the size the same, or increase the size in 1 GB increments up to 32768 (32 TB). You can't decrease the size.
- **High-Performance Volume:**(Optional) By default, the clone has the same performance setting as the source volume. Use this button to change the performance setting for this clone.
- **Backup Policy:**(Optional) Select a backup policy from the drop-down list. You might need to change the compartment.

  Oracle defined policies are listed, and any user defined policies. For information about Oracle defined policies (`bronze`, `silver`, and `gold`) see Oracle Provided Backup Policies on page 748.

  Backup policies can be assigned or changed after the volume is cloned, or you can back up this volume manually. A volume can only have only one volume backup policy assigned at a time. For information about creating, editing, and assigning backup policies, see Managing Backup Policies on page 747. You can also back up this volume manually as described in Creating a Manual Boot or Block Volume Backup on page 745.
- **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.

5. Select **Create Clone**.

Using the OCI CLI

Use the oci bv volume create command and required parameters to clone a volume.

```
oci bv volume create --availability-domain <availability_domain_name> --
compartment-id <compartment_OCID> --source-volume-id <source_volume_OCID> --
display-name <display_name>  [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateVolume operation to clone a volume.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Detaching a Block Volume

On Private Cloud Appliance, when an instance no longer needs access to a volume, you can detach the volume from the instance without affecting the volume's data.

> **Caution:**
>
> If you later reattach the detached volume, the volume might be associated with a different device name and the instance operating system might not recognize the volume.

Prerequisites

- Perform administrative tasks to remove any dependencies that any instances have for the block volume.

  For example, ensure that no applications are accessing the volume. Unmount the volume and remove it from the `/etc/fstab` file, and so on.

Using the Compute Web UI

1. Perform administrative tasks to remove any dependencies that any instances have for the block volume.

   For example, ensure that no applications are accessing the volume. Unmount the volume and remove it from the `/etc/fstab` file, and so on.
2. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
3. At the top of the page, select the compartment that contains the instance with the volume that you want to detach.

4. Select the name of the instance that has the volume attached.
5. Under Resources, select **Attached Block Volumes**.
6. Next to the volume you want to detach, select the Actions menu (⋮), then select **Detach**.
7. Confirm when prompted.

Using the OCI CLI

Use the oci compute volume-attachment detach command and required parameters to detach a block volume from an instance.

```
oci compute volume-attachment detach --volume-attachment-id <volume-attachment_OCID>  [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DetachVolume operation to detach a block volume from an instance.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Block Volume

On Private Cloud Appliance, you can delete a volume that's no longer needed.

> **Caution:**
>
> You can't undo this operation. Any data on a volume is permanently deleted when the volume is deleted.

> **Caution:**
>
> All policy-based (scheduled) backups expire. A manual backup expires if a scheduled backup of the same volume is created after the manual backup was created. To keep a volume backup indefinitely, cancel all future scheduled backups and create a manual backup before you delete the source volume. See Backing Up Block Volumes on page 744.

The result of deleting a block volume differs depending on the following conditions:

- **The volume has no backups or clones:** The volume is immediately deleted and the volume capacity is returned to the system for reuse. The volume is marked TERMINATED and eventually is no longer listed.
- **The volume has a backup or a clone:** The volume is marked TERMINATED, but the volume is not deleted and the capacity is not returned to the system until all of the backups and clones of the volume are deleted.
- **The volume is part of a DR configuration and replicating:** The volume is marked TERMINATED, but the volume is not deleted and the capacity is not returned to the system until DR completes the replication.

Using the Compute Web UI

1. Perform administrative tasks to remove any dependencies that any instances have for the block volume.

   For example, ensure that no applications are accessing the volume. Unmount the volume and remove it from the `/etc/fstab` file, and so on.
2. In the Compute Web UI navigation menu, select **Block Storage**, then select **Block Volumes**.
3. At the top of the page, select the compartment that contains the block volume that you plan to delete.
4. For the volume you plan to delete, select the Actions menu (⋮), then select **Terminate**.
5. Confirm the termination.

   The block volume enters the TERMINATED state, and remains in that state for about 24 hours until it's removed.

Using the OCI CLI

Use the oci bv volume delete command and required parameters to delete a block volume.

```
oci bv volume delete --volume-id <volume_OCID>  [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteVolume operation to delete a volume.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing Boot Volumes

On Private Cloud Appliance, when you launch an instance, a new boot volume for the instance is created in the same compartment and attached to the instance.

You can accept the image's default boot volume size, or you can specify a larger size, up to 16 TB. If you specify a size smaller than the default boot volume size, your request is ignored and the boot volume is the default size.

The boot volume is associated with the instance until you terminate the instance. When you terminate the instance, you can preserve and reuse the boot volume and its data.

Boot volumes are encrypted by default.

You can list, detach, reattach, back up, clone, and delete boot volumes.

For Linux-based images, the custom boot volume size must be at least the image's default boot volume size or 50 GB, whichever is larger.

For Microsoft Windows-based images, the custom boot volume size must be at least the image's default boot volume size or 256 GB, whichever is larger. The minimum size requirement for Microsoft Windows images is to ensure that enough space is available for patches and updates.

If you specify a custom boot volume size, extend the volume to take advantage of the larger size.

Listing Boot Volumes

On conkeyref="cnvVariables/cnv-prodname-short"/>, you can list all the boot volumes in a compartment.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Boot Volumes**.
2. At the top of the page, select the compartment that contains the boot volume that you want to see.

   A list of boot volumes is displayed.
3. To view details about a boot volume, select the boot volume name.

Using the OCI CLI

Use the oci bv boot-volume list command and required parameters to list the boot volumes in a compartment.

```
oci bv boot-volume list --availability-domain <availability_domain_name> --
compartment-id <compartment_OCID>  [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListBootVolumes operation to list the boot volumes in a compartment.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Listing Boot Volume Attachments

On Private Cloud Appliance, you can list the boot volume that is attached to an instance.

Using the API

Use the ListBootVolumeAttachments operation to list the boot volume attached to an instance.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. At the top of the page, select the compartment that contains the instance with boot volume attachment you want to see.
3. Select the name of the instance for which you want to view the boot volume attachment.
4. Under **Resources**, select **Attached Boot Volumes**.

   The boot volume attachments are displayed.
5. To view the details about an attachment, select the boot volume attachment name.

Using the OCI CLI

Use the oci compute boot-volume-attachment list command and required parameters to list the boot volume attached to an instance.

```
oci compute boot-volume-attachment list --availability-
domain <availability_domain_name> --compartment-id <compartment_OCID>
 [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListBootVolumeAttachments operation to list the boot volume attached to an instance.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Detaching a Boot Volume

On Private Cloud Appliance, if you think a boot volume issue is causing a compute instance problem, you can stop the instance and detach the boot volume using the steps described in this topic. Then you can attach the boot volume to another instance as a data volume to troubleshoot it.

Prerequisite:

The instance must be in the Stopped state to detach a boot volume. You must reattach a boot volume before you can start the instance. See Stopping, Starting, and Resetting an Instance on page 661.

When you stop the instance, any application that's running on the instance is immediately stopped, possibly resulting in data corruption. To avoid stopping the instance while applications are running, manually shut down the instance by using the commands available in the instance OS. After the instance is shut down from the OS, then stop the instance.

Using the Compute Web UI

1. Stop the instance:
   a. Shut down the instance from the instance OS.
   b. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
   c. At the top of the page, select the compartment that contains the instance you plan to stop.
   d. Select the name of the instance.
   e. On the instance details page, select **Controls** (upper right corner), then select **Stop**.

Wait for the status to change from **Stopping** to **Stopped**. Stopping an instance can take up to five minutes.

2. Under **Resources**, select **Boot Volumes**.
3. Select the boot volume Actions menu (⋮), then select **Detach**.
4. Confirm when prompted.

Using the OCI CLI

Use the oci compute boot-volume-attachment detach command and required parameters to detach a boot volume from an instance.

```
oci compute boot-volume-attachment detach --boot-volume-attachment-
id <boot_volume-attachment_OCID>  [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DetachBootVolume operation to detach a boot volume from an instance.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Reattaching a Boot Volume

On Private Cloud Appliance, you can reattach a boot volume to an instance.

A boot volume is automatically attached to an instance when the instance is launched. Sometimes you need to detach and reattach a boot volume as a data volume for troubleshooting purposes. For that information, see Detaching a Boot Volume on page 732 and Attaching a Volume on page 719. This procedure describes how to reattach a boot volume to an instance.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. At the top of the page, select the compartment that contains the instance for which you want to reattach a boot volume.
3. Select the instance name.
4. Under **Resources**, select **Boot Volumes**.
5. Select the boot volume Actions menu (⋮), then select **Attach**.

   Confirm when prompted.

   Wait for the state of the boot volume to be **Attached**. Then you can restart the instance as described in Stopping, Starting, and Resetting an Instance on page 661.

Using the OCI CLI

Use the oci compute boot-volume-attachment attach command and required parameters to reattach a boot volume to an instance.

```
oci compute boot-volume-attachment attach --boot-volume-id
  <boot_volume_OCID> --instance-id <instance_OCID>  [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the AttachBootVolume operation to reattach a boot volume to an instance.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Cloning a Boot Volume

On Private Cloud Appliance, you can create a clone from a boot volume using the Block Volume service. Cloning enables you to make a copy of an existing boot volume without needing to go through the backup and restore process.

Any subsequent changes to the data on the source boot volume are not copied to the boot volume clone. The clone is the same size as the source boot volume unless you specify a larger volume size when you create the clone.

The clone operation occurs immediately and you can use the cloned boot volume when the state changes to available.

There is a single point-in-time reference for a source boot volume while it is being cloned. If you clone a boot volume while the associated compute instance is running, you need to wait for the first clone operation to complete before creating more clones. You also need to wait for any backup operations to complete.

You can only create a clone for a boot volume within the same tenant. You can create a clone of a boot volume in a different compartment from the source volume compartment if you have the required access permissions.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Boot Volumes**.
2. At the top of the page, select the compartment that contains the boot volume that you want to clone.
3. For the volume you want to clone, select the Actions menu (⋮), then select **Create Clone**.
4. In the dialog box, enter the following information:
   - **Name:** A name or description for the volume. Avoid entering confidential information.
   - **Compartment:** Select the compartment in which to clone the block volume.
   - **High-Performance Volume:** (Optional) By default, the clone has the same performance setting as the source volume. Use this button to change the performance setting for this clone.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
5. Select **Create Clone**.

   The new boot volume is ready to use when it reaches the Available state. For example, you can select the Actions menu and then select **Create Instance** to use this new boot volume to create a new instance.

Using the OCI CLI

Use the oci bv boot-volume create command and required parameters to clone a boot volume.

```
oci bv boot-volume create --source-boot-volume-id OCID_of_volume_to_clone
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

This task is not available in the API.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Boot Volume

On Private Cloud Appliance, you can delete a boot volume that's detached from an instance.

> **Caution:**
>
> Any data on a volume is permanently deleted when the volume is deleted. You can't restart the associated instance.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Compute**, then select **Instances**.
2. At the top of the page, select the compartment that contains the instance with the boot volume you want to delete.

3. For the instance that has the boot volume you want to delete, select the Actions menu (⋮), then select **Stop**.
4. When prompted, select **Force stop the instance by immediately powering off**, then select **Force Stop Instance**.
5. Wait for the instance to reach the **Stopped** state.
6. Select the name of the instance that has the boot volume you want to delete.
7. Under **Resources**, select **Boot Volumes**.
8. For the boot volume that you want to delete, select the Actions menu (⋮), then select **Detach**.

   Confirm when prompted.
9. Select the name of the boot volume.
10. Select the **Controls** button (upper right), then select **Terminate**.
11. Confirm the termination.

    The block volume enters the TERMINATED state, and remains in that state for about 24 hours until it's removed.

Using the OCI CLI

Use the command and required parameters to delete a boot volume.

```
oci bv boot-volume delete --boot-volume-id <boot_volume_OCID>  [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteBootVolume operation to delete a boot volume.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Resizing Volumes

On Private Cloud Appliance, the Block Volume service lets you expand the size of block volumes and boot volumes.

Use one of the following options to increase the size of a volume:

- Expand an existing volume in place with online resizing.
- Restore from a volume backup to a larger volume.
- Clone an existing volume to a new, larger volume.
- Expand an existing volume in place with offline resizing.

You can't decrease the size of a volume.

Online Block Volume Resizing

This topic describes how to resize a block volume while it is online.

> **Caution:**
>
> Before you resize a boot or block volume, create a full backup of the volume.
> See Backing Up Block Volumes on page 744.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Block Volumes**.
2. At the top of the page, select the compartment that contains the block volume that you want to resize.
3. For the volume you plan to resize, select the Actions menu (⋮), then select **Edit**.
4. Change the size:
   - **Size (in GBs):** You can increase the size in 1 GB increments up to 16384 (16 TB). You can't decrease the size.
5. Select **Save Changes**.

6. Perform disk administrative tasks. See Completing the Online Resizing Operation on page 737.

Using the OCI CLI

Use the oci bv volume update command and required parameters to resize the volume.

```
oci bv volume update --volume-id <volume_OCID> --size-in-gbs <size_in_GBS>
 [OPTIONS]
```

<size_in_GBS> is the size of the block volume. You can increment the size. You can't decrease the size. The value must be between 50 GB and 32 TB and specified in 1 GB increments.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

After resizing, perform disk administrative tasks. See Completing the Online Resizing Operation on page 737.

Using the API

Use the UpdateVolumeDetails operation to resize a volume.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Online Boot Volume Resizing

This topic describes how to resize a boot volume while it's online.

> **Important:**
>
> Before you resize a boot or block volume, create a full backup of the volume. See Backing Up Block Volumes on page 744.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Boot Volumes**.
2. At the top of the page, select the compartment that contains the boot volume.
3. For the volume you plan to resize, select the Actions menu (⋮), then select **Edit**.
4. Change the size:
   - **Size (in GBs):** You can keep the size the same, or increase the size in 1 GB increments up to 16384 (16 TB). You can't decrease the size.
5. Select **Save Changes**.
6. Rescan the disk.

   For details, consult the OS documentation for the OS type and version running in the instance.
7. Extend the partition.

   For details, consult the OS documentation for the OS type and version running in the instance.

Using the OCI CLI

Use the oci bv boot-volume update command and required parameters to resize a boot volume.

```
oci bv boot-volume update --boot-volume-id <volume_OCID> --size-in-
gbs <size_in_GBSs  [OPTIONS]
```

For *size _in_GBs* specify the size of the boot volume. You can increase the size in 1 GB increments up to 16384 (16 TB). You can't decrease the size.

> **Important:**
>
> The `path` option is required and its value must be `AUTOSELECT`. The value must be given in all uppercase letters. Providing any other value will cause the `export create` command to fail.

The export path is always automatically generated. See the `path` property in the command output for the export path.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

After resizing, perform disk administrative tasks. See Completing the Online Resizing Operation on page 737.

Using the API

Use the UpdateBootVolume operation to resize a boot volume.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Completing the Online Resizing Operation

After resizing the online block volume or boot volume, you need to perform the following disk administrative tasks:

1. Rescan the disk.

   For details, consult the OS documentation for the OS type and version running in the instance. Also see Rescanning the Disk for a Block Volume or Boot Volume
2. Extend the partition.

   For details, consult the OS documentation for the OS type and version running in the instance. Also see Extending the Partition for a Block Volume and Extending the Partition for a Boot Volume.

Offline Block Volume Resizing

On Private Cloud Appliance, for offline resizing, you detach the volume from an instance before you expand the volume size. When the volume is resized and reattached, you need to extend the partition, but you do not need to rescan the disk.

Considerations When Resizing an Offline Volume

Whenever you detach and reattach volumes, there are complexities and risks for both UNIX based and Microsoft Windows instances. Be aware of the following cautions.

> **Caution:**
>
> Before you resize a boot or block volume, create a full backup of the volume. See Backing Up Block Volumes on page 744.

> **Caution:**
>
> When you reattach a volume to an instance after resizing, if you are not using consistent device paths, or if the instance does not support consistent device paths, device order and path might change. If you are using a tool such as Logical Volume Manager (LVM), you might need to fix the device mappings.

Prerequisite

- Block Volumes: Detach the block volume. See Detaching a Block Volume on page 729.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Block Volumes**.
2. At the top of the page, select the compartment that contains the block volume you want to resize.
3. For the volume you plan to resize, select the Actions menu (⋮), then select **Edit**.
4. Change the size:
   - **Size (in GBs):** You can increase the size in 1 GB increments up to 32768 (32 TB). You can't decrease the size.
5. Select **Save Changes**.
6. Complete the resizing operation.

See Completing the Offline Resizing Operation for Block Volumes on page 738

Using the OCI CLI

Use the oci bv boot-volume update command and required parameters to resize the block volume.

```
oci bv volume update --volume-id <volume_OCID> --size-in-gbs <size_in_GBS>
 [OPTIONS]
```

<size_in_GBs> is the size of the block volume. You can increase the size in 1 GB increments up to 32768. You can't decrease the size.

Next, complete the resizing operation.

See Completing the Offline Resizing Operation for Block Volumes on page 738

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateVolume operation to resize a block volume.

Next, complete the resizing operation.

See Completing the Offline Resizing Operation for Block Volumes on page 738

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Completing the Offline Resizing Operation for Block Volumes

After resizing the offline block volume or boot volume, you need to perform the following administrative tasks.

1. Reattach the volume.

   See Attaching a Volume on page 719.
2. Rescan the disk.

   For details, consult the OS documentation for the OS type and version running in the instance. Also see Rescanning the Disk for a Block Volume or Boot Volume
3. Extend the partition.

   For details, consult the OS documentation for the OS type and version running in the instance. Also see Extending the Partition for a Block Volume and Extending the Partition for a Boot Volume.

Offline Boot Volume Resizing

On Private Cloud Appliance, for offline boot volume resizing, you must stop the instance and detach the volume from an instance before you expand the volume size. When the volume is resized and reattached, you need to extend the partition, but you don't need to rescan the disk.

Considerations When Resizing an Offline Volume

Whenever you detach and reattach volumes, there are complexities and risks for both UNIX based and Microsoft Windows instances. Be aware of the following cautions.

> **Caution:**
>
> Before you resize a boot or block volume, create a full backup of the volume. See Backing Up Block Volumes on page 744.

> **Caution:**
>
> When you reattach a volume to an instance after resizing, if you aren't using consistent device paths, or if the instance doesn't support consistent device paths, device order and path might change. If you're using a tool

such as Logical Volume Manager (LVM), you might need to fix the device mappings.

Prerequisites

- Stop the instance that has the boot volume. See Stopping, Starting, and Resetting an Instance on page 661
- Detach the boot volume. See Detaching a Boot Volume on page 732.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Boot Volumes**.
2. At the top of the page, select the compartment that contains the boot volume you want to resize.
3. For the volume you plan to resize, select the Actions menu (⋮), then select **Edit**.
4. Change the size:
   - **Size (in GBs):** You can keep the size the same, or increment the size. You can't decrease the size. The value must be between 50 GB and 32 TB and specified in 1 GB increments.
5. Select **Save Changes**.
6. Complete the resizing operation.

   See Completing the Offline Resizing Operation for Boot Volumes on page 739

Using the OCI CLI

Use the oci bv boot-volume update command and required parameters to resize a boot volume.

```
oci bv boot-volume update --boot-volume-id <volume_OCID> --size-in-
gbs <size_in_GBs>  [OPTIONS]
```

The *size_in_GBs* value is the size of the boot volume. You can increment the size. You can't decrease the size. The value must be between 50 GB and 32 TB and specified in 1 GB increments.

Next, complete the resizing operation.

See Completing the Offline Resizing Operation for Boot Volumes on page 739.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateBootVolume operation to resize a boot volume.

Next, complete the resizing operation.

See Completing the Offline Resizing Operation for Boot Volumes on page 739

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Completing the Offline Resizing Operation for Boot Volumes

After resizing the online block volume or boot volume, you need to perform the following administrative tasks.

1. Attach the boot volume to a second instance as a data volume.

   See Attaching a Volume on page 719.
2. Extend the partition and grow the file system.

   For details, consult the OS documentation for the OS type and version running in the instance.
3. Detach the data volume.

   See Detaching a Block Volume on page 729.
4. Reattach the boot volume.

   See Reattaching a Boot Volume on page 733.
5. Restart the instance.

See Stopping, Starting, and Resetting an Instance on page 661.

Managing Volume Groups

On Private Cloud Appliance, the Block Volume service enables you to organize multiple volumes into a volume group. A volume group can include both block and boot volumes.

You can use volume groups to create volume group backups and clones that are point-in-time and crash-consistent. This simplifies the process to create time-consistent backups of running enterprise applications that span multiple storage volumes across multiple instances. You can then restore an entire group of volumes from a volume group backup.

Similarly, you can also clone an entire volume group in a time-consistent and crash-consistent manner. A deep disk-to-disk and fully isolated clone of a volume group, with all the volumes associated in it, becomes available for use within a matter of seconds. This speeds up the process of creating new environments for development, quality assurance, user acceptance testing, and troubleshooting.

When working with volume groups and volume group backups, keep the following in mind:

- You can only add a volume to a volume group when the volume status is Available.
- A volume group can include up to 32 volumes, up to a maximum size of 128 TB. For example, if you wanted to add 32 volumes of equal size to a volume group, the maximum size for each volume would be 4 TB. You could add volumes that vary in size, however the overall combined size of all the block and boot volumes in the volume group cannot be more than 128 TB. Ensure you account for the size of any boot volumes in your volume group when considering volume group size limits.
- A volume can only be in one volume group.
- When you clone a volume group, a new group with new volumes is created. For example, if you clone a volume group containing three volumes, once this operation is complete, you will now have two separate volume groups and six different volumes with nothing shared between the volume groups.
- When you update a volume group using the OCI CLI, SDKs, or REST APIs, specify all the volumes to include in the volume group. The list of volumes to include replaces the existing list. If you do not include a volume OCID in the updated list, that volume will be removed from the volume group.
- When you delete a volume group, the individual volumes in the group are not deleted.
- When you delete a volume that is part of a volume group, you must first remove the volume from the volume group and then delete the volume.
- When you delete a volume group backup, all the volume backups in the volume group backup are deleted.

Viewing the Volumes in a Volume Group

On Private Cloud Appliance, you can view volume groups using the Compute Web UI, OCI CLI, and API.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Volume Groups**.
2. At the top of the page, select the compartment that contains the volume group you want to see.
3. In the **Volume Groups** list, select the volume group you want to see.
4. To view the block volumes in the volume group, under **Resources**, select **Volumes**.

Using the OCI CLI

Use the oci bv volume-group list command and required parameters to list volume groups.

```
oci bv volume-group list --compartment-id <compartment_OCID>  [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListVolumeGroups operation to list volume groups.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Creating a Volume Group

On Private Cloud Appliance, you can create volume groups that include block and boot volumes using the Compute Web UI, OCI CLI, and API.

Prerequisite

Ensure that the volumes you plan to group are created. See Creating a Block Volume on page 716.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Volume Groups**.
2. Select **Create Volume Group**.
3. Provide the following information:
   - **Name**: A user-friendly name or description for the volume group. Avoid entering confidential information.
   - **Compartment**: The compartment for the volume group.
   - **Volumes**: Select a volume to add to the group from the volume drop-down list. You might need to select a different compartment above the volume drop-down list. Select **+ Add Volume** to add another volume.
   - **Backup Policy:** (Optional) Select a backup policy from the drop-down list. You might need to change the compartment.
4. **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
5. Select **Create Volume Group**.

Using the OCI CLI

Use the oci bv volume-group create command and required parameters to create a volume group.

```
oci bv volume-group create --availability-domain <availability_domain_name>
 --compartment-id <compartment_OCID> --source-details <json_string> or
 file://<path_to_JSON_file>  [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateVolumeGroup operation to create a volume group:

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Adding Volumes to a Group

On Private Cloud Appliance, you can add block and boot volumes to an existing volume group using the Compute Web UI, OCI CLI, and API.

> **Note:**
>
> You can't add a volume with an existing backup policy assignment to a volume group with a backup policy assignment. You must first remove the backup policy assignment from the volume before you can add it to the volume group.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Volume Groups**.
2. At the top of the page, select the compartment that contains the volume group.
3. In the **Volume Groups** list, select the volume group you want to add the volume to.
4. Under **Resources**, select **Volumes**.
5. Select **Add Volumes**.

6. Select a volume to add to the group from the volume drop-down list. You might need to select a different compartment above the volume drop-down list. Select **+ Add Volume** to add another volume.
7. Select **Update Volume Group**.

Using the OCI CLI

Use the oci bv volume-group update command and required parameters to update a volume group.

```
oci bv volume-group update --volume-group-id <volume_group_OCID> --volume-
ids <volume_OCIDs_JSON> [OPTIONS]
```

For details about the JSON format, run this command:

```
oci bv volume-group update --generate-param-json-input volume-ids
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateVolumeGroup operation to update a volume group.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Removing Volumes from a Group

On Private Cloud Appliance, you can remove block and boot volumes from an existing volume group using the Compute Web UI, OCI CLI, and API.

When you remove the last volume in a volume group, the volume group is deleted.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Volume Groups**.
2. At the top of the page, select the compartment that contains the volume group that has the volume you want to remove.
3. In the **Volume Groups** list, select the volume group that contains the volume you plan to remove.
4. Under **Resources**, select **Volumes**.
5. For the volume that you want to remove, select the Actions menu (⋮), then select **Remove**.
6. Confirm the removal.

Using the OCI CLI

Use the oci bv volume-group update command and required parameters to update a volume group.

To remove a volume from a volume group, use the `oci bv volume-group update` command, and remove the volume from the list of volumes in the argument for the `--volume-ids` option as described in Adding Volumes to a Group on page 741.

```
oci bv volume-group update --volume-group-id <volume_group_OCID> --volume-
ids <volume_OCIDs_JSON> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateVolumeGroup operation to update a volume group:

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Cloning a Volume Group

On Private Cloud Appliance, you can create a clone of an existing volume group.

A cloned volume group is a point-in-time direct disk-to-disk deep copy of the source volume group, so all the data that's in the source volume group is copied to the clone volume group.

Any subsequent changes to the data on the source volume group are not copied to the clone.

For additional details about clones and how they differ from backups, see Volume Backups and Clones on page 714.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Volume Groups**.
2. At the top of the page, select the compartment that contains the volume group that you want to clone.
3. Select the name of the Volume Group you plan to clone.
4. Under **Resources**, select **Volume Group Clones**.
5. Select **Create Volume Group Clone**.
6. Enter the following information:
   - **Volume Group Clone Name:** Enter a descriptive name for the clone. Avoid entering confidential information.
   - **Create in Compartment:** Select the compartment where the clone will be created.
7. Select **Create Volume Group Clone**.

Using the OCI CLI

This task is not available in the OCI CLI.

Using the API

Use the CreateVolume operation and specify VolumeSourceFromVolumeDetails for CreateVolumeDetails.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Volume Group

On Private Cloud Appliance, you can delete a volume group using the Compute Web UI, OCI CLI, and API.

When you delete a volume group the individual volumes in the group are not deleted, only the volume group is deleted.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Volume Groups**.
2. At the top of the page, select the compartment that contains the volume group that you want to delete.
3. In the **Volume Groups** list, select the volume group you want to delete.
4. On the details page for the volume group, select **Terminate**.
5. Confirm the termination.

Using the OCI CLI

Use the oci bv volume-group delete command and required parameters to delete a volume group.

```
oci bv volume-group delete --volume-group-id <volume_group_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteVolumeGroup operation to delete a volume group.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Backing Up Block Volumes

On Private Cloud Appliance, the backup feature for the Block Volume service enables you make a point-in-time snapshot of the data on a block or boot volume. These backups can then be restored to new volumes any time.

On Private Cloud Appliance, all backups are full backups.

There are two ways to initiate a backup:

- **Manual Backups:**These are full backups that you can launch immediately. See Creating a Manual Boot or Block Volume Backup on page 745.
- **Policy-Based Backups:**These are automated scheduled backups defined by the backup policy assigned to the volume. See Managing Backup Policies on page 747.

For information to help you decide whether to create a backup or a clone of a boot volume, see Volume Backups and Clones on page 714.

Viewing Volume Backups

On Private Cloud Appliance, you can view volume backups using the Compute Web UI, CLI, and API.

Backups that are created by backup policies (also called automatic or scheduled backups) can take up to five minutes to show in the backups list.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Block Volumes**, **Boot Volumes**, or **Volume Groups**.

   Both automatic and manual backups are listed.
2. If you don't see your backup listed, ensure you're viewing the correct compartment, which is displayed at the top of the page.
3. To view the details of a backup, select the backup name.

Using the OCI CLI

Use the oci bv backup list command and required parameters to view the volume backups.

```
oci bv backup list --compartment-id <Compartment_OCID> [OPTIONS]
```

Use the oci bv backup get command and required parameters to view the details of a volume backup.

```
oci bv volume-group-backup list --compartment-id <Compartment_OCID>
```

When there are many volume backups, for example hundreds of backups, it can take a long time to list. The `backup list` commands support pagination of results. Use the `--limit` option to specify the number of backups to list. To show the next page of results, use the value of the `opc-next-page` property with the `--page` option. Specify the `--limit` option again for each new page command. Otherwise, all remaining resources are listed. When no `opc-next-page` property is shown, all results have been listed.

By default, the list is shown in ascending order by display name, and the display name is case sensitive. You can change the list order to try to move specific resources closer to the top of the list. You can sort by (`--sort-by`) `displayname` or `timecreated` and set the sort order (`--sort-order`) to either `asc` (ascending) or `desc` (descending). The default sort order for `timecreated` is descending.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Viewing Volume Backups Using the API

Use the ListVolumeBackups operation to view the volume backups.

Use the GetVolumeBackup command and required parameters to view the details of a volume backup.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Creating a Manual Boot or Block Volume Backup

On Private Cloud Appliance, you can create a block or boot volume backup manually using the Compute Web UI, OCI CLI, and API.

To create scheduled backups, see Managing Backup Policies on page 747

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select either **Block Volumes** or **Boot Volumes**.
2. At the top of the page, select the compartment that contains the volume that you want to back up.
3. For the volume you plan to back up, select the Actions menu (⋮), then select **Create Manual Backup**.
4. Enter the following information:
   - **Name:** Enter a descriptive name for the backup.
   - **Compartment:** For a boot volume backup, select the compartment where you want this backup created.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
5. For a block volume backup, select **Create Manual Backup**. For a boot volume backup, select **Create**.

Using the OCI CLI

Use the oci bv backup create command and required parameters to create a manual backup.

```
oci bv backup create --volume-id <block_volume_OCID> --display-
name <Backup_Name> --type FULL|INCREMENTAL [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateVolumeBackup operation to create a manual backup.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Creating a Manual Backup of a Volume Group

On Private Cloud Appliance, you can create a volume group backup manually using the Compute Web UI, OCI CLI, and API.

A volume group backup provides coordinated point-in-time-consistent backups of all the volumes in a volume group automatically. You can perform most of the same backup operations and tasks with volume groups that you can perform with individual block volumes and boot volumes.

You can restore a volume group backup to a volume group, or you can restore individual volumes in the volume group from volume backups. With volume group backups, you can manage the backup settings for several volumes in one place, consistently. This simplifies the process to create time-consistent backups of running enterprise applications that span multiple storage volumes across multiple instances.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Volume Groups**.
2. At the top of the page, select the compartment that contains the volume group that you want to back up.
3. For the volume group you want to back up, select the Actions menu (⋮), then select **Create Volume Group Backup**.
4. Enter a descriptive name for the backup. Avoid entering confidential information.
5. Select **Create Volume Group Backup**.

Using the OCI CLI

Use the oci bv volume-group-backup create command and required parameters to back up a volume group.

```
oci bv volume-group-backup create --volume-group-id <volume_group_OCID> --
compartment-id <compartment_OCID> --display-name <display_name> --type FULL
 [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateVolumeGroupBackup operation to back up a volume group.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Restoring a Backup to a New Volume

On Private Cloud Appliance, you can restore a volume backup to a new volume using the Compute Web UI, CLI, and API.

You can restore a volume from any of your incremental or full volume backups. Both backup types enable you to restore the full volume contents to the point-in-time snapshot of the volume when the backup was taken. You don't need to keep the initial full backup or subsequent incremental backups in the backup chain and restore them in sequence, you only need to keep the backups taken for the times you care about.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Block Volume Backups**.
2. At the top of the page, select the compartment that contains the block volume backup that you want to restore.
3. For the block volume backup that you want to restore, select the Actions menu (⋮), then select **Restore Block Volume**.
4. Enter the following information:
   - **Name:** A name or description for the volume. Avoid entering confidential information.
   - **Compartment:** Select the compartment in which to restore the block volume.
   - **Size (in GBs):** To change the size, enter a value from 50 to 32768 (50 GB to 32 TB) in 1 GB increments. You cannot enter a smaller value than the value that is shown.
   - **High Performance Volume:** (Optional) By default, the volume uses balanced performance. To create a block volume that uses the high performance feature, select the Enable High Performance button.

     This selection can't be changed after the volume is created.
   - **Backup Policy:** (Optional) Select a backup policy from the drop-down list. You might need to change the compartment.

     For more information about backup policies, see Managing Backup Policies on page 747.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
5. Select **Create Block Volume**.

   The volume is ready to attach after its icon no longer lists it as PROVISIONING in the volume list.

Using the OCI CLI

Use the oci bv volume create command and required parameters to restore a backup to a new volume.

```
oci bv volume create --availability-domain  <availability_domain_name> --
volume-backuup-id <source_volume_backup_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateVolume operation to restore a backup to a new volume. The API has an optional `volumeBackupId` parameter that you can use to define the backup from which the data should be restored on the newly created volume.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Restoring a Volume Group from a Volume Group Backup

On Private Cloud Appliance, you can restore a volume group from a volume group backup using the Compute Web UI, CLI, and API.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Volume Group Backups**.
2. At the top of the page, select the compartment that contains the volume group backup that you want to restore.
3. For the volume group backup that you want to restore, select the Actions menu (⋮), then select **Restore Volume Group**.
4. Enter the following information:
   - **Name**: Enter a descriptive name for the group.
   - **Compartment**: Select the compartment for the volume group.
   - **Backup Policy**: (Optional) Select a backup policy from the drop-down list. You might need to change the compartment.

     For more information about backup policies, see Managing Backup Policies on page 747.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
5. Select **Create Volume Group**.

Using the OCI CLI

Use the oci bv volume-group create command and required parameters to create a new volume group from a volume group backup.

```
oci bv volume-group create --availability-domain <availability_domain_name>
 --compartment-id  <compartment_OCID>--source-details <json_string> or
 file://<path_to_JSON_file> [OPTIONS]
```

Using the API

Use the CreateVolumeGroupBackup operation to create a new volume group from a volume group backup.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing Backup Policies

On Private Cloud Appliance, you can perform volume backups and volume group backups automatically using a schedule, and retain them based on the retention setting in the backup policy.

The Block Volume service enables you to perform volume backups and volume group backups automatically according to a schedule that's defined in a backup policy. The backup policy also specifies how long to retain the backup.

You can use one of the backup policies defined by Oracle and available in every compartment, or you can create your own user-defined backup policy as described in Creating a Backup Policy on page 748. You can't modify policies that are provided by Oracle. You can't use an Oracle defined backup policy to back up a volume group.

On Private Cloud Appliance, all backups are full backups.

A particular resource can't be assigned more than one backup policy. A particular policy can have more than one backup schedule.

Schedule Notes

These notes apply to both user-defined and Oracle-defined backup policies.

- Start time: A backup might not start at its scheduled start time. A backup can be delayed for hours if the system is very busy.
- Conflicts: The Block Volume service will not run more than one scheduled backup of a particular resource in one day. If more than one backup is scheduled to run on the same day (for example, daily, weekly, and monthly backups are all scheduled to run this Sunday), the Block Volume service will run the backup that has the longest schedule period.

> **Caution:**
>
> All policy-based (scheduled) backups expire. A manual backup expires if a scheduled backup of the same volume is created after the manual backup was created. To keep a volume backup indefinitely, cancel all future scheduled backups and create a manual backup as described in Backing Up Block Volumes on page 744.

Oracle Provided Backup Policies

Oracle defined backup policies can be used to back up individual block volumes and boot volumes. These policies can't be used to back up a volume group.

Oracle defined policies can't be changed. If you need a different backup time or retention time, for example, create a user defined backup policy as described in Creating a Backup Policy on page 748.

The following backup policies are provided by Oracle and available in every compartment. A resource can have only one backup policy assigned, but a backup policy can have multiple schedules. The Oracle Bronze policy has one schedule, the Silver policy has two schedules, and the Gold policy has three schedules. All schedule times are your regional data center time. All backup types are full backups.

Bronze

1. Monthly backups that run at 00:00 on the first day of the month and are retained for twelve months.

Silver

1. Weekly backups that run at 00:00 every Monday and are retained for four weeks.
2. Monthly backups that run at 00:00 on the first day of the month and are retained for twelve months.

Gold

1. Daily backups that run at 00:00 and are retained for seven days.
2. Weekly backups that run at 00:00 every Monday and are retained for four weeks.
3. Monthly backups that run at 00:00 on the first day of the month and are retained for twelve months.

The Block Volume service will not run more than one scheduled backup of a particular resource in one day. When schedules conflict, for example, daily, weekly, and monthly backups are scheduled to run at the same time, the backup with the longest period runs.

Creating a Backup Policy

On Private Cloud Appliance, you can use an Oracle defined backup policy, or you can follow the procedures described in this section to create your own backup policy.

For details about Oracle defined backup policies, see Viewing Backup Policies on page 752.

Using the Compute Web UI

When you create a backup policy by using the Compute Web UI, creating the backup policy schedule is a separate step. A policy can't be assigned to a resource until the policy schedule is defined.

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Backup Policies**.
2. Select **Create Backup Policy**.
3. Enter the following information:
   - **Name:** Enter a descriptive name for the policy.
   - **Create in Compartment:** Select the compartment for the policy.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
4. Select **Create Backup Policy**.

   A policy can't be assigned to a resource until at least one schedule is defined.
5. Select the name of the new backup policy to go to the details page for the policy.
6. On the policy details page, under Resources, select **Add Schedule**.
7. Enter the following schedule parameters:
   - **Schedule Type:** Select **Daily**, **Weekly**, or **Monthly**, and then specify the time for the backup to run and the length of time to retain the backup.
     - **Hourly:** Select the Retention Time In Hours (1 - 24).
     - **Daily:** Select the Hour of the Day (0 - 23) and the Retention Time In Days (1 - 365).
     - **Weekly:** Select the Day of the Week and the Hour of the Day, and select the Retention Time In Weeks (1 - 52).
     - **Monthly:** Select the Day of the Month (1 - 31) and the Hour of the Day, and select the Retention Time In Months (1 - 144).
   - **Time Zone:** Select either UTC or your Regional Time.
8. Select **Add Schedule**.

   The policy can now be assigned to one or more resources. See Assigning a Backup Policy to a Volume or Volume Group on page 750.

   A policy can have more than one schedule. To add another schedule to this policy, select **Add Schedule** again. See the schedule notes in Managing Backup Policies on page 747.

Using the OCI CLI

Use the oci bv volume-backup-policy create command and required parameters to create a new user defined backup policy.

```
oci bv volume-backup-policy create --compartment-id <compartment_OCID>
 --display-name <discriptive_name> --schedule <json_string> or
 file://<path_to_JSON_file> [OPTIONS]
```

Once the policy is created, it can be applied to one or more volumes and volume groups. See Assigning a Backup Policy to a Volume or Volume Group on page 750.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateVolumeBackupPolicy operation to create a new user defined backup policy.

Once the policy is created, it can be applied to one or more volumes and volume groups. See Assigning a Backup Policy to a Volume or Volume Group on page 750.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Assigning a Backup Policy to a Volume or Volume Group

On Private Cloud Appliance, after creating a backup policy, you assign the policy to one or more volumes or volume groups.

You can assign a backup policy to a volume or volume group at resource creation or later:

- During volume or volume group creation. Select from the Backup Policy list in the Compute Web UI or specify the `--backup-policy-id` option in the OCI CLI.
- After the volume or volume group is created. Follow the procedures described in this section to add a backup policy if no backup policy is assigned, or to change the backup policy that is assigned to a volume or volume group.

A volume or volume group can have only one backup policy assigned. If a backup policy is already assigned to this volume or volume group when you assign a new policy, the existing assignment is replaced with the new assignment.

A volume group cannot be assigned a backup policy if any of the volumes in the group already is assigned a backup policy. To remove a backup policy assignment from a volume, see Removing a Backup Policy Assignment on page 750.

> **Note:**
>
> Oracle defined backup policies cannot be used for volume group backups. Only user defined backup policies can be assigned to a volume group.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Block Volumes**, **Boot Volumes**, or **Volume Groups**.
2. At the top of the page, select the compartment that contains the volume or volume group to which you want to assign a backup policy.
3. For the volume or volume group to which you want to assign a backup policy, select the Actions menu (⋮), then select **Assign Backup Policy**.
4. In the **Assign Backup Policy dialog**, select a backup policy from the drop-down list.

   You can't assign an Oracle defined policy to a volume group.
5. Select **Assign Backup Policy**.

Using the OCI CLI

Use the oci bv volume-backup-policy-assignment create command and required parameters to assign a volume backup policy to the specified volume.

```
oci bv volume-backup-policy-assignment create --asset-id <volume_OCID> --
policy-id <backup-policyABC> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateVolumeBackupPolicy operation to assign a volume backup policy to the specified volume.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Removing a Backup Policy Assignment

On Private Cloud Appliance, you can remove a backup policy assignment from a volume or volume group. A volume group cannot be assigned a backup policy if any of the volumes in the group already is assigned a backup policy.

Using the OCI CLI

1. Use the appropriate `list` command to get the OCID of the volume or volume group from which you want to remove the backup policy assignment. For example: `oci bv volume list`.

2. Use the volume or volume group OCID from the preceding step as the argument of the `--asset-id` option in the following command to get the backup policy assignment OCID.

```
$ oci bv volume-backup-policy-assignment get-volume-backup-policy-asset-
assignment \
--asset-id ocid1.volume.unique_ID
{
  "data": [
    {
      "asset-id": "ocid1.volume.unique_ID",
      "id": "ocid1.backuppolicyassignment.unique_ID",
      "policy-id": "ocid1.volumebackuppolicy.unique_ID",
      "time-created": "2023-06-07T02:03:53.466062+00:00"
    }
  ]
}
```

3. Use the backup policy assignment OCID from the preceding step to delete this assignment so that this resource has no backup policy assigned.

```
$ oci bv volume-backup-policy-assignment delete \
--policy-assignment-id
 ocid1.backuppolicyassignment.AK00661530.scasg01.unique_ID \
--force
{
  "etag": "7a0ca7dd-50f7-4d60-9689-aaa442ac4348",
  "opc-work-request-id": "ocid1.workrequest.unique_ID"
}
```

The following command shows that this asset has no backup policy assigned.

```
$ oci bv volume-backup-policy-assignment get-volume-backup-policy-asset-
assignment \
--asset-id ocid1.volume.unique_ID
$
```

Accessing the Backups

On Private Cloud Appliance, after a manual or automatic backup takes place, you can use the topics in this section to locate the backup.

> **Note:**
>
> For scheduled or policy-based backups, there might be a delay of up to 5 minutes for backups to show up in the list. This delay applies to viewing backups in the Compute Web UI, OCI CLI, and API.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select one of the following items:
   - **Block Volume Backups**
   - **Volume Group Backups**

   The manual and automatic backups are displayed.
2. To see the details of a backup, select the backup name.

Using the OCI CLI

Use the oci bv backup list command and required parameters to list the volume backups in the specified compartment.

```
oci bv backup list --compartment-id <compartment_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListVolumeBackups operation to list the volume backups in the specified compartment.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing Backup Policies

On Private Cloud Appliance, after creating a backup policy, you can use the Compute Web UI, OCI CLI, or API to view the policy.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Backup Policies**.

   At a minimum, the Oracle defined policies are listed. If the compartment has any user defined policies, then both Oracle defined and user defined policies are listed.
2. To view a particular user defined policy, you might need to select a different compartment from the top of the page.
3. To see details about a policy, select the policy name. Scroll to the **Resources** section to see the schedules.

Using the OCI CLI

Use the oci bv volume-backup-policy list command and required parameters to list all the volume backup policies available in the specified compartment.

```
oci bv volume-backup-policy list --compartment-id <compartment_OCID>
 [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListVolumeBackupPolicies operation to list all the volume backup policies available in the specified compartment.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Editing a Backup Policy Schedule

On Private Cloud Appliance, you cannot edit an Oracle defined backup policy. You can only edit user defined backup policies.

Avoid entering confidential information.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Backup Policies**.
2. At the top of the page, select the compartment that contains the backup policy.
3. Select the name of the backup policy that has the schedule you want to edit.
4. On the backup policy details page, scroll to the **Resources** section.
5. For the schedule that you want to edit in the **Schedules** list, select the Actions menu (⋮), then select **Edit**.
6. Make your changes. See Creating a Backup Policy on page 748 for details.
7. Select **Update Schedule**.

Using the OCI CLI

Use the oci bv volume-backup-policy update command and required parameters to update a user defined backup policy.

> **Caution:**
>
> Running this command replaces all schedule items with the items you specify in the --schedule JSON. Include all the schedule parameters, not just the schedule parameter changes.

```
oci bv volume-backup-policy update --policy-id <backup_policy_OCID> --
schedule <json_string> or file://<path_to_JSON_file> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateVolumeBackupPolicy operation to update a user defined backup policy.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Backup Policy Schedule

On Private Cloud Appliance, you cannot change an Oracle defined backup policy. You can only change or delete the schedules of user defined backup policies.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Backup Policies**.
2. At the top of the page, select the compartment that contains the backup policy that has the schedule you want to delete.
3. Select the name of the backup policy that has the schedule you want to delete.
4. On the **Backup Policy** page, scroll to the **Resources** section.
5. For the schedule you want to delete in the **Schedules** list, select the Actions menu (⋮), then select **Delete**.
6. Confirm the deletion.

Using the OCI CLI

Use the oci bv volume-backup-policy delete command and required parameters to delete a backup policy.

```
oci bv volume-backup-policy delete \
--policy-id ocid1.volumebackuppolicy.unique_ID --force
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

This task is not available in the API.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Backup Policy

On Private Cloud Appliance, you cannot delete an Oracle defined backup policy. You can only delete user defined backup policies.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Block Storage**, then select **Backup Policies**.
2. At the top of the page, select the compartment that contains the backup policy.
3. For the policy you want to delete, select the Actions menu (⋮), then select **Delete**.
4. Confirm the deletion.

Using the OCI CLI

Use the oci bv volume-backup-policy delete command and required parameters to delete a user defined backup policy.

```
oci bv volume-backup-policy delete --policy-id <policy_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteVolumeBackupPolicy operation to delete a user defined backup policy.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

File Storage

On Private Cloud Appliance, the File Storage service provides a durable, scalable, secure network file system. You can connect to a File Storage service file system from any Private Cloud Appliance compute instance in your Virtual Cloud Network (VCN).

File Storage Overview

The File Storage service supports these protocols:

- Network File System version 4.1 (NFSv4.1)
- Network File System version 4.0 (NFSv4)
- Network File System version 3.0 (NFSv3)
- Server Message Block (SMBv2 - SMBv3.1) – Requires Active Directory

File Storage Connectivity

You can connect to a File Storage service file system from any instance in your Virtual Cloud Network (VCN).

Suitable Workloads

The File Storage service is designed to meet the needs of applications and users that need an enterprise file system across a wide range of use cases, including the following:

- **General Purpose File Storage:** Access to a pool of file systems to manage growth of structured and unstructured data.
- **Big Data and Analytics:** Run analytic workloads and use shared file systems to store persistent data.
- **Lift and Shift of Enterprise Applications:** Migrate existing applications that need NFS storage, such as Oracle Cloud Infrastructure E-Business Suite and PeopleSoft.
- **Databases and Transactional Applications:** Run test and development workloads with Oracle Cloud Infrastructure, MySQL, or other databases.
- **Backups, Business Continuity, and Disaster Recovery:** Host a secondary copy of relevant file systems from on premises to the cloud for backup and disaster recovery purposes.
- **Portable OS Interface (POSIX)-compliant file system**
- **MicroServices and Docker:** Deliver stateful persistence for containers. Easily scale as your container-based environments grow.

Data Protection with Snapshots

The File Storage service supports snapshots for data protection of your file system. Snapshots are a consistent, point-in-time view of your file systems. Snapshots are copy-on-write, and scoped to the entire file system. The File Storage service encrypts all file system and snapshot data at rest. You can take as many snapshots as you need.

For data protection, you can use a tool that supports NFS to copy your data to a different file system, object storage, or remote location.

For best performance, use the parallel tar (`partar`) and parallel copy (`parcp`) tools provided in the File Storage Parallel File Toolkit for this purpose. These tools work best with parallel workloads and requests. The Parallel File Toolkit is available for Oracle Linux, Red Hat Enterprise Linux, and CentOS. You can use `rsync` or regular `tar` for other OS types. See Installing the Parallel File Tools for more information.

File Storage Objects

Mount Target

A mount target is an NFS endpoint in a subnet of your choice. The mount target provides the IP address that's used in the mount command when connecting NFS clients to a file system. File systems are exported (made available) through mount targets.

For an instance to mount a file system, the instance's Virtual Cloud Network (VCN) must have a mount target. A VCN can only have one mount target.

You can reuse the same mount target to make many file systems available. To reuse the same mount target for multiple file systems, create an export in the mount target for each file system.

Export

Exports control how NFS clients access file systems when they connect to a mount target. File systems are exported (made available) through mount targets. Each mount target maintains an export set which contains one or many exports. A file system must have at least one export in one mount target for compute instances to mount the file system.

Export Set

An export set is a collection of one or more exports that control what file systems the mount target exports and how those file systems are found using the NFS mount protocol. Each mount target has an export set. Each file system associated with the mount target has at least one export in the export set.

Export Path

The export path uniquely identifies the file system within the mount target. The export path is used by a compute instance to mount (logically attach to) the file system. For more information, see File Storage Paths on page 756.

Export Options

NFS export options are a set of parameters within the export that specify the level of access granted to NFS clients when they connect to a mount target. An NFS export options entry within an export defines access for a single IP address or CIDR block range.

File System

In Private Cloud Appliance, file system refers to a file system that's accessed by one or more clients over the network. File systems are associated with a single compartment. File systems must have at least one export in one mount target for any client to mount and use the file system. Data is added to a file system from the client that has mounted (has access to) the file system.

Data is added to a file system from the client that has mounted (has access to) the file system. A file system can have a quota of up to 8 petabytes.

The total number of file systems is limited to 100 per tenancy.

Virtual Cloud Network (VCN)

A private network that you set up in Private Cloud Appliance, with firewall rules and specific types of communication gateways that you can choose to use. A VCN covers a single, contiguous IPv4 CIDR block of your choice.

Subnet

Subnets are subdivisions you define in a VCN (for example, 10.0.0.0/24 and 10.0.1.0/24). Subnets contain virtual network interface cards (VNICs), which attach to compute instances. A subnet consists of a contiguous range of IP addresses that don't overlap with other subnets in the VCN.

Security Rules

Security rules are virtual firewall rules for your VCN. Your VCN comes with a default security list, and you can add more. These security lists provide ingress and egress rules that specify the types of traffic allowed in and out of the compute instances. You can choose whether a particular rule is stateful or stateless. Security list rules must be set up so that clients can connect to file system mount targets.

Another method for applying security rules is to set them up in a network security group (NSG), and then add the mount target to the NSG. Unlike security list rules that apply to all VNICs in the subnet, NSGs apply only to resource VNICs you add to the NSG.

Snapshots

Snapshots provide a consistent, point-in-time view of your file system, and you can take as many snapshots as you need. Each snapshot reflects only data that changed from the previous snapshot.

File Storage Paths

The File Storage service uses these kinds of paths:

- **Export Paths** are part of the information contained in an export that makes a file system available through a mount target.

  The export path is automatically generated when you create an export, and it uniquely identifies the file system within the mount target.

  Note – When you create an export from the OCI CLI, you must specify a `--path` <*path*> argument. The path you specify is recorded but not used for mounting file systems. Private Cloud Appliance auto-generates a path that's used to mount the file system.

  Export path syntax:

  ```
  /export/<file-system-OCID-unique-string>
  ```

  where:

  - /export/ – Is the beginning of the export path.
  - <*file-system-OCID-unique-string*> – Is the unique character string portion of the file system's OCID.

  For example, a file system with this OCID . . .

  ```
  ocid1.filesystem.oc1.pca.d0v812zdp48onybubehhx1c67i4p3mjfth5avt3z2rkn50uqpbce3fhsa8nm
  ```

  . . . has an export path that looks like this:

  ```
  /export/d0v812zdp48onybubehhx1c67i4p3mjfth5avt3z2rkn50uqpbce3fhsa8nm
  ```

  The export path is used by a file system client to mount (logically attach to) the file system. This path is unrelated to any path within the file system or the client instance. It exists solely as a way to distinguish one file system from another within a single mount target.

  Example of an export path in a client's mount command:

  ```
  sudo mount -t nfs  \
   -o nfsvers=4.0 192.0.2.0:/export/
  d0v812zdp48onybubehhx1c67i4p3mjfth5avt3z2rkn50uqpbce3fhsa8nm /mnt/fs
  ```

  In this mount command example, `192.0.2.0` is the mount target IP address.`/export/d0v812zdp48onybubehhx1c67i4p3mjfth5avt3z2rkn50uqpbce3fhsa8nm` is the unique export path that was specified when the file system was associated with a mount target during creation.

  Export paths can't be edited after the export is created.

- **Mount Point Paths** are paths within a client instance to a locally accessible directory to which the remote file system is mounted.

  In this mount command example, `/mnt/fs` is the path to the directory on the client instance on which the external file system is mounted.

  ```
  sudo mount -t nfs  \
   -o nfsvers=4.0 192.0.2.0:/export/
  d0v812zdp48onybubehhx1c67i4p3mjfth5avt3z2rkn50uqpbce3fhsa8nm /mnt/fs
  ```

- **File System Paths** are paths to directories within the file system, and contain the contents of the file system. When the file system is mounted, you can create any directory structure within it.

VCN Security Rules for File Storage

On Private Cloud Appliance, before you can mount a file system, you must configure security rules to allow traffic to the mount target's VNIC using specific protocols and ports.

Security rules enable traffic for the following:

- Open Network Computing Remote Procedure Call (ONC RPC) rpcbind utility protocol
- Network File System (NFS) protocol
- Network File System (MOUNT) protocol
- Network Lock Manager (NLM) protocol

Ways to Enable Security Rules for File Storage

The Networking service offers two virtual firewall features that both use security rules to control traffic at the packet level. The two features are:

- **Security lists:** The original virtual firewall feature from the Networking service. When you create a VCN, a default security list is also created. Add the required rules to the security list for the subnet that contains the mount target.
- **Network security groups (NSGs):** A subsequent feature designed for application components that have different security postures. Create an NSG that contains the required rules, and then add the mount target to the NSG. Each mount target can belong to up to five (5) NSGs.

> **Important:**
>
> You can use security lists alone, network security groups alone, or both together. It depends on your particular security needs.
>
> If you choose to use both security lists and network security groups, the set of rules that applies to a given mount target VNIC is the combination of these items:
>
> - The security rules in the security lists associated with the VNIC subnet
> - The security rules in all NSGs that the VNIC is in
>
> It doesn't matter which method you use to apply security rules to the mount target VNIC, as long as the ports for protocols necessary for File Storage are correctly configured in the rules applied.

For additional conceptual information, see Virtual Firewall on page 485.

For instructions on how to create security rules and NSGs for the File Storage service, see Controlling Access to File Storage on page 771.

For general instructions for creating security lists and NSGs, see Controlling Traffic with Security Lists on page 496.

File Storage Network Ports

On Private Cloud Appliance, when configuring the File Storage service, you might need to know the ports that are used for a service or protocol.

This table lists the port numbers used for networking services that support File Storage.

When you configure security lists and network security groups for the File Storage service, use the port numbers for the particular service you are using. For example, for NFS configure ports 2048–2050 for TCP and UDP.

Service	Protocol	Port
RPC	TCP, UDP	111
NFS	TCP, UDP	2048, 2049, 2050
lockd	TCP, UDP	4045
mountd	TCP, UDP	20048
SMB	TCP, UDP	445
LDAP	TCP, UDP	389
Kerberos	TCP, UDP	88
DNS	TCP, UDP	53

Deleting File System Resources Overview

On Private Cloud Appliance, you can't delete a file system resource that has dependencies.

File Systems

You can delete a file system if it's not a parent file system. If it's a parent file system, all descendant snapshots and clones must first be deleted.

Snapshots

A parent snapshot can't be deleted. A snapshot that isn't a parent can be deleted.

Clones

A parent clone can't be deleted. A clone that isn't a parent can be deleted.

Export Options for File Storage

On Private Cloud Appliance, NFS export options enable you to create more granular access control than is possible using security list rules to limit VCN access. You can use NFS export options to specify access levels for IP addresses or CIDR blocks connecting to file systems through exports in a mount target. Access can be restricted so that each client’s file system is inaccessible and invisible to the other, providing better security controls in multitenant environments.

Using NFS export option access controls, you can limit clients' ability to connect to the file system and view or write data. For example, if you want to allow clients to consume but not update resources in your file system, you can set access to read-only. You can also reduce client root access to your file systems and map specified User IDs (UIDs) and Group IDs (GIDs) to a single anonymous UID/GID of your choice.

Export Options

Exports control how NFS clients access file systems when they connect to a mount target. File systems are exported (made available) through mount targets.

NFS export options are a set of parameters within the export that specify the level of access granted to NFS clients when they connect to a mount target. An NFS export options entry within an export defines access for a single IP address or CIDR block range. You can have up to 100 options per file system.

Each separate client IP address or CIDR block you want to define access for needs a separate export options entry in the export. For example, if you want to set options for NFS client IP addresses 10.0.0.6, 10.0.0.08, and 10.0.0.10, you need to create three separate entries, one for each IP address.

When there is more than one export using the same file system and same mount target, the export options that are applied to an instance are the options with a source that most closely matches the instance IP address. The smallest (most specific) match takes precedence across all the exports. Therefore, you can determine which export options are applied to an instance by looking at source value of all the exports.

For example, consider the following two export options entries specifying access for an export:

Entry 1: Source: 10.0.0.8/32, Access: Read/Write

Entry 2: Source: 10.0.0.0/16, Access: Read-only

In this case, clients who connect to the export from IP address 10.0.0.8 have read/write access. When there are multiple export options, the most specific match is applied.

> **Important:**
>
> When more than one file system is exported to the same mount target, you must export to the mount target with the smallest network (largest CIDR number) first. For detailed information and instructions, refer to My Oracle Support Doc ID 2823994.1.

> **Important:**
>
> File systems can be associated with one or more exports, contained within one or more mount targets.
>
> If the client source IP address does not match any entry on the list for a single export, then that export is not visible to the client. However, the file system could be accessed through other exports on the same or other mount targets. To completely deny client access to a file system, be sure that the client source IP address or CIDR block is not included in any export for any mount target associated with the file system.

For information about how to configure export options for various file sharing scenarios, see NFS Access Control Scenarios on page 760.

For more information about configuring export options, refer to the section titled Setting NFS Export Options on page 772.

NFS Export Option Defaults

When you create a file system and export, the NFS export options for that file system are set to the following defaults, which allow full access for all NFS client source connections. These defaults must be changed if you want to restrict access:

Note – When using the OCI CLI to set the export options, if you set the options with an empty array (no options specified), the export is not accessible to any clients.

Export Option in the Compute Web UI	Export Option in the OCI CLI	Default Value	Description
Source:	`source`	0.0.0.0/0	The IP address or CIDR block of a connecting NFS client.

Export Option in the Compute Web UI	Export Option in the OCI CLI	Default Value	Description
Ports:	`require-privileged-source-port`	Any	Always set to: • Compute Web UI: Any • OCI CLI: `false`
Access:	`access`	Read/Write	Specifies the source NFS client access. Can be set to one of these values: • `READ_WRITE` • `READ_ONLY`
Squash:	`identity-squash`	None	Determines whether the clients accessing the file system as root have their User ID (UID) and Group ID (GID) remapped to the squash UID/GID. These are the possible values: • Root – Only the root user is remapped. • None – No users are remapped.
Squash UID/GID:	`anonymous-uid`and `anonymous-gid`	65534	This setting is used along with the Squash option. When remapping a root user, you can use this setting to change the default anonymousUid and anonymousGid to any user ID of your choice.

NFS Access Control Scenarios

On Private Cloud Appliance, Learn different ways to control NFS access by reviewing some scenarios.

- Scenario A: Control Host Based Access on page 760: Provides a managed environment for two clients. The clients share a mount target, but each has their own file system, and can't access each others data.
- Scenario B: Limit the Ability to Write Data on page 761: Provides data to clients for consumption, but doesn't allow them to update the data.
- Scenario C: Improve File System Security on page 762: Increases security by limiting the root user's privileges when connecting to a file system.

Scenario A: Control Host Based Access

On Private Cloud Appliance, provide a managed hosted environment for two clients. The clients share a mount target, but each has their own file system, and can't access each other's data.

For example:

- Client A is assigned to CIDR block 10.0.0.0/24, and requires read/write access to file system A, but not file system B.
- Client B is assigned to CIDR block 10.1.1.0/24, and requires read/write access to file system B, but not file system A.
- Client C is assigned to CIDR block 10.2.2.0/24, and has no access of any kind to file system A or file system B.

- Both file systems A and B are associated with a single mount target, MT1. Each file system has an export contained in the export set of MT1.

Because Client A and Client B access the mount target from different CIDR blocks, you can set the client options for both file system exports to allow access to only a single CIDR block. Client C is denied access by not including its IP address or CIDR block in the NFS export options for any export of either file system.

Compute Web UI Example

Set the export options for file system A to allow read/write access only to Client A, who is assigned to CIDR block 10.0.0.0/24. Client B and Client C aren't included in this CIDR block, and can't access the file system.

> **Note:**
>
> To learn how to access the NFS export options in the Compute Web UI see Setting NFS Export Options on page 772.

Source	Ports	Access	Squash	Squash UID/GID
10.0.0.0/24	Any	Read/Write	None	(not used)

Set the export options for file system B to allow read/write access only to Client B, who is assigned to CIDR block 10.1.1.0/24. Client A and Client C aren't included in this CIDR block, and can't access the file system.

Source	Ports	Access	Squash	Squash UID/GID
10.1.1.0/24	Any	Read/Write	None	(not used)

OCI CLI Example

> **Note:**
>
> To learn how to access the NFS export options in the OCI CLI, see Setting NFS Export Options on page 772.

Set the export options for file system A to allow `Read_Write` access only to Client A, who is assigned to CIDR block 10.0.0.0/24. Client B and Client C aren't included in this CIDR block, and can't access the file system.

```
oci fs export update --export-id <File_system_A_export_ID> --export-options
 \
'[{"source":"10.0.0.0/24","require-privileged-source-
port":"false","access":"READ_WRITE","identity-squash":"NONE","anonymous-
uid":"65534","anonymous-gid":"65534"}]'
```

Set the export options for file system B to allow `Read_Write` access only to Client B, who is assigned to CIDR block 10.1.1.0/24. Client A and Client C are not included in this CIDR block, and can't access the file system.

```
oci fs export update --export-id <File_system_B_export_ID> --export-options
 \
'[{"source":"10.1.1.0/24 ","require-privileged-source-
port":"false","access":"READ_WRITE","identity-squash":"NONE","anonymous-
uid":"65534","anonymous-gid":"65534"}]'
```

Scenario B: Limit the Ability to Write Data

On Private Cloud Appliance, provide data to customers for consumption, but don't allow them to update the data.

For example, you'd like to publish a set of resources in file system A for an application to consume, but not change. The application connects from IP address 10.0.0.8.

Compute Web UI Example

Set the source IP address 10.0.0.8 to read-only in the export for file system A.

> **Note:**
>
> To learn how to access the NFS export options in the Compute Web UI see Setting NFS Export Options on page 772.

Source	Ports	Access	Squash	Squash UID/GID
10.0.0.8	Any	Read-only	None	(not used)

OCI CLI Example

> **Note:**
>
> To learn how to access the NFS export options in the OCI CLI, see Setting NFS Export Options on page 772.

Set the source IP address 10.0.0.8 to `READ_ONLY` in the export for file system A.

```
oci fs export update --export-id <File_System_A_export_OCID> --export-
options \
'[{"source":"10.0.0.8","require-privileged-source-
port":"false","access":"READ_ONLY","identitysquash":"NONE","anonymousuid":"65534","anony
```

Scenario C: Improve File System Security

On Private Cloud Appliance, to increase security, you can limit the root user's privileges when connecting to File System A. Use Identity Squash to remap root users to UID/GID 65534.

In UNIX-like systems, this UID/GID combination is reserved for '*nobody*', a user with no system privileges.

Compute Web UI Example

Set the source IP address 10.0.0.8 to read-only in the export for file system A.

> **Note:**
>
> To learn how to access the NFS export options in the Compute Web UI see Setting NFS Export Options on page 772.

Source	Ports	Access	Squash	Squash UID/GID
0.0.0.0/0	Any	Read/Write	Root	65534

OCI CLI Example

> **Note:**
>
> To learn how to access the NFS export options in the OCI CLI, see Setting NFS Export Options on page 772.

```
oci fs export update --export-id <File_System_A_export_OCID> --export-
options   \
'[{"source":"0.0.0.0/0","require-privileged-source-
port":"false","access":"READ_WRITE","identitysquash":"ROOT","anonymousuid":"65534","anon
```

Creating a File System, Mount Target, and Export

On Private Cloud Appliance, you can use this task flow to perform all the tasks that are required to create a file system and make it available for instances.

Task Flow

No.	Description	Links to Procedures
1.	Ensure that a mount target exists for the subnet that the instance where you want to mount a file system will use and the backing store pool that the file system will use. **Note** The file system and mount target must be in the same compartment and the same backing store pool when you create an export.	Creating a Mount Target on page 763
2.	Create the file system.	Creating a File System on page 766
3.	Create a file system export in the mount target.	Creating an Export for a File System on page 769
4.	Enable Security Rules for File Storage.	Controlling Access to File Storage on page 771
5.	Change NFS export options to control access to the file system.	Setting NFS Export Options on page 772

After the file system is exported, on the NFS client, perform these tasks to mount the file system:

1. (If needed) Install NFS client software.
2. Create a mount point.
3. On the client, mount the file system to the mount point.
4. On the client, add whatever files, directories, and data that you want in the file system.

For more information about mounting file systems, see Mounting File Systems on UNIX-based Instances on page 774.

Creating a Mount Target

On Private Cloud Appliance, A mount target is an NFS endpoint assigned to a subnet of your choice. The mount target provides the IP address or DNS name that's used in the mount command when connecting NFS clients to a file system.

You can create at most two mount targets per VCN: at most one mount target per pool type. Pool type refers to the backing store pool for the file system, which can be either the default pool of the attached ZFS Storage Appliance or a high performance pool. See the *Backing store pool* in Creating a File System on page 766. Two mount targets in a VCN are counted as one with regard to resource limits.

You can reuse a mount target to make many file systems available on the network. To reuse the same mount target for multiple file systems, create an export in the mount target for each file system. The file system and mount target must be in the same compartment and the same backing store pool when you create an export.

> **Caution:**
>
> Don't use /30 or smaller subnets for mount target creation because they might not have enough available IP addresses.

> **Important:**

> When exporting file systems to overlapping CIDRs in a VCN, exports to the longest CIDR (smallest network) must be done first. For more information and an example, see My Oracle Support article PCA File system as a Service Exports (Doc ID 2823994.1).

Before you can create a mount target, ensure that these items are configured:

- At least one Virtual Cloud Network (VCN) and subnet is configured. See Managing VCNs and Subnets on page 478
- (Required for Mounting File Systems Across Private Cloud Appliance Appliances on page 770) A Dynamic Routing Gateway (DRG) with a route rule in the VCN. See Connecting to the On-Premises Network through a Dynamic Routing Gateway (DRG) on page 511.
- (Optional) Security rules for the file system mount target. Security rules can be created in the security list for the mount target subnet, or in a Network Security Group (NSG) that you add the mount target to. See Controlling Access to File Storage on page 771.

  You don't need security rules to create a mount target, but you do need security rules to eventually mount file systems that are associated with this mount target.
- To ensure that the mount target is created using high performance storage, first configure the OraclePCA tag namespace and backing store pool (`poolName`) attribute in your OCI tenancy. See Using Resource Tags to Extend Functionality (IAM on the Appliance) on page 460. Then when you create the mount target, assign the OraclePCA tag namesapce and `poolName` attribute to the mount target. The poolName property can be set only when the mount target is created. You cannot set or change this property value after the mount target is created.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **File Storage**, then select **Mount Targets**.
2. In the compartment drop-down menu above the mount targets list, select the compartment where you plan to create the file system.

   If a mount target is listed, select the name of the mount target to open the details page and check the following parameters:

   - The mount target must be on the same subnet as the instance where you want to mount the file system.
   - Select the **Select Tag(s)** tab. The mount target must be in the same backing store pool that's specified for the file system. If the value of the **OraclePCA.poolName** tag is **PCA_POOL_HIGH**, then the mount target is in the high performance pool. If the value of the **OraclePCA.poolName** tag is **PCA_POOL**, or if no **OraclePCA.poolName** tag, then the mount target is in the default pool. For more information about these tags, see Using Resource Tags to Extend Functionality (IAM on the Appliance) on page 460.

   If the mount target meets your needs, skip this procedure and go to Creating a File System on page 766.
3. Select **Create Mount Target**.
4. Enter the mount target information:

   - **Name:** It doesn't have to be unique. An Oracle Cloud Identifier (OCID) uniquely identifies the mount target. Avoid entering confidential information.

     > **Note:**
     >
     > The mount target name is different than the DNS hostname.
   - **Create in Compartment:** Specify the compartment.
   - **VCN:** Select the VCN where you want to create the new mount target.
   - **Subnet:** Select a subnet to attach the mount target to.
   - **IP Address:** (Optional) You can specify an unused IP address in the subnet you selected for the mount target. If left blank, an IP address is automatically assigned.
   - **Host Name:** (Optional) You can specify a hostname you want to assign to the mount target.

> **Note:**
>
> The File Storage service constructs a fully qualified domain name (FQDN) by combining the hostname with the FQDN of the mount target subnet.
>
> For example, `myhostname.subnet123.dnslabel.examplevcn.com`.

- **Enable Network Security Groups:** Select this option to add this mount target to an existing NSG.

  > **Important:**
  >
  > Rules for the NSG that you select must be configured to allow traffic to the mount target's VNIC using specific protocols and ports. For more information, seeControlling Access to File Storage on page 771 and Configuring VCN Security Rules for File Storage on page 771.

- (Optional) Add one or more tags to this resource. For more information about tagging resources, see Resource Tags.

  By default, the mount target is created on the default storage pool. To create a mount target for a high performance pool, select the **OraclePCA** tag namespace, the **poolName** tag key, and the value **PCA_POOL_HIGH**. The poolName property can be set only when the mount target is created. You can't set or change this property value after the mount target is created.

5. Select **Create Mount Target**.

   Next, create a file system. See Creating a File System on page 766.

Using the OCI CLI

Use the oci fs mount-target create command and required parameters to create a new mount target in the specified compartment and subnet.

```
oci fs mount-target create --availability-domain <availability_domain_name>
 --compartment-id <compartment_OCID>--subnet-id <subnet_OCID> --display-
name <name_to_assign_to_mount-target> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Procedure

1. Gather the information that you need to run the command:
   - OCID of the compartment where you plan to create the file system (`oci iam compartment list`)
   - OCID of the subnet of the instance where you want to mount a file system (`oci network subnet list`)
2. Run the create mount target command.

   By default, the mount target is for the default pool of the attached ZFS Storage Appliance. To create a mount target for a high performance pool, specify the `OraclePCA.poolName` tag with a value of `PCA_POOL_HIGH` as shown in the following example. The `poolName` property can be set only when the mount target is created. You can't set or change this property value with the `update` command.

   Example:

```
oci fs mount-target create --availability-domain AD-1 \
--compartment-id ocid1.compartment.uniqueID --subnet-id
 ocid1.subnet.uniqueID \
--defined-tags '{"OraclePCA":{"poolName":"PCA_POOL_HIGH"}}' \
--display-name HighPerfPoolMT
{
  "data": {
    "availability-domain": "AD-1",
    "compartment-id": "ocid1.compartment.uniqueID",
```

```
    "defined-tags": {
      "Oracle-Tags": {
        "CreatedBy": "pca_user",
        "CreatedOn": "2024-07-03T14:56:29.92Z"
      },
      "OraclePCA":{
        "poolName":"PCA_POOL_HIGH"
      }
    },
    "display-name": "HighPerfPoolMT",
    "export-set-id": "ocid1.exportset.uniqueID",
    "freeform-tags": {},
    "id": "ocid1.mounttarget.uniqueID",
    "lifecycle-details": null,
    "lifecycle-state": "CREATING",
    "nsg-ids": [],
    "private-ip-ids": [],
    "subnet-id": "ocid1.subnet.uniqueID",
    "time-created": "2024-07-03T14:56:29.921587+00:00"
  },
  "etag": "2d278b37-a74a-4fec-b74a-fd9e9a1c72de"
```

3. Next, create a file system. See Creating a File System on page 766.

Using the API

Use the CreateMountTarget operation to create a new mount target in the specified compartment and subnet.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Creating a File System

On Private Cloud Appliance, you can create a shared file system using the File Storage service.

(Optional) To create a file system with specific values for the following attributes, first configure the attributes as described in Using Resource Tags to Extend Functionality (IAM on the Appliance) on page 460. Then use the following information when you create the file system.

- **Quota**

  The default value of `quota` is 0, which means no quota is set. A quota that you set includes the data in the file system and all snapshots created under the file system. You can specify a quota value in gigabytes from 0 to 8000000 (8 petabytes). Any fractional portion of the gigabyte value is rounded to the next larger megabyte. The file system quota can be changed with the file system `update` command.

  To apply a quota, during file system creation, assign this defined tag:

  - **Tag namespace:** `OraclePCA`
  - **Tag name:** `quota`
  - **Value:** *<quota value in gigabytes>*
- **Database record size**

  The default database record size is `131072` bytes. You can specify one of the following values (in bytes): `512`, `1024`, `2048`, `4096`, `8192`, `16384`, `32768`, `65536`, `131072`, `262144`, `524288`, `1048576`. The record size can be set only when the file system is created. You can't set or change this property value later.
- To use a nondefault value, during file system creation, assign this defined tag:

  - **Tag namespace:** `OraclePCA`
  - **Tag name:** `databaseRecordSize`
  - **Value** (select one): `512`, `1024`, `2048`, `4096`, `8192`, `16384`, `32768`, `65536`, `131072`, `262144`, `524288`, `1048576`
- **Backing store pool**

By default, the backing store of a file system is the default pool of the attached ZFS Storage Appliance, specified as **PCA_POOL**. You can specify `PCA_POOL_HIGH` to indicate that you want to use a high performance pool for the backing store. Before you specify `PCA_POOL_HIGH`, verify that a high performance pool is available. This property can be set only when the file system is created. You can't set or change this property value later.

To use a nondefault value, during file system creation, assign this defined tag:

- **Tag namespace:** `OraclePCA`
- **Tag name:** `databaseRecordSize`
- **Value** (select one): `PCA_POOL` (default) or `PCA_POOL_HIGH`

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **File Storage**, then select **File Systems**.
2. Ensure that the correct compartment is selected in the compartment drop-down menu above the file systems list. The file system and mount target must be in the same compartment and the same backing store pool when you create an export.
3. Select **Create File System**.
4. In the **Create File System** dialog box, enter the following information:
   - **Name:** It doesn't have to be unique. An Oracle Cloud Identifier (OCID) uniquely identifies the file system. Avoid entering confidential information.
   - **Create in Compartment:** Select the compartment where the file system is created.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.

     If the OraclePCA tag namespace and OraclePCA tag key definitions for file systems are configured in your OCI tenancy (see Using Resource Tags to Extend Functionality (IAM on the Appliance) on page 460), you can specify the **OraclePCA** Tag Namespace, select a key and a value. Your file system is created with the applicable attribute. The OraclePCA tag namespace can't be added later.
5. Select **Create File System**.

   The file system is created.

   Next, create an export for the file system. See Creating an Export for a File System on page 769.

Using the OCI CLI

Use the oci fs file-system create command and required parameters to create a new file system in the specified compartment and availability domain.

```
oci fs file-system create --availability-domain <availability_domain_name>
  --compartment-id <compartment_id> --display-name <fs_display_name>
 [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Procedure

1. Gather the information that you need to run the command:
   - Availability Domain Name (`oci iam availability-domain list`)
   - Compartment OCID (`oci iam compartment list`)
   - File System Name: The display name you want assigned to this file system
2. To use nondefault values for the quota, database record size, or backend store pool, specify the appropriate tags to set the values for these attributes. See Adding Tags at Resource Creation on page 458.

   See the following step for an example of setting these values.
3. Run the create file system command.

Example:

```
oci fs file-system create --availability-domain AD-1 --compartment-id
 ocid1.compartment.unique_ID --display-name MyFileSystem

{
  "data": {
    "availability-domain": "AD-1",
    "compartment-id": "ocid1.compartment.unique_ID",
    "defined-tags": {
      "Oracle-Tags": {
        "CreatedBy": "pca_user",
        "CreatedOn": "2024-07-05T13:15:11.19Z"
      }
    },
    "display-name": "MyFileSystem",
    "freeform-tags": {},
    "id": "ocid1.filesystem.unique_ID",
    "is-clone-parent": false,
    "is-hydrated": true,
    "is-targetable": null,
    "kms-key-id": "",
    "lifecycle-details": "",
    "lifecycle-state": "CREATING",
    "metered-bytes": 0,
    "source-details": {
      "parent-file-system-id": "",
      "source-snapshot-id": ""
    },
    "time-created": "2024-07-05T13:15:11.234434+00:00"
  },
  "etag": "58dec47e-4732-4730-9e18-6b5db1ac30d6"
}
```

Example using defined tags to set additional properties:

To set a quota for the file system, change the default database record size, or specify a high performance pool for the file system backing store, use `OraclePCA` defined tags as shown in the following example.

```
oci fs file-system create --availability-domain AD-1
 --compartment-id ocid1.compartment.unique_ID  --
display-name myfilesystem --defined-tags '{"OraclePCA":
{"quota":100000,"databaseRecordSize":8192,"poolName":"PCA_POOL_HIGH"}}'
```

Alternatively, you can specify these properties in a JSON file.

```
{
  "OraclePCA": {
    "quota": 100000,
    "databaseRecordSize": 8192,
    "poolName": "PCA_POOL_HIGH"
  }
}
```

Then specify the file as the argument of the `--defined-tags` option.

```
--defined-tags file://./fs_options.json
```

4. Next, create an export for the file system. See Creating an Export for a File System on page 769.

Using the API

Use the CreateFileSystem operation to create a new file system in the specified compartment and availability domain.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Creating an Export for a File System

On Private Cloud Appliance, exports control how NFS clients access file systems when they connect to a mount target. A file system must have at least one export in one mount target for instances to mount the file system.

> **Important:**
>
> When exporting file systems to overlapping CIDRs in a VCN, exports to the longest CIDR (smallest network) must be done first. For more information and an example, see My Oracle Support article PCA File system as a Service Exports (Doc ID 2823994.1).

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **File Storage**, then select **File Systems**.
2. In the left panel, select **File Systems**.
3. At the top of the page, select the compartment that contains the file system for which you plan to create an export.
4. Select the name of the file system.
5. Select **Create Export**.
6. Enter the required information:
   - **Mount Target**: Select a mount target from the list.
   - **Source CIDR**: Enter the longest CIDR (smallest network) in the CIDR range. Starting with the smallest CIDR range (largest network) will result in an error later in the process, because CIDR ranges larger than existing ones will not be accepted. For example, 10.0.0.0/29 is a longer CIDR than 10.0.0.0/28, so 10.0.0.0/29 must be added first.
7. Select **Create Export**.

   The file system export is created and the export details page is displayed.
8. In the export details page, copy the export path. The export path is used to mount the file system on an instance.
9. In the lower panel, review the **NFS Export Options**.

   The NFS export options for that file system are set to the default values, which allow full access for all NFS client source connections. These defaults must be changed to restrict access.
10. Consider your next action:
    - Mount the file system from an NFS client. See Mounting File Systems on UNIX-based Instances on page 774.
    - Configure NFS options to secure the exported file system. See Setting NFS Export Options on page 772.

Using the OCI CLI

Use the oci fs export create command and required parameters to create a new export in the specified export set, path, and file system.

```
oci fs export create --export-set-id <export_set_OCID> --file-system-
id <file_system_OCID>
--path "</pathname>" [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateExport operation to create a new export in the specified export set, path, and file system.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Mounting File Systems Across Private Cloud Appliance Appliances

You can create a file system on one Private Cloud Appliance appliance, and mount the file system from an instance that's on another appliance. To achieve this scenario, you must configure certain network parameters on each appliance.

Restriction

The Private Cloud Appliance appliance hosting the file system and the instance on the remote appliance that mounts the file system can't have overlapping VCN CIDR blocks.

On the Appliance Hosting the File System

1. Configure these network parameters:
   a. Create a Dynamic Routing Gateway (DRG).

      See Creating a Dynamic Routing Gateway on page 511.
   b. For the VCN subnet that will be used by the mount target, attach the VCN to the DRG.

      See Attaching VCNs to a Dynamic Routing Gateway on page 512.
   c. For the VCN subnet, add a route rule with the DRG as the target, and assign a destination CIDR that matches the remote appliance VCN CIDR.

      For example, if the remote instance that will mount the file system has a VCN with a 10.11.0.0/16 CIDR, the set the route rule destination CIDR to 10.11.0.0/16.

      > **Important:**
      >
      > Don’t specify 0.0.0.0/0 as the destination. Doing so causes serious internal network issues.

      > **Note:**
      >
      > This route rule configuration is only required for mounting file systems across Private Cloud Appliance racks. This configuration isn't required for file system mounts within the same rack.

      See Working with Route Tables on page 493.
2. Create a mount target. See Creating a Mount Target on page 763.
3. Create a file system. See Creating a File System on page 766.
4. Create an export that exports to both the source and remote VCN CIDR.

   The export CIDR must be big enough to cover both VCN subnet CIDRs. For example, if the host VCN CIDR is 10.10.0.0/16, and the remote instance VCN is 10.11.0.0/16, you can configure the export to use 10.10.0.0/15. This requirement only applies to the Private Cloud Appliance hosting the file system.

   See Creating an Export for a File System on page 769.

On the Remote Appliance

1. Configure these network parameters:
   a. Create a Dynamic Routing Gateway (DRG).

      See Creating a Dynamic Routing Gateway on page 511.
   b. For the VCN subnet, attach the VCN to the DRG.

      See Attaching VCNs to a Dynamic Routing Gateway on page 512.
   c. For the VCN subnet, add a route rule with the DRG as the target, and assign a destination CIDR that matches the host VCN CIDR.

For example, if the host mount target has a VCN 10.0.0.0/16 CIDR, set the route rule destination CIDR to 10.0.0.0/16.

Important:

Don’t specify 0.0.0.0/0 as the destination. Doing so causes serious internal network issues.

Note:

This route rule configuration is only required for mounting file systems across Private Cloud Appliance racks. not required for file system mounts within the same Private Cloud Appliance rack.

See Working with Route Tables on page 493.

2. Log in to the instance and mount the file system.

   See:

   - Mounting File Systems on UNIX-based Instances on page 774
   - Mounting File Systems on Microsoft Windows Instances on page 778

Controlling Access to File Storage

On Private Cloud Appliance, before you can mount a file system, you must configure security rules to allow traffic to the mount target's VNIC using specific protocols and ports.

Security rules enable traffic for the following protocols:

- Open Network Computing Remote Procedure Call (ONC RPC) rpcbind utility protocol
- Network File System (NFS) protocol
- Network File System (MOUNT) protocol

Configuring VCN Security Rules for File Storage

On Private Cloud Appliance, you can add the required rules to a preexisting security list associated with a subnet, such as the default security list that is created along with the VCN.

For specific information about which security rules are required for the File Storage service, see File Storage Network Ports on page 758.

For more information about managing VCNs and subnets, see Managing VCNs and Subnets on page 478.

Adding File Storage to a Network Security Group

On Private Cloud Appliance, adding File Storage to a network security group is accomplished by performing several procedures. This section lists the procedures.

Task Flow

No.	Description	Links to Procedures
1.	Create an NSG with the required security rules. (Alternatively, you can add them to a previously existing NSG.)	Controlling Traffic with Network Security Groups on page 501
2.	Add the mount target (or more specifically, the mount target's VNIC) to the NSG. You can do this task when you create the mount target, or you can update the mount target and add it to one or more NSGs that contain the required security rules.	Adding a Mount Target to a Network Security Group on page 772

No.	Description	Links to Procedures
3.	If you're setting up a mount target and instance in different subnets, add the instance (or more specifically, the instance's primary VNIC) to the NSG that contains the required security rules. You can do this task when you create the instance, or you can directly update the instance's primary VNIC.	Updating a VNIC on page 522

Adding a Mount Target to a Network Security Group

On Private Cloud Appliance, you can add the mount target to one or more Network Security Groups (NSGs). File storage requires specific rules to be configured for NSGs that are associated with mount targets.

Using the Compute Web UI

1. Ensure that an NSG with ingress and egress rules has been configured. See Configuring VCN Security Rules for File Storage on page 771.
2. Ensure that a mount target is created. See Creating a Mount Target on page 763.
3. In the Compute Web UI navigation menu, select **File Storage**, then select **Mount Targets**.
4. At the top of the page, select the compartment that contains the mount target.
5. Select the mount target name to see the details page.
6. Select **Edit**.
7. Enable Network Security Groups.
8. Select the NSG from the list.
9. Select **Save Changes**.

Using the OCI CLI

Use the oci fs mount-target update command and required parameters to update the specified mount target's information.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

```
oci fs mount-target update --mount-target-id <mount_target_OCID>  --nsg-ids
 '["<nsg1_OCID>","i"]' [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateMountTarget operation to update the specified mount target's information.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Setting NFS Export Options

On Private Cloud Appliance, when you create a file system and export, the NFS export options for that file system are set to the defaults. The default values allow full access for all NFS client source connections. These defaults must be changed if you want to restrict access.

> **Caution:**
>
> If a file system is mounted by any clients, creating, deleting, or editing the Source value can disrupt file system I/O operations.

Export Option in the Compute Web UI	Export Option in the OCI CLI	Default Value	Description
Source:	`source`	0.0.0.0/0	The IP address or CIDR block of a connecting NFS client.
Ports:	`require-privileged-source-port`	Any	Always set to: • UI: Any • OCI CLI: `false`
Access:	`access`	Read/Write	Specifies the source NFS client access. Can be set to one of these values: • `READ_WRITE` • `READ_ONLY`
Squash:	`identity-squash`	None	Determines whether the clients accessing the file system as root have their User ID (UID) and Group ID (GID) remapped to the squash UID/GID. Possible values: • Root – Only the root user is remapped. • None – No users are remapped.
Squash UID/GID:	`anonymous-uid` and `anonymous-gid`	65534	This setting is used along with the Squash option. When remapping a root user, you can use this setting to change the default anonymousUid and anonymousGid to any user ID of your choice.

Note – If you change the RW/RO permissions of an export option for an SMB share, the changes are only enforced for newly network-mapped drives of that share. Any previously mapped drives of the same share retain the original permissions. To have the changed permissions enforced on previously mapped drives on SMB clients, disconnect the shares and map them again.

For more information about configuring the options to suit various access scenarios, see Export Options for File Storage on page 758.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **File Storage**, then select **File Systems**.
2. At the top of the page, select the compartment that contains the file system.
3. Select the file system name.
4. Under **Resources**, select **Exports**.
5. Select the export's export path.

   The NFS Export Options are displayed.
6. Select **Edit Options**.
7. In the **NFS Export Options** dialog box, configure the NFS options.
8. Select **Update Options**.

Using the OCI CLI

Use the oci fs export update command and required parameters to update the export information.

```
oci fs export update --export-id <export_id> --export-options <file:// json_file or json_string> [OPTIONS]
```

Note – The `require-privileged-source-port` option can only be set to `false`.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateExport operation to update the export information.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Mounting File Systems on UNIX-based Instances

On Private Cloud Appliance, instance users of UNIX based operating systems, such as Linux and Oracle Solaris, can use OS commands to mount and access file systems.

Mount targets serve as network access points for file systems. After your mount target is assigned an IP address, you can use it together with the export path to mount the file system.

On the instance from which you want to mount the file system, you need to install an NFS client package and create a mount point. When you mount the file system, the mount point effectively represents the root directory of the File Storage file system, allowing you to write files to the file system from the instance.

Prerequisites

- The file system must be created and have at least one export in a mount target. See Creating a File System, Mount Target, and Export on page 763.
- The mount target must have correctly configured security rules or be assigned to an NSG. See Configuring VCN Security Rules for File Storage on page 771.

> **Note:**
>
> **Only for NFSv4 Mounts in Oracle Linux instances** – If you find that the file system owner is assigned as `nobody` instead of the actual user who mounts the file system, and if you haven't set identity squash, you might need to edit the `/etc/idmapd.conf` file. In the file, set the DOMAIN entry to either `localdomain` or to the Active Directory domain name, if applicable. After the change, run `service rpcidmapd restart` to restart the `rpcidmapd` service.
>
> Defining settings in the `/etc/idmapd.conf` file is specific to Oracle Linux, and there are other ways to configure the domain depending on the OS in use. Consult your OS documentation.

Obtaining the Mount Target IP Address

To mount a file system, you need to know the private IP address of the mount target that has the export for the file system.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **File Storage**, then select **Mount Target**.
2. At the top of the page, select the compartment that contains the mount target.
3. Select the mount target name to see the details page.

   The private IP address is displayed.

Using the OCI CLI

Use the oci fs mount-target get command and required parameters to get mount target information.

```
oci fs mount-target get --mount-target-id <mount_target_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the GetMountTarget operation to get mount target information.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Mounting a File System on Linux, Red Hat, or CentOS

1. Log into the instance where you want to mount the file system.

   See Connecting to an Instance.

   Example:

   ```
   ssh user@192.0.2.0
   ```

2. Install the NFS client using this command:

   ```
   sudo yum install nfs-utils
   ```

3. Create a directory that will be used as the mount point.

   Replace <*yourmountpoint*> with a directory name of your choice. Example: `/mnt/mountpoint-A`

   ```
   sudo mkdir -p <yourmountpoint>
   ```

4. Mount the file system.

   > **Caution:**
   >
   > Omitting the `-o nosuid` option can allow unprivileged users to escalate their permissions to 'root'. The `nosuid` option disables set-user-identifier or set-group-identifier bits within the mounted system, which are rarely used.

   Example:

   ```
   sudo mount -t nfs -o nfsvers=<version>,nosuid <10.x.x.x>:<fs-export-path> <yourmountpoint>
   ```

   - Replace <*version*> with one of the following, based on the NFS protocol version you want to use:
     - 3,noacl
     - 4.0
     - 4.1
   - Replace <*10.x.x.x*> with the mount target's private IP address. See Obtaining the Mount Target IP Address on page 774.
   - Replace <*fs-export-path*> with the export path that was generated when the export was created. See Creating an Export for a File System on page 769.
   - Replace <*yourmountpoint*>with the full path to the local mount point.

5. View the mounted file system.

   ```
   df -h
   ```

6. Write a file to the file system.

   Replace <*yourmountpoint*> with the path to the local mount point and <*filename*>with your file name.

   ```
   sudo touch /mnt/<yourmountpoint>/<filename>
   ```

7. Verify that you can access the file system and view the file.

   Replace yourmountpoint with the path to the local mount point.

   ```
   cd <yourmountpoint>
   ls
   ```

8. Add the file system mount information to the appropriate mount file for your OS.

   So far, the file system is manually mounted to the client. If the client is rebooted, the file system won't automatically mount unless you add it to the mount file (for example the `/etc/fstab` or `/etc/vfstab` file).

Mounting a File System on Ubuntu or Debian

Operating Systems and versions of operating systems differ in the way software is added. Consult the documentation for our specific operating system for details.

1. On the NFS client, open a command window, and install the NFS client using this command:

   ```
   sudo apt-get install nfs-common
   ```

2. Create a directory that will be used as the mount point.

   Replace <*yourmountpoint*> with a directory name of your choice. Example: `/mnt/mountpoint-A`

   ```
   sudo mkdir -p <yourmountpoint>
   ```

3. Mount the file system.

   > **Caution:**
   >
   > Omitting the `-o nosuid` option might allow unprivileged users to escalate their permissions to 'root'. The `nosuid` option disables set-user-identifier or set-group-identifier bits within the mounted system, which are rarely used.

   Example:

   ```
   sudo mount -t nfs -o nfsvers=<version>,nosuid <10.x.x.x>:<fs-export-path> <yourmountpoint>
   ```

   - Replace <*version*> with one of the following, based on the NFS protocol version you want to use:
     - 3,noacl
     - 4.0
     - 4.1
   - Replace <*10.x.x.x*> with the mount target's private IP address. See Obtaining the Mount Target IP Address on page 774.
   - Replace **<*fs-export-path*>** with the export path that was generated when the export was created.

     See Creating an Export for a File System on page 769.
   - Replace <*yourmountpoint*>with the full path to the local mount point.

4. View the file system.

   ```
   df -h
   ```

5. Write a file to the file system.

Replace <*yourmountpoint*> with the path to the local mount point and <*filename*>with your file name.

```
sudo touch /mnt/<yourmountpoint>/<filename>
```

6. Verify that you can access the file system and view the file.

   Replace yourmountpoint with the path to the local mount point.

   ```
   cd <yourmountpoint>
   ls
   ```

7. Add the file system mount information to the appropriate mount file for your OS.

   So far, the file system is manually mounted to the client. If the client is rebooted, the file system won't automatically mount unless you add it to the mount file (for example the `/etc/fstab` or `/etc/vfstab` file).

Configuring a File System to Automatically Mount (Linux Instances)

On Linux instances, if you want to automatically mount exported file systems during an instance boot, you need to add the mount information in the `/etc/fstab` file.

1. Log into the instance where you want the file system mounted.

   See Connecting to an Instance.

2. Create a mount point, if one hasn't been created.

   Example:

   ```
   mkdir /mnt/fs01
   ```

3. Open the `/etc/fstab` file in an editor and add a line for the nfs file systems you want automatically mounted.

   This is an example of an `/etc/fstab` file entry.

   ```
   192.0.2.0:/
   export/3ywflz8hhqfde81miewqwjfd049zju69502t9ouo6shzidr4dndaz1hd6qfi /mnt/
   fs01 nfs nfsvers=4.1,nosuid,nofail 0 0
   ```

   The `/etc/fstab` file space-separated fields are specified with these entries:

   - **Field 1:** Device to mount. For network file systems, specify: <***mount target IP***> : <***export_path***>

     See Obtaining the Mount Target IP Address on page 774 and Creating an Export for a File System on page 769.
   - **Field 2:** Full path of the mount point on the instance.
   - **Field 3:** File system type. In this case, specify `nfs`.
   - **Field 4:** NFS mount options separated with commas, such as:

     ```
     nfsvers=<version>,nosuid,nofail
     ```

     - `nfsvers=` where <***version***> is one of the following:
       - `3,noacl`
       - `4.0`
       - `4.1`
     - `nosuid` – prevents unprivileged users from escalating their permissions to root.
     - `nofail` – Ensures that an unavailable file system does not cause the instance reboot process to fail.

     In this case, use the same options as described in Mounting a File System on Linux, Red Hat, or CentOS on page 775. Each option is separated by a comma (no spaces).
   - **Field 5:** Obsolete option for dump backups. Specify `0` (zero) for no dump backup.
   - **Field 6:** File system check (fsck) order. Specify `0` (zero) for no check.

4. Use this command to mount the volumes that are in the `/etc/fstab` file:

```
sudo mount -a
```

   If you get any error messages, fix the cause before proceeding.
5. Verify that the file systems are mounted:

```
mount | grep nfs
```

6. To verify that the file system will automatically mount, reboot the instance.

```
sudo reboot
```

7. After the reboot, log into the instance and check to see if the nfs file system is mounted.

```
mount | grep nfs
```

Mounting File Systems on Microsoft Windows Instances

On Private Cloud Appliance, you can make file systems available to Microsoft Windows instances by mapping a network drive to the mount target IP address and export path provided by the File Storage service. You can accomplish this task using NFS or SMB protocols.

Prerequisite

Using the SMB protocol requires that the Microsoft Windows instances and Private Cloud Appliance belong to the same Active Directory domain. Before performing any tasks in this section, ensure the following requirements are set up:

- Installed and configured Microsoft Active Directory Federation Services for your organization.
- Set up groups in Active Directory that will map to groups in Private Cloud Appliance.
- Created users in Active Directory who will sign in to the Compute Web UI.

> **Note:**
>
> Consider naming Active Directory groups that you intend to map to Private Cloud Appliance groups with a common prefix to make it easy to apply a filter rule, for example, `CCC_Administrators`, `CCC_NetworkAdmins`, `CCC_InstanceLaunchers`.

- Open an Oracle support request to request an Active Directory configuration in the Private Cloud Appliance appliance. See Creating a Support Request. To access support, sign in to the Oracle Cloud Console as described in Sign In to the OCI Console.

Mounting a File system On a Windows Instance Using NFS

Prerequisites

- The file system must be created and have at least one export in a mount target. See Creating a File System, Mount Target, and Export on page 763.
- The mount target must have correctly configured security rules or be assigned to an NSG. See Configuring VCN Security Rules for File Storage on page 771.
- You must know the mount target's IP address. See Obtaining the Mount Target IP Address on page 774.
- You must be able to log into the Microsoft Windows OS on the instance with superuser or administrator privileges.

Before You Begin

The following tasks are included in this procedure, and you might want to be aware of them before you begin.

- **Installation of the Microsoft Windows NFS Client** – This service must be installed on the instance from which you want to mount the file system. Installing the client often requires a restart of the instance.

- **The `AnonymousGid` and `AnonymousUid` identity values must be configured to allow write access.**
  – Access to NFS file systems requires UNIX user and group identities, which are not the same as Microsoft Windows user and group identities. By default, file systems write permissions are only granted to the root user. To enable user access to NFS shared resources, the Microsoft Windows client for NFS accesses file systems anonymously, using `AnonymousGid` and `AnonymousUid`.

  > **Caution:**
  >
  > Updating the AnonymousGid and AnonymousUid values require registry changes to your instance.

Choose one the following methods:

Using the Microsoft Windows Command Prompt

1. Log into your Microsoft Windows instance.

   See Connecting to an Instance.
2. Open Microsoft Windows PowerShell and run as Administrator:
   a. Go to Start and open PowerShell.
   b. In PowerShell, type the following to run as Administrator:

   ```
   Start-Process powershell -Verb runAs
   ```

   c. In the User Account Control window, select Yes. A new Administrator: PowerShell window opens. You can close the standard PowerShell window to avoid confusing them.
3. In Administrator: PowerShell, get the NFS client and update the registry by typing the following:

   ```
   Install-WindowsFeature -Name NFS-Client
   Set-ItemProperty HKLM:\SOFTWARE\Microsoft\ClientForNFS\CurrentVersion
   \Default -Name AnonymousUid -Value 0
   Set-ItemProperty HKLM:\SOFTWARE\Microsoft\ClientForNFS\CurrentVersion
   \Default -Name AnonymousGid -Value 0
   Stop-Service -Name NfsClnt
   Restart-Service -Name NfsRdr
   Start-Service -Name NfsClnt
   ```
4. Open a standard Command Prompt Window.

   > **Important:**
   >
   > NFS file systems mounted as Administrator are not available to standard users.
5. From the Command Prompt window, mount the file system.

   See the cautions and notes below the example.

   In the following example, replace:

   - `10.x.x.x` with the mount point IP address (see Obtaining the Mount Target IP Address on page 774)
   - `fs-export-path` with the file system export path (see Creating a File System, Mount Target, and Export on page 763)
   - `X` with the drive letter of any available drive you want to map the file system to.

   Example:

   ```
   mount 10.x.x.x:/fs-export-path X:
   ```
6. Verify that you can access and write to the file system.
   a. Access the file system.

In the example, replace `X` with the drive letter you used to mount the file system.

```
X:
```

b. Write a file.

```
echo > myfile.txt
```

c. Verify that you can view the file.

```
dir
```

Using the Microsoft Windows File Explorer

1. Log into your Microsoft Windows instance.

   See Connecting to an Instance.
2. Open Microsoft Windows PowerShell and run as Administrator:
   a. Go to Start and open PowerShell.
   b. In PowerShell, type the following to run as Administrator:

   ```
   Start-Process powershell -Verb runAs
   ```

   c. In the User Account Control window, select Yes. A new Administrator: PowerShell window opens. You can close the standard PowerShell window to avoid confusing them.
3. In Administrator: PowerShell, get the NFS client by typing the following:

   ```
   Install-WindowsFeature -Name NFS-Client
   ```
4. If necessary, restart your system.
5. Open the registry editor (regedit) to map the AnonymousGid and AnonymousUid to the root user.

   > **Caution:**
   >
   > User identity mapping requires changes to your system registry.

   a. Select Windows Search.
   b. Enter `regedit` in the Search field and press Enter.
   c. Select Yes to allow changes to your device.
   d. Select `HKEY_LOCAL_MACHINE`. Then, browse to: `Software\Microsoft\ClientForNFS\CurrentVersion\Default`.
6. Add a new DWORD32 registry entry for `AnonymousGid`:
   a. Select Edit, and select New DWORD (32 bit) Value.
   b. In the Name field, enter `AnonymousGid`. Leave the value at `0`.
7. Repeat the previous step to add a second DWORD32 registry entry named `AnonymousUid` with a value of `0`.
8. Open Microsoft Windows Command Line (CMD) and run as Administrator:
   a. Go to Start and scroll down to Apps.
   b. In the Windows System section, press CTRL+Shift and select Command Prompt.
9. In the Microsoft Windows Command Line (CMD) window, restart the NFS Client by typing the following:

   ```
   nfsadmin client stop
   ```

   ```
   nfsadmin client start
   ```
10. Open File Explorer and select This PC. In the Computer tab, select Map network drive.
11. Select the Drive letter that you want to assign to the file system.
12. In the Folder field, enter the following line, replacing:

- `10.x.x.x` with the mount point IP address (see Obtaining the Mount Target IP Address on page 774)
- `fs-export-path` with the file system export path (see Creating a File System, Mount Target, and Export on page 763)

Line:

```
\\10.x.x.x\fs-export-path
```

13. Select Finish.

Mounting a File System on a Windows Instance Using SMB

On Private Cloud Appliance, you can make file systems available to Microsoft Windows instances by mapping a network drive to the mount target IP address and export path provided by the File Storage service. You can accomplish this task using NFS or SMB protocols. The SMB process is described here.

> **Note:**
>
> These instructions apply only to file systems created On Private Cloud Appliance.

General Prerequisites

- The file system must be created and have at least one export in a mount target. See Creating a File System, Mount Target, and Export on page 763.
- The mount target must have correctly configured security rules or be assigned to an NSG. See Configuring VCN Security Rules for File Storage on page 771.
- You must know the mount target's IP address. See Obtaining the Mount Target IP Address on page 774.
- You must be able to log into the Microsoft Windows OS on the instance with superuser or administrator privileges.

Specific Prerequisites for SMB Support

Using the SMB protocol requires that the Microsoft Windows instances and Private Cloud Appliance belong to the same Active Directory domain.

This procedure assumes that the AD service is already configured in your data center.

To add a Microsoft Windows instance to your AD service, perform the necessary administrative tasks according to the documentation for your version of Microsoft Windows OS.

To add the Private Cloud Appliance to your AD service, Oracle must add the AD domain name to the Oracle Cloud Infrastructure Active Directory Domain configuration. To request this administrative action, submit a support request. See Creating a Support Request. To access support, sign in to the Oracle Cloud Console as described in Sign In to the OCI Console.

Relaxing File System Permissions Before Network Mapping with SMB

By default, write permissions to a file system are limited to the UNIX superuser and group identity. To provide write permission to AD domain users, the permissions need to be relaxed.

1. Mount the network drive using NFS protocol.

   See Mounting a File system On a Windows Instance Using NFS on page 778.
2. Relax the file system permissions:
   a. Open File Explorer, select the mapped drive and right-select it, then select Properties.
   b. Select the NFS Attributes tab.
   c. Change File permissions by checking all RWX check boxes to relax the permissions for Owner, Group, and Other.
   d. Select OK.
3. Disconnect the NFS-mounted drive.

Now that the file system permissions are relaxed, you can mount the file system using the SMB protocol.

Mounting a File System Using SMB

1. Log into your Microsoft Windows instance.
2. Open File Explorer and select This PC.
3. In the Computer tab, select Map network drive.
4. In the Folder field, enter the following line and replace these items:
   - `10.x.x.x` with the mount target IP address.
   - `fs-export-path-ID` with the file system export path (see Creating an Export for a File System on page 769).

     Note – Do not include `\export` in the `fs-export-path-ID` string when mounting using SMB.

   ```
   \\10.x.x.x\fs-export-path-ID
   ```

   Example:

   ```
   \\192.0.2.0\39u21btystm8x1axizezb9a3lfnpzjho98evi3ij450i96vj0a8jpf36au26
   ```
5. select the 'Drive' letter of any available drive you want to map the file system to.
6. If needed, select the Connect using different credentials check box.
7. Select Finish.
8. When prompted, provide the user name and password of the AD domain user used for mapping the network drive.
9. Select OK.
10. In a Command Prompt window (cmd), verify that the drive is properly mapped using this command:

```
C:\>net use
New connections will be remembered.
Status       Local     Remote                    Network
-------------------------------------------------------------------------------
OK           Z:        \
\10.0.0.2\uvj1iw6ytyecqijcbdgpy7ec15mgsv044i7609giqx7ukfn6t2pwgfqot0ma
                                                 Microsoft Windows Network
The command completed successfully.
C:\>
```

Managing Mount Targets and Exports

On Private Cloud Appliance, a mount target is an NFS endpoint assigned to a VCN subnet of your choice and provides network access for file systems. The mount target provides the IP address or DNS name that's used together with a unique export path to mount the file system.

For an instance to mount a file system, the instance's VCN must have a mount target.

You can reuse the same mount target to make as many file systems available on the network as you want. To reuse the same mount target for multiple file systems, create an export in the mount target for each file system.

> **Important:**
>
> When more than one file system is exported to the same mount target, you must export to the mount target with the smallest network (largest CIDR number) first. For detailed information and instructions, refer to My Oracle Support Doc ID 2823994.1.

For instructions to create a mount target, see Creating a File System, Mount Target, and Export on page 763.

This section provides instructions for administering mount targets.

Listing Mount Targets and Viewing Details

On Private Cloud Appliance, you can list File System mount targets.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **File Storage**, then select **Mount Targets**.
2. At the top of the page, select the compartment that contains the mount target.

   The mount targets are displayed.
3. To see the mount target details, select the mount target name.

Using the OCI CLI

Listing Mount Targets

Use the oci fs mount-target list command and required parameters to list the mount target resources in the specified compartment.

```
oci fs mount-target list --availability-domain <availability_domain_name>
 --compartment-id <compartment_id> [OPTIONS]
```

Listing Details About a Mount Target

Use the oci fs mount-target get command and required parameters to get the specified mount target's information.

```
oci fs mount-target get --mount-target-id <mount-target_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListMountTargets operation to list the mount target resources in the specified compartment.

Use the GetMountTarget operation to get the specified mount target's information.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Updating a Mount Target

On Private Cloud Appliance, when you update a mount target, you can set or change the Network Security Groups and display name. You can't change the backing store pool type assignment (default or high performance).

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **File Storage**, then select **Mount Targets**.
2. At the top of the page, select the compartment that contains the mount target.
3. Select the Actions menu (⋮) for the mount target, and select **Edit**.
4. Make changes.
5. Select **Save**.

Using the OCI CLI

Use the oci fs mount-target update command and required parameters to change the mount target.

```
oci fs mount-target update --mount-target-id <mount_target_OCID> <parameters
 to change> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateMountTarget command and required parameters to update a mount target.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Listing Exports

On Private Cloud Appliance, you can list File System exports.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **File Storage**, then select **Mount Targets**.
2. At the top of the page, select the compartment that contains the mount targets.
3. Select the mount target name.

   The exports are displayed at the bottom of the page.
4. To see the export details, select the export name.

Using the OCI CLI

Use the oci fs export list command and required parameters to list exports for the specified compartment.

```
oci fs export list --compartment-id <compartment_id> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListExports operation to list exports for the specified compartment.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Listing Export Sets

On Private Cloud Appliance, you can list File System export sets using the OCI CLI and API.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci fs export-set list command and required parameters to list the export set resources in the specified compartment.

```
oci fs export-set list --availability-domain <availability_domain_name> --
compartment-id <compartment_id> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Procedure

1. Get the compartment where you want to list export sets (`oci iam compartment list`)
2. Run the command.

   Example:

```
oci fs export-set list --availability-domain AD-1 \
--compartment-id ocid1.compartment.uniqueID

{
  "data": [
    {
      "availability-domain": "AD-1",
      "compartment-id": "ocid1.compartment.uniqueID",
      "display-name": "MyMountTarget2 - export set",
      "id": "ocid1.exportset.uniqueID",
      "lifecycle-state": "ACTIVE",
```

```
      "time-created": "2021-06-17T19:01:37+00:00",
      "vcn-id": "ocid1.vcn.uniqueID"
    }
  ]
}
```

Using the API

Use the ListExportSets operation to list the export set resources in the specified compartment.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting an Export

On Private Cloud Appliance, deleting an export deletes the file system path that clients use to mount the file system. Deleting an export does not delete any file systems.

> **Caution:**
>
> When you delete an export, you can no longer mount the file system using the file path specified in the deleted export. Any clients that use the export path to mount a file system will not be able to access the file system.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **File Storage**, then select **File Systems**.
2. At the top of the page, select the compartment that contains the file system with the export you plan to delete.
3. Select the name of a file system that uses the export you plan to delete.
4. Select the Actions menu (⋮) for the export, then select **Delete**.
5. Confirm the deletion.

Deleting an Export Using the OCI CLI

Use the oci fs export delete command and required parameters to delete the specified export.

```
oci fs export delete --export-id <export_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Deleting an Export Using the API

Use the DeleteExport operation to delete the specified export.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Moving a Mount Target to a Different Compartment

On Private Cloud Appliance, you can move a Mount Target to a different compartment.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci fs mount-target change-compartment command and required parameters to move a mount target to a different compartment.

```
oci fs mount-target change-compartment --mount-target-id <mount_target_OCID>
 --compartment-id <destination_compartment_OCID>
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

This task is not available in the API.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Mount Target

On Private Cloud Appliance, deleting a mount target deletes all the exports that are associated with the mount target.

> **Caution:**
>
> Deleting the mount target also deletes all of its exports of associated file systems. File systems are no longer available through the deleted mount target.
>
> **Deleting a mount target has no effect on file system data or file system snapshots.**

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **File Storage**, then select **Mount Targets**.
2. At the top of the page, select the compartment that contains the mount target that you want to delete.
3. Select the Actions menu (⋮) for the mount target you plan to delete, then select **Delete**.
4. Confirm the deletion.

Using the OCI CLI

Use the oci fs mount-target delete command and required parameters to delete the specified mount target.

```
ci fs mount-target delete --mount-target-id <mount_target_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteMountTarget command and required parameters to delete the specified mount target.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing File Systems

On Private Cloud Appliance, a file system in the File Storage service represents a network file system that's mounted by one or more clients. File systems are associated with a single compartment. File systems must have at least one export in one mount target for any client to mount and use the file system. Data is added to a file system from the client.

This section describes how to manage file systems after they're created. For instructions to create a file system, see Creating a File System, Mount Target, and Export on page 763.

Listing File Systems

On Private Cloud Appliance, file systems are associated with a single compartment. When you select a compartment, the Compute Web UI displays all file systems in the compartment. You can also see exports and snapshots associated with each file system.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **File Storage**, then select **File Systems**.
2. At the top of the page, select the compartment that contains the file system.

   The file systems for the compartment are listed.

3. To see file system details, select the name of the file system.

Using the OCI CLI

Use the oci fs file-system list command and required parameters to list file systems.

```
oci fs file-system list --availability-domain <availability_domain_name> --
compartment-id <compartment_OCID> [OPTIONS]
```

Use the oci fs file-system get command and required parameters to view file system details.

```
oci fs file-system get --file-system-id <file-system_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListFileSystems operation to list file systems.

Use the GetFileSystem operation to get more information about the specified file system.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Updating a File System

On Private Cloud Appliance, you can change some file system parameters.

You can change the file system name and quota. You can't change the database record size or the backing store pool type assignment (default or high performance).

To reduce the quota for the file system, first check the current file system usage. The quota can't be set smaller than the current usage. Current usage includes the data in the file system and all snapshots created under the file system. To check the current usage, check the value of **Metered Bytes** on the file system details page using the Compute Web UI or `metered-bytes` in the file system `get` output using the OCI CLI.

> **Note:**
>
> The **Metered Bytes** value can take up to 15 minutes to refresh on a system with active I/O.

Fifteen minutes after setting a lower quota, compare the quota and metered bytes values for the file system. Check the value of quota in the OraclePCA defined tag on the Tags tab on the file system details page in the Compute Web UI or in `defined-tags` in the file system `get` output in the OCI CLI. If the quota is less than metered bytes, then the quota isn't enforced and you should set a higher quota.

Avoid entering confidential information.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **File Storage**, then select **File Systems**.
2. At the top of the page, select the compartment that contains the file system.
3. For the file system you want to update, select the Actions menu (⋮), then select **Edit**.
4. Enter a new name in the name field, or enter a new quota in the OraclePCA:quota defined tag as described in Creating a File System on page 766.
5. Select **Save Changes**.

Using the OCI CLI

Use the oci fs file-system update command and required parameters to update the specified file system's information. You can use this operation to rename a file system.

```
oci fs file-system update --file-system-id <file_system_OCID> --display-
name <new_file-system_name> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Procedure

1. Get the OCID of the file system that you want to update:`oci fs file-system list`
2. Specify a new name for the file system, or set a new quota in the OraclePCA:quota defined tag as described in Creating a File System on page 766.
3. Run the update file system command.

   Example:

   ```
   oci fs file-system update --file-system-id ocid1.filesystem.unique_ID --
   defined-tags '{"OraclePCA":{"quota":500000}}'
   ```

Using the API

Use the UpdateFileSystem operation to update the specified file system's information. You can use this operation to rename a file system.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Moving a File System to a Different Compartment

On Private Cloud Appliance, you can move a file system to another compartment using the OCI CLI or API.

The file system is moved immediately. Moving a file system doesn't affect mounted instances.

For information about moving resources between compartments, see Moving Resources to a Different Compartment.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci fs file-system change-compartment command and required parameters to move a file system and its associated snapshots into a different compartment within the same tenancy.

```
oci fs file-system change-compartment --file-system-id <file-system_OCID> --
compartment-id <destination_compartment_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ChangeFileSystemCompartment operation to move a file system and its associated snapshots into a different compartment within the same tenancy.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a File System

On Private Cloud Appliance, you can delete a file system that doesn't have any dependencies.

> **Caution:**
>
> You can't undo this operation. Any data in a file system is permanently deleted with the file system. You can't recover a deleted file system or its snapshots.

A file system that has an export can't be deleted. To delete the export, see Deleting an Export on page 785.

You can't delete file systems that have dependencies. For example, if you have created a snapshot of this file system and then created a new file system from the snapshot, you can't delete the source file system.

For more information, see Deleting File System Resources Overview on page 758.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **File Storage**, then select **File Systems**.
2. At the top of the page, select the compartment that contains the file system.
3. Select the Actions menu (⋮) for the file system and select **Delete**.
4. Confirm the deletion.

Using the OCI CLI

Use the oci fs file-system delete command and required parameters to delete the specified file system. Before you delete the file system, verify that no remaining export resources still reference it.

```
oci fs file-system delete --file-system-id<file-system_OCID>[OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteFileSystem operation to delete the specified file system. Before you delete the file system, verify that no remaining export resources still reference it.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing File System Snapshots

On Private Cloud Appliance, the File Storage service supports snapshots for file system data protection.

Snapshots are a consistent, point-in-time view of your file systems. Snapshots are copy-on-write, and scoped to the entire file system. The File Storage service encrypts all file system and snapshot data at rest. You can take as many snapshots as you need.

The snapshot is accessible at `.zfs/snapshot/`*name*.

Listing Snapshots

On Private Cloud Appliance, you can list snapshots and view snapshot details.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **File Storage**, then select **File Systems**.
2. At the top of the page, select the compartment that contains the file system with the snapshot.
3. Select the file system name.
4. Under **Resources**, select **Snapshots**.

   The file system snapshots are listed.
5. To get the details for a specific snapshot, select the snapshot name.

Using the OCI CLI

Use the oci fs snapshot list command and required parameters to list snapshots of the specified file system.

```
oci fs snapshot list --file-system-id <file-system_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListSnapshots operation to list snapshots of the specified file system.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Creating a Snapshot

On Private Cloud Appliance, You can create a snapshot of a file system. A snapshot is a point-in-time view of the file system. The snapshot is accessible at `.zfs/snapshot/`*name*.

Avoid entering confidential information.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **File Storage**, then select **File Systems**.
2. At the top of the page, select the compartment that contains the file system for which you want to create a snapshot.
3. Select the file system name.
4. Under **Resources**, select **Snapshots**.
5. Select **Create Snapshot**.
6. Enter a name for the snapshot.

   The name is limited to 64 characters and it must be unique among all other snapshots for this file system. The name can't be changed. Avoid entering confidential information.
7. Select **Create Snapshot**.

   The snapshot is accessible under the root directory of the file system at `.zfs/snapshot/`*name*.

Using the OCI CLI

Use the oci fs snapshot createoci_fs_snapshot_create command and required parameters to create a new snapshot of the specified file system.

```
oci fs snapshot create --file-system-id <file-system_OCID> --
name <snapshot_name> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateSnapshot operation to create a new snapshot of the specified file system.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Accessing a Snapshot on the Mounted File System

On Private Cloud Appliance, when a file system snapshot is created, the snapshot is placed in the file system. If the file system is mounted in a client system, you can access the snapshot on the client system.

The snapshot is accessible in this directory path: `<`*mount-point*`>/.zfs/snapshot/<`*snapshot-name*`>`.

From a UNIX OS Client

1. Log into the instance OS that has the mounted the file system from which the snapshot was made.

2. List the snapshots.

   Syntax:

   ```
   ls -la <mount-point>/.zfs/snapshot/
   ```

   Example:

   ```
   ls -la  /mnt/MyMountPoint/.zfs/snapshot
   total 17
   dr-xr-xr-x. 4 root root 4 Sep  8 15:54 .
   dr-xr-xr-x. 4 root root 4 Sep  1 17:27 ..
   drwxr-xr-x. 4 root root 7 Sep  8 15:53 file-system-FS-snapshot-02
   drwxr-xr-x. 4 root root 6 Sep  1 18:12 file-system-FS-snapshot-01
   ```

3. Change to the directory of a snapshot.

   Example:

   ```
   cd /mnt/MyMountPoint/.zfs/snapshot/file-system-FS-snapshot-02
   ```

4. List the contents of the snapshot.

   ```
   ls -la
   total 3027
   drwxr-xr-x. 4 root root       7 Sep  8 15:53 .
   dr-xr-xr-x. 4 root root       4 Sep  8 15:54 ..
   -rwxr-xr-x. 1 root root     429 Sep  8 15:53 example1
   drwxr-x---. 2 root sys        3 Sep  1 17:28 .$EXTEND
   drwxr-xr-x. 2 root root       2 Sep  1 18:10 ABC-directory
   -rw-r--r--. 1 root root       0 Sep  1 18:10 xyz-file
   -rw-r--r--. 1 root root 3073219 Sep  1 18:12 zap.zip
   ```

Restoring a Snapshot (UNIX-Based Instances)

On Private Cloud Appliance, you can restore individual snapshot files or an entire snapshot using the `cp` command.

> **Note:**
>
> Optionally, you can use `rsync`, `tar`, or another tool that supports NFS to copy your data to another remote location.

Using the Instance OS

1. Log into the instance OS that has the mounted the file system from which the snapshot was made.
2. List the snapshots.

   Syntax:

   ```
   ls -la <mount-point>/.zfs/snapshot/
   ```

   Example:

   ```
   ls -la  /mnt/MyMountPoint/.zfs/snapshot
   total 17
   dr-xr-xr-x. 4 root root 4 Sep  8 15:54 .
   dr-xr-xr-x. 4 root root 4 Sep  1 17:27 ..
   drwxr-xr-x. 4 root root 7 Sep  8 15:53 file-system-FS-snapshot-02
   drwxr-xr-x. 4 root root 6 Sep  1 18:12 file-system-FS-snapshot-01
   ```

3. Use the `cp` command to copy individual snapshot files, or the entire snapshot to a location of your choice.

   Use the `-r` option when restoring a snapshot that contains subdirectories.

Example:

```
cp -r /mnt/MyMountPoint/.zfs/snapshot/<snapshot_name>/
* <destination_directory>
```

Deleting a Snapshot

On Private Cloud Appliance, you can delete file system snapshots.

There are dependencies between file systems, snapshots, and clones. Private Cloud Appliance doesn't allow you to delete any resources for which there is a dependency. For details, see Deleting File System Resources Overview on page 758.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **File Storage**, then select **File Systems**.
2. At the top of the page, select the compartment that contains the file system snapshot that you want to delete.
3. Select the name of the file system where the snapshot resides.
4. Under **Resources**, select **Snapshots**.
5. Select the Actions menu (⋮), then select **Delete**.
6. Confirm the deletion.

Using the OCI CLI

Use the oci fs snapshot delete command and required parameters to delete the specified snapshot.

```
oci fs snapshot delete --snapshot-id <snapshot_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteSnapshot operation to delete the specified snapshot.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing File System Clones

On Private Cloud Appliance, a clone is a new file system that is created based on a snapshot of an existing file system. Snapshots preserve the state of the data of a file system at a particular point in time. If you take snapshots of a file system at regular intervals, you can create clones of the file system as it existed at multiple points in its lifetime.

Cloned file systems are managed in the same way that any other file system is managed. See Managing File Systems on page 786.

Creating a File System Clone

On Private Cloud Appliance, you can create a file system clone by cloning the file system snapshot.

Prerequisite

A snapshot of the file system must exist. See Creating a Snapshot on page 790.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **File Storage**, then select **File Systems**.
2. At the top of the page, select the compartment that contains the file system you want to clone.
3. Select the name of the file system that you want to clone.
4. Under **Resources**, select **Snapshots**.
5. For the snapshot that you want to clone, Select the Actions menu (⋮), and select **Clone**.
6. In the **Clone Snapshot from File System** dialog box, provide the following information:

- **Name**: Enter a name for the clone. Avoid entering confidential information.
- **Create in Compartment**: Select the compartment where you want to create the clone.
- **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.

7. Select **Create File System**.

   The clone is created and the details page of the new file system clone is displayed.

   In the **Clones** section of the file system clone details page, **Clone Parent** is false and **Clone Root** is false. The name of the **Parent File System** is shown and is selectable, and the name of the **Source Snapshot** is shown and is selectable.

   If you select the name of the parent file system, the **Clones** section of the details page of that parent file system shows **Clone Parent** is true, and **Parent File System** and **Source Snapshot** have no values.

Using the OCI CLI

Use the oci fs file-system create command and required parameters to create a file system clone of a file system snapshot.

```
oci fs file-system create --availability-domain <availability_domain_name>
 --compartment-id <compartment_id> --display-name <fs_clone_display_name> --
source-snapshot-id <fs_snapshot_id> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateFileSystem operation to create a file system clone of a file system snapshot.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a File System Clone

On Private Cloud Appliance, cloned file systems are managed; you delete a clone the same way you delete a file system. However, dependencies exist between file systems, snapshots, and clones. You can't delete any resources that have a dependency.

See Deleting a File System on page 789.

For details about dependencies, see Deleting File System Resources Overview on page 758.

Object Storage

On Private Cloud Appliance, the Object Storage service provides reliable and cost-efficient data durability.

The Object Storage service stores unstructured data of any content type, including analytic data and rich content, such as images and videos.

The data is stored as an object in a bucket. Buckets are associated with a compartment within a tenancy.

An Object Storage namespace serves as the top-level container for all buckets and objects. At account creation time, each tenant is assigned one unique system-generated and immutable Object Storage namespace name. The namespace spans all compartments.

With Object Storage, you can safely and securely store and retrieve data directly from the internet or from within Private Cloud Appliance.

Object Storage Resources

The Object Storage service uses these components to organize the data stored in object storage:

Objects

Any type of data, regardless of content type, is stored as an object.

An object is composed of the object itself and metadata about the object. Each object is stored in a bucket. The object is processed as a single entity.

Buckets

Buckets are logical containers for storing objects.

Users or systems create buckets as needed. A bucket is associated with a single compartment that has policies that determine what actions a user can perform on a bucket and on all the objects in the bucket. Buckets can't be nested.

You determine the bucket names, but each bucket name must be unique within a namespace.

Namespace

An Object Storage namespace serves as the top-level container for all buckets and objects.

When your tenancy is provisioned, the tenancy is assigned one unique system-generated and immutable Object Storage namespace name. The namespace spans all compartments within the tenant.

Within a namespace, buckets and objects exist in flat hierarchy, but you can simulate a directory structure to help navigate a large set of objects.

Compartment

A compartment is the primary building block used to organize your cloud resources.

When your tenancy is provisioned, a root compartment is created for you. You can then create compartments under your root compartment to organize your resources. You control access by creating policies that specify what actions groups of users can take on the resources in those compartments. An Object Storage bucket can only exist in one compartment.

Obtaining the Object Storage Namespace

On Private Cloud Appliance, an Object Storage namespace serves as the top-level container for all buckets and objects. Each tenant is assigned one unique system-generated and immutable Object Storage namespace name. The namespace spans all compartments. The namespace name is a required argument for many Object Storage OCI CLI commands.

Using the Compute Web UI

1. In the Compute Web UI, select your user name (upper right corner), then select **Tenancy**.

   The namespace string is listed under Object Storage Settings.

Using the OCI CLI

Run the following command and add the `iaas` endpoint to the command. For example:

```
oci os ns get --endpoint https://iaas.<mysystem>.example.com
{
  "data": "<myobjstor_namespace_name>"
}
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Managing Object Storage Buckets

On Private Cloud Appliance, a bucket is a container for storing objects in a compartment within an Object Storage namespace.

A bucket is associated with a single compartment. The compartment has policies that indicate what actions you can perform on a bucket and all objects in the bucket.

A bucket can't contain other buckets.

Creating a Bucket

On Private Cloud Appliance, you can create Object Storage buckets.

When you create a bucket, the bucket doesn't provide public access. To make the bucket publicly available, see Using Preauthenticated Requests on page 811.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Object Storage**, then select **Object Storage**.
2. Select **Create Bucket**.
3. Enter the following details:
   - **Name:** Enter a name for the bucket.

     Specify a name that's unique within your tenancy Object Storage namespace.
   - **Create in Compartment:** Select the compartment in which to create this bucket.
   - **Enable Object Versioning:** Optionally, you can enable object versioning.

     For more information, see Managing Object Versioning on page 807.
   - **Tagging:** (Optional) Add defined or free-form tags for this resource as described in Adding Tags at Resource Creation on page 458. Tags can also be applied later.
4. Select **Create Bucket**.

   The bucket is created immediately and you can start uploading objects. See Uploading an Object on page 800.

Using the OCI CLI

Use the oci os bucket create command and required parameters to create a bucket in the particular namespace with a bucket name and optional user-defined metadata.

```
oci os bucket create --namespace-name <object_storage_namespace> --
compartment-id <compartment_OCID> --name <bucket_name> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateBucket operation to create a bucket in the given namespace with a bucket name and optional user-defined metadata.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Listing Buckets

On Private Cloud Appliance, you can list buckets.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Object Storage**, then select **Object Storage**.

   A list of the buckets in the compartment you're viewing is displayed.
2. If you don’t see the bucket you're looking for, ensure that you’re viewing the correct compartment (select the compartment at the top of the page).

   The page shows only the resources in that compartment.

Using the OCI CLI

Use the oci os bucket list command and required parameters to get a list of all bucket summary items in a compartment. A bucket summary contains only summary fields for the bucket and does not contain fields like the user-defined metadata.

```
oci os bucket list --namespace-name <object_storage_namespace> --
compartment-id <compartment_OCID> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListBuckets operation to get a list of all bucket summary items in a compartment. A bucket summary contains only summary fields for the bucket and doesn't contain fields such as the user-defined metadata.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Getting Bucket Details

On Private Cloud Appliance, you can view bucket details.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Object Storage**, then select **Object Storage**.
2. At the top of the page, select the compartment that contains the bucket.
3. Select the bucket name to display the details.
4. Select **View** or **Copy**.

Using the OCI CLI

Use the oci os bucket get command and required parameters to get the current representation of the given bucket in the given Object Storage namespace.

```
oci os bucket get --namespace-name <object_storage_namespace> --bucket-
name <bucket_name> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the GetBucket operation to get the current representation of the given bucket in the given Object Storage namespace.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Moving a Bucket to a Different Compartment

On Private Cloud Appliance, you can move a bucket from one compartment to another as long as both the source and target compartments are in the same tenancy. This capability includes moving a bucket from one compartment level down to a lower level within the source compartment.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os bucket update command and required parameters to move a bucket from one compartment to another within the same tenancy. Supply the compartment Id of the compartment that you want to move the bucket to.

```
oci os bucket update --namespace-name <object_storage_namespace> --
compartment-id <target_compartment_id> --bucket-name <bucket_name> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateBucket operation to move a bucket from one compartment to another within the same tenancy. Supply the compartment Id of the compartment that you want to move the bucket to.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Bucket

On Private Cloud Appliance, you can permanently delete an empty bucket.

> **Caution:**
>
> You can't recover a deleted bucket.

You can't delete a bucket that contains any of the following:

- Any objects
- Previous versions of an object
- A multipart upload in progress
- A preauthenticated request

> **Tip:**
>
> When you delete an object in a version-enabled bucket, a previous version of that object is created. Select Show Deleted Objects to display the object versions that might prevent you from deleting the bucket. For more information, see Managing Object Versioning on page 807.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os bucket delete command and required parameters to delete a bucket if the bucket is empty.

```
oci os bucket delete --namespace-name <object_storage_namespace> --bucket-
name <bucket_name>[OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteBucket operation to delete a bucket if the bucket is empty.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing Storage Objects

On Private Cloud Appliance, you can manage objects in a bucket using the Compute Web UI, OCI CLI, and API.

Object Naming Prefixes and Hierarchies

Within an Object Storage namespace, buckets and objects exist in a flat structure. However, you can simulate a directory structure by adding a prefix string that includes one or more forward slashes (/) to an object name. Doing so lets you list one directory at a time, which is helpful when navigating a large set of objects.

For example:

```
marathon/finish_line.jpg
```

```
marathon/participants/p_21.jpg
```

If you add prefixes to object names, you can:

- Use the OCI CLI or API to perform bulk downloads and bulk deletes of all objects at a specified level of the hierarchy.
- Use the Compute Web UI to display a hierarchical view of your objects in virtual folders. In the previous example, `marathon` would be displayed as a folder containing an object named `finish_line.jpg` and `participants` would be a subfolder of `marathon`, containing an object named `p_21.jpg`. You can bulk upload objects to any level of the hierarchy and perform bulk deletes of all the objects in a bucket or folder.

Bulk operations at a specified level of the hierarchy do not affect objects in any level above.

When naming objects, you can also use prefix strings without a delimiter. No delimiters allow search operations and certain bulk operations in to match on the prefix portion of the object name. For example, in the object names below, the string `gloves_27_` can serve as a prefix for matching purposes when performing bulk operations:

```
gloves_27_dark_green.jpg
gloves_27_light_blue.jpg
```

When you perform bulk uploads, you can add a prefix string to the names of the files you are uploading.

Object Names

Unlike other resources, objects don't have Cloud Identifiers (OCIDs). Instead, you define an object name when you upload an object.

Use the following guidelines when naming an object:

- The maximum length for an object and bucket name is 255 characters.
- Valid characters are letters (upper or lowercase), numbers, and characters other than line feed, carriage return, and NULL.
- Bucket names and object names are case-sensitive.
- Use only Unicode characters for which the UTF-8 encoding doesn't exceed 1024 bytes. Clients are responsible for URL-encoding characters.
- Make the name unique within the bucket.
- Object names can include one or more slash (/) characters in the name. See Object Naming Prefixes and Hierarchies on page 797.
- At the top of the page, select the compartment in names.

Optional Response Headers and Metadata

When you upload objects, you can provide optional response headers and user-defined metadata. Response headers are HTTP headers sent from Object Storage to Object Storage clients when objects are downloaded.

User-defined metadata are name-value pairs stored with an object.

> **Important:**
>
> No validation is performed on the response headers or metadata you provide.

You can specify values for the following response headers:

- **Content-Disposition**

  Defines presentation only information for the object. Specifying values for this header has no effect on Object Storage behavior. Programs that read the object determine what to do based on the value provided. For example, you could use this header to let users download objects with custom file names in a browser. For Example:

  ```
  attachment; filename="fname.ext"
  ```

- **Cache-Control**

Defines the caching behavior for the object. Specifying values for this header has no effect on Object Storage behavior. Programs that read the object determine what to do based on the value provided. For example, you could use this header to identify objects that require caching restrictions. For Example:

```
no-cache, no-store
```

Metadata

You specify user-defined metadata in the form of name-value pairs. User-defined metadata names are stored and returned to Object Storage clients with the mandatory prefix of `opc-meta-`.

Viewing Objects in a Bucket

On Private Cloud Appliance, you can view objects in a bucket.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Object Storage**, then select **Object Storage**.
2. At the top of the page, select the compartment that contains the bucket that contains your object.

   A list of buckets is displayed.
3. Select the bucket name that contains your object.
4. Under **Resources**, select **Objects**.

   The objects are displayed.

Using the OCI CLI

Use the oci os object list command and required parameters to list objects in a bucket.

```
oci os object list --namespace-name <object_storage_namespace> --bucket-
name <bucket_name> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Viewing Objects Using the API

Use the ListObjects operation to list objects in a bucket.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Creating a Folder or Subfolder

On Private Cloud Appliance, you can create a folder in a bucket or you can create a subfolder in an existing folder or subfolder using the OCI CLI and API.

Avoid entering confidential information in names and tags.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os object put command and required parameters to create a folder in a bucket or a subfolder in an existing folder or subfolder.

Syntax:

```
oci os object put --namespace-name<object_storage_namespace>--bucket-
name<bucket_name>--file<file_location>--name<object_name>[OPTIONS]
```

Example:

```
oci os object put  --namespace-name examplenamespace  --bucket-name
 Bucket1_objv-enabl  --file /home/log_files/install.log  --name /home/
log_files/install.log
```

```
oci os object put  --namespace-name examplenamespace  --bucket-name
 Bucket1_objv-enabl  --file myfile  --name /home/log_files/install.log
```

```
oci os object put  --namespace-name examplenamespace  --
bucket-name Bucket1_objv-enabl  --file /home/log_files/
install.log  --name /home/log_files/install.log Uploading
 object [####################################] 100% { "etag":
 "bae04836d4ea5d521c23cbee70566cf2", "last-modified":
 "2021-05-13T15:37:18.000Z", "opc-content-md5": "GWZbZ8CXPCjLcPxBs6cPCQ==" }
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the PutObject operation to create a folder in a bucket or a subfolder in an existing folder or subfolder.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Uploading an Object

On Private Cloud Appliance, you can upload an object using the OCI CLI and API.

An object can be uploaded as a single part or as multiple parts. Use the `--no-multipart` option to upload as a single part. For detailed information on multipart uploads, see Performing a Multipart Upload on page 800.

> **Note:**
>
> Objects stored in Private Cloud Appliance Object Storage are compressed. You might notice that the size of an uploaded object is smaller that the size of the original object before it was uploaded.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os object put command and required parameters to upload an object to a bucket.

```
oci os object put --namespace-name <object_storage_namespace> --bucket-
name <bucket_name> --file <file_location> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the PutObject operation to upload an object to a bucket.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Performing a Multipart Upload

On Private Cloud Appliance, you can upload objects in multiple parts.

With multipart uploads, individual parts of an object can be uploaded in parallel to reduce the amount of time you spend uploading.

Multipart uploads accommodate objects that are too large for a single upload operation. Object parts must be no larger than 50 GiB.

You can pause between the uploads of individual parts, and resume the upload when your schedule and resources allow.

Object Parts

With multipart upload, you split the object you want to upload into individual parts. Individual parts can be as large as 50 GiB. The maximum size for an uploaded object is 10 TiB.

Decide what part number you want to use for each part. Part numbers can range from 1 to 10,000. You do not need to assign contiguous numbers, but Object Storage constructs the object by ordering part numbers in ascending order.

Multipart Upload API

Before you use the multipart upload API, you are responsible for creating the parts to upload. Object Storage provides API operations for the remaining steps.

A multipart upload performed using the API consists of the following steps:

1. Start an upload.
2. Upload object parts.
3. Commit the upload.

The service also provides API operations for listing in-progress multipart uploads, listing the object parts in an in-progress multipart upload, and aborting in-progress multipart uploads initiated through the API.

Multipart Upload OCI CLI

When you perform a multipart upload using the OCI CLI, you don't need to split the object into parts as you are required to do by the API. Instead, you specify the part size of your choice, and Object Storage splits the object into parts and performs the upload of all parts automatically. You can choose to set the maximum number of parts that can be uploaded in parallel. By default, the OCI CLI limits the number of parts that can be uploaded in parallel to three. When using the OCI CLI, you don't have to perform a commit when the upload is complete.

You can also use the OCI CLI to list in-progress multipart uploads, and to cancel multipart uploads initiated through the API.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os object put command and required parameters to upload an object in multiple parts.

Syntax:

```
oci os object put --namespace-name <object_storage_namespace> --
bucket-name <bucket_name> --file <file_location> --parallel-upload-
count <maximum_number_parallel_uploads> --part-size <upload_part_size_in_MB>
 --force [OPTIONS]
```

Example:

```
oci os object put   \
--namespace-name examplenamespace  \
--file /boot/initramfs-0-rescue-e542c19f0fbf4e41a41428d933a7357f.img   \
--parallel-upload-count 5   \
--part-size 15   \
--force
```

```
Upload ID: a21bba2c-8922-4b9c-a98a-9ef3569c0138
Split file into 6 parts for upload.
Uploading object [####################################] 100%
{
  "etag": "0964effc8dc4394fd317f03a025ae5d0",
  "last-modified": "2021-05-11T21:35:19",
  "opc-multipart-md5": "UIVRhiwSHY6o0E4pi/yfGg==-6"
}
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

A multipart upload performed using the API consists of the following steps:

1. Starting an upload.
2. Uploading object parts.
3. Committing the upload.

Before you use the multipart upload API, you're responsible for creating the parts to upload. Object Storage provides API operations for the remaining steps. The service also provides API operations for listing in-progress multipart uploads, listing the object parts in an in-progress multipart upload, and aborting in-progress multipart uploads initiated through the API.

> **Note:**
>
> These multipart upload API requirements do not apply to the OCI CLI.

For more information about using the multipart Upload API, see Using the Multipart Upload API.

Use the PutObject operation to upload an object in multiple parts.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Listing Multipart Uploads in Object Storage

On Private Cloud Appliance, you can view a list of the in-progress objects in multipart uploads for an Object Storage bucket using the OCI CLI and API.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os multipart list command and required parameters to list all the in-progress objects in a multipart upload.

Syntax:

```
oci os multipart list --namespace-name <object_storage_namespace> --bucket-
name <bucket_name> [OPTIONS]
```

Example

```
oci os multipart list --namespace-name examplenamespace  --bucket-name
 MyBucket
{
  "data": [
    {
      "bucket": "MyBucket",
      "namespace": "examplenamespace",
      "object": "MyObject",
      "time-created": "2019-07-25T21:55:21.973000+00:00",
      "upload-id": "0b7abd48-9ff2-9d5f-2034-63a02fdd7afa"
```

```
    },
    {
      "bucket": "MyBucket",
      "namespace": "examplenamespace",
      "object": "MyObject",
      "time-created": "2019-07-25T21:53:09.246000+00:00",
      "upload-id": "1293ac9d-83f8-e055-a5a7-d1e13277b5c0"
    },
    {
      "bucket": "MyBucket",
      "namespace": "examplenamespace",
      "object": "MyObject",
      "time-created": "2019-07-25T21:46:34.981000+00:00",
      "upload-id": "33e7a875-9e94-c3bc-6577-2ee5d8226b53"
    }
...
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListMultipartUploadParts operation to list all the in-progress objects in a multipart upload.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Performing a Bulk Object Upload

On Private Cloud Appliance, you can perform bulk object uploads to a bucket using the OCI CLI.

Bulk operations at a specific level of the hierarchy do not affect objects in any level above.

Using the OCI CLI

Use the oci os object bulk-upload command and required parameters to bulk upload objects.

Syntax:

```
oci os object bulk-upload --namespace-name <object_storage_namespace> --
bucket-name <bucket_name> --src-dir <source_directory_location> [OPTIONS]
```

Example:

```
oci os object bulk-upload
--namespace-name examplenamespace  \
--bucket-name MyBucket    \
--src-dir /home/log-dir/

Uploaded Jan-logs [####################################] 100%
Uploaded Feb-logs [####################################] 100%
Uploaded Mar-logs [####################################] 100%
Uploaded Apr-logs [####################################] 100%

{
  "skipped-objects": [],
  "upload-failures": {},
  "uploaded-objects": {
    "Jan-logs": {
      "etag": "33ed1aff724eac56f00616552fc61f3e",
      "last-modified": "2021-06-01T20:42:50.000Z",
      "opc-content-md5": "Ucf+fZbCK/RN5gGsEl7G5w=="
    },
    "Feb-logs": {
      "etag": "e1875449257cc6ac6ab93cc9c7921c87",
```

```
      "last-modified": "2021-06-01T20:42:50.000Z",
      "opc-content-md5": "1B2M2Y8AsgTpgAmY7PhCfg=="
    },
    "Mar-logs": {
      "etag": "c784ac5216d889f55138ecfb428eee3c",
      "last-modified": "2021-06-01T20:42:51.000Z",
      "opc-content-md5": "1B2M2Y8AsgTpgAmY7PhCfg=="
    },
    "Apr-logs": {
      "etag": "3b4571c73bdb9e44bec0512a5e48fba7",
      "last-modified": "2021-06-01T20:42:51.000Z",
      "opc-content-md5": "1B2M2Y8AsgTpgAmY7PhCfg=="
    }
  }
}
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Copying an Object to a Different Bucket

On Private Cloud Appliance, you can copy an object to a different bucket as long as the target bucket is located in the same Private Cloud Appliance appliance.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os object copy command and required parameters to copy an object to a different bucket.

```
oci os object copy --namespace-name <object_storage_namespace> --
bucket-name <source_bucket_name> --source-object-name <source_object>
 --destination-bucket <destination_bucket_name> --destination-object-
name <destination_object_name> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CopyObject operation to copy an object to a different bucket.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Downloading an Object

On Private Cloud Appliance, you can download an object using the OCI CLI and API.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os object get command and required parameters to download an object.

Syntax:

```
oci os object get --namespace-name <object_storage_namespace> --bucket-
name <bucket_name> --name<object_name> --file<file_location> [OPTIONS]
```

<*file_location*> is the destination path for the file being downloaded, such as `C:\workspace\Downloads\MyFile.txt` or `/home/user/Documents/Downloads/MyFile.txt`.

Example:

```
oci os object get  \
--namespace-name examplenamespace  \
--bucket-name MyBucket  \
--name photos  \
--file /home/photos_backup

Downloading object [#----------------------------------] 100%

# ls -l
total 8
-rw-r--r-- 1 root root 1363 Jun 1 17:56 photo1
-rw-r--r-- 1 root root 1363 Jun 1 21:40 photo1_backup
-rw-r--r-- 1 root root 0 Jun 1 20:42 photo2
-rw-r--r-- 1 root root 0 Jun 1 20:42 photo3
-rw-r--r-- 1 root root 0 Jun 1 20:42 photo4
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the GetObject operation to download an object.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Performing a Multipart Download

On Private Cloud Appliance, you can download an object in multiple parts using OCI CLI and API operations.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os object get command and required parameters to operation to download an object in multiple parts.

Syntax (entered on a single line):

```
oci os object get --namespace-name <object_storage_namespace> --bucket-
name <bucket_name> --name <object_name> --file <file_location> --range
 bytes=<byte_range> [OPTIONS]
```

The byte-range for the download. Multipart object downloading is available using the byte-range request standard defined in RFC 7233, section 2.1

Example:

```
oci os object get    \
--namespace-name examplenamespace  \
--bucket-name MyBucket    \
--name MyObject.mp4    \
--file c:\workspace\Downloads\MyObject.mp4    \
--range bytes=0-50

cusobjstorenamespace --range bytes=0-50
Downloading object [#----------------------------------] 3%

# ls -l
total 12
-rw-r--r-- 1 root root 1363 Jun 1 17:56 abc.mp41
-rw-r--r-- 1 root root 51 Jun 1 21:50 def.mp4
-rw-r--r-- 1 root root 1363 Jun 1 21:40 ghi.mp4
```

```
-rw-r--r-- 1 root root 0 Jun 1 20:42 jkl.mp4
-rw-r--r-- 1 root root 0 Jun 1 20:42 mno.mp4
-rw-r--r-- 1 root root 0 Jun 1 20:42 pqr.mp4
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use theGetObject operation to download an object in multiple parts.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Performing a Bulk Download

On Private Cloud Appliance, you can download objects in bulk.

Using the OCI CLI

Use the oci os object copy command and required parameters to perform a bulk download.

```
oci os object bulk-download
--namespace-name <object_storage_namespace>
--bucket-name <bucket_name>
--download-dir <download_directory_location> [OPTIONS]
```

The download directory (<*download_directory_location*>) is the destination path for the objects being downloaded, such as `C:\workspace\Downloads\` or `/home/user/Documents/Downloads/`. If the directory does not exist, Object Storage creates the directory when the command runs.

Example:

```
oci os object bulk-download    \
--namespace-name examplenamespace  \
--bucket-name MyBucket   \
--download-dir c:\workspace\Downloads

Downloaded MyFile.txt [####################################] 100%
Downloaded logFile.log [####################################] 100%

{
  "download-failures": {},
  "skipped-objects": []
}
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Deleting an Object

On Private Cloud Appliance, you can permanently delete an object from a bucket or folder.

> **Caution:**
>
> You can't recover a deleted object unless you have object versioning enabled. See Managing Object Versioning on page 807 for details.

You can't delete an object that has an active retention rule. See Defining Retention Rules on page 817 for details.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os object delete command and required parameters to delete an object or folder.

```
oci os object delete --namespace-name <object_storage_namespace> --bucket-
name <bucket_name> --object-name <object_name> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteObject operation to delete an object or folder.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Bulk Deleting All Objects in a Bucket

On Private Cloud Appliance, you can delete all objects in a bucket using the OCI CLI.

> **Caution:**
>
> You can't recover a deleted object unless you have object versioning enabled. See Managing Object Versioning on page 807 for details.

You can't delete an object that has an active retention rule. See Defining Retention Rules on page 817 for details.

Using the OCI CLI

Use the oci os object bulk-delete command and required parameters to delete an object or folder.

To see a list of the files impacted by a bulk delete command without actually deleting the files, use the `--dry-run` option. Perform the command without the `--dry-run` option to bulk delete all objects in the bucket.

```
oci os object bulk-delete --namespace-name <object_storage_namespace> --
bucket-name <bucket_name> --dry-run
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Managing Object Versioning

On Private Cloud Appliance, object versioning provides data protection against accidental or malicious object update, overwrite, or deletion.

Object versioning is enabled at the bucket level. Versioning directs Object Storage to automatically create an object version each time one of these actions takes place:

- A new object is uploaded.
- An existing object is overwritten.
- When an object is deleted.

You can enable object versioning at bucket creation time or later.

A bucket that is versioning-enabled can have many versions of an object. There is always one latest version of the object and zero or more previous versions.

Scope and Constraints

- Versioning can be enabled on a bucket in Object Storage.
- You can rename the latest version of an object, but you can't rename a previous object version. Renaming an object creates a new object.

Interaction Between Versioning and Other Object Storage Features

This section describes some key things you need to know about the interaction between object versioning and other Object Storage features.

Copying Objects

If you copy the latest version of an object to a different bucket, only the object is copied. None of the object's previous versions are copied. You can copy a previous version of an object to another bucket, but that action creates either the latest version of a new object or a new object version in the destination bucket.

Retention Rules

- You can't add retention rules to a bucket that has versioning enabled.
- You can't enable versioning on a bucket with active retention rules.
- You can add retention rules to bucket that has versioning suspended. However, you can't resume versioning with active retention rules.

Enabling Versioning During Bucket Creation

On Private Cloud Appliance, you can enable object versioning during bucket creation using the OCI CLI and API.

Object versioning provides data protection against accidental or malicious object updates and deletions.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os bucket create command and required parameters to enable object versioning during bucket creation.

```
oci os bucket create --namespace-name <object_storage_namespace> --
compartment-id <target_compartment_id> --name <bucket_name> --versioning
 enabled [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateBucket operation to enable object versioning during bucket creation.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Enabling or Suspending Versioning

On Private Cloud Appliance, object versioning provides data protection against accidental or malicious object updates and deletions.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os bucket update command and required parameters to enable or suspend versioning.

For the `--versioning` option, choose one of these values: `Enabled` or `Suspended`.

```
oci os bucket update --namespace-name <object_storage_namespace> --
compartment-id <target_compartment_id> --bucket-name <bucket_name> --
versioning <enabled | suspended> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateBucket operation to enable or suspend versioning.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing Object Versions and Details

On Private Cloud Appliance, you can list object versions and details using the OCI CLI and API.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os object list-object-versions command and required parameters to list the object versions in a bucket.

```
oci os object list-object-versions --namespace-
name <object_storage_namespace> --bucket-name <bucket_name> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListObjectVersion operation to list the object versions in a bucket.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting the Previous Version of an Object

On Private Cloud Appliance, when versioning is enabled, deleting an object without targeting a specific version creates a delete marker and a previous version of the object that object can be recovered. However, deleting a previous version of an object is a permanent deletion.

No object is physically deleted from a bucket that has versioning enabled until you take explicit action to do so.

When you delete an object without targeting a specific version, the latest object version becomes a previous object version and a special delete marker is created that marks the deletion point. A delete marker contains only minimal metadata. If you delete a folder, a delete marker is created for each object in the folder. You can simply delete the delete marker to make that deleted version become the latest object version.

When you upload an object with the same name as the delete marker, the uploaded object becomes the latest version of the object. The delete marker remains. There can be multiple delete markers for an object and you can recover any of the previous object versions.

Object version deletion is different. When you delete an object version, the version is permanently deleted. Permanent deletion also happens if you explicitly delete the latest version by version ID. All delete operations that target a specific object version ID permanently deletes the data.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os object delete command and required parameters to delete the previous version of an object.

If an object has a `version-id` of `null`, there is only one version of the object. To delete this object, omit the `--version-id` argument.

```
oci os object delete --namespace-name <object_storage_namespace> --
bucket-name <bucket_name> --version-id <bucket_version_id> --object-
name <object_name> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteObject operation to delete the previous version of an object.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Recovering a Deleted Object Version

On Private Cloud Appliance, recovering a deleted object version is as simple as deleting the delete marker that was created when you deleted the latest version of an object. The previous version of the object listed just below the delete marker is recovered and becomes the latest version of the object.

To perform this task, you need to know which object is marked for deletion. To obtain that information, list the objects in the bucket. See Viewing Object Versions and Details on page 809. In the output, locate the object version that has `"is-delete-marker": true`.

Use the version-id of that object with the delete command to delete the delete marker.

> **Note:**
>
> If an object has a `version-id` of `null`, there is only one version of the object. To delete this object marker, omit the `--version-id` argument.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os object delete command and required parameters to remove the delete marker from an object.

Syntax:

```
oci os object delete --namespace-name <object_storage_namespace>
 --bucket-name <bucket_name> --object-name <object_name> --version-
id <bucket_version_id> [OPTIONS]
```

Example:

```
oci os object delete
--namespace-name examplenamespace  \
--bucket-name  MyBucket
--object-name application.log
--version-id  6ce3eb93-8850-4732-8949-uniqueID
Are you sure you want to delete this resource? [y/N]: y
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteObject operation to remove the delete marker from an object.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Troubleshooting Versioning

Identify and address common issues that can occur while working with object versioning On Private Cloud Appliance.

Unable to Enable Versioning

If enabling versioning fails, the most likely cause is missing or incomplete IAM permissions. Enabling versioning requires:

- User permissions that let you use the bucket and manage the objects in that bucket.
- BUCKET_UPDATE permissions.

Unable to Delete a Bucket

If deleting a bucket fails, the most likely cause is that the bucket is not empty.

You can permanently delete an empty bucket. You can't delete a bucket that contains any of the following:

- Any objects
- Previous versions of an object
- A multipart upload in progress
- A preauthenticated request

> **Tip:**
>
> When you delete an object in a version-enabled bucket, a previous version of that object is created. Select Show Deleted Objects to display the object versions that might prevent you from deleting the bucket.

Unable to Delete a Previous Version

If deleting a previous object version fails, the most likely cause is missing or incomplete IAM permissions. Object version deletion requires:

- User permissions that let you use the bucket and manage the objects in that bucket.
- At minimum, OBJECT_VERSION_DELETE permissions.

Using Preauthenticated Requests

On Private Cloud Appliance, preauthenticated requests provide a way to let users access a bucket or an object without having their own credentials, as long as the request creator has permissions to access those objects.

For example, you can create a request that lets an operations support user upload backups to a bucket without owning API keys. Or, you can create a request that lets a business partner update shared data in a bucket without owning API keys.

When you create a preauthenticated request, a unique URL is generated. Anyone you provide this URL to can access the Object Storage resources identified in the preauthenticated request, using standard HTTP tools like curl and wget.

> **Important:**
>
> Assess the business requirement for and the security ramifications of preauthenticated access to a bucket or objects.
>
> A preauthenticated request URL gives anyone who has the URL access to the targets identified in the request. Carefully manage the distribution of the URL.

Required Permissions

To Create a Preauthenticated Request

You need the PAR_MANAGE permission to the target bucket or object.

You must also have the appropriate permissions for the access type that you are granting. For example:

- If you are creating a preauthenticated request for uploading objects to a bucket, you need the OBJECT_CREATE and OBJECT_OVERWRITE permissions.
- If you are creating a preauthenticated request for read/write access to objects in a bucket, you need the OBJECT_READ, OBJECT_CREATE, and OBJECT_OVERWRITE permissions.

> **Important:**
>
> If the creator of a preauthenticated request is deleted or loses the required permissions after they created the request, the request will no longer work.

To Use a Preauthenticated Request

Permissions of the preauthenticated request creator are checked each time you use a preauthenticated request.

The preauthenticated request no longer works when any of the following occurs:

- Permissions of the preauthenticated request creator have changed.
- The user who created the preauthenticated request is deleted.
- A Federated user who created the preauthenticated request has lost the user capabilities that they had when they created the request.
- Preauthenticated request has expired.

Types of Preauthenticated Requests

When creating a preauthenticated request, you have the following options:

- You can specify the name of a bucket that a preauthenticated request user has write access to and can upload one or more objects to.
- You can specify the name of an object that a preauthenticated request user can read from, write to, or read from and write to.

Scope and Constraints

Understand the following scope and constraints regarding preauthenticated requests:

- Users can't list bucket contents.
- You can create an unlimited number of preauthenticated requests.
- There is no time limit to the expiration date that you can set.
- You can't edit a preauthenticated request. If you want to change user access options in response to changing requirements, you must create a new preauthenticated request.
- The target and actions for a preauthenticated request are based on the creator's permissions. The request isn't, however, bound to the creator's account login credentials. If the creator's login credentials change, a preauthenticated request isn't affected.
- You can't delete a bucket that has a preauthenticated request associated with that bucket or with an object in that bucket.

> **Important:**
>
> The unique URL provided by the system when you create a preauthenticated request is the only way a user can access the bucket or object specified as the request target. Copy the URL to durable storage. The URL is displayed only at the time of creation and can't be retrieved later.

Creating a Preauthenticated Request for All Objects in a Bucket

On Private Cloud Appliance, you can create preauthenticated requests for all objects in a bucket using the OCI CLI and API.

> **Important:**
>
> Immediately after creating the request, copy the `access-uri` to durable storage.
>
> The unique `access-uri` provided by the system is the only way to construct a URL that a user can use to access the bucket or object specified as the request target.
>
> **The `access-uri` is displayed only at the time of creation and can't be retrieved later.**

To construct a URL from the unique `access-uri`, see Constructing the Preauthenticated Request URL on page 815.

> **Note:**
>
> Listing objects in a bucket is denied by default. If the `--access-type` is `AnyObjectRead` or `AnyObjectReadWrite`, you can specify the optional `--bucket-listing-action ListObjects` parameter when creating the preauthenticated request that lets users list the objects in the bucket.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os preauth-request create command and required parameters to create a preauthenticated request for all objects in a bucket.

```
oci os preauth-request create --namespace-name <object_storage_namespace> --
bucket-name <bucket_name> --name <preauthenticated_request_name> --access-
type <access_value> --time-expires <timestamp> [OPTIONS]
```

For access type, use one of these values:

- `AnyObjectRead` permits reads on all objects in the bucket.
- `AnyObjectWrite` permits writes to all objects in the bucket.
- `AnyObjectReadWrite` permits reads and writes to all objects in the bucket.

`time-expires` is a required argument and must be an RFC 3339 timestamp. For example: `2017-09-01T00:09:51.000+02:00`.

This example creates a preauthenticated request that allows reads and writes to all objects in the bucket:

```
 oci os preauth-request create --namespace-name examplenamespace --bucket-
name MyBucket --name PAR-all-objectsRW --access-type AnyObjectWrite --time-
expires '2021-07-30 23:55'
{
  "data": {
    "access-type": "AnyObjectWrite",
    "access-uri": "/p/KOCRWzqBilJmIsaBbJNelKLWcOxwRLq/n/examplenamespace/b/
MyBucket/o/",
    "id": "783cd56b-9df5-4518-aacf-f523deae5102",
    "name": "PAR-all-objectsRW",
    "object-name": null,
    "time-created": "2021-06-10T20:49:11+00:00",
    "time-expires": "2021-07-30T23:54:59+00:00"
```

```
  }
}
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreatePreauthenticatedRequest operation to create a preauthenticated request for all objects in a bucket.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Creating a Preauthenticated Request for a Specific Object

On Private Cloud Appliance, you can create a preauthenticated request for a specific object in a bucket using the OCI CLI and API.

> **Important:**
>
> Immediately after creating the request, copy the `access-uri` to durable storage.
>
> The unique `access-uri` provided by the system is the only way to construct a URL that a user can use to access the bucket or object specified as the request target.
>
> **The `access-uri` is displayed only at the time of creation and can't be retrieved later.**

To construct a URL from the unique `access-uri`, see Constructing the Preauthenticated Request URL on page 815.

> **Note:**
>
> Listing objects in a bucket is denied by default. If the `--access-type` is `AnyObjectRead` or `AnyObjectReadWrite`, you can specify the optional `--bucket-listing-action ListObjects` parameter when creating the preauthenticated request that lets users list the objects in the bucket.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os preauth-request create command and required parameters to create a preauthenticated request for a specific object in a bucket.

```
oci os preauth-request create --namespace-name <object_storage_namespace> --
bucket-name <bucket_name> --name <preauthenticated_request_name> --access-
type <access_value> --time-expires <timestamp> -on <object_name_or_null>
 [OPTIONS]
```

For access type, use one of these values:

- `AnyObjectRead` permits reads on all objects in the bucket.
- `AnyObjectWrite` permits writes to all objects in the bucket.
- `AnyObjectReadWrite` permits reads and writes to all objects in the bucket.

`--time-expires` is a required argument and must be an RFC 3339 timestamp. For example: `2017-09-01T00:09:51.000+02:00`.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreatePreauthenticatedRequest operation to create a preauthenticated request for a specific object in a bucket.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Constructing the Preauthenticated Request URL

On Private Cloud Appliance, after you have a unique `access-uri`, you can construct the access URL that enables users to access preauthenticated objects. The `access-uri` is generated when you create a preauthenticated request.

Construct the URL using this syntax.

```
https://objectstorage.<ccc_fqdn><access-uri>
```

where:

- *<ccc_fqdn>* is the fully qualified domain name of your Private Cloud Appliance appliance.
- *<access-uri>* is the access URI that was obtained from one of these procedures:
  - Creating a Preauthenticated Request for All Objects in a Bucket on page 813
- Creating a Preauthenticated Request for a Specific Object on page 814

Example:

```
https://objectstorage.myccc01.us.example.com/p/
MrxLFkKlFkIlNDhvhcZnrjbUAlsoeah/n/mynamespace/b/my-bucket/o/my-object
```

Listing Preauthenticated Requests

On Private Cloud Appliance, you can list preauthenticated requests using the unique request URL.

For information about the unique request URL, see Constructing the Preauthenticated Request URL on page 815.

Using curl

- Syntax:

  ```
  $ curl -X GET <unique-PAR-URL>
  ```

  Example:

  ```
  $ curl -X GET    \
  https://objectstorage.us-example-1.example.com/p/
  CoO26YkSARiRevWlDWJD_QUvtFPUocn/n/examplenamespace/b/MyParBucket/o/

  {"objects":[{"name":"InfoWorld DeepDive - Tips  for Git and GitHub
   Users.pdf"},{"name":"OCI_User_Guide.pdf"},
  {"name":"OracleCorporateTerminologyUsageGuideRedwood.pdf"},
  {"name":"VPN.png"},{"name":"eventslogreference.htm"},
  {"name":"functionslogreference.htm"},{"name":"glob.txt"},
  {"name":"loadbalancerreference.htm"},
  {"name":"objectstoragelogreference.htm"},
  {"name":"servicechanges.html"},{"name":"servicediscovery.dita"},
  {"name":"serviceessentials.html"},{"name":"servicelogreference.htm"},
  {"name":"services.html"}]}
  ```

Listing Objects for Preauthenticated Requests

On Private Cloud Appliance, using the unique request URL, you can use a tool like curl to list, read, and write data using the preauthenticated request.

Using curl

- Syntax (entered on a single line):

```
$ curl -X GET <unique-PAR-URL>
```

Example:

```
$ curl -X GET    \
https://objectstorage.us-example-1.example.com/p/
CoO26YkSARiRevWlDWJD_QUvtFPUocn/n/examplenamespace/b/MyParBucket/o/

{"objects":[{"name":"InfoWorld DeepDive - Tips  for Git and GitHub
 Users.pdf"},{"name":"developer-reference.pdf"},
{"name":"OracleCorporateTerminologyUsageGuideRedwood.pdf"},
{"name":"VPN.png"},{"name":"eventslogreference.htm"},
{"name":"functionslogreference.htm"},{"name":"glob.txt"},
{"name":"loadbalancerreference.htm"},
{"name":"objectstoragelogreference.htm"},
{"name":"servicechanges.html"},{"name":"servicediscovery.dita"},
{"name":"serviceessentials.html"},{"name":"servicelogreference.htm"},
{"name":"services.html"}]}
```

Uploading an Object Using a Preauthenticated Request

On Private Cloud Appliance, using the unique request URL, you can use a tool like curl to read and write data using the preauthenticated request.

Using curl

- Syntax (entered on a single line):

```
$ curl -X PUT --data-binary '@<local-filename>' <unique-PAR-URL>
```

Example:

```
$ curl -X PUT    \
--data-binary '@using-dita-guide.pdf'    \
https://objectstorage.us-example-1.example.com/p/lnaqMuXWef_lhTxCiS9ngCw/
n/examplenamespace/b/MyParBucket/o/using-dita-guide.pdf
```

Downloading an Object Using a Preauthenticated Request

On Private Cloud Appliance, using the unique request URL, you can use a tool like curl to read and write data using the preauthenticated request.

Using curl

- Syntax (entered on a single line):

```
$ curl -X GET <unique-PAR-URL>
```

Example:

```
$ curl -X GET https://objectstorage.us.example.com/p/
tnjDhazP9o6s2KzLyFUxILQzSamEp/n/examplenamespace/b/MyParBucket/o/
developer_reference.pdf
'@data.1''@data.2''@data.3'
```

Deleting a Preauthenticated Request

On Private Cloud Appliance, you can delete preauthenticated requests.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os preauth-request delete command and required parameters to delete a preauthenticated request.

```
oci os preauth-request delete --namespace-name <object_storage_namespace> --
bucket-name <bucket_name> --par-id <preauthenticated_request_id> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeletePreauthenticatedRequest operation to delete a preauthenticated request.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Defining Retention Rules

On Private Cloud Appliance, retention rules provide immutable storage options for data written to Object Storage for data governance, regulatory compliance, and legal hold requirements. Retention rules can also protect your data from accidental or malicious writes or deletion. Retention rules can be locked to prevent rule modification and data deletion or modification even by administrators.

Retention rules are configured at the bucket level and are applied to all individual objects in the bucket.

Object Storage provides a flexible approach to data retention that supports the following use cases.

- **Regulatory Compliance**

  Your industry might require you to retain a certain class of data for a defined length of time. Your data retention regulations might also require that you lock the retention settings. If you lock the settings, the only change you can make is to increase the retention duration.

  For Object Storage regulatory compliance, you create a time-bound retention rule and specify a duration. Object modification and deletion are prevented for the duration specified. Duration is applied to each object individually, and is based on the object's Last Modified timestamp. Lock the rule as required.
- **Data Governance**

  You might need to protect certain datasets as a part of internal business process requirements. While retaining the data for a defined length of time is necessary, that time period could change.

  Create a time-bound retention rule and specify a duration. Object modification and deletion are prevented for the duration specified. Duration is applied to each object individually, and is based on the object's Last Modified timestamp. To be able to delete the rule and allow changes to the duration as required, do not lock the rule.
- **Legal Hold**

  You might need to preserve certain business data in response to potential or on-going lawsuits. A legal hold does not have a defined retention period and remains in effect until removed.

For Object Storage legal holds, you create an indefinite retention rule. Object modification and deletion are prevented until you delete the rule. You can't lock an indefinite retention rule because the rule has no duration.

It's important to understand retention duration for time-bound rules. Even though you are creating retention rules for a bucket, the duration of a rule is applied to each object in the bucket individually, and is based on the object's Last Modified timestamp. Let's say you have two objects in the bucket, ObjectX and ObjectY. ObjectX was last modified 14 months ago and ObjectY was last modified 3 months ago. You create a retention rule with a duration of 1 year. This rule prevents the modification or deletion of ObjectY for the next 9 months. The rule allows the modification or deletion of ObjectX because the retention rule duration (1 year) is less that the object's Last Modified timestamp (14 months). If ObjectX is overwritten some time in the coming year, modification and deletion would be prevented for the rule duration time remaining.

Locking a retention rule is an irreversible operation. Not even a tenancy administrator can delete a locked rule. There is a mandatory 14-day delay before a rule is locked. This delay lets you thoroughly test, modify, or delete the rule or the rule lock before the rule is permanently locked. A rule is active at the time of creation. The lock only controls whether the rule itself can be modified. After a rule is locked, only increases in the duration are allowed. Object modification is prevented and the rule can only be deleted by deleting the bucket. A bucket must be empty before it can be deleted.

Scope and Constraints

- Retention rules can be applied to a bucket in the Object Storage.
- The actions that you can perform on a bucket with active retention rules are limited. You can't update, overwrite, or delete objects or object metadata until the retention rule is deleted (indefinite rule) or for the duration specified (time-bound rules). The duration for time-bound rules is applied to each object individually, and is based on the object's Last Modified timestamp.
- You can create multiple retention rules for a bucket. Indefinite retention rule is applied before any time-bound rule is considered.
- When a retention rule is locked, the rule can only be deleted by deleting the bucket. A bucket must be empty before it can be deleted.

Interaction Between Retention and Other Object Storage Features

Carefully review the policies and rules that you have in place for the other Object Storage features that you are using. Some of these policies and rules might not make sense with retention rules. This section describes some key things you need to know about the interaction between retention rules and other Object Storage features.

Multipart Uploads

Uncommitted (unfinished or failed) multipart uploads are not protected by retention rules and can be deleted at any time.

Versioning

- You can't add retention rules to a bucket that has versioning enabled.
- You can't enable versioning on a bucket with active retention rules.
- You can add retention rules to bucket that has versioning suspended. However, you can't resume versioning with active retention rules.

Creating a Retention Rule

On Private Cloud Appliance, you can create Object Storage retention rules.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os retention-rule create command and required parameters to create a retention rule.

```
oci os retention-rule create --namespace-name <object_storage_namespace> --
bucket-name <bucket_name> --display-name <name_displayed_for_rule> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateRetentionRule operation to create a retention rule.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Viewing Retention Rules and Details

On Private Cloud Appliance, you can view Object Storage retention rules and details.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os retention-rule list command and required parameters to list retention rules.

```
oci os retention-rule list --namespace-name <object_storage_namespace> --
bucket-name <bucket_name> [OPTIONS]
```

Use the oci os retention-rule get command and required parameters to see the details for a retention rule.

```
oci os retention-rule get --namespace-name <object_storage_namespace> --
bucket-name <bucket_name>
--retention-rule-id <retention_rule_identifier> [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the ListRetentionRules operation to list retention rules.

Use the GetRetentionRule operation to get the details for a retention rule.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Modifying a Retention Rule

On Private Cloud Appliance, you can modify Object Storage retention rules.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os retention-rule update command and required parameters to update a retention rule.

Syntax:

```
oci os retention-rule update --namespace-name <object_storage_namespace> --
bucket-name <bucket_name> --retention-rule-id <retention_rule_id>
```

Followed by the retention rule item that you plan to change:

```
--time-amount <time_integer> --time-unit <days|years>
```

Example:

```
oci os retention-rule update  \
--namespace-name examplenamespace  \
--bucket-name MyBucket  \
--retention-rule-id 344a9c205187408699b51c7769dc1bb4  \
--time-amount 60  \
--time-unit days
{
  "data": {
    "display-name": "TempHold",
    "duration": {
      "time-amount": 60,
      "time-unit": "DAYS"
    },
    "etag": "344a9c205187408699b51c7769dc1bb4",
    "id": "344a9c205187408699b51c7769dc1bb4",
    "time-created": "2021-06-10T22:17:50+00:00",
    "time-modified": "2021-06-10T22:45:16+00:00",
    "time-rule-locked": null
  }
}
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateRetentionRule operation to update a retention rule.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting a Retention Rule

On Private Cloud Appliance, you can delete Object Storage retention rules.

Using the Compute Web UI

This task isn't available in the Compute Web UI.

Using the OCI CLI

Use the oci os retention-rule delete command and required parameters to delete a retention rule.

```
oci os retention-rule delete --namespace-name <object_storage_namespace> --
bucket-name <bucket_name> --retention-rule-id <retention_rule_identifier>
 [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteRetentionRule operation to delete a retention rule.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Troubleshooting Retention Rules

Identify and address common issues that can occur while working with Object Storage retention rules On Private Cloud Appliance.

Unable to Create a Retention Rule

If creating a retention rule fails, the most likely cause is missing or incomplete IAM permissions. Rule creation requires:

- User permissions that let you access the bucket and manage the objects in those buckets.
- Minimally, BUCKET_UPDATE and RETENTION_RULE_MANAGE permissions.
- Minimally, BUCKET_UPDATE and RETENTION_RULE_MANAGE permissions.

Unable to Lock a Retention Rule

If locking a retention rule fails, the most likely cause is missing or incomplete IAM permissions. Minimally, BUCKET_UPDATE, RETENTION_RULE_MANAGE, and RETENTION_RULE_LOCK permissions are required to lock retention rules.

Unable to Delete a Retention Rule

You can't delete a time-bound retention rule that's locked. When a retention rule is locked, the rule can only be deleted by deleting the bucket. A bucket must be empty before it can be deleted.

If deleting an indefinite retention rule fails, the most likely cause is missing or incomplete IAM permissions. Rule deletion requires:

- User permissions that let you access the bucket and manage the objects in those buckets.
- Minimally, BUCKET_UPDATE and RETENTION_RULE_MANAGE permissions.

Kubernetes Engine (OKE) On Private Cloud Appliance

The Oracle Kubernetes Engine (OKE) is a scalable, highly available service that can be used to deploy any containerized application to Private Cloud Appliance.

The OKE service uses Cluster API Provider (CAPI) and Cluster API Provider for Oracle Cloud Infrastructure (CAPOCI) to orchestrate the cluster on the Private Cloud Appliance.

The OKE service uses Kubernetes, the open-source system for automating deployment, scaling, and management of containerized applications across clusters of hosts. Kubernetes groups the containers that make up an application into logical units called pods for easy management.

For more information about Kubernetes in Oracle, see What is Kubernetes?. For more general information about Kubernetes, see the Kubernetes site.

Using the OKE service

You can access the OKE service to create OKE clusters by using the Private Cloud Appliance Compute Web UI, the OCI CLI, and the API.

You can access OKE clusters by using the Kubernetes command line (`kubectl`), the Kubernetes Dashboard, and the Kubernetes API.

On Private Cloud Appliance, the OKE service manages all OKE cluster nodes—including both control plane and worker node compute instances—within your tenancy. These instances are fully managed and controlled by the OKE service, and users should not patch, update, or otherwise modify them directly for security fixes or any other purposes. Any user-initiated changes can render the clusters unsupported and may negatively affect the functioning of the OKE service.

To ensure security and compatibility, each release of Private Cloud Appliance provides the latest images for OKE control plane and worker node compute instances. Customers can apply new security patches and functionality by upgrading their OKE clusters to the latest supported version after upgrading Private Cloud Appliance. This upgrade process is the supported and recommended method for maintaining secure, up-to-date OKE clusters.

Supported Versions of Kubernetes

The OKE service uses versions of Kubernetes that are certified as conformant by the Cloud Native Computing Foundation (CNCF). The OKE service is itself ISO-Compliant (ISO-IEC 27001, 27017, 27018).

Supported versions of Kubernetes are 1.30.3, 1.29.9, and 1.28.8.

Best practice is to keep clusters upgraded so that they are always running versions of Kubernetes that are currently supported by OKE. Viewing a cluster tells you if a newer Kubernetes version is available for that cluster. See Updating an OKE Cluster on page 914.

Supported Versions of the OCI Terraform Provider

The Creating OKE Network Resources on page 833 topic provides example Terraform scripts to configure the network resources. To use these scripts, you must install both Terraform and the Oracle Cloud Infrastructure (OCI) Terraform provider.

In the Terraform `provider` block, specify the version of the OCI Terraform provider to install as at least v4.50.0 but no greater than v6.36.0:

```
provider "oci" {
    version          = ">= 4.50.0, <= 6.36.0"
...
}
```

OKE Service Limits

The following table shows the service limits for the OKE service on Private Cloud Appliance.

Service	Limit
Maximum number of clusters per tenancy	10
Maximum number of worker nodes (compute instances) per cluster. These nodes can be distributed across multiple node pools.	128
Maximum number of nodes per node pool/group	128
Maximum number of node pools/groups per cluster	No limit on number of node pools as long as total nodes per cluster does not exceed 128.
Maximum number of pods per node	110. This is the Kubernetes default.

OKE Workflow

Private Cloud Appliance has specific requirements to enable the OKE service and to configure OKE.

Most steps to configure and use the OKE service can be performed by any Private Cloud Appliance user with sufficient authorization in the Compute Enclave: creating VCNs and subnets, creating clusters, enabling add-ons, creating node pools, adding storage for containerized applications, and exposing containerized applications to users.

Some steps require Compute Enclave authorizations that might be restricted to fewer people: creating user groups and their policies, creating dynamic groups and their policies, and creating tags that all users in the tenancy can use.

Some steps can only be performed by a Service Enclave administrator: configuring the appliance administration network and making platform images available to Compute Enclave users.

Private Cloud Appliance Administrator Tasks

These OKE setup tasks can only be performed by a Private Cloud Appliance Service Enclave administrator.

1. If the administration network is enabled, ensure that it is configured correctly. Verify that the administration network and the data center network are configured to allow traffic to and from the cluster control plane. See the following resources:
   - OKE Cluster Management Across Networks on page 835
   - Updating the Administration Network Configuration on page 237
   - OKE Cluster Management in Administration Network Guidelines on page 66
   - Access Configuration With Administration Network
2. Provide platform images. Platform images include images required by OKE that have Kubernetes installed on them. Platform images should be imported to all tenancies in the Compute Enclave during appliance installation, upgrade, or patching. If this was not done, a Service Enclave administrator must import images as described in Providing Platform Images.
3. Provide sufficient public IP addresses. Public clusters require at least three free public IP addresses. For more information, see Public Clusters on page 834.

Cluster Administrator Tasks

These OKE setup tasks can be performed by Private Cloud Appliance users who are creating and managing OKE clusters.

Perform the following prerequisite tasks in the Private Cloud Appliance Compute Enclave. These tasks require authorization to create users, user groups, dynamic groups, policies, and tag namespaces.

1. Create an OKE users group. See Create an OKE User Group and Policies on page 824.
2. Create an OKE dynamic group. See Create a Cluster Dynamic Group and Policies on page 824.
3. Create the OraclePCA-OKE defined tag. See Create the OraclePCA-OKE.cluster_id Tag on page 827
4. Create OraclePCA tags that are required to create a node pool. See Creating OraclePCA Tags on page 828.

Perform the following cluster user tasks in the Private Cloud Appliance Compute Enclave. To get the required authorizations, you should be a member of an OKE users group as described in Create an OKE User Group and Policies on page 824.

1. On your local system, configure OCI CLI access. See Configuring the OCI CLI on page 400.

   If you work in more than one tenancy, create a profile for each tenancy as described in Using Multiple Profiles on page 403. If you already have OCI CLI installed, use `oci -v` to check the version. The minimum required version for using OKE is 3.48.0.

   To configure OCI CLI access, you need the Certificate Authority (CA) bundle for the Private Cloud Appliance. The CA bundle includes the certificate, private and public keys, and other authorization information. The CA bundle is automatically updated on the Private Cloud Appliance when regular certificate rotation occurs or when the appliance is upgraded. To use the OCI CLI or the Compute Enclave API, copy the new CA bundle to your local system as described in Obtaining the Certificate Authority Bundle on page 403.

   The CA bundle is automatically downloaded and made available to a cluster when the cluster is created. If you need to update the CA bundle for a cluster manually, see the procedure in Updating the Certificate Authority Bundle on page 830.
2. Create network resources. See Creating OKE Network Resources on page 833.

   Create a VCN, subnets, route tables, and security lists. Create gateways as needed: internet gateway, NAT gateway, Local Peering gateway, Dynamic Routing gateway. See Public and Private Clusters on page 834.
3. Create a cluster. See Creating an OKE Cluster on page 905.
   - On your local system, install the Kubernetes client command line tool, `kubectl`, and create a Kubernetes configuration file for the cluster. See Creating a Kubernetes Configuration File on page 912.

If you already have `kubectl` installed, ensure the version is within one minor version of the Kubernetes version that you are using.

- Deploy a Kubernetes Dashboard to manage the cluster and to manage and troubleshoot applications running in the cluster. On the https://kubernetes.io/ site, see Deploy and Access the Kubernetes Dashboard.
- Enable add-ons as necessary. See Managing OKE Cluster Add-ons on page 916.

4. Create a worker node pool. See Creating an OKE Worker Node Pool on page 932.

- Configure any registries or repositories that the worker nodes need. Ensure you have access to a self-managed public or intranet container registry to use with the OKE service and your application images.
- Create a service to expose containerized applications outside the Private Cloud Appliance. See Exposing Containerized Applications on page 947.
- Create persistent storage for applications to use. See Adding Storage for Containerized Applications on page 949.

Create an OKE User Group and Policies

In the Private Cloud Appliance OKE tenancy, create a user group and policies that authorize users to use OKE.

OKE users groups have a policy that authorizes their members to use OKE. Create separate OKE users groups to authorize different users to use OKE in different compartments.

See Creating and Managing User Groups to create a group or update an existing group.

Include the `manage cluster-family` authorization in the user group policy. The following is an example policy for an OKE user group. Depending on your organization, for example if you have a separate team who manage network resources, some of the following "manage" authorizations could be "read" or "use" authorizations, or you might need to add authorizations.

```
allow group group-name to read all-resources in tenancy
allow group group-name to manage cluster-family in compartment compartment-
name
allow group group-name to manage instance-family in compartment compartment-
name
allow group group-name to manage network-load-balancers in
 compartment compartment-name
allow group group-name to manage virtual-network-family in
 compartment compartment-name
```

For general information about policies, see Managing Policies.

What's Next:

Create a Cluster Dynamic Group and Policies on page 824

Create a Cluster Dynamic Group and Policies

In the Private Cloud Appliance OKE tenancy, create a dynamic group and policies to authorize member instances to manage OKE resources.

For instructions for creating a dynamic group and writing matching rules using the Compute Web UI and the OCI CLI see Creating and Managing Dynamic Groups. For an example of creating an OKE dynamic group using Terraform, see Using Terraform to Create a Dynamic Group on page 825.

Specify the following matching rule to define the group:

```
tag.OraclePCA-OKE.cluster_id.value
```

All cluster nodes that have this tag are members of the dynamic group.

The following is an example policy for the dynamic group. In this example, `oke_dyn_grp` is the name of the dynamic group and `oke` is the name of the compartment where resources are created. Note that all policy statements

are for the same compartment. If clusters in this group require access to resources in other compartments, change the policy accordingly. For general information about policies, see Managing Policies.

```
allow dynamic-group oke_dyn_grp to manage file-family in compartment oke
allow dynamic-group oke_dyn_grp to manage volume-family in compartment oke
allow dynamic-group oke_dyn_grp to manage load-balancers in compartment oke
allow dynamic-group oke_dyn_grp to manage instance-family in compartment oke
allow dynamic-group oke_dyn_grp to manage virtual-network-family in
 compartment oke
allow dynamic-group oke_dyn_grp to use tag-namespaces in compartment oke
```

For information about the purpose of the `use tag-namespaces` policy, see Exposing Containerized Applications on page 947.

What's Next:

Create the OraclePCA-OKE.cluster_id Tag on page 827

Using Terraform to Create a Dynamic Group

The following example shows how to use Terraform to create a dynamic group.

variables.tf

```
variable "oci_config_file_profile" {
  type    = string
  default = "DEFAULT"
}

variable "tenancy_ocid" {
    description = "tenancy OCID"
    type        = string
    nullable    = false
}

variable "compartment_name" {
    description = "compartment name"
    type        = string
    nullable    = false
}

variable "oke_dyn_grp" {
    description = "Dynamic group that needs to be created for instance
 principal"
    default = "oke-dyn-ip-grp"
}

variable "oke_policy_name" {
    description = "Policy set name for dynamic group"
    default = "oke-instance-principal-policy"
}
```

terraform.tfvars

```
# Name of the profile to use from $HOME/.oci/config
oci_config_file_profile = "DEFAULT"

# Tenancy OCID from the oci_config_file_profile profile.
tenancy_ocid = "ocid1.tenancy.UNIQUE_ID"

# Compartment name
compartment_name = "oke"

# Dynamic Group Name
```

```
oke_dyn_grp = "oke-dyn-ip-group"

# OKE Dynamic Group Policy Name
oke_policy_name = "oke-dyn-grp-policy"
```

provider.tf

```
provider "oci" {
  config_file_profile = var.oci_config_file_profile
  tenancy_ocid        = var.tenancy_ocid
}
```

main.tf

```
terraform {
  required_providers {
    oci = {
      source  = "oracle/oci"
      version = ">= 4.50.0, <= 6.36.0"
      # If necessary, you can pin a specific version here
      # version = "6.36.0"
    }
  }
  required_version = ">= 1.1"
}
```

oke-dyn-grp.tf

```
resource "oci_identity_dynamic_group" "oke-dynamic-grp" {
    compartment_id = "${var.tenancy_ocid}"
    description    = "PCA OKE worker dynamic group for instance principal"
    matching_rule  = "tag.${oci_identity_tag_namespace.oracle-pca.name}.
${oci_identity_tag.cluster-id.name}.value"
    name           = "${var.oke_dyn_grp}"
    depends_on = [oci_identity_tag.cluster-id]
}
```

oke-policy.tf

```
resource "oci_identity_policy" "oke-dyn-grp-policy" {
    compartment_id = "${var.tenancy_ocid}"
    description    = "Dynamic group policies for OKE Resources"
    name           = "${var.oke_policy_name}"
    statements     = [
        "allow dynamic-group ${oci_identity_dynamic_group.oke-dynamic-
grp.name} to manage load-balancers in compartment ${var.compartment_name}",
        "allow dynamic-group ${oci_identity_dynamic_group.oke-dynamic-
grp.name} to manage volume-family in compartment ${var.compartment_name}",
        "allow dynamic-group ${oci_identity_dynamic_group.oke-dynamic-
grp.name} to manage file-family in compartment ${var.compartment_name}",
        "allow dynamic-group ${oci_identity_dynamic_group.oke-dynamic-
grp.name} to manage instance-family in compartment ${var.compartment_name}",
        "allow dynamic-group ${oci_identity_dynamic_group.oke-
dynamic-grp.name} to manage virtual-network-family in compartment
 ${var.compartment_name}",
        "allow dynamic-group ${oci_identity_dynamic_group.oke-dynamic-
grp.name} to use tag-namespaces in compartment ${var.compartment_name}"
    ]

    depends_on = [oci_identity_dynamic_group.oke-dynamic-grp]
}
```

oke-tag-ns.tf

Create the OraclePCA-OKE.cluster_id tag, which is also described in Create the OraclePCA-OKE.cluster_id Tag on page 827.

```
resource "oci_identity_tag" "cluster-id" {
    description      = "Default tag key definition"
    name             = "cluster_id"
    tag_namespace_id = "${oci_identity_tag_namespace.oracle-pca.id}"
    depends_on = [oci_identity_tag_namespace.oracle-pca]
}

resource "oci_identity_tag_namespace" "oracle-pca" {
    compartment_id = "${var.tenancy_ocid}"
    description    = "Default Tag namespace for Oracle PCA OKE"
    name           = "OraclePCA-OKE"
}
```

What's Next?

Create the OraclePCA-OKE.cluster_id Tag on page 827

Create the OraclePCA-OKE.cluster_id Tag

The OraclePCA-OKE.cluster_id defined tag is required to create or update an OKE cluster or node pool On Private Cloud Appliance.

The cluster_id tag value is provided automatically when you create a cluster. When you create a node pool, or update the node pool to add nodes, this tag is applied to every node to identify instances as members of the dynamic group.

This procedure describes how to create the OraclePCA-OKE tag namespace and the cluster_id tag key. You must create both the tag namespace and the tag key.

> **Important:**
>
> Do not delete this tag key definition. The tag namespace name must be exactly OraclePCA-OKE, and the tag key name must be exactly cluster_id.

Using the Compute Web UI

1. In the navigation menu, select **Governance**, and then select **Tag Namespaces**.
2. If OraclePCA-OKE is not shown in the Tag Namespaces list, select the **Create Namespace Definitions** button.
3. In the Create Namespace Definition dialog box, scroll down to the Tagging section and select the Tag Namespace field.
   - If the OraclePCA-OKE tag namespace is not listed, continue with Step 4 on page 827 of this procedure.
   - If the OraclePCA-OKE tag namespace is listed, then it has already been created in a different compartment and is available to all compartments.

     If the OraclePCA-OKE tag namespace is already available, select the Cancel button in the Create Namespace Definition dialog box. You will receive an error message if you try to create this tag namespace when it is already available.
4. In the Create Namespace Definition dialog box, enter the following information:
   - **Compartment**: Select the compartment where you want to create the OraclePCA-OKE tag namespace definition.
   - **Name**: Enter OraclePCA-OKE.
   - **Description**: Enter a description for the tag namespace. For example, "Required to create or update an OKE cluster or node pool." Avoid entering confidential information.
   - Select the **Create Namespace Definition** button in the dialog box.

     The details page for the OraclePCA-OKE tag namespace definition is displayed.

5. On the OraclePCA-OKE details page, scroll to the Resources section and select the **Create Tag Key Definition** button.
6. In the Create Tag Key Definition dialog box, enter the following information:
   - **Name**: Enter `cluster_id`
   - **Description**: Enter a description for the tag key. For example, "Required to create or update an OKE cluster or node pool." Avoid entering confidential information.
   - **Tag Value Type**: Select Static Value.
7. Select the **Create Tag Key Definition** button in the dialog box.

Using the CLI

1. Use the oci iam tag-namespace create command to create the OraclePCA-OKE tag namespace.

   The `--compartment-id` value is the compartment where you want to create the OraclePCA-OKE.cluster_id defined tag. If this tag namespace has already been created in a different compartment in the tenancy, then it is available to use in any other compartment in the tenancy and you will receive an error message from this create tag namespace command.

   Example:

   ```
   oci iam tag-namespace create --compartment-id compartment_OCID --name
    OraclePCA-OKE --description "Required to create or update an OKE cluster
    or node pool."
   ```

2. Use the oci iam tag create command to create the cluster_id tag in the OraclePCA-OKE tag namespace.

   You do not need to specify the `--validator` option because you want the default value.

   Example:

   ```
   $ oci iam tag create --tag-namespace-id tag_namespace_OCID --name
    "cluster_id" --description "Required to create or update an OKE cluster
    or node pool."
   ```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the CreateTagNamespace operation to create the OraclePCA-OKE tag namespace in your OKE compartment. You must specify "OraclePCA-OKE" as the name of the tag namespace. If this tag namespace has already been created in a different compartment in the tenancy, then it is available to use in any other compartment in the tenancy and you will receive an error message from this create tag namespace operation.

Use the CreateTag operation to create the cluster_id tag in the OraclePCA-OKE tag namespace. You must specify "cluster_id" as the name of the tag.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Creating OraclePCA Tags

Private Cloud Appliance uses the OraclePCA tag namespace to set attributes that aren't available as OCI CLI options or API attributes. For OKE, some cluster attributes must be set by using OraclePCA tags.

Other attributes that can only be set by using OraclePCA tags, such as some block volume and file system attributes, are documented in Using Resource Tags to Extend Functionality (IAM on the Appliance) on page 460. You might want to set some of these for nodes in a node pool.

When you use the OCI CLI or API, you can specify the OraclePCA tag namespace, tag key, and values for the attributes that you want to set. You don't need to first create the OraclePCA tag namespace and tag keys.

To use the OCI CLI to set these attributes, you must first create the OraclePCA tag namespace, tag keys, and value choices.

> **CAUTION:**
>
> Don't delete these tag keys. Don't create this tag namespace and these keys unless you need to use the OCI CLI to create clusters. If you create this tag namespace and these keys, create them exactly as shown here, don't modify them, and don't delete them.

The following sections describe how to create the OraclePCA tag namespace, and how to create the tag key definitions for the OKE attributes.

Creating the OraclePCA Tag Namespace

1. In the Compute Web UI navigation menu, select **Governance**, and then select **Tag Namespaces**.

   Ensure that the compartment that's selected in the compartment drop-down menu above the tag namespaces list is the compartment where you want to create the OraclePCA tag namespace.
2. If OraclePCA isn't shown in the **Tag Namespaces** list, select **Create Namespace Definitions**.
3. In the **Create Namespace Definition** dialog box, scroll down to the **Tagging** section and click in the **Tag Namespace** field.
   - If the OraclePCA tag namespace isn't listed, continue with Step 4 of this procedure.
   - If the tag you need is already available, select **Cancel**. You get an error message if you try to create this tag namespace when it's already available.
4. In **Create Namespace Definition**, enter the following information:
   - **Namespace Definition Name**: Enter "OraclePCA".
   - **Description**: For example, "Support block volume, cluster, and file system parameters that are only available On Private Cloud Appliance."
5. Select **Create Namespace Definition**.

   The details page for the new OraclePCA tag namespace definition is displayed.

Creating the OraclePCA Tag Key Definitions

1. Navigate to the OraclePCA tag namespace details page, under **Resources**, and select **Tag Key Definitions**.
2. If the tag key you need isn't listed, select **Create Tag Key Definition**.
3. In the **Create Tag Key Definition** dialog box, enter the following information for the tag key that you're creating.

   SSH Key

   - **Name**: Enter "sshkey".
   - **Description**: For example, "Your public SSH key."
   - **Tag Value Type**: Select "Static Value".

   Number of Control Plane Nodes

   - **Name**: Enter "cpNodeCount".
   - **Description**: For example, "Number of nodes in the control plane."
   - **Tag Value Type**: Select "A List of Values".
   - **Values**: Enter "1", newline, "3", newline, and "5".

   Shape of Control Plane Nodes

   - **Name**: Enter "cpNodeShape".
   - **Description**: For example, "The shape of the control plane nodes."
   - **Tag Value Type**: Select "Static Value".

   Shape Configuration of Control Plane Nodes

   - **Name**: Enter "cpNodeShapeConfig".

- **Description**: For example, "The number of OCPUs and optionally amount of memory for a flexible node shape."
- **Tag Value Type**: Select "Static Value".

4. Select **Create Tag Key Definition**.

Updating the Certificate Authority Bundle

The Certificate Authority (CA) bundle for Private Cloud Appliance is downloaded and made available to a cluster when the cluster is created. The CA bundle includes the certificate, private and public keys, and other authorization information.

The CA bundle is automatically updated when regular certificate rotation occurs or when the Private Cloud Appliance is upgraded.

When the CA bundle is updated on the appliance, it must be updated on the local system that you use to manage the OKE service. For example, the CA bundle authorizes the use of the OCI CLI and API operations. You replace the CA bundle in your `~/.oci` configuration so that you can run OCI CLI commands. For clusters, the CA bundle update is automated.

A process runs every hour to check the validity of the CA bundle in OKE clusters and updates the CA bundle if it has been updated on the appliance.

If you need to update the CA bundle between these hourly checks, you can run the process manually:

1. Log onto the management node of the Private Cloud Appliance as a system administrator with root privilege.
2. Get the name of an OKE pod.

   The following command lists the OKE pods in the `oke` namespace:

   ```
   # kubectl get pod -n oke -l app=oke
   ```

3. Run the command to update the CA bundle.

   Use one of the `oke-`*`uniqueID`* pod names from the preceding step.

   ```
   # kubectl exec -it oke-6c4d85d6f-72fxs -n oke -c oke -- /usr/bin/pca-oke-
   cluster-tool
   ```

You can check Loki logs in Grafana for any errors that might have occurred when this process ran either automatically or manually. See Loki Logs.

OKE Best Practices

Follow these OKE best practices for Private Cloud Appliance for best results using OKE clusters.

Cluster Management Best Practices

Upgrade clusters.

Keep your clusters upgraded so that they are always running versions of Kubernetes that are listed as currently supported by OKE. Viewing a cluster tells you if a newer Kubernetes version is available for that cluster. See Supported Versions of Kubernetes on page 822 and Updating an OKE Cluster on page 914.

Use Kubernetes labels.

Use Kubernetes labels to organize the many Kubernetes resources (such as services, pods, containers, and networks) that comprise a cluster.

Kubernetes labels are key-value pairs that help you to maintain these resources and keep track of how they interact with each other in a cluster.

Use resource tagging.

Use resource tagging to organize the many resources (such as worker nodes, VCNs, load balancers, and block volumes) used by the Kubernetes clusters that you create with OKE.

When a large number of resources is spread across multiple compartments in a tenancy, it can be challenging to track the resources that are used for specific purposes. It can also be challenging to aggregate the resources, report on them, and take bulk actions on them.

Tagging enables you to define keys and values, and associate those tags with resources. You can then use the tags to organize and list resources based on your business needs.

For more information, see Resource Tag Management.

Set resource requests and limits.

- Set resource requests to specify the minimum amount of resources a container can use.
- Set resource limits to specify the maximum amount of resources a container can use.

Sometimes an application fails to deploy on a Kubernetes cluster because of limited availability of resources on that cluster. The failure of the application to deploy can be avoided by correctly setting resource requests and resource limits.

If you do not set resource requests and limits, pods in a cluster can start utilizing more resources than necessary. If a pod starts consuming more CPU or memory on a node, then the Kubernetes scheduler (`kube-scheduler`) might not be able to place new pods on the node, and the node might even crash.

For more information, see Resource Management for Pods and Containers on the `kubernetes.io` site.

Provide dedicated nodes by using taints and tolerations.

Use Kubernetes taints and tolerations to limit resource-intensive applications to specific worker nodes.

Using taints and tolerations enables you to keep node resources available for workloads that require them, and prevents the scheduling of other workloads on the nodes.

For more information, see Taints and Tolerations on the `kubernetes.io` site.

Control pod scheduling by using node selectors and affinity.

Several different methods are available to constrain a pod to run on particular nodes, or to specify a preference for a pod to run on particular nodes. The recommended approaches all use label selectors to specify the node selection.

Often, the `kube-scheduler` makes a reasonable placement when constraints and preferences are not specified. However, there are some circumstances where you might want to control the node on which a pod runs. In these situations, best practice is to control the scheduling of pods on nodes using Kubernetes node selectors, node affinity, and inter-pod affinity.

Using node selectors, node affinity, and inter-pod affinity enables the `kube-scheduler` to logically isolate workloads, such as according to the node's hardware.

Use third-party tools for backup and disaster recovery.

Use third-party tools such as Velero with OKE for backup and disaster recovery.

The combined backup and disaster recovery capabilities of these tools and OKE can provide a reliable, robust, and scalable Kubernetes platform that is production-ready.

Networking Best Practices

Create separate compartments for each team.

If you expect multiple teams to create clusters, create a separate compartment for each team.

Size your VCN appropriately.

Allow for possible future cluster and node pool scaling requirements when sizing the VCN in which you want to create and deploy Kubernetes clusters.

Ensure that the VCN has a CIDR block that is large enough to allocate network addresses to all the resources that a cluster requires: subnets, Kubernetes API endpoint, worker nodes, pods, load balancers.

Select the pod networking CNI plug-in that best suits your needs.

Consider pod networking requirements carefully, and then select the pod networking CNI plug-in that best suits your needs.

- If applications require the use of base networking requirements (and not the use of IP addresses from the VCN), or require a high density of pods per worker node, best practice is to use the Flannel Overlay CNI plug-in. See Creating Flannel Overlay Network Resources on page 836.
- If applications require pods to have an IP address from the VCN CIDR, or require the consistent network performance offered by virtual machines (regardless of the nodes on which the pods are running) with no additional overlay, best practice is to use the OCI VCN-Native Pod Networking CNI plugin. See Creating VCN-Native Pod Network Resources on page 864.

Configure externalTrafficPolicy appropriately when exposing applications.

Carefully consider the most appropriate value for the `externalTrafficPolicy` setting when provisioning a network load balancer for a Kubernetes service of type LoadBalancer.

Avoid overlapping pod and service CIDR blocks with an on-premises CIDR block and when using the Flannel Overlay CNI plug-in.

Avoid the situation where the CIDR block used by the Flannel Overlay network to provision pods and services with IP addresses overlaps with a CIDR block used to provision external compute instances with IP addresses.

Kubernetes clusters require a unique IP address for every pod. Therefore, IP address planning is necessary because addresses cannot overlap with the private IP address space used on-premises or in other connected VCNs.

Plan the number of nodes you will need.

Create a plan for the number of nodes in a cluster that takes into account node size, the application profile of pods, and the selected pod networking CNI plug-in.

Use separate subnets and security rules.

Use separate subnets and security rules when configuring network resources. The VCN in which you want to create and deploy clusters must have at least two different subnets, and can have more:

- A Kubernetes API endpoint subnet
- A worker nodes subnet
- One regional, or two AD-specific, load balancer subnets (optional)
- A pods subnet (when using the OCI VCN-Native Pod Networking CNI plugin)
- A bastion subnet (optional)

You can choose to combine the subnets, and also to combine security rules. However, this approach makes security harder to manage and is therefore not recommended unless you are using network security groups to control access to clusters.

Security Best Practices

Plan exposure level.

Answer the following questions before implementing a security plan for the clusters you create with OKE:

- How much internet exposure do you want clusters to have?
- How do you plan to expose workloads internally to your VCN, and externally to the internet?
- How do you plan to scale workloads?
- Which types of Oracle services will the cluster consume?

Create private clusters.

If your cluster does not require direct access from the internet, create a private cluster. In a private cluster, the Kubernetes API server and worker nodes are assigned only private IP addresses.

Optionally use a NAT gateway for outbound internet access, a Dynamic Routing Gateway (DRG) to enable access from the on-premises network, a Local Peering Gateway (LPG) to allow access from other VCNs.

Place all applications in private subnets.

If the applications running on worker nodes don't require direct access to the internet, both the worker nodes subnet and the worker load balancer subnet should be private.

Restrict cluster traffic using Network Security Groups.

Define security rules in network security groups (NSGs), rather than in security lists, for the VCN in which you want to create and deploy clusters. See Controlling Traffic with Network Security Groups on page 501.

General security best practices.

- Apply security patches regularly.
- Use a combination of Kubernetes network policies and NSGs.
- Use NSGs with appliance-as-code tools (such as Terraform).
- Rotate secrets and certificates regularly.
- Run all applications as a non-privileged user.
- Treat containers as immutable.

Auditing, logging, and monitoring.

- Check logs regularly.
- Enable audit logging.
- Use Kubernetes cluster-based logging.
- Monitor cluster components.
- Log network traffic metadata and analyze it regularly.
- Use small and secure container images.
- Limit credential exposure.

Storage Best Practices

- Select the appropriate storage type.
- Create and use storage classes to define application needs.
- Create and use volumes for persistent storage.
- Limit storage resource consumption.
- Secure and back up data.

Upgrade Best Practices

- Use the latest supported version of Kubernetes.
- Set up test and production environments.
- Cordon and drain worker nodes in preparation for maintenance.
- Treat worker nodes as immutable.

Creating OKE Network Resources

Some network resources used by Kubernetes Engine (OKE) On Private Cloud Appliance must be configured in specific ways.

The resource definitions in the following sections create a working example set of network resources for workload clusters. Use this configuration as a guide when you create these resources. You can change the values of properties such as CIDR blocks and IP addresses. Don't change the values of properties such as the network protocol, the stateful setting, or the private/public setting.

> **Note:**
>
> For subnets used to create or run an OKE cluster (including node pool, load balancer, and pod subnets), don’t configure custom DNS servers in the subnet’s DHCP options. OKE depends on the default DNS resolver for reliable name resolution to required Kubernetes and platform services

(including the on-rack region registry used to pull images during cluster and node provisioning). Using custom DHCP resolvers can cause cluster provisioning failures and ongoing node or add-on connectivity issues. If you need custom name resolution, use supported OCI DNS capabilities (for example, Private DNS) without overriding the subnet DHCP DNS settings.

See Workload Cluster Network Ports (Flannel Overlay) on page 838 and Workload Cluster Network Ports (VCN-Native Pod) on page 866 for specific ports that must be open for specific purposes.

Note:

If the appliance administration network is enabled, ask your system administrator to verify that the administration network and the data center network are configured to allow traffic to and from the cluster control plane.

You can create network resources for two networking types:

- Creating Flannel Overlay Network Resources on page 836
- Creating VCN-Native Pod Network Resources on page 864

Public and Private Clusters on page 834 summarizes which network resources you need to create a public cluster and which network resources you need to create a private cluster.

Pod Networking

The Kubernetes networking model assumes containers (pods) have unique and routable IP addresses within a cluster. In the Kubernetes networking model, pods use those IP addresses to communicate with other pods on the same node in a cluster or on a different node, with pods on other clusters, with the cluster's control plane nodes, with other services (such as storage services), and with the internet.

By default, pods accept traffic from any source. To enhance cluster security, control access to and from pods using security rules defined as part of network security groups (recommended) or security lists. The security rules apply to all pods in all the worker nodes connected to the pod subnet specified for a node pool. See Controlling Traffic with Network Security Groups on page 501 and Controlling Traffic with Security Lists on page 496.

Public and Private Clusters

On Private Cloud Appliance, before you create a cluster, decide what kind of network access the cluster requires: whether you need a public cluster or a private cluster. You cannot create both public and private clusters in one VCN.

The key difference between a public cluster and a private cluster is whether you configure public or private subnets for the Kubernetes API endpoint and the worker load balancer.

Note:

The subnets for the worker nodes and control plane nodes are always private.

For the worker nodes and control plane nodes, you can configure route rules that allow access only within the VCN or outside the VCN. This documentation names those route tables "vcn_private" and "nat_private." You can choose either of these private subnet configurations for your worker nodes and control plane nodes whether the cluster is private or the cluster is public.

Public Clusters

A public cluster requires the following network resources:

- A public subnet for the Kubernetes API endpoint. See the instructions for creating a public "control-plane-endpoint" subnet in Creating an OKE Control Plane Subnet (Flannel Overlay) on page 858 and Creating a Control Plane Subnet (VCN-Native Pod) on page 895.
- A public subnet for the worker load balancer. See the instructions for creating a public "service-lb" subnet in Creating a Worker Load Balancer Subnet (Flannel Overlay) on page 856 and Creating a Worker Load Balancer Subnet (VCN-Native Pod) on page 893.

- An internet gateway to connect resources on a public subnet to the internet using public IP addresses.
- A NAT gateway. Use a NAT gateway for outbound internet access. A NAT gateway connects resources on a private subnet to the internet without exposing private IP addresses.
- At least three free public IP addresses. Free public IP addresses are required for the NAT gateway, control plane load balancer, and worker load balancer.

  The worker load balancer requires a free public IP address to expose applications. The worker load balancer might require more free public IP addresses depending on the applications running on the pods.

Private Clusters

If you create multiple OKE VCNs, each CIDR must be unique. CIDRs of different VCNs for private clusters cannot overlap with any other VCN CIDRs or any on-premises CIDR. The IP addresses used must be exclusive to each VCN.

A private cluster has the following network resources:

- A private subnet for the Kubernetes API endpoint. See the instructions for creating a private "control-plane-endpoint" subnet in Creating an OKE Control Plane Subnet (Flannel Overlay) on page 858 and Creating a Control Plane Subnet (VCN-Native Pod) on page 895.
- A private subnet for the worker load balancer. See the instructions for creating a private "service-lb" subnet in Creating an OKE Control Plane Load Balancer Subnet (Flannel Overlay) on page 861 and Creating a Control Plane Load Balancer Subnet (VCN-Native Pod) on page 900.
- A route table with no route rules. This route table allows access only within the VCN.
- (Optional) A Local Peering Gateway (LPG). Use an LPG to allow access from other VCNs. An LPG allows access to the cluster from an instance running on a different VCN. Create an LPG on the OKE VCN, and create an LPG on a second VCN on the Private Cloud Appliance. Use the LPG connect command to peer the two LPGs. Peered VCNs can be in different tenancies. CIDRs for the peered VCNs cannot overlap. See Connecting VCNs through a Local Peering Gateway on page 510.

  Create a route rule to steer VCN subnet traffic to and from the LPGs, and security rules to allow or deny certain types of traffic. See Creating a VCN (Flannel Overlay) on page 848 or Creating a VCN (VCN-Native Pod Networking) on page 879 for the route table to add to the OKE VCN and similar route table to add to the second VCN. Add the same route rule on the second VCN, specifying the OKE VCN CIDR as the destination.

  Install the OCI SDK and `kubectl` on the instance on the second VCN and connect to the private cluster. See Creating a Kubernetes Configuration File on page 912.
- (Optional) A Dynamic Routing Gateway (DRG). Use a DRG to enable access from the on-premises network. A DRG allows traffic between the OKE VCN and the on-premises network's IP address space. Create the DRG in the OKE VCN compartment, and then attach the OKE VCN to that DRG. See Connecting to the On-Premises Network through a Dynamic Routing Gateway (DRG) on page 511.

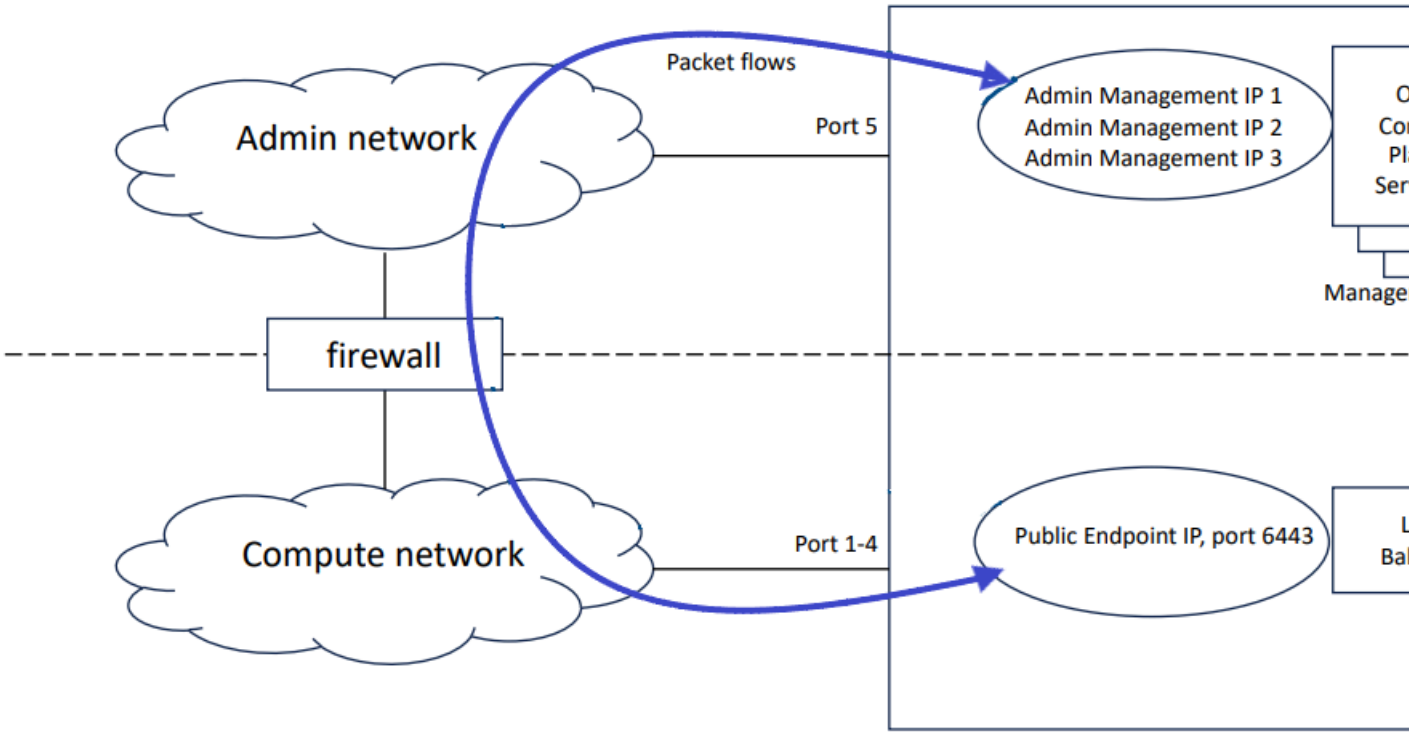
  Create a route rule to steer traffic to the on-premises data center network's IP address space. See Creating a VCN (Flannel Overlay) on page 848 or Creating a VCN (VCN-Native Pod Networking) on page 879 for the route table to add to the OKE VCN.

OKE Cluster Management Across Networks

Learn about OKE cluster management across the admin and data networks On Private Cloud Appliance.

Private Cloud Appliance admin and data networks are configured by Oracle, when Oracle installs the Private Cloud Appliance in the data center.

The OKE service runs on the management nodes in the admin network, while the OKE clusters are deployed in the data network as shown in the following diagram.



The management interface of an OKE cluster is port 6443 on its load balancer public IP address. This address is assigned from the data center IP range that was reserved and configured as public IPs during the Private Cloud Appliance appliance installation.

Because of the network segregation, traffic from the OKE service must exit the appliance through the admin network, and reenter through the data network to reach the OKE cluster.

> **Important:**
>
> The data center network appliance must allow traffic in both directions. Without the necessary firewall and routing rules, users cannot deploy OKE clusters.

See Workload Cluster Network Ports (Flannel Overlay) on page 838 and Workload Cluster Network Ports (VCN-Native Pod) on page 866 for ports needed for OKE.

For information about other network ports, see Access Configuration With Administration Network.

Creating Flannel Overlay Network Resources

On Private Cloud Appliance, the Flannel Overlay network type encapsulates communication between pods in the Flannel Overlay network.

The Flannel Overlay network is a simple private overlay virtual network that satisfies the requirements of the OKE networking model by attaching IP addresses to containers. The pods in the private overlay network are only accessible from other pods in the same cluster.

The resource definitions in the following sections create a working example set of network resources for workload clusters when you are using Flannel Overlay networking. Use this configuration as a guide when you create these resources. You can change the values of properties such as CIDR blocks and IP addresses. Don't change the values of properties such as the network protocol, the stateful setting, or the private/public setting.

Create all the network resources in the same compartment.

Use the following sections to manually configure flannel overlay network resources:

- Workload Cluster Network CIDR Ranges (Flannel Overlay) on page 837
- Workload Cluster Network Ports (Flannel Overlay) on page 838
- Creating a VCN (Flannel Overlay) on page 848
- Creating a Worker Subnet (Flannel Overlay) on page 852
- Creating a Worker Load Balancer Subnet (Flannel Overlay) on page 856
- Creating an OKE Control Plane Subnet (Flannel Overlay) on page 858
- Creating an OKE Control Plane Load Balancer Subnet (Flannel Overlay) on page 861

To use Terraform scripts to automate the flannel overlay network configuration, see Creating a Control Plane Load Balancer Subnet (VCN-Native Pod) on page 900.

Workload Cluster Network CIDR Ranges (Flannel Overlay)

Review the example CIDR ranges used for OKE network resources for Flannel Overlay networking On Private Cloud Appliance.

Throughout this documentation, variables are used to represent CIDR ranges for instances in different subnets. The following table lists the CIDR variables and example values for use with Flannel Overlay networking.

> **Note:**
>
> These are examples only. The CIDR ranges you use depend on the number of clusters you have, the number of nodes in each cluster, and the type of networking you are using.

For Flannel Overlay networking, IP addresses are managed by the underlying Container service. Pods are not assigned IP addresses from the IP address pool that is defined in the pod subnet CIDR. This is the reason you do not need a pod subnet when you are using Flannel Overlay networking.

The primary difference between IP address requirements of Flannel Overlay networking and VCN-Native Pod Networking is that VCN-Native Pod Networking requires more IP addresses to be available. The table in Workload Cluster Network CIDR Ranges (VCN-Native Pod) on page 865 shows larger CIDR ranges than the following table for Flannel Overlay CIDR ranges. The CIDR ranges used with Flannel Overlay networking can be much smaller than the CIDR ranges used with VCN-Native Pod Networking.

Variable Name	Description	Example Value
vcn_cidr	VCN CIDR range	172.31.252.0/23
worker_cidr	Worker subnet CIDR	172.31.253.0/24
workerlb_cidr	Worker load balancer subnet CIDR	172.31.252.0/25
kmi_cidr	OKE control plane subnet CIDR	172.31.252.224/28
kmilb_cidr	OKE control plane load balancer subnet CIDR	172.31.252.240/28
kube_client_cidr	CIDR for clients that are allowed to contact the Kubernetes API server	10.0.0.0/8
public_ip_cidr	Public IP CIDR configured in the Private Cloud Appliance Service Enclave	10.0.0.0/8
kube_internal_cidr	CIDR used by the Kubernetes appliance to allocate IP addresses for various internal services and components	253.255.0.0/16

The IP Subnet Calculator on Calculator.net is one tool for finding all available networks for a given IP address and prefix length.

What's Next:

Review the network ports used for Flannel Overlay networking. See Workload Cluster Network Ports (Flannel Overlay) on page 838.

Workload Cluster Network Ports (Flannel Overlay)

Review the ports that are required for OKE for Flannel Overlay networking On Private Cloud Appliance. These ports must be available to configure workload cluster networking. You might need to open more ports for other purposes.

The following table lists ports that are used by workload clusters when you use Flannel Overlay networking. These ports must be available to configure workload cluster networking. You might need to open more ports for other purposes.

All protocols are TCP. All port states are Stateful. Port 6443 is the port used for the Kubernetes API and is also referred to as *`kubernetes_api_port`*.

See also the table in Access Configuration With Administration Network.

Ports that Must Be Available for Use by Workload Clusters for Flannel Overlay Networking

Source IP Address	Destination IP Address	Port	Description
bastion host: *`vcn_cidr`*	Worker nodes subnet: *`worker_cidr`*	22	Outbound connections from the bastion host to the worker CIDR.
bastion host: *`vcn_cidr`*	Control plane subnet: *`kmi_cidr`*	22	Outbound connections from the bastion host to the control plane nodes.
Worker nodes subnet: *`worker_cidr`*	yum repository	80	Outbound connections from the worker CIDR to external applications.
Worker nodes subnet: *`worker_cidr`*	Secure yum repository	443	Secure outbound traffic from the worker CIDR to external applications.
Worker nodes subnet: *`worker_cidr`*	Container registry	5000	Outbound connections from the worker CIDR to the container registry.
Worker nodes subnet: *`worker_cidr`*	Control plane subnet: *`kmi_cidr`*	6443	Outbound connections from the worker CIDR to the Kubernetes API. This is necessary to allow nodes to join through either a public IP address on one of the nodes or the load balancer public IP address. Port 6443 is called the ***`kubernetes_api_port`***.
Worker nodes subnet: *`worker_cidr`*	Control plane load balancer	6443	Inbound connections from the worker CIDR to the Kubernetes API.
CIDR for clients: *`kube_client_cidr`*	Control plane load balancer	6443	Inbound connections from clients to the Kubernetes API server.
Worker nodes subnet: *`worker_cidr`*	Control plane subnet: *`kmi_cidr`*	6443	Private outbound connections from the worker CIDR to `kubeapi` on the control plane subnet.
`kube_client_cidr`	Worker nodes subnet: *`worker_cidr`*	30000-32767	Inbound traffic for applications from Kubernetes clients.

What's Next:

Depending on how you want to create the network resources, see one of the following sections:

- Use Terraform scripts to automate the creation of the network resources. See Example Terraform Scripts for Network Resources (Flannel Overlay) on page 839.
- Begin to create the network resources individually using the Compute Web UI, CLI, or API. See Example Terraform Scripts (VCN-Native Pod) on page 867.

Example Terraform Scripts for Network Resources (Flannel Overlay)

On Private Cloud Appliance, you can use Terraform scripts to automate the creation of OKE network resources for Flannel Overlay networking.

The following Terraform scripts create the network resources that are required by Kubernetes Engine (OKE) when you're using Flannel Overlay networking. Subsequent sections in this topic show other ways to define these same network resources.

Most of the values shown in these scripts, such as resource display names and CIDRs, are examples. Some ports must be specified as shown (see Workload Cluster Network Ports (Flannel Overlay) on page 838), and the OKE control plane subnet must be named `control-plane`. See Workload Cluster Network CIDR Ranges (Flannel Overlay) on page 837 for comments about CIDR values.

Terraform Script Examples

- variables.tf on page 839
- terraform.tfvars on page 840
- provider.tf on page 841
- main.tf on page 841
- oke_vcn.tf on page 843
- oke_worker_seclist.tf on page 844
- oke_worker_subnet.tf on page 845
- oke_kmi_seclist.tf on page 846
- oke_kmi_subnet.tf on page 847

variables.tf

This file creates several variables that are used to configure OKE network resources when you're using Flannel Overlay networking. Many of these variables aren't assigned values in this file. One port and five CIDRs are assigned values. The `kubernetes_api_port`, port 6443, is the port used to access the Kubernetes API. See also Workload Cluster Network Ports (Flannel Overlay) on page 838. The five CIDRs that are defined in this file are for the OKE VCN, worker subnet, worker load balancer subnet, control plane subnet, and control plane load balancer subnet.

```
variable "oci_config_file_profile" {
  type    = string
  default = "DEFAULT"
}

variable "tenancy_ocid" {
  description = "tenancy OCID"
  type        = string
  nullable    = false
}

variable "compartment_id" {
  description = "compartment OCID"
  type        = string
  nullable    = false
}

variable "vcn_name" {
  description = "VCN name"
```

```
  nullable    = false
}

variable "kube_client_cidr" {
  description = "CIDR of Kubernetes API clients"
  type        = string
  nullable    = false
}

variable "public_ip_cidr" {
  description = "Public IP CIDR configured in the Service Enclave"
  type        = string
  nullable    = false
}

variable "kubernetes_api_port" {
  description = "Port used for Kubernetes API"
  type        = string
  default     = "6443"
}

variable "vcn_cidr" {
  default = "172.31.252.0/23"
}

# Subnet for KMIs where kube-apiserver and other control
# plane applications run, max 9 nodes
variable "kmi_cidr" {
  description = " Kubernetes control plane subnet CIDR"
  default     = "172.31.252.224/28"
}

# Subnet for KMI load balancer
variable "kmilb_cidr" {
  description = "Kubernetes control plane LB subnet CIDR"
  default     = "172.31.252.240/28"
}

# Subnet for worker nodes, max 128 nodes
variable "worker_cidr" {
  description = "Kubernetes worker subnet CIDR"
  default     = "172.31.253.0/24"
}

# Subnet for worker load balancer (for use by CCM)
variable "workerlb_cidr" {
  description = "Kubernetes worker LB subnet CIDR"
  default     = "172.31.252.0/25"
}

# Flag to Enable private endpoint
variable "enable_private_endpoint" {
  description = "Flag to create private control plane endpoint/service-lb"
  type = bool
  default = false
  nullable = false
}
```

terraform.tfvars

This file assigns values to some of the variables that were created in `variables.tf`.

```
# name of the profile to use from $HOME/.oci/config
```

```
oci_config_file_profile = "DEFAULT"

# tenancy ocid from the above profile
tenancy_ocid = "tenancy_OCID"

# compartment in which to build the OKE cluster
compartment_id = "compartment_OCID"

# display-name for the OKE VCN
vcn_name = "oketest"
```

provider.tf

This file is required to use the OCI provider. The file initializes the OCI module using the OCI profile configuration file.

```
provider "oci" {
  config_file_profile = var.oci_config_file_profile
  tenancy_ocid        = var.tenancy_ocid
}
```

main.tf

This file specifies the provider to use (`oracle/oci`), defines several security list rules, and initializes required local variables.

The version of the OCI provider that you use must be at least v4.50.0 but no greater than v6.36.0.

```
terraform {
  required_providers {
    oci = {
      source  = "oracle/oci"
      version = ">= 4.50.0, <= 6.36.0"
      # If necessary, you can pin a specific version here
      # version = "4.71.0"
    }
  }
  required_version = ">= 1.1"
}

locals {
  kube_internal_cidr = "253.255.0.0/16"
  worker_lb_ingress_rules = [
    {
      source   = var.kube_client_cidr
      port_min = 80
      port_max = 80
    },
    {
      source   = var.kube_client_cidr
      port_min = 443
      port_max = 443
    }
  ]
  worker_ingress_rules = [
     {
      source   = var.kube_client_cidr
      port_min = 30000
      port_max = 32767
    },
    {
      source   = var.vcn_cidr
```

```
      port_min = 22
      port_max = 22
    },
    {
      source   = var.workerlb_cidr
      port_min = 30000
      port_max = 32767
    },
    {
      source   = var.workerlb_cidr
      port_min = 10256
      port_max = 10256
    },
    {
      source   = var.kmi_cidr
      port_min = 22
      port_max = 65535
    },
  ]

  kmi_lb_ingress_rules = [
    {
      source   = local.kube_internal_cidr
      port_min = var.kubernetes_api_port
      port_max = var.kubernetes_api_port
    },
    {
      source   = var.kube_client_cidr
      port_min = var.kubernetes_api_port
      port_max = var.kubernetes_api_port
    },
    {
      source   = var.vcn_cidr
      port_min = var.kubernetes_api_port
      port_max = var.kubernetes_api_port
    },
  ]
  kmi_ingress_rules = [
    {
      source   = var.kube_client_cidr
      port_min = var.kubernetes_api_port
      port_max = var.kubernetes_api_port
    },
    {
      source   = var.kmilb_cidr
      port_min = var.kubernetes_api_port
      port_max = var.kubernetes_api_port
    },
    {
      source   = var.worker_cidr
      port_min = 1024
      port_max = 65535
    },
    {
      source   = var.kmi_cidr
      port_min = 1024
      port_max = 65535
    },
  ]
  worker_ingress_udp_rules = [
    {
      source   = var.worker_cidr
      port_min = 8285
      port_max = 8472
```

```
    },
    {
      source   = var.kmi_cidr
      port_min = 8285
      port_max = 8472
    },
  ]
  kmi_ingress_udp_rules = [
    {
      source   = var.worker_cidr
      port_min = 8285
      port_max = 8472
    },
    {
      source   = var.kmi_cidr
      port_min = 8285
      port_max = 8472
    },
  ]
}
```

oke_vcn.tf

This file defines a VCN, NAT gateway, internet gateway, private route table, and public route table. The private route table is the default route table for the VCN.

```
resource "oci_core_vcn" "oke_vcn" {
  cidr_block     = var.vcn_cidr
  dns_label      = var.vcn_name
  compartment_id = var.compartment_id
  display_name   = "${var.vcn_name}-vcn"
}

resource "oci_core_nat_gateway" "vcn_ngs" {
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id
  count          = var.enable_private_endpoint ? 0:1

  display_name = "VCN nat g6s"
}

resource "oci_core_internet_gateway" "vcn_igs" {
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id
  count          = var.enable_private_endpoint ? 0:1

  display_name = "VCN i6t g6s"
  enabled      = true
}

resource "oci_core_default_route_table" "default_private" {
  manage_default_resource_id = oci_core_vcn.oke_vcn.default_route_table_id
  display_name               = "Default - private"
  count                      = var.enable_private_endpoint ? 1:0
}

resource "oci_core_default_route_table" "private" {
  count          = var.enable_private_endpoint ? 0:1
  manage_default_resource_id = oci_core_vcn.oke_vcn.default_route_table_id
  display_name               = "Default - private"

  route_rules {
```

```
    destination       = "0.0.0.0/0"
    destination_type  = "CIDR_BLOCK"
    network_entity_id = oci_core_nat_gateway.vcn_ngs[0].id
  }
}

resource "oci_core_route_table" "public" {
  count          = var.enable_private_endpoint ? 0:1
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id

  display_name = "public"
  route_rules {
    destination       = "0.0.0.0/0"
    destination_type  = "CIDR_BLOCK"
    network_entity_id = oci_core_internet_gateway.vcn_igs[0].id
  }
}
```

oke_worker_seclist.tf

This file defines the security lists for both the worker subnet and the worker load balancer subnet. The rules for these security lists were defined in other Terraform files in this set.

```
resource "oci_core_security_list" "workerlb" {
  display_name   = "${var.vcn_name}-workerlb"
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id

  dynamic "ingress_security_rules" {
    iterator = port
    for_each = local.worker_lb_ingress_rules

    content {
      source      = port.value.source
      source_type = "CIDR_BLOCK"
      protocol    = "6"
      tcp_options {
        min = port.value.port_min
        max = port.value.port_max
      }
    }
  }
}

resource "oci_core_security_list" "worker" {
  display_name   = "${var.vcn_name}-worker"
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id

  dynamic "ingress_security_rules" {
    iterator = port
    for_each = local.worker_ingress_rules

    content {
      source      = port.value.source
      source_type = "CIDR_BLOCK"
      protocol    = "6"
      tcp_options {
        min = port.value.port_min
        max = port.value.port_max
      }
```

```
    }
  }

  dynamic "ingress_security_rules" {
    iterator = port
    for_each = local.worker_ingress_udp_rules

    content {
      source      = port.value.source
      source_type = "CIDR_BLOCK"
      protocol    = "17"
      udp_options {
        min = port.value.port_min
        max = port.value.port_max
      }
    }
  }
}
```

oke_worker_subnet.tf

This file defines the worker and worker load balancer subnets. The worker load balancer subnet is named `service-lb`.

```
resource "oci_core_subnet" "worker" {
  cidr_block     = var.worker_cidr
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id

  display_name   = "worker"
  dns_label      = "worker"
  prohibit_public_ip_on_vnic = true

  security_list_ids = [
    oci_core_default_security_list.oke_vcn.id,
    oci_core_security_list.worker.id
  ]
}

resource "oci_core_subnet" "worker_lb" {
  cidr_block     = var.workerlb_cidr
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id

  display_name               = "service-lb"
  dns_label                  = "servicelb"
  prohibit_public_ip_on_vnic = var.enable_private_endpoint
  route_table_id             = var.enable_private_endpoint==false ?
 oci_core_route_table.public[0].id :
 oci_core_vcn.oke_vcn.default_route_table_id

  security_list_ids = [
    oci_core_default_security_list.oke_vcn.id,
    oci_core_security_list.workerlb.id
  ]
}
```

oke_kmi_seclist.tf

This file defines the security lists for the control plane and control plane load balancer subnets. This file also defines updates to make to the default security list for the VCN.

```
resource "oci_core_default_security_list" "oke_vcn" {
  manage_default_resource_id = oci_core_vcn.oke_vcn.default_security_list_id

  egress_security_rules {
    destination      = "0.0.0.0/0"
    destination_type = "CIDR_BLOCK"
    protocol         = "all"
  }

  dynamic "ingress_security_rules" {
    iterator = icmp_type
    for_each = [3, 8, 11]

    content {
      # ping from VCN; unreachable/TTL from anywhere
      source      = (icmp_type.value == "8" ? var.vcn_cidr : "0.0.0.0/0")
      source_type = "CIDR_BLOCK"
      protocol    = "1"
      icmp_options {
        type = icmp_type.value
      }
    }
  }
}

resource "oci_core_security_list" "kmilb" {
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id

  display_name = "${var.vcn_name}-kmilb"

  dynamic "ingress_security_rules" {
    iterator = port
    for_each = local.kmi_lb_ingress_rules

    content {
      source      = port.value.source
      source_type = "CIDR_BLOCK"
      protocol    = "6"
      tcp_options {
        min = port.value.port_min
        max = port.value.port_max
      }
    }
  }

  dynamic "ingress_security_rules" {
    for_each = var.enable_private_endpoint ? [] : [0]

    content {
      source      = var.public_ip_cidr
      source_type = "CIDR_BLOCK"
      protocol    = "6"
      tcp_options {
        min = var.kubernetes_api_port
        max = var.kubernetes_api_port
      }
    }
```

```
  }
}

resource "oci_core_security_list" "kmi" {
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id

  display_name = "${var.vcn_name}-kmi"

  dynamic "ingress_security_rules" {
    iterator = port
    for_each = local.kmi_ingress_rules

    content {
      source      = port.value.source
      source_type = "CIDR_BLOCK"
      protocol    = "6"
      tcp_options {
        min = port.value.port_min
        max = port.value.port_max
      }
    }
  }

  dynamic "ingress_security_rules" {
    iterator = port
    for_each = local.kmi_ingress_udp_rules

    content {
      source      = port.value.source
      source_type = "CIDR_BLOCK"
      protocol    = "17"
      udp_options {
        min = port.value.port_min
        max = port.value.port_max
      }
    }
  }
}
```

oke_kmi_subnet.tf

This file defines the control plane and control plane load balancer subnets.

> **Important:**
>
> The name of the `kmi` subnet must be exactly `control-plane`.

```
resource "oci_core_subnet" "kmi" {
  cidr_block                 = var.kmi_cidr
  compartment_id             = var.compartment_id
  display_name               = "control-plane"
  dns_label                  = "kmi"
  vcn_id                     = oci_core_vcn.oke_vcn.id
  prohibit_public_ip_on_vnic = true
  security_list_ids = [
    oci_core_default_security_list.oke_vcn.id,
    oci_core_security_list.kmi.id
  ]
}

resource "oci_core_subnet" "kmi_lb" {
  cidr_block                 = var.kmilb_cidr
```

```
  compartment_id             = var.compartment_id
  dns_label                  = "kmilb"
  vcn_id                     = oci_core_vcn.oke_vcn.id
  display_name               = "control-plane-endpoint"
  prohibit_public_ip_on_vnic = var.enable_private_endpoint
  route_table_id             = var.enable_private_endpoint==false ?
 oci_core_route_table.public[0].id :
 oci_core_default_route_table.default_private[0].id
  security_list_ids = [
    oci_core_default_security_list.oke_vcn.id,
    oci_core_security_list.kmilb.id
  ]
}
```

Creating a VCN (Flannel Overlay)

Learn how to create a VCN for Flannel Overlay networking On Private Cloud Appliance.

Create the following resources in the order listed:

1. Create the VCN on page 848
2. Create a route table with the following route rules:
   - Public clusters:
     - Internet gateway and a route table with a route rule that references that internet gateway.
     - NAT gateway and a route table with a route rule that references that NAT gateway.
   - Private clusters:
     - Route table with no route rules.
     - (Optional) Dynamic Routing Gateway (DRG) and a route table with a route rule that references that DRG. See Private Clusters on page 835.
     - (Optional) Local Peering Gateway (LPG) and a route table with a route rule that references that LPG. See Private Clusters on page 835.
3. Modify the VCN Default Security List on page 851

Resource names and CIDR blocks are example values.

Create the VCN

To create the VCN, follow the instructions in Creating a VCN on page 479, and use the parameters listed in this section. For Terraform input, see Example Terraform Scripts for Network Resources (Flannel Overlay) on page 839.

> **Note:**
>
> Subnets are created later, and described in later sections.

For this example, use the following input to create the VCN. The VCN covers one contiguous CIDR block. The CIDR block can't be changed after the VCN is created.

Compute Web UI property	OCI CLI property
• Name: oketest-vcn • CIDR Block: *vcn_cidr* • DNS Label: oketest This label must be unique across all VCNs in the tenancy.	• `--display-name`: `oketest-vcn` • `--cidr-blocks`: `'["`*vcn_cidr*`"]'` • `--dns-label`: `oketest` This label must be unique across all VCNs in the tenancy.

Note the OCID of the new VCN for use later. In the examples in this guide, this VCN OCID is `ocid1.vcn.`*oke_vcn_id*.

Next Steps

- Public internet access. For traffic on a public subnet that connects to the internet using public IP addresses, create an internet gateway and a route rule that references that internet gateway.
- Private internet access. For traffic on a private subnet that needs to connect to the internet without exposing private IP addresses, create a NAT gateway and a route rule that references that NAT gateway.
- VCN-only access. To restrict communication to only other resources on the same VCN, use the default route table, which has no route rules.
- Instances in another VCN. To enable communication between the cluster and an instance running on a different VCN, create a Local Peering Gateway (LPG) and a route rule that references that LPG.
- On-premises IP address space. To enable communication between the cluster and the on-premises network IP address space, create a Dynamic Routing Gateway (DRG), attach the OKE VCN to that DRG, and create a route rule that references that DRG.

Edit the VCN Private Route Table

Edit the default route table that was created when you created the VCN. Change the name of the route table to vcn_private. This route table does not have any route rules. Do not add any route rules.

Create a NAT Private Route Table

Create a NAT gateway and a route table with a route rule that references the NAT gateway.

NAT Gateway

To create the NAT gateway, use the instructions in Configuring a NAT Gateway on page 506. For Terraform input, see Example Terraform Scripts for Network Resources (Flannel Overlay) on page 839.

Note the name and OCID of the NAT gateway for assignment to the private route rule.

Private Route Rule

Modify the default route table, using the following input to create a private route rule that references the NAT gateway that was created in the preceding step. See Creating a Route Table on page 493.

For this example, use the following input to create the route table with a private route rule that references the NAT gateway that was created in the preceding step.

Compute Web UI property	OCI CLI property
• Name: nat_private	• `--display-name`: nat_private
Route rule	`--route-rules`
• Target Type: NAT Gateway • NAT Gateway: Name of the NAT gateway that was created in the preceding step • CIDR Block: 0.0.0.0/0 • Description: NAT private route rule	• `networkEntityId`: OCID of the NAT gateway that was created in the preceding step • `destinationType`: `CIDR_BLOCK` • `destination`: `0.0.0.0/0` • `description`: NAT private route rule

Note the name and OCID of this route table for assignment to private subnets.

Create a Local Peering Gateway

Create a Local Peering gateway (LPG) and a route table with a route rule that references the LPG.

Local Peering Gateway

Create the LPG. See Connecting VCNs through a Local Peering Gateway (LPG) on page 509.

Note the name and OCID of the LPG for assignment to the private route rule.

Private Route Rule

Create a route table. See Creating a Route Table on page 493.

For this example, use the following input to create the route table with a private route rule that references the LPG that was created in the preceding step.

Compute Web UI property	OCI CLI property
• Name: lpg_rt	• `--display-name`: lpg_rt
Route rule	`--route-rules`
• Target Type: Local Peering Gateway • Local Peering Gateway: Name of the LPG that was created in the preceding step • CIDR Block: *`CIDR_for_the_second_VCN`* • Description: LPG private route rule	• `networkEntityId`: OCID of the LPG that was created in the preceding step • `destinationType`: `CIDR_BLOCK` • `destination`: *`CIDR_for_the_second_VCN`* • `description`: LPG private route rule

Note the name and OCID of this route table for assignment to the "control-plane-endpoint" subnet (Creating an OKE Control Plane Load Balancer Subnet (Flannel Overlay) on page 861).

Add the same route rule on the second VCN (the peered VCN), specifying the OKE VCN CIDR as the destination.

Create a Dynamic Routing Gateway

Create a Dynamic Routing gateway (DRG) and a route table with a route rule that references the DRG.

Dynamic Routing Gateway

To create the DRG and attach the OKE VCN to that DRG See Connecting to the On-Premises Network through a Dynamic Routing Gateway (DRG) on page 511. Create the DRG in the OKE VCN compartment, and then attach the OKE VCN to that DRG.

Note the name and OCID of the DRG for assignment to the private route rule.

Private Route Rule

Create a route table. See Creating a Route Table on page 493.

For this example, use the following input to create the route table with a private route rule that references the DRG that was created in the preceding step.

Compute Web UI property	OCI CLI property
• Name: drg_rt	• `--display-name`: drg_rt
Route rule	`--route-rules`
• Target Type: Dynamic Routing Gateway • Dynamic Routing: Name of the DRG that was created in the preceding step • CIDR Block: 0.0.0.0/0 • Description: DRG private route rule	• `networkEntityId`: OCID of the DRG that was created in the preceding step • `destinationType`: `CIDR_BLOCK` • `destination`: `0.0.0.0/0` • `description`: DRG private route rule

Note the name and OCID of this route table for assignment to the "control-plane-endpoint" subnet (Creating an OKE Control Plane Load Balancer Subnet (Flannel Overlay) on page 861).

Create a Public Route Table

Create an Internet gateway and a route table with a route rule that references the Internet gateway. This enables internet access for OKE nodes.

Create an Internet Gateway

To create the internet gateway, use the instructions in Configuring an Internet Gateway on page 507. For Terraform input, see Example Terraform Scripts for Network Resources (Flannel Overlay) on page 839.

Note the name and OCID of the internet gateway for assignment to the public route rule.

Create a Public Route Rule

Create a public route rule for the internet gateway you just created. To create a route table, use the instructions in Creating a Route Table on page 493. For Terraform input, see Example Terraform Scripts for Network Resources (Flannel Overlay) on page 839.

For this example, use the following input to create the route table with a public route rule that references the internet gateway that was created in the preceding step.

Compute Web UI property	**OCI CLI property**
• Name: public Route rule • Target Type: Internet Gateway • Internet Gateway: Name of the internet gateway that was created in the preceding step • CIDR Block: 0.0.0.0/0 • Description: OKE public route rule	• `--vcn-id`: `ocid1.vcn.`*`oke_vcn_id`* • `--display-name`: public `--route-rules` • `networkEntityId`: OCID of the internet gateway that was created in the preceding step • `destinationType`: `CIDR_BLOCK` • `destination`: `0.0.0.0/0` • `description`: OKE public route rule

Modify the VCN Default Security List

To modify a security list, see Updating a Security List on page 499. For Terraform input, see Example Terraform Scripts for Network Resources (Flannel Overlay) on page 839.

Delete all the default rules, then create the rules shown in the following table.

Compute Web UI property	**OCI CLI property**
• Name: Default	`--security-list-id`: `ocid1.securitylist.`*`default_securitylist_id`*
One egress security rule: • Stateless: clear the box • Egress CIDR: 0.0.0.0/0 • IP Protocol: All protocols • Description: "Allow all outgoing traffic."	**One egress security rule:** `--egress-security-rules` • `isStateless`: `false` • `destination`: `0.0.0.0/0` • `destinationType`: `CIDR_BLOCK` • `protocol`: `all` • `description`: "Allow all outgoing traffic."
Three ingress security rules:	**Three ingress security rules:** `--ingress-security-rules`

Compute Web UI property	**OCI CLI property**

Ingress Rule 1
• Stateless: clear the box
• Ingress CIDR: *vcn_cidr*
• IP Protocol: ICMP
  • Parameter Type: 8: Echo
• Description: "Allow ping from VCN."
	Ingress Rule 1
	• `isStateless`: `false`
	• `source`: *vcn_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `1`
	• `icmpOptions`
	  • `type`: `8`
	• `description`: "Allow ping from VCN."

Ingress Rule 2
• Stateless: clear the box
• Ingress CIDR: 0.0.0.0/0
• IP Protocol: ICMP
  • Parameter Type: 3: Destination Unreachable
• Description: "Blocks incoming requests from any source."
	Ingress Rule 2
	• `isStateless`: `false`
	• `source`: `0.0.0.0/0`
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `1`
	• `icmpOptions`
	  • `type`: `3`
	• `description`: "Blocks incoming requests from any source."

Ingress Rule 3
• Stateless: clear the box
• Ingress CIDR: 0.0.0.0/0
• IP Protocol: ICMP
  • Parameter Type: 11: Time Exceeded
• Description: "Time exceeded."
	Ingress Rule 3
	• `isStateless`: `false`
	• `source`: `0.0.0.0/0`
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `1`
	• `icmpOptions`
	  • `type`: `11`
	• `description`: "Time exceeded."

Note the name and OCID of this default security list for assignment to subnets.

What's Next:

Creating a Worker Subnet (Flannel Overlay) on page 852

Creating a Worker Subnet (Flannel Overlay)

Learn how to create a worker subnet for Flannel Overlay networking On Private Cloud Appliance.

Create the following resources in the order listed:

1. Create a Worker Security List on page 852.
2. Create the Worker Subnet on page 855.

Create a Worker Security List

To create a security list, use the instructions in Creating a Security List on page 496. For Terraform input, see Example Terraform Scripts for Network Resources (Flannel Overlay) on page 839.

This security list defines traffic that's allowed to contact worker nodes directly.

For this example, use the following input for the worker subnet security list.

Compute Web UI	OCI CLI property
• Name: worker-seclist	• `--vcn-id`: `ocid1.vcn.`*`oke_vcn_id`* • `--display-name`: `worker-seclist`
Seven ingress security rules:	**Seven ingress security rules:** `--ingress-security-rules`

Ingress Rule 1
• Stateless: clear the box
• Ingress CIDR: *`vcn_cidr`*
• IP Protocol: TCP
  • Destination Port Range: 22
• Description: "Allow intra-VCN `ssh`."
	Ingress Rule 1
	• `isStateless`: `false`
	• `source`: *`vcn_cidr`*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `22`
	  • `min`: `22`
	• `description`: "Allow intra-VCN `ssh`."

Ingress Rule 2
• Stateless: clear the check box
• Ingress CIDR: *`kube_client_cidr`*
• IP Protocol: TCP
  • Destination Port Range: 30000-32767
• Description: "Allow clients to contact the node port range."
	Ingress Rule 2
	• `isStateless`: `false`
	• `source`: *`kube_client_cidr`*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `32767`
	  • `min`: `30000`
	• `description`: "Allow clients to contact the node port range."

Ingress Rule 3
• Stateless: clear the box
• Ingress CIDR: *`workerlb_cidr`*
• IP Protocol: TCP
  • Destination Port Range: 30000-32767
• Description: "Allow the worker load balancer to contact the worker nodes."
	Ingress Rule 3
	• `isStateless`: `false`
	• `source`: *`workerlb_cidr`*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `32767`
	  • `min`: `30000`
	• `description`: "Allow the worker load balancer to contact the worker nodes."

Compute Web UI	OCI CLI property

Ingress Rule 4
• Stateless: clear the box
• Ingress CIDR: *workerlb_cidr*
• IP Protocol: TCP
  • Destination Port Range: 10256
• Description: "Allow the worker load balancer to contact the worker nodes."
	Ingress Rule 4
	• `isStateless`: `false`
	• `source`: *workerlb_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `10256`
	  • `min`: `10256`
	• `description`: "Allow the worker load balancer to contact the worker nodes."

Ingress Rule 5
• Stateless: clear the box
• Ingress CIDR: *kmi_cidr*
• IP Protocol: TCP
  • Destination Port Range: 22-65535
• Description: "Allow the control plane to contact the worker nodes."
	Ingress Rule 5
	• `isStateless`: `false`
	• `source`: *kmi_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `65535`
	  • `min`: `22`
	• `description`: "Allow the control plane to contact the worker nodes."

Ingress Rule 6
• Stateless: clear the box
• Ingress CIDR: *worker_cidr*
• IP Protocol: UDP
  • Destination Port Range: 8285-8472
• Description: "Allow flannel traffic."
	Ingress Rule 6
	• `isStateless`: `false`
	• `source`: *worker_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `17`
	• `udpOptions`
	`destinationPortRange`
	  • `max`: `8472`
	  • `min`: `8285`
	• `description`: "Allow flannel traffic."

Compute Web UI	OCI CLI property

Ingress Rule 7
• Stateless: clear the box
• Ingress CIDR: *kmi_cidr*
• IP Protocol: UDP
  • Destination Port Range: 8285-8472
• Description: "Allow flannel traffic."
	Ingress Rule 7
	• `isStateless`: `false`
	• `source`: *kmi_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `17`
	• `udpOptions`
	`destinationPortRange`
	  • `max`: `8472`
	  • `min`: `8285`
	• `description`: "Allow flannel traffic."

Create the Worker Subnet

To create a subnet, use the instructions in Creating a Subnet on page 480. For Terraform input, see Example Terraform Scripts for Network Resources (Flannel Overlay) on page 839.

For this example, use the following input for the worker subnet security list. Use the OCID of the VCN that was created in Example Terraform Scripts for Network Resources (Flannel Overlay) on page 839. Create the worker subnet in the same compartment where you created the VCN.

Create either a NAT private worker subnet or a VCN private worker subnet. Create a NAT private worker subnet to communicate outside the VCN.

Create a NAT Private Worker Subnet

Compute Web UI property	OCI CLI property

• Name: worker
• CIDR Block: *worker_cidr*
• Route Table: Select "nat_private" from the list
• Private Subnet: check the box
• DNS Hostnames:
  Use DNS Hostnames in this Subnet: check the box
  • DNS Label: worker
• Security Lists: Select "worker-seclist" and "Default Security List for oketest-vcn" from the list
	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id*
	• `--display-name`: `worker`
	• `--cidr-block`: *worker_cidr*
	• `--dns-label`: `worker`
	• `--prohibit-public-ip-on-vnic`: `true`
	• `--route-table-id`: OCID of the "nat_private" route table
	• `--security-list-ids`: OCIDs of the "worker-seclist" security list and the "Default Security List for oketest-vcn" security list

The difference in the following private subnet is the VCN private route table is used instead of the NAT private route table.

Create a VCN Private Worker Subnet

Compute Web UI property	**OCI CLI property**
• Name: worker • CIDR Block: *worker_cidr* • Route Table: Select "vcn_private" from the list • Private Subnet: check the box • DNS Hostnames: Use DNS Hostnames in this Subnet: check the box • DNS Label: worker • Security Lists: Select "worker-seclist" and "Default Security List for oketest-vcn" from the list	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id* • `--display-name`: `worker` • `--cidr-block`: *worker_cidr* • `--dns-label`: `worker` • `--prohibit-public-ip-on-vnic`: `true` • `--route-table-id`: OCID of the "vcn_private" route table • `--security-list-ids`: OCIDs of the "worker-seclist" security list and the "Default Security List for oketest-vcn" security list

What's Next:

Creating a Worker Load Balancer Subnet (Flannel Overlay) on page 856

Creating a Worker Load Balancer Subnet (Flannel Overlay)

Learn how to create a worker load balancer subnet for Flannel Overlay networking On Private Cloud Appliance.

Create the following resources in the order listed:

1. Create a Worker Load Balancer Security List on page 856
2. Create the Worker Load Balancer Subnet on page 857

Create a Worker Load Balancer Security List

Create a security list and a worker load balancer subnet. To create a security list, use the instructions in Creating a Security List on page 496. For Terraform input, see Example Terraform Scripts for Network Resources (Flannel Overlay) on page 839.

This security list defines traffic, such as applications, that's allowed to contact the worker load balancer.

For this example, use the following input for the worker load balancer subnet security list. These sources and destinations are examples; adjust these for your applications.

> **Note:**
>
> When you create an external load balancer for your containerized applications (see Exposing Containerized Applications on page 947), remember to add that load balancer service front-end port to this security list.

Compute Web UI property	**OCI CLI property**
• Name: workerlb-seclist	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id* • `--display-name`: `workerlb-seclist`
Two ingress security rules:	**Two ingress security rules:** `--ingress-security-rules`

Compute Web UI property	**OCI CLI property**

Ingress Rule 1
• Stateless: clear the box
• Ingress CIDR: *kube_client_cidr*
• IP Protocol: TCP
  • Destination Port Range: 80
• Description: "Allow inbound traffic for applications."
	Ingress Rule 1
	• `isStateless`: `false`
	• `source`: *kube_client_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `80`
	  • `min`: `80`
	• `description`: "Allow inbound traffic for applications."

Ingress Rule 2
• Stateless: clear the box
• Ingress CIDR: *kube_client_cidr*
• IP Protocol: TCP
  • Destination Port Range: 443
• Description: "Allow inbound traffic for applications."
	Ingress Rule 2
	• `isStateless`: `false`
	• `source`: *kube_client_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `443`
	  • `min`: `443`
	• `description`: "Allow inbound traffic for applications."

Create the Worker Load Balancer Subnet

To create a subnet, use the instructions in Creating a Subnet on page 480 For Terraform input, see Example Terraform Scripts for Network Resources (Flannel Overlay) on page 839.

For this example, use the following input to create the worker load balancer subnet. Use the OCID of the VCN that was created in Creating a VCN (Flannel Overlay) on page 848. Create the worker load balancer subnet in the same compartment where you created the VCN.

Create either a private or a public worker load balancer subnet. Create a public worker load balancer subnet to use with a public cluster. Create a private worker load balancer subnet to expose applications in a private cluster.

Create a Public Worker Load Balancer Subnet

Compute Web UI property	**OCI CLI property**
• Name: service-lb • CIDR Block: *workerlb_cidr* • Route Table: Select "public" from the list • Public Subnet: check the box • DNS Hostnames: Use DNS Hostnames in this Subnet: check the box • DNS Label: servicelb • Security Lists: Select "workerlb-seclist" and "Default Security List for oketest-vcn" from the list	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id* • `--display-name`: `service-lb` • `--cidr-block`: *workerlb_cidr* • `--dns-label`: `servicelb` • `--prohibit-public-ip-on-vnic`: `false` • `--route-table-id`: OCID of the "public" route table • `--security-list-ids`: OCIDs of the "workerlb-seclist" security list and the "Default Security List for oketest-vcn" security list

The difference in the following private subnet is the VCN private route table is used instead of the public route table.

Create a Private Worker Load Balancer Subnet

Compute Web UI property	**OCI CLI property**
• Name: service-lb • CIDR Block: *workerlb_cidr* • Route Table: Select "vcn_private" from the list • Private Subnet: check the box • DNS Hostnames: Use DNS Hostnames in this Subnet: check the box • DNS Label: servicelb • Security Lists: Select "workerlb-seclist" and "Default Security List for oketest-vcn" from the list	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id* • `--display-name`: `service-lb` • `--cidr-block`: *workerlb_cidr* • `--dns-label`: `servicelb` • `--prohibit-public-ip-on-vnic`: `true` • `--route-table-id`: OCID of the "vcn_private" route table • `--security-list-ids`: OCIDs of the "workerlb-seclist" security list and the "Default Security List for oketest-vcn" security list

What's Next:

Creating an OKE Control Plane Subnet (Flannel Overlay) on page 858

Creating an OKE Control Plane Subnet (Flannel Overlay)

Learn how to create a control plane subnet for Flannel Overlay networking On Private Cloud Appliance.

Create the following resources in the order listed:

1. Create a Control Plane Security List on page 858
2. Create the Control Plane Subnet on page 860

Create a Control Plane Security List

To create a security list, use the instructions in Creating a Security List on page 496. For Terraform input, see Example Terraform Scripts for Network Resources (Flannel Overlay) on page 839.

For this example, use the following input for the control plane subnet security list. Use this configuration as a guide when you create these resources. You can change the values of properties such as CIDR blocks and IP addresses. You should not change the values of properties such as the network protocol, the stateful setting, or the private/public setting.

Compute Web UI property	**OCI CLI property**
• Name: kmi-seclist	• `--vcn-id`: `ocid1.vcn.`*`oke_vcn_id`* • `--display-name`: `kmi-seclist`
Six ingress security rules:	**Six ingress security rules:** `--ingress-security-rules`

Ingress Rule 1
• Stateless: clear the box
• Ingress CIDR: *`kube_client_cidr`*
• IP Protocol: TCP
  • Destination Port Range: *`kubernetes_api_port`*
• Description: "Allow inbound connections to the Kubernetes API server."
	Ingress Rule 1
	• `isStateless`: `false`
	• `source`: *`kube_client_cidr`*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: *`kubernetes_api_port`*
	  • `min`: *`kubernetes_api_port`*
	• `description`: "Allow inbound connections to the Kubernetes API server."

Ingress Rule 2
• Stateless: clear the box
• Ingress CIDR: *`kmilb_cidr`*
• IP Protocol: TCP
  • Destination Port Range: *`kubernetes_api_port`*
• Description: "Allow inbound connections from the control plane load balancer."
	Ingress Rule 2
	• `isStateless`: `false`
	• `source`: *`kmilb_cidr`*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: *`kubernetes_api_port`*
	  • `min`: *`kubernetes_api_port`*
	• `description`: "Allow inbound connections from the control plane load balancer."

Ingress Rule 3
• Stateless: clear the box
• Ingress CIDR: *`worker_cidr`*
• IP Protocol: TCP
  • Destination Port Range: 1024-65535
• Description: "Allow inbound connections from worker nodes to the control plane."
	Ingress Rule 3
	• `isStateless`: `false`
	• `source`: *`worker_cidr`*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `65535`
	  • `min`: `1024`
	• `description`: "Allow inbound connections from worker nodes to the control plane."

Compute Web UI property	**OCI CLI property**

Ingress Rule 4
• Stateless: clear the box
• Ingress CIDR: *kmi_cidr*
• IP Protocol: TCP
  • Destination Port Range: 1024-65535
• Description: "Allow inbound connections within the control plane."
	Ingress Rule 4
	• `isStateless`: `false`
	• `source`: *kmi_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `65535`
	  • `min`: `1024`
	• `description`: "Allow inbound connections within the control plane."

Ingress Rule 5
• Stateless: clear the box
• Ingress CIDR: *worker_cidr*
• IP Protocol: UDP
  • Destination Port Range: 8285-8472
• Description: "Allow flannel traffic."
	Ingress Rule 5
	• `isStateless`: `false`
	• `source`: *worker_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `17`
	• `udpOptions`
	`destinationPortRange`
	  • `max`: `8472`
	  • `min`: `8285`
	• `description`: "Allow flannel traffic."

Ingress Rule 6
• Stateless: clear the box
• Ingress CIDR: *kmi_cidr*
• IP Protocol: UDP
  • Destination Port Range: 8285-8472
• Description: "Allow flannel traffic."
	Ingress Rule 6
	• `isStateless`: `false`
	• `source`: *kmi_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `17`
	• `udpOptions`
	`destinationPortRange`
	  • `max`: `8472`
	  • `min`: `8285`
	• `description`: "Allow flannel traffic."

Create the Control Plane Subnet

To create a subnet, use the instructions in Creating a Subnet on page 480. For Terraform input, see Example Terraform Scripts for Network Resources (Flannel Overlay) on page 839.

Use the following input to create the control plane subnet. Use the OCID of the VCN that was created in Creating a VCN (Flannel Overlay) on page 848. Create the control plane subnet in the same compartment where you created the VCN.

Create either a NAT private control plane subnet or a VCN private control plane subnet. Create a NAT private control plane subnet to communicate outside the VCN.

> **Important:**
>
> The name of this subnet must be exactly `control-plane`.

Create a NAT Private Control Plane Subnet

Compute Web UI property	**OCI CLI property**
• Name: control-plane • CIDR Block: *kmi_cidr* • Route Table: Select "nat_private" from the list • Private Subnet: check the box • DNS Hostnames: Use DNS Hostnames in this Subnet: check the box • DNS Label: kmi • Security Lists: Select "kmi-seclist" and "Default Security List for oketest-vcn" from the list	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id* • `--display-name`: `control-plane` • `--cidr-block`: *kmi_cidr* • `--dns-label`: `kmi` • `--prohibit-public-ip-on-vnic`: `true` • `--route-table-id`: OCID of the "nat_private" route table • `--security-list-ids`: OCIDs of the "kmi-seclist" security list and the "Default Security List for oketest-vcn" security list

Create a VCN Private Control Plane Subnet

Compute Web UI property	**OCI CLI property**
• Name: control-plane • CIDR Block: *kmi_cidr* • Route Table: Select "vcn_private" from the list • Private Subnet: check the box • DNS Hostnames: Use DNS Hostnames in this Subnet: check the box • DNS Label: kmi • Security Lists: Select "kmi-seclist" and "Default Security List for oketest-vcn" from the list	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id* • `--display-name`: `control-plane` • `--cidr-block`: *kmi_cidr* • `--dns-label`: `kmi` • `--prohibit-public-ip-on-vnic`: `true` • `--route-table-id`: OCID of the "vcn_private" route table • `--security-list-ids`: OCIDs of the "kmi-seclist" security list and the "Default Security List for oketest-vcn" security list

What's Next:

Creating an OKE Control Plane Load Balancer Subnet (Flannel Overlay) on page 861

Creating an OKE Control Plane Load Balancer Subnet (Flannel Overlay)

Learn how to create a control plane load balancer subnet for Flannel Overlay networking On Private Cloud Appliance.

Create the following resources in the order listed:

1. Create a Control Plane Load Balancer Security List on page 861.
2. Create the Control Plane Load Balancer Subnet on page 863.

Create a Control Plane Load Balancer Security List

To create a security list, use the instructions in Creating a Security List on page 496. For Terraform input, see Example Terraform Scripts for Network Resources (Flannel Overlay) on page 839.

The control plane load balancer accepts traffic on port 6443, which is also called *kubernetes_api_port* in these instructions. Adjust this security list to only accept connections from where you expect the network to run. Port 6443 must accept connections from the cluster control plane instances and worker instances.

For this example, use the following input for the control plane load balancer subnet security list.

Compute Web UI property	**OCI CLI property**
• Name: kmilb-seclist	• `--vcn-id`: `ocid1.vcn.`*`oke_vcn_id`* • `--display-name`: kmilb-seclist
Four ingress security rules:	**Four ingress security rules:** `--ingress-security-rules`

Ingress Rule 1:
• Stateless: clear the box
• Ingress CIDR:
`kube_internal_cidr`
This value is required. Do not change this CIDR value.
• IP Protocol: TCP
  • Destination Port Range: *`kubernetes_api_port`*
• Description: "Allow inbound connections to the control plane load balancer."
	Ingress Rule 1:
	• `isStateless`: `false`
	• `source`:
	`kube_internal_cidr`
	This value is required. Do not change this CIDR value.
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: *`kubernetes_api_port`*
	  • `min`: *`kubernetes_api_port`*
	• `description`: "Allow inbound connections to the control plane load balancer."

Ingress Rule 2:
• Stateless: clear the box
• Ingress CIDR: *`kube_client_cidr`*
• IP Protocol: TCP
  • Destination Port Range: *`kubernetes_api_port`*
• Description: "Allow inbound connections to the control plane load balancer."
	Ingress Rule 2:
	• `isStateless`: `false`
	• `source`: *`kube_client_cidr`*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: *`kubernetes_api_port`*
	  • `min`: *`kubernetes_api_port`*
	• `description`: "Allow inbound connections to the control plane load balancer."

Ingress Rule 3:
• Stateless: clear the box
• Ingress CIDR: *`vcn_cidr`*
• IP Protocol: TCP
  • Destination Port Range: *`kubernetes_api_port`*
• Description: "Allow inbound connections to the control plane load balancer."
	Ingress Rule 3:
	• `isStateless`: `false`
	• `source`: *`vcn_cidr`*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: *`kubernetes_api_port`*
	  • `min`: *`kubernetes_api_port`*
	• `description`: "Allow inbound connections to the control plane load balancer."

Compute Web UI property	OCI CLI property

Ingress Rule 4: Public Endpoint
• Stateless: clear the box
• Ingress CIDR: *public_ip_cidr*
• IP Protocol: TCP
  • Destination Port Range: *kubernetes_api_port*
• Description: "Used to access the control plane endpoint from the public CIDR. The public IP CIDR is configured in the Service Enclave. If you do not know what your public IP CIDR is, ask your Service Enclave administrator."
	Ingress Rule 4: Public Endpoint
	• `isStateless`: `false`
	• `source`: *public_ip_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: *kubernetes_api_port*
	  • `min`: *kubernetes_api_port*
	• `description`: "Used to access the control plane endpoint from the public CIDR. The public IP CIDR is configured in the Service Enclave. If you do not know what your public IP CIDR is, ask your Service Enclave administrator."

Create the Control Plane Load Balancer Subnet

To create a subnet, use the instructions in Creating a Subnet on page 480. For Terraform input, see Example Terraform Scripts for Network Resources (Flannel Overlay) on page 839.

For this example, use the following input to create the control plane load balancer subnet. Use the OCID of the VCN that was created in Creating a VCN (Flannel Overlay) on page 848. Create the control plane load balancer subnet in the same compartment where you created the VCN.

Create either a private or a public control plane load balancer subnet. Create a public control plane load balancer subnet to use with a public cluster. Create a private control plane load balancer subnet to use with a private cluster.

See Private Clusters on page 835 for information about using Local Peering Gateways to connect a private cluster to other instances on the Private Cloud Appliance and using Dynamic Routing Gateways to connect a private cluster to the on-premises IP address space. To create a private control plane load balancer subnet, specify one of the following route tables (see Creating a VCN (Flannel Overlay) on page 848):

- vcn_private
- lpg_rt
- drg_rt

Create a Public Control Plane Load Balancer Subnet

Compute Web UI property	OCI CLI property

• Name: control-plane-endpoint
• CIDR Block: *kmilb_cidr*
• Route Table: Select "public" from the list
• Public Subnet: check the box
• DNS Hostnames:
  Use DNS Hostnames in this Subnet: check the box
  • DNS Label: kmilb
• Security Lists: Select "kmilb-seclist" and "Default Security List for oketest-vcn" from the list
	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id*
	• `--display-name`: `control-plane-endpoint`
	• `--cidr-block`: *kmilb_cidr*
	• `--dns-label`: `kmilb`
	• `--prohibit-public-ip-on-vnic`: `false`
	• `--route-table-id`: OCID of the "public" route table
	• `--security-list-ids`: OCIDs of the "kmilb-seclist" security list and the "Default Security List for oketest-vcn" security list

The difference in the following private subnet is the VCN private route table is used instead of the public route table. Depending on your needs, you could specify the LPG route table or the DRG route table instead.

Create a Private Control Plane Load Balancer Subnet

Compute Web UI property	**OCI CLI property**
• Name: control-plane-endpoint • CIDR Block: *kmilb_cidr* • Route Table: Select "vcn_private" from the list • Private Subnet: check the box • DNS Hostnames: Use DNS Hostnames in this Subnet: check the box • DNS Label: kmilb • Security Lists: Select "kmilb-seclist" and "Default Security List for oketest-vcn" from the list	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id* • `--display-name`: `control-plane-endpoint` • `--cidr-block`: *kmilb_cidr* • `--dns-label`: `kmilb` • `--prohibit-public-ip-on-vnic`: `true` • `--route-table-id`: OCID of the "vcn_private" route table • `--security-list-ids`: OCIDs of the "kmilb-seclist" security list and the "Default Security List for oketest-vcn" security list

What's Next:

Creating an OKE Cluster on page 905

Creating VCN-Native Pod Network Resources

On Private Cloud Appliance, VCN-Native Pod Networking enables you to directly manage the traffic from pods. VCN-Native Pod Networking offers more flexibility and control over the traffic and allows you to use different security rules.

With VCN-Native Pod Networking, pod IP addresses come directly from the VCN CIDR block and not from a network overlay such as Flannel Overlay. VCN-Native Pod Networking connects nodes in a Kubernetes cluster to pod subnets in the OKE VCN. Pod IP addresses within the OKE VCN are directly routable from other VCNs that are connected (peered) to the OKE VCN, and from on-premises networks.

When you create a cluster that uses VCN-Native Pod Networking, the VCN that you specify must have a subnet named `pod`. You must provide a subnet named `pod` so that the system can find that subnet. The pod subnet has security rules that enable pods on control plane nodes to communicate directly with pods on worker nodes and with other pods and other resources. See Creating a Pod Subnet (VCN-Native Pod) on page 884. If you select VCN-Native Pod Networking and do not have a subnet named `pod`, the cluster creation will fail.

When you create a node pool for a cluster that is using VCN-Native Pod Networking, the pod subnet that you specify (Pod Communication > Pod Communication Subnet or `--pod-subnet-ids`) serves the function of a pod subnet for pods on worker nodes. That pod subnet should have security rules that enable pods on worker nodes to communicate directly with other pods on worker nodes and control plane nodes. You can optionally specify the worker node subnet as the pod subnet. The CIDR of the pod subnet that you specify must be larger than /25. The pod subnet should be larger than the worker node subnet.

In general, when you use VCN-Native Pod Networking, security rules can enable pods to communicate directly with other pods on the same node or on other nodes in the cluster, with other clusters, with other services, and with the internet.

Node Shapes and Number of Pods

When using the OCI VCN-Native Pod Networking CNI plugin, each pod needs a private IP address. By default, 31 IP addresses are assigned to a VNIC for use by pods running on the worker node.

You can specify the maximum number of pods that you want to run on a worker node. The default maximum is 31 pods per worker node. You can specify up to 110.

A node shape, and therefore a worker node, has a minimum of two VNICs. The first VNIC is connected to the worker subnet. The second VNIC is connected to the pod subnet. Therefore a worker node can support at least 31 pods. If you want more than 31 pods on a single worker node, specify a shape for the node pool that supports three or more VNICs: one VNIC to connect to the worker node subnet, and at least two VNICs to connect to the pod subnet.

A VM.PCAStandard1.4 standard node shape can have a maximum of four VNICs, and the worker node can support up to 93 pods. A VM.PCAStandard.E5.Flex node shape with five OCPUs can have a maximum of five VNICs, and the worker node can support up to 110 pods. A node cannot have more than 110 pods. See OKE Service Limits on page 822.

The following formula summarizes the maximum number of pods supported per node:

```
MIN( (Number of VNICs - 1) * 31 ), 110)
```

For information about all node shapes, see Compute Shapes.

VCN-Native Pod Networking Resources

The resource definitions in the following sections in this topic create a working example set of network resources for workload clusters when you are using VCN-Native Pod Networking. Use this configuration as a guide when you create these resources. You can change the values of properties such as CIDR blocks and IP addresses. Don't change the values of properties such as the network protocol, the stateful setting, or the private/public setting.

See Workload Cluster Network Ports (VCN-Native Pod) on page 866 for specific ports that must be open for specific purposes.

Create the following network resources. To use Terraform, see Example Terraform Scripts (VCN-Native Pod) on page 867.

> **Note:**
>
> Create all network resources in the same compartment.

- VCN. See Creating a VCN (VCN-Native Pod Networking) on page 879.
- Internet Gateway
- NAT Gateway
- Dynamic Routing Gateway
- Local Peering Gateway
- Route rules
- Security lists
- The following five subnets:
  - Pod. See Creating a Pod Subnet (VCN-Native Pod) on page 884.
  - Worker. See Creating a Worker Subnet (VCN-Native Pod) on page 887.
  - Worker load balancer. See Creating a Worker Load Balancer Subnet (VCN-Native Pod) on page 893.
  - Control plane. See Creating a Control Plane Subnet (VCN-Native Pod) on page 895.
  - Control plane load balancer. See Creating a Control Plane Load Balancer Subnet (VCN-Native Pod) on page 900.

Workload Cluster Network CIDR Ranges (VCN-Native Pod)

Review the example CIDR ranges used for OKE network resources for VCN-Native Pod Networking On Private Cloud Appliance.

Throughout this documentation, variables are used to represent CIDR ranges for instances in different subnets. The following table lists the CIDR variables and example values for use with VCN-Native Pod Networking.

> **Note:**
>
> These are examples only. The CIDR ranges you use depend on the number of clusters you have, the number of nodes in each cluster, the shape you select for the worker nodes, and the type of networking you're using.

For VCN-Native Pod Networking, every pod gets an IP address assigned from the IP address pool that's defined in the pod subnet CIDR. The shape you specify for the node pool determines the maximum number of VNICs (pods) for each worker node, as described in Node Shapes and Number of Pods on page 864.

The primary difference between IP address requirements of VCN-Native Pod Networking and Flannel Overlay networking is that VCN-Native Pod Networking requires more IP addresses to be available. The table in Workload Cluster Network CIDR Ranges (Flannel Overlay) on page 837 shows smaller CIDR ranges than the following table for VCN-Native Pod Networking CIDR ranges.

Example CIDR Values to Use with VCN-Native Pod Networking

Variable Name	Description	Example Value
vcn_cidr	VCN CIDR range This Is a small VCN with 8192 IP's for creating OKE appliance.	172.31.0.0/19
worker_cidr	Worker subnet CIDR	172.31.8.0/21
workerlb_cidr	Worker load balancer subnet CIDR	172.31.0.0/23
kmi_cidr	OKE control plane subnet CIDR	172.31.4.0/22
kmilb_cidr	OKE control plane load balancer subnet CIDR	172.31.2.0/23
pod_cidr	Pod subnet CIDR	172.31.16.0/20
kube_client_cidr	CIDR for clients that are allowed to contact the Kubernetes API server	10.0.0.0/8
public_ip_cidr	Public IP CIDR configured in the Private Cloud Appliance Service Enclave	10.0.0.0/8
kube_internal_cidr	CIDR used by the Kubernetes appliance to allocate IP addresses for various internal services and components	253.255.0.0/16

The IP Subnet Calculator on Calculator.net is one tool for finding all available networks for a given IP address and prefix length.

Workload Cluster Network Ports (VCN-Native Pod)

Review the ports that are required for OKE for VCN-Native Pod Networking On Private Cloud Appliance. These ports must be available to configure workload cluster networking. You might need to open more ports for other purposes.

All protocols are TCP. All port states are Stateful. Port 6443 is the port used for Kubernetes API and is also known as ***kubernetes_api_port*** in this guide.

See also the tables in Port Matrix.

Ports that Must Be Available for Use by Workload Clusters for VCN-Native Pod Networking

Source IP Address	Destination IP Address	Port	Description
bastion host: *vcn_cidr*	Worker nodes subnet: *worker_cidr*	22	Outbound connections from the bastion host to the worker CIDR.
bastion host: *vcn_cidr*	Control plane subnet: *kmi_cidr*	22	Outbound connections from the bastion host to the control plane nodes.
Worker nodes subnet: *worker_cidr*	yum repository	80	Outbound connections from the worker CIDR to external applications.
Worker nodes subnet: *worker_cidr*	Control plane subnet: *kmi_cidr*	6443	Outbound connections from the worker CIDR to the Kubernetes API. This is necessary to allow nodes to join through either a public IP address on one of the nodes or the load balancer public IP address. Port 6443 is called the ***kubernetes_api_port***.
Worker nodes subnet: *worker_cidr*	Control plane load balancer	6443	Inbound connections from the worker CIDR to the Kubernetes API.
CIDR for clients: *kube_client_cidr*	Control plane load balancer	6443	Inbound connections from clients to the Kubernetes API server.
Worker nodes subnet: *worker_cidr*	Control plane subnet: *kmi_cidr*	6443	Private outbound connections from the worker CIDR to `kubeapi` on the control plane subnet.
kube_client_cidr	Worker nodes subnet: *worker_cidr*	30000-32767	Inbound traffic for applications from Kubernetes clients.
kmi_cidr	*worker_cidr*, *pod_cidr*	10250	Kubernetes API endpoint to worker node communication.
kmi_cidr	*worker_cidr*, *pod_cidr*	10256	Allow load balancer or network load balancer to communicate with `kube-proxy` on worker nodes or pod subnet.
pod_cidr	*kmilb_cidr*	12250	Pod to Kubernetes API endpoint communication.
kmi_cidr	*kmi_cidr*	2379-2381	Communication between the `etcd` server and metrics services.
kmi_cidr	*kmi_cidr*	10257-10260	Inbound connection for Kubernetes components.

Example Terraform Scripts (VCN-Native Pod)

On Private Cloud Appliance, you can use Terraform scripts to automate the creation of OKE network resources for VCN-Native Pod Networking.

Most of the values shown in these scripts, such as resource display names and CIDRs, are examples. Some ports must be specified as shown (see Workload Cluster Network Ports (VCN-Native Pod) on page 866), the OKE pod subnet must be named `pod`, and the OKE control plane subnet must be named `control-plane`. See Workload Cluster Network CIDR Ranges (VCN-Native Pod) on page 865 for comments about CIDR values.

- variables.tf on page 868

- terraform.tfvars on page 869
- provider.tf on page 870
- main.tf on page 870
- oke_vcn.tf on page 873
- oke_pod_seclist.tf on page 874
- oke_pod_subnet.tf on page 875
- oke_worker_seclist.tf on page 875
- oke_worker_subnet.tf on page 876
- oke_kmi_seclist.tf on page 877
- oke_kmi_subnet.tf on page 878

variables.tf

This file creates several variables that are used to configure OKE network resources when you're using VCN-Native Pod Networking. Many of these variables aren't assigned values in this file. One port and five CIDRs are assigned values. The `kubernetes_api_port`, port 6443, is the port used to access the Kubernetes API. See also Workload Cluster Network Ports (VCN-Native Pod) on page 866. The six CIDRs that are defined in this file are for the OKE VCN, pod subnet, worker subnet, worker load balancer subnet, control plane subnet, and control plane load balancer subnet.

```
variable "oci_config_file_profile" {
  type    = string
  default = "DEFAULT"
}

variable "tenancy_ocid" {
  description = "tenancy OCID"
  type        = string
  nullable    = false
}

variable "compartment_id" {
  description = "compartment OCID"
  type        = string
  nullable    = false
}

variable "vcn_name" {
  description = "VCN name"
  nullable    = false
}

variable "kube_client_cidr" {
  description = "CIDR of Kubernetes API clients"
  type        = string
  nullable    = false
}

variable "kubernetes_api_port" {
  description = "Port used for Kubernetes API"
  type        = string
  default     = "6443"
}

# IP network addressing
variable "vcn_cidr" {
  default = "172.31.0.0/19"
}

# Subnet for KMIs where kube-apiserver and other control
```

```
# plane applications run, max 9 nodes
variable "kmi_cidr" {
  description = "Kubernetes control plane subnet CIDR"
  default     = "172.31.4.0/22"
}

# Subnet for KMI load balancer
variable "kmilb_cidr" {
  description = "Kubernetes control plane LB subnet CIDR"
  default     = "172.31.2.0/23"
}

# Subnet CIDR configured for VCN public IP for NAT in Network
variable "public_ip_cidr" {
  description = "Public IP CIDR configured in the Service Enclave"
  type        = string
  nullable    = false
}

# Subnet for worker nodes, max 128 nodes
variable "worker_cidr" {
  description = "Kubernetes worker subnet CIDR"
  default     = "172.31.8.0/21"
}

# Subnet for worker load balancer (for use by CCM)
variable "workerlb_cidr" {
  description = "Kubernetes worker LB subnet CIDR"
  default     = "172.31.0.0/23"
}

# Subnet for pod communication
variable "pod_cidr" {
  description = "Kubernetes pod communication subnet CIDR"
  default     = "172.31.16.0/20"
}

# Flag to Enable private endpoint
variable "enable_private_endpoint" {
  description = "Flag to create private control plane endpoint/service-lb"
  type = bool
  default = false
  nullable = false
}
```

terraform.tfvars

This file assigns values to some of the variables that were created in `variables.tf`.

```
# name of the profile to use from $HOME/.oci/config
oci_config_file_profile = "DEFAULT"

# tenancy ocid from the above profile
tenancy_ocid = "tenancy_OCID"

# compartment in which to build the OKE cluster
compartment_id = "compartment_OCID"

# display-name for the OKE VCN
vcn_name = "oketest"
```

provider.tf

This file is required in order to use the OCI provider. The file initializes the OCI module using the OCI profile configuration file.

```
provider "oci" {
  config_file_profile = var.oci_config_file_profile
  tenancy_ocid        = var.tenancy_ocid
}
```

main.tf

This file specifies the provider to use (`oracle/oci`), defines several security list rules, and initializes required local variables.

The version of the OCI provider that you use must be at least v4.50.0 but no greater than v6.36.0.

```
terraform {
  required_providers {
    oci = {
      source  = "oracle/oci"
      version = ">= 4.50.0, <= 6.36.0"
      # If necessary, you can pin a specific version here
      # version = "4.71.0"
    }
  }
  required_version = ">= 1.1"
}

locals {
  kube_internal_cidr      = "253.255.0.0/16"
  worker_lb_ingress_rules = [
    {
      source   = var.kube_client_cidr
      port_min = 80
      port_max = 80
    },
    {
      source   = var.kube_client_cidr
      port_min = 443
      port_max = 443
    }
  ]
  worker_ingress_rules = [
    {
      source   = var.kube_client_cidr
      port_min = 30000
      port_max = 32767
    },
    {
      source   = var.kmi_cidr
      port_min = 22
      port_max = 22
    },
    {
      source   = var.worker_cidr
      port_min = 22
      port_max = 22
    },
    {
      source   = var.worker_cidr
      port_min = 10250
      port_max = 10250
    },
```

```
    {
      source   = var.worker_cidr
      port_min = 10256
      port_max = 10256
    },
    {
      source   = var.worker_cidr
      port_min = 30000
      port_max = 32767
    },
    {
      source   = var.workerlb_cidr
      port_min = 10256
      port_max = 10256
    },
    {
      source   = var.workerlb_cidr
      port_min = 30000
      port_max = 32767
    },
    {
      source   = var.kmi_cidr
      port_min = 10250
      port_max = 10250
    },
    {
      source   = var.kmi_cidr
      port_min = 10256
      port_max = 10256
    },
    {
      source   = var.pod_cidr
      port_min = 30000
      port_max = 32767
    },
  ]
  kmi_lb_ingress_rules = [
    {
      source   = local.kube_internal_cidr
      port_min = var.kubernetes_api_port
      port_max = var.kubernetes_api_port
    },
    {
      source   = var.kube_client_cidr
      port_min = var.kubernetes_api_port
      port_max = var.kubernetes_api_port
    },
    {
      source   = var.kmi_cidr
      port_min = var.kubernetes_api_port
      port_max = var.kubernetes_api_port
    },
    {
      source   = var.worker_cidr
      port_min = var.kubernetes_api_port
      port_max = var.kubernetes_api_port
    },
    {
      source   = var.worker_cidr
      port_min = 12250
      port_max = 12250
    },
    {
      source   = var.pod_cidr
```

```
      port_min = 12250
      port_max = 12250
    },
  ]
  kmi_ingress_rules = [
    {
      source   = var.kube_client_cidr
      port_min = var.kubernetes_api_port
      port_max = var.kubernetes_api_port
    },
    {
      source   = var.kmilb_cidr
      port_min = var.kubernetes_api_port
      port_max = var.kubernetes_api_port
    },
    {
      source   = var.kmilb_cidr
      port_min = 12250
      port_max = 12250
    },
    {
      source   = var.worker_cidr
      port_min = var.kubernetes_api_port
      port_max = var.kubernetes_api_port
    },
    {
      source   = var.worker_cidr
      port_min = 12250
      port_max = 12250
    },
    {
      source   = var.kmi_cidr
      port_min = var.kubernetes_api_port
      port_max = var.kubernetes_api_port
    },
    {
      source   = var.kmi_cidr
      port_min = 2379
      port_max = 2381
    },
    {
      source   = var.kmi_cidr
      port_min = 10250
      port_max = 10250
    },
    {
      source   = var.kmi_cidr
      port_min = 10257
      port_max = 10260
    },
    {
      source   = var.pod_cidr
      port_min = var.kubernetes_api_port
      port_max = var.kubernetes_api_port
    },
    {
      source   = var.pod_cidr
      port_min = 12250
      port_max = 12250
    },
  ]
  pod_ingress_rules = [
    {
      source   = var.vcn_cidr
```

```
      port_min = 22
      port_max = 22
    },
    {
      source   = var.workerlb_cidr
      port_min = 10256
      port_max = 10256
    },
    {
      source   = var.worker_cidr
      port_min = 10250
      port_max = 10250
    },
    {
      source   = var.worker_cidr
      port_min = 10256
      port_max = 10256
    },
    {
      source   = var.worker_cidr
      port_min = 80
      port_max = 80
    },
  ]
}
```

oke_vcn.tf

This file defines a VCN, NAT gateway, internet gateway, private route table, and public route table. The private route table is the default route table for the VCN.

```
resource "oci_core_vcn" "oke_vcn" {
  cidr_block     = var.vcn_cidr
  dns_label      = var.vcn_name
  compartment_id = var.compartment_id
  display_name   = "${var.vcn_name}-vcn"
}

resource "oci_core_nat_gateway" "vcn_ngs" {
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id
  count          = var.enable_private_endpoint ? 0:1

  display_name = "VCN nat g6s"
}

resource "oci_core_internet_gateway" "vcn_igs" {
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id
  count          = var.enable_private_endpoint ? 0:1

  display_name = "VCN i6t g6s"
  enabled      = true
}

resource "oci_core_default_route_table" "default_private" {
  manage_default_resource_id = oci_core_vcn.oke_vcn.default_route_table_id
  display_name   = "Default - private"
  count          = var.enable_private_endpoint ? 1:0
}

resource "oci_core_default_route_table" "private" {
```

```
  count          = var.enable_private_endpoint ? 0:1
  manage_default_resource_id = oci_core_vcn.oke_vcn.default_route_table_id
  display_name   = "Default - private"

  route_rules {
    destination       = "0.0.0.0/0"
    destination_type  = "CIDR_BLOCK"
    network_entity_id = oci_core_nat_gateway.vcn_ngs[0].id
  }
}

resource "oci_core_route_table" "public" {
  count          = var.enable_private_endpoint ? 0:1
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id

  display_name = "public"
  route_rules {
    destination       = "0.0.0.0/0"
    destination_type  = "CIDR_BLOCK"
    network_entity_id = oci_core_internet_gateway.vcn_igs[0].id
  }
}
```

oke_pod_seclist.tf

This file defines the security list for the pod subnet. The rules for this security list were defined in other Terraform files in this set.

```
resource "oci_core_security_list" "pod" {
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id

  display_name = "${var.vcn_name}-pod"

  dynamic "ingress_security_rules" {
    iterator = port
    for_each = local.pod_ingress_rules

    content {
      source      = port.value.source
      source_type = "CIDR_BLOCK"
      protocol    = "6"
      tcp_options {
        min = port.value.port_min
        max = port.value.port_max
      }
    }
  }

   dynamic "ingress_security_rules" {
    iterator = icmp_type
    for_each = [0, 8]

    content {
      # ping from VCN; unreachable/TTL from anywhere
      source      = var.kmi_cidr
      source_type = "CIDR_BLOCK"
      protocol    = "1"
      icmp_options {
        type = icmp_type.value
      }
    }
```

```
  }

  dynamic "ingress_security_rules" {
    for_each = var.pod_cidr != null ? [var.pod_cidr] : []

    content {
      source      = ingress_security_rules.value
      source_type = "CIDR_BLOCK"
      protocol    = "all"
    }
  }
}
```

oke_pod_subnet.tf

This file defines the pod subnet.

> **Important:**
>
> The name of the `pod` subnet must be exactly `pod`.

```
resource "oci_core_subnet" "pod" {
  cidr_block     = var.pod_cidr
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id

  display_name               = "pod"
  dns_label                  = "pod"
  prohibit_public_ip_on_vnic = true

  security_list_ids = [
    oci_core_default_security_list.oke_vcn.id,
    oci_core_security_list.pod.id
  ]
}
```

oke_worker_seclist.tf

This file defines the security lists for both the worker subnet and the worker load balancer subnet. The rules for these security lists were defined in other Terraform files in this set.

```
resource "oci_core_security_list" "workerlb" {
  display_name   = "${var.vcn_name}-workerlb"
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id

  dynamic "ingress_security_rules" {
    iterator = port
    for_each = local.worker_lb_ingress_rules

    content {
      source      = port.value.source
      source_type = "CIDR_BLOCK"
      protocol    = "6"
      tcp_options {
        min = port.value.port_min
        max = port.value.port_max
      }
    }
  }
}

resource "oci_core_security_list" "worker" {
```

```
  display_name   = "${var.vcn_name}-worker"
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id

  dynamic "ingress_security_rules" {
    iterator = port
    for_each = local.worker_ingress_rules

    content {
      source      = port.value.source
      source_type = "CIDR_BLOCK"
      protocol    = "6"
      tcp_options {
        min = port.value.port_min
        max = port.value.port_max
      }
    }
  }

  dynamic "ingress_security_rules" {
    iterator = icmp_type
    for_each = [0, 8]

    content {
      # ping from VCN; unreachable/TTL from anywhere
      source      = var.kmi_cidr
      source_type = "CIDR_BLOCK"
      protocol    = "1"
      icmp_options {
        type = icmp_type.value
      }
    }
  }
}
```

oke_worker_subnet.tf

This file defines the worker and worker load balancer subnets. The worker load balancer subnet is named `service-lb`.

```
resource "oci_core_subnet" "worker" {
  cidr_block     = var.worker_cidr
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id

  display_name               = "worker"
  dns_label                  = "worker"
  prohibit_public_ip_on_vnic = true

  security_list_ids = [
    oci_core_default_security_list.oke_vcn.id,
    oci_core_security_list.worker.id
  ]
}

resource "oci_core_subnet" "worker_lb" {
  cidr_block     = var.workerlb_cidr
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id

  display_name               = "service-lb"
  dns_label                  = "servicelb"
```

```
  prohibit_public_ip_on_vnic = var.enable_private_endpoint
  route_table_id             = var.enable_private_endpoint==false ?
 oci_core_route_table.public[0].id :
 oci_core_vcn.oke_vcn.default_route_table_id

  security_list_ids = [
    oci_core_default_security_list.oke_vcn.id,
    oci_core_security_list.workerlb.id
  ]
}
```

oke_kmi_seclist.tf

This file defines the security lists for the control plane and control plane load balancer subnets. This file also defines updates to make to the default security list for the VCN.

```
resource "oci_core_default_security_list" "oke_vcn" {
  manage_default_resource_id = oci_core_vcn.oke_vcn.default_security_list_id

  egress_security_rules {
    destination      = "0.0.0.0/0"
    destination_type = "CIDR_BLOCK"
    protocol         = "all"
  }

  dynamic "ingress_security_rules" {
    iterator = icmp_type
    for_each = [3, 8, 11]

    content {
      # ping from VCN; unreachable/TTL from anywhere
      source      = (icmp_type.value == "8" ? var.vcn_cidr : "0.0.0.0/0")
      source_type = "CIDR_BLOCK"
      protocol    = "1"
      icmp_options {
        type = icmp_type.value
      }
    }
  }
}

resource "oci_core_security_list" "kmilb" {
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id

  display_name = "${var.vcn_name}-kmilb"

  dynamic "ingress_security_rules" {
    iterator = port
    for_each = local.kmi_lb_ingress_rules

    content {
      source      = port.value.source
      source_type = "CIDR_BLOCK"
      protocol    = "6"
      tcp_options {
        min = port.value.port_min
        max = port.value.port_max
      }
    }
  }
```

```
  dynamic "ingress_security_rules" {
    for_each = var.enable_private_endpoint ? [1] : []
    content {
      source   = var.kmilb_cidr
      source_type = "CIDR_BLOCK"
      protocol = "6"
      tcp_options {
        min = var.kubernetes_api_port
        max = var.kubernetes_api_port
      }
    }
  }

  dynamic "ingress_security_rules" {
    for_each = var.enable_private_endpoint ? [] : [0]
    content {
      source   = var.public_ip_cidr
      source_type = "CIDR_BLOCK"
      protocol = "6"
      tcp_options {
        min = var.kubernetes_api_port
        max = var.kubernetes_api_port
      }
    }
  }

  depends_on = []
}

resource "oci_core_security_list" "kmi" {
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id

  display_name = "${var.vcn_name}-kmi"

  dynamic "ingress_security_rules" {
    iterator = port
    for_each = local.kmi_ingress_rules

    content {
      source      = port.value.source
      source_type = "CIDR_BLOCK"
      protocol    = "6"
      tcp_options {
        min = port.value.port_min
        max = port.value.port_max
      }
    }
  }
}
```

oke_kmi_subnet.tf

This file defines the control plane and control plane load balancer subnets.

> **Important:**
>
> The name of the `kmi` subnet must be exactly `control-plane`.

```
resource "oci_core_subnet" "kmi" {
  cidr_block                 = var.kmi_cidr
  compartment_id             = var.compartment_id
  display_name               = "control-plane"
```

```
  dns_label                  = "kmi"
  vcn_id                     = oci_core_vcn.oke_vcn.id
  prohibit_public_ip_on_vnic = true
  security_list_ids = [
    oci_core_default_security_list.oke_vcn.id,
    oci_core_security_list.kmi.id
  ]
}

resource "oci_core_subnet" "kmi_lb" {
  cidr_block                 = var.kmilb_cidr
  compartment_id             = var.compartment_id
  dns_label                  = "kmilb"
  vcn_id                     = oci_core_vcn.oke_vcn.id
  display_name               = "control-plane-endpoint"
  prohibit_public_ip_on_vnic = var.enable_private_endpoint
  route_table_id             = var.enable_private_endpoint==false ?
 oci_core_route_table.public[0].id :
 oci_core_default_route_table.default_private[0].id
  security_list_ids = [
    oci_core_default_security_list.oke_vcn.id,
    oci_core_security_list.kmilb.id
  ]
}
```

Creating a VCN (VCN-Native Pod Networking)

Learn how to create a VCN for VCN-Native Pod Networking On Private Cloud Appliance.

Create the following resources in the order listed:

1. VCN
2. Route rules
   - Public clusters:
     - Internet gateway and a route table with a route rule that references that internet gateway.
     - NAT gateway and a route table with a route rule that references that NAT gateway.
   - Private clusters:
     - Route table with no route rules.
     - (Optional) Dynamic Routing Gateway (DRG), attach the OKE VCN to that DRG, and create a route table with a route rule that references that DRG. See Private Clusters on page 835.
     - (Optional) Local Peering Gateway (LPG) and a route table with a route rule that references that LPG. See Private Clusters on page 835.
3. Security list. Modify the VCN default security list.

Resource names and CIDR blocks are example values.

VCN

To create the VCN, use the instructions inCreating a VCN on page 479. For Terraform input, see Example Terraform Scripts (VCN-Native Pod) on page 867.

For this example, use the following input to create the VCN. The VCN covers one contiguous CIDR block. The CIDR block can't be changed after the VCN is created.

Compute Web UI property	OCI CLI property
• Name: oketest-vcn • CIDR Block: *vcn_cidr* • DNS Label: oketest This label must be unique across all VCNs in the tenancy.	• `--display-name`: `oketest-vcn` • `--cidr-blocks`: `'["vcn_cidr"]'` • `--dns-label`: `oketest` This label must be unique across all VCNs in the tenancy.

Note the OCID of the new VCN. In the examples in this guide, this VCN OCID is `ocid1.vcn.oke_vcn_id`.

Next Steps

- Public internet access. For traffic on a public subnet that connects to the internet using public IP addresses, create an internet gateway and a route rule that references that internet gateway.
- Private internet access. For traffic on a private subnet that needs to connect to the internet without exposing private IP addresses, create a NAT gateway and a route rule that references that NAT gateway.
- VCN-only access. To restrict communication to only other resources on the same VCN, use the default route table, which has no route rules.
- Instances in another VCN. To enable communication between the cluster and an instance running on a different VCN, create a Local Peering Gateway (LPG) and a route rule that references that LPG.
- Data center IP address space. To enable communication between the cluster and the on-premises network IP address space, create a Dynamic Routing Gateway (DRG) and a route rule that references that DRG.

VCN Private Route Table

Edit the default route table that was created when you created the VCN. Change the name of the route table to vcn_private. This route table does not have any route rules. Do not add any route rules.

NAT Private Route Table

Create a NAT gateway and a route table with a route rule that references the NAT gateway.

NAT Gateway

To create the NAT gateway, use the instructions in Enabling Public Connections through a NAT Gateway on page 506. For Terraform input, see Example Terraform Scripts (VCN-Native Pod) on page 867.

Note the name and OCID of the NAT gateway for assignment to the private route rule.

Private Route Rule

Create a route table. See Creating a Route Table on page 493. For Terraform input, see Example Terraform Scripts for Network Resources (Flannel Overlay) on page 839.

For this example, use the following input to create the route table with a private route rule that references the NAT gateway that was created in the preceding step.

Compute Web UI property	OCI CLI property
• Name: nat_private	• `--display-name`: nat_private
Route rule	`--route-rules`
• Target Type: NAT Gateway • NAT Gateway: Name of the NAT gateway that was created in the preceding step • CIDR Block: 0.0.0.0/0 • Description: NAT private route rule	• `networkEntityId`: OCID of the NAT gateway that was created in the preceding step • `destinationType`: `CIDR_BLOCK` • `destination`: `0.0.0.0/0` • `description`: NAT private route rule

Note the name and OCID of this route table for assignment to private subnets.

Local Peering Gateway

Create a Local Peering gateway (LPG) and a route table with a route rule that references the LPG.

Local Peering Gateway

Create the LPG. See Connecting VCNs through a Local Peering Gateway (LPG) on page 509.

Note the name and OCID of the LPG for assignment to the private route rule.

Private Route Rule

Create a route table. See Creating a Route Table on page 493.

For this example, use the following input to create the route table with a private route rule that references the LPG that was created in the preceding step.

Compute Web UI property	OCI CLI property
• Name: lpg_rt	• `--display-name`: lpg_rt
Route rule	`--route-rules`
• Target Type: Local Peering Gateway • Local Peering Gateway: Name of the LPG that was created in the preceding step • CIDR Block: *`CIDR_for_the_second_VCN`* • Description: LPG private route rule	• `networkEntityId`: OCID of the LPG that was created in the preceding step • `destinationType`: `CIDR_BLOCK` • `destination`: *`CIDR_for_the_second_VCN`* • `description`: LPG private route rule

Note the name and OCID of this route table for assignment to the "control-plane-endpoint" subnet (Creating a Control Plane Load Balancer Subnet (VCN-Native Pod) on page 900).

Add the same route rule on the second VCN (the peered VCN), specifying the OKE VCN CIDR as the destination.

Dynamic Routing Gateway

Create a Dynamic Routing gateway (DRG) and a route table with a route rule that references the DRG.

Dynamic Routing Gateway

Create the DRG and attach the OKE VCN to that DRG. See Connecting to the On-Premises Network through a Dynamic Routing Gateway (DRG) on page 511. Create the DRG in the OKE VCN compartment, and then attach the OKE VCN to that DRG.

Note the name and OCID of the DRG for assignment to the private route rule.

Private Route Rule

Create a route table. See Creating a Route Table on page 493.

For this example, use the following input to create the route table with a private route rule that references the DRG that was created in the preceding step.

Compute Web UI property	OCI CLI property
• Name: drg_rt Route rule • Target Type: Dynamic Routing Gateway • Dynamic Routing: Name of the DRG that was created in the preceding step • CIDR Block: 0.0.0.0/0 • Description: DRG private route rule	• `--display-name`: drg_rt `--route-rules` • `networkEntityId`: OCID of the DRG that was created in the preceding step • `destinationType`: `CIDR_BLOCK` • `destination`: `0.0.0.0/0` • `description`: DRG private route rule

Note the name and OCID of this route table for assignment to the "control-plane-endpoint" subnet (Creating a Control Plane Load Balancer Subnet (VCN-Native Pod) on page 900).

Public Route Table

Create an Internet gateway and a route table with a route rule that references the Internet gateway.

Internet Gateway

To create the internet gateway, use the instructions in Configuring an Internet Gateway on page 507. For Terraform input, see Example Terraform Scripts (VCN-Native Pod) on page 867.

Note the name and OCID of the internet gateway for assignment to the public route rule.

Public Route Rule

To create a route table, use the instructions in Creating a Route Table on page 493. For Terraform input, see Example Terraform Scripts (VCN-Native Pod) on page 867.

For this example, use the following input to create the route table with a public route rule that references the internet gateway that was created in the preceding step.

Compute Web UI property	OCI CLI property
• Name: public Route rule • Target Type: Internet Gateway • Internet Gateway: Name of the internet gateway that was created in the preceding step • CIDR Block: 0.0.0.0/0 • Description: OKE public route rule	• `--vcn-id`: `ocid1.vcn.`***`oke_vcn_id`*** • `--display-name`: public `--route-rules` • `networkEntityId`: OCID of the internet gateway that was created in the preceding step • `destinationType`: `CIDR_BLOCK` • `destination`: `0.0.0.0/0` • `description`: OKE public route rule

Note the name and OCID of this route table for assignment to public subnets.

VCN Default Security List

Modify the default security list, using the input shown in the following table. Delete all of the default rules and create the rules shown in the following table.

To modify a security list, use the instructions in Updating a Security List on page 499. For Terraform input, see Example Terraform Scripts (VCN-Native Pod) on page 867.

Compute Web UI property	OCI CLI property
• Name: Default	`--security-list-id`: `ocid1.securitylist.`***`default_securitylist_id`***
One egress security rule: • Stateless: clear the box • Egress CIDR: 0.0.0.0/0 • IP Protocol: All protocols • Description: "Allow all outgoing traffic."	**One egress security rule:** `--egress-security-rules` • `isStateless`: `false` • `destination`: `0.0.0.0/0` • `destinationType`: `CIDR_BLOCK` • `protocol`: `all` • `description`: "Allow all outgoing traffic."
Three ingress security rules:	**Three ingress security rules:** `--ingress-security-rules`

Ingress Rule 1
• Stateless: clear the box
• Ingress CIDR: *`vcn_cidr`*
• IP Protocol: ICMP
  • Parameter Type: 8: Echo
• Description: "Allow ping from VCN."
	Ingress Rule 1
	• `isStateless`: `false`
	• `source`: ***`vcn_cidr`***
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `1`
	• `icmpOptions`
	  • `type`: `8`
	• `description`: "Allow ping from VCN."

Ingress Rule 2
• Stateless: clear the box
• Ingress CIDR: 0.0.0.0/0
• IP Protocol: ICMP
  • Parameter Type: 3: Destination Unreachable
• Description: "Blocks incoming requests from any source."
	Ingress Rule 2
	• `isStateless`: `false`
	• `source`: `0.0.0.0/0`
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `1`
	• `icmpOptions`
	  • `type`: `3`
	• `description`: "Blocks incoming requests from any source."

Ingress Rule 3
• Stateless: clear the box
• Ingress CIDR: 0.0.0.0/0
• IP Protocol: ICMP
  • Parameter Type: 11: Time Exceeded
• Description: "Time exceeded."
	Ingress Rule 3
	• `isStateless`: `false`
	• `source`: `0.0.0.0/0`
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `1`
	• `icmpOptions`
	  • `type`: `11`
	• `description`: "Time exceeded."

Note the name and OCID of this default security list for assignment to subnets.

Creating a Pod Subnet (VCN-Native Pod)

Learn how to create a pod subnet for VCN-Native Pod Networking On Private Cloud Appliance.

The instructions in this topic create a pod subnet named "pod" in the VCN that provides the private IP addresses for pods running on the control plane nodes. The number of IP addresses in this subnet should be equal to or greater than the number of IP addresses in the control plane subnet. The pod subnet must be a private subnet.

The pod subnet supports communication between pods and direct access to individual pods using private pod IP addresses. The pod subnet must be private. The pod subnet enables pods to communicate with other pods on the same worker node, with pods on other worker nodes, with OCI services (through a service gateway) and with the internet (through a NAT gateway).

Create the following resources in the order listed:

1. Pod security list
2. Pod subnet

Create a Pod Security List

Create a security list. See Creating a Security List on page 496. For Terraform input, see Example Terraform Scripts (VCN-Native Pod) on page 867.

The security rules shown in the following table define traffic that is allowed to contact pods directly. Use these security rules as part of network security groups (NSGs) or in security lists. We recommend using NSGs.

The security rules apply to all pods in all the worker nodes connected to the pod subnet specified for a node pool.

Route incoming requests to pods based on routing policies specified by routing rules and route tables. See the route tables defined in Creating a VCN (VCN-Native Pod Networking) on page 879.

For this example, use the following input for the pod subnet security list.

Compute Web UI property	**OCI CLI property**
• Name: pod-seclist	• `--vcn-id`: `ocid1.vcn.`*`oke_vcn_id`* • `--display-name`: `pod-seclist`
One egress security rule: • Stateless: clear the box • Egress CIDR: 0.0.0.0/0 • IP Protocol: All protocols • Description: "Allow all outgoing traffic."	**One egress security rule:** `--egress-security-rules` • `isStateless`: `false` • `destination`: `0.0.0.0/0` • `destinationType`: `CIDR_BLOCK` • `protocol`: `all` • `description`: "Allow all outgoing traffic."
Eight ingress security rules:	**Eight ingress security rules:** `--ingress-security-rules`

Compute Web UI property	OCI CLI property

Ingress Rule 1
• Stateless: clear the box
• Ingress CIDR: *vcn_cidr*
• IP Protocol: TCP
  • Destination Port Range: 22
• Description: "Allow SSH connection to the pod subnet from all subnets in the VCN."
	Ingress Rule 1
	• `isStateless`: `false`
	• `source`: *vcn_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `22`
	  • `min`: `22`
	• `description`: "Allow SSH connection to the pod subnet from all subnets in the VCN."

Ingress Rule 2
• Stateless: clear the box
• Ingress CIDR: *workerlb_cidr*
• IP Protocol: TCP
  • Destination Port Range: 10256
• Description: "Allow the worker load balancer to contact the pods."
	Ingress Rule 2
	• `isStateless`: `false`
	• `source`: *workerlb_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `10256`
	  • `min`: `10256`
	• `description`: "Allow the worker load balancer to contact the pods."

Ingress Rule 3
• Stateless: clear the box
• Ingress CIDR: *worker_cidr*
• IP Protocol: TCP
  • Destination Port Range: 10250
• Description: "Allow Kubernetes API endpoint to pod (via worker node) communication."
	Ingress Rule 3
	• `isStateless`: `false`
	• `source`: *worker_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `10250`
	  • `min`: `10250`
	• `description`: "Allow Kubernetes API endpoint to pod (via worker node) communication."

Compute Web UI property	OCI CLI property

Ingress Rule 4
• Stateless: clear the box
• Ingress CIDR: *worker_cidr*
• IP Protocol: TCP
  • Destination Port Range: 10256
• Description: "Allow Load Balancer or Network Load Balancer to communicate with the `kube-proxy` pod (via the worker subnet)."
	Ingress Rule 4
	• `isStateless`: `false`
	• `source`: *worker_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `10256`
	  • `min`: `10256`
	• `description`: "Allow Load Balancer or Network Load Balancer to communicate with the `kube-proxy` pod (via the worker subnet)."

Ingress Rule 5
• Stateless: clear the box
• Ingress CIDR: *worker_cidr*
• IP Protocol: TCP
  • Destination Port Range: 80
• Description: "Allow the worker node to contact the pods."
This ingress is optional. This port is open for an end user application. This rule could be different based on what applications are deployed.
	Ingress Rule 5
	• `isStateless`: `false`
	• `source`: *worker_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `80`
	  • `min`: `80`
	• `description`: "Allow the worker node to contact the pods."
	This ingress is optional. This port is open for an end user application. This rule could be different based on what applications are deployed.

Ingress Rule 6
• Stateless: clear the box
• Ingress CIDR: *kmi_cidr*
• IP Protocol: ICMP
  • Parameter Type: 8: Echo
• Description: "Test the reachability of a network pod from *kmi_cidr* by sending a request."
	Ingress Rule 6
	• `isStateless`: `false`
	• `source`: *kmi_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `1`
	• `icmpOptions`
	  • `type`: `8`
	• `description`: "Test the reachability of a network pod from *kmi_cidr* by sending a request."

Compute Web UI property	OCI CLI property

Ingress Rule 7
• Stateless: clear the box
• Ingress CIDR: *kmi_cidr*
• IP Protocol: ICMP
  • Parameter Type: 0: Echo Reply
• Description: "If the destination pod is reachable from *kmi_cidr*, respond with an ICMP Echo Reply."
	Ingress Rule 7
	• `isStateless`: `false`
	• `source`: *kmi_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `1`
	• `icmpOptions`
	  • `type`: `0`
	• `description`: "If the destination pod is reachable from *kmi_cidr*, respond with an ICMP Echo Reply."

Ingress Rule 8 • Stateless: clear the box • Ingress CIDR: *pod_cidr* • IP Protocol: All protocols • Description: "Allow the pod CIDR to communicate with itself."	**Ingress Rule 8** • `isStateless`: `false` • `source`: *pod_cidr* • `sourceType`: `CIDR_BLOCK` • `protocol`: `all` • `description`: "Allow the pod CIDR to communicate with itself."

Create the Pod Subnet

Create a subnet. See Creating a Subnet on page 480. For Terraform input, see Example Terraform Scripts (VCN-Native Pod) on page 867.

For this example, use the following input to create the pod subnet. Use the OCID of the VCN that was created in Creating a VCN (VCN-Native Pod Networking) on page 879. Create the pod subnet in the same compartment where you created the VCN.

> **Important:**
>
> The name of this subnet must be exactly "pod".

Compute Web UI property	OCI CLI property

• Name: pod
• CIDR Block: *pod_cidr*
• Route Table: Select "nat_private" from the list
• Private Subnet: check the box
• DNS Hostnames:
  Use DNS Hostnames in this Subnet: check the box
  • DNS Label: pod
• Security Lists: Select "pod-seclist" and "Default Security List for oketest-vcn" from the list
	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id*
	• `--display-name`: `pod`
	• `--cidr-block`: *pod_cidr*
	• `--dns-label`: `pod`
	• `--prohibit-public-ip-on-vnic`: `true`
	• `--route-table-id`: OCID of the "nat_private" route table
	• `--security-list-ids`: OCIDs of the "pod-seclist" security list and the "Default Security List for oketest-vcn" security list

Creating a Worker Subnet (VCN-Native Pod)

Learn how to create a worker subnet for VCN-Native Pod Networking On Private Cloud Appliance.

Create the following resources in the order listed:

1. Worker security list

2. Worker subnet

Create a Worker Security List

Create a security list. See Creating a Security List on page 496. For Terraform input, see Example Terraform Scripts (VCN-Native Pod) on page 867.

This security list defines traffic that is allowed to contact worker nodes directly.

For this example, use the following input for the worker subnet security list.

Compute Web UI property	**OCI CLI property**
• Name: worker-seclist	• `--vcn-id`: `ocid1.vcn.`*`oke_vcn_id`* • `--display-name`: `worker-seclist`
One egress security rule: • Stateless: clear the box • Egress CIDR: 0.0.0.0/0 • IP Protocol: All protocols • Description: "Allow all outgoing traffic."	**One egress security rule:** `--egress-security-rules` • `isStateless`: `false` • `destination`: `0.0.0.0/0` • `destinationType`: `CIDR_BLOCK` • `protocol`: `all` • `description`: "Allow all outgoing traffic."
Thirteen ingress security rules:	**Thirteen ingress security rules:** `--ingress-security-rules`

Ingress Rule 1
• Stateless: clear the box
• Ingress CIDR: *`kube_client_cidr`*
• IP Protocol: TCP
  • Destination Port Range: 30000-32767
• Description: "Allow worker nodes to receive connections through the pod subnet."
	Ingress Rule 1
	• `isStateless`: `false`
	• `source`: *`kube_client_cidr`*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `32767`
	  • `min`: `30000`
	• `description`: "Allow worker nodes to receive connections through the pod subnet."

Ingress Rule 2
• Stateless: clear the box
• Ingress CIDR: *`kmi_cidr`*
• IP Protocol: TCP
  • Destination Port Range: 22
• Description: "Allow SSH connection from the control plane subnet."
	Ingress Rule 2
	• `isStateless`: `false`
	• `source`: *`kmi_cidr`*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `22`
	  • `min`: `22`
	• `description`: "Allow SSH connection from the control plane subnet."

Compute Web UI property	OCI CLI property

Ingress Rule 3
• Stateless: clear the box
• Ingress CIDR: *worker_cidr*
• IP Protocol: TCP
  • Destination Port Range: 22
• Description: "Allow SSH connection from the worker subnet."
	Ingress Rule 3
	• `isStateless`: `false`
	• `source`: *worker_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `22`
	  • `min`: `22`
	• `description`: "Allow SSH connection from the worker subnet."

Ingress Rule 4
• Stateless: clear the box
• Ingress CIDR: *worker_cidr*
• IP Protocol: TCP
  • Destination Port Range: 10250
• Description: "Allow Kubernetes API endpoint to worker node communication."
	Ingress Rule 4
	• `isStateless`: `false`
	• `source`: *worker_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `10250`
	  • `min`: `10250`
	• `description`: "Allow Kubernetes API endpoint to worker node communication."

Ingress Rule 5
• Stateless: clear the box
• Ingress CIDR: *worker_cidr*
• IP Protocol: TCP
  • Destination Port Range: 10256
• Description: "Allow Load Balancer or Network Load Balancer to communicate with `kube-proxy` on worker nodes."
	Ingress Rule 5
	• `isStateless`: `false`
	• `source`: *worker_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `10256`
	  • `min`: `10256`
	• `description`: "Allow Load Balancer or Network Load Balancer to communicate with `kube-proxy` on worker nodes."

Compute Web UI property	OCI CLI property

Ingress Rule 6
• Stateless: clear the box
• Ingress CIDR: *worker_cidr*
• IP Protocol: TCP
  • Destination Port Range: 30000-32767
• Description: "Allow traffic to worker nodes."
	Ingress Rule 6
	• `isStateless`: `false`
	• `source`: *worker_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	  `destinationPortRange`
	  • `max`: `32767`
	  • `min`: `30000`
	• `description`: "Allow traffic to worker nodes."

Ingress Rule 7
• Stateless: clear the box
• Ingress CIDR: *workerlb_cidr*
• IP Protocol: TCP
  • Destination Port Range: 10256
• Description: "Allow Load Balancer or Network Load Balancer to communicate with `kube-proxy` on worker nodes."
	Ingress Rule 7
	• `isStateless`: `false`
	• `source`: *workerlb_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	  `destinationPortRange`
	  • `max`: `10256`
	  • `min`: `10256`
	• `description`: "Allow Load Balancer or Network Load Balancer to communicate with `kube-proxy` on worker nodes."

Ingress Rule 8
• Stateless: clear the box
• Ingress CIDR: *workerlb_cidr*
• IP Protocol: TCP
  • Destination Port Range: 30000-32767
• Description: "Allow worker nodes to receive connections through Network Load Balancer."
	Ingress Rule 8
	• `isStateless`: `false`
	• `source`: *workerlb_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	  `destinationPortRange`
	  • `max`: `32767`
	  • `min`: `30000`
	• `description`: "Allow worker nodes to receive connections through Network Load Balancer."

Compute Web UI property	OCI CLI property

Ingress Rule 9
• Stateless: clear the box
• Ingress CIDR: *kmi_cidr*
• IP Protocol: TCP
  • Destination Port Range: 10250
• Description: "Allow Kubernetes API endpoint to worker node communication."
	Ingress Rule 9
	• `isStateless`: `false`
	• `source`: *kmi_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `10250`
	  • `min`: `10250`
	• `description`: "Allow Kubernetes API endpoint to worker node communication."

Ingress Rule 10
• Stateless: clear the box
• Ingress CIDR: *kmi_cidr*
• IP Protocol: TCP
  • Destination Port Range: 10256
• Description: "Allow Load Balancer or Network Load Balancer to communicate with `kube-proxy` on worker nodes."
	Ingress Rule 10
	• `isStateless`: `false`
	• `source`: *kmi_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `10256`
	  • `min`: `10256`
	• `description`: "Allow Load Balancer or Network Load Balancer to communicate with `kube-proxy` on worker nodes."

Ingress Rule 11
• Stateless: clear the box
• Ingress CIDR: *pod_cidr*
• IP Protocol: TCP
  • Destination Port Range: 30000-32767
• Description: "Allow worker nodes to receive connections through the pod subnet."
	Ingress Rule 11
	• `isStateless`: `false`
	• `source`: *pod_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `32767`
	  • `min`: `30000`
	• `description`: "Allow worker nodes to receive connections through the pod subnet."

Compute Web UI property	OCI CLI property

Ingress Rule 12
• Stateless: clear the box
• Ingress CIDR: *kmi_cidr*
• IP Protocol: ICMP
  • Parameter Type: 8: Echo
• Description: "Test the reachability of a network pod from *kmi_cidr* by sending a request."
	Ingress Rule 12
	• `isStateless`: `false`
	• `source`: *kmi_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `1`
	• `icmpOptions`
	  • `type`: `8`
	• `description`: "Test the reachability of a network pod from *kmi_cidr* by sending a request."

Ingress Rule 13
• Stateless: clear the box
• Ingress CIDR: *kmi_cidr*
• IP Protocol: ICMP
  • Parameter Type: 0: Echo Reply
• Description: "If the destination pod is reachable from *kmi_cidr*, respond with an ICMP Echo Reply."
	Ingress Rule 13
	• `isStateless`: `false`
	• `source`: *kmi_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `1`
	• `icmpOptions`
	  • `type`: `0`
	• `description`: "If the destination pod is reachable from *kmi_cidr*, respond with an ICMP Echo Reply."

Create the Worker Subnet

Create a subnet. See Creating a Subnet on page 480. For Terraform input, see Example Terraform Scripts (VCN-Native Pod) on page 867.

For this example, use the following input to create the worker subnet. Use the OCID of the VCN that was created in Creating a VCN (VCN-Native Pod Networking) on page 879. Create the worker subnet in the same compartment where you created the VCN.

Create either a NAT private worker subnet or a VCN private worker subnet. Create a NAT private worker subnet to communicate outside the VCN.

Create a NAT Private Worker Subnet

Compute Web UI property	OCI CLI property

• Name: worker
• CIDR Block: *worker_cidr*
• Route Table: Select "nat_private" from the list
• Private Subnet: check the box
• DNS Hostnames:
  Use DNS Hostnames in this Subnet: check the box
  • DNS Label: worker
• Security Lists: Select "worker-seclist" and "Default Security List for oketest-vcn" from the list
	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id*
	• `--display-name`: `worker`
	• `--cidr-block`: *worker_cidr*
	• `--dns-label`: `worker`
	• `--prohibit-public-ip-on-vnic`: `true`
	• `--route-table-id`: OCID of the "nat_private" route table
	• `--security-list-ids`: OCIDs of the "worker-seclist" security list and the "Default Security List for oketest-vcn" security list

The difference in the following private subnet is the VCN private route table is used instead of the NAT private route table.

Create a VCN Private Worker Subnet

Compute Web UI property	OCI CLI property
• Name: worker • CIDR Block: *worker_cidr* • Route Table: Select "vcn_private" from the list • Private Subnet: check the box • DNS Hostnames: Use DNS Hostnames in this Subnet: check the box • DNS Label: worker • Security Lists: Select "worker-seclist" and "Default Security List for oketest-vcn" from the list	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id* • `--display-name`: `worker` • `--cidr-block`: *worker_cidr* • `--dns-label`: `worker` • `--prohibit-public-ip-on-vnic`: `true` • `--route-table-id`: OCID of the "vcn_private" route table • `--security-list-ids`: OCIDs of the "worker-seclist" security list and the "Default Security List for oketest-vcn" security list

Creating a Worker Load Balancer Subnet (VCN-Native Pod)

Learn how to create a worker load balancer subnet for VCN-Native Pod Networking On Private Cloud Appliance.

Create the following resources in the order listed:

1. Worker load balancer security list
2. Worker load balancer subnet

Create a Worker Load Balancer Security List

Create a security list. See Creating a Security List on page 496. For Terraform input, see Example Terraform Scripts (VCN-Native Pod) on page 867.

This security list defines traffic, such as applications, that is allowed to contact the worker load balancer.

For this example, use the following input for the worker load balancer subnet security list. These sources and destinations are examples; adjust these for your applications.

> **Note:**
>
> When you create an external load balancer for your containerized applications (see Exposing Containerized Applications on page 947), remember to add that load balancer service front-end port to this security list.

Compute Web UI property	OCI CLI property
• Name: workerlb-seclist	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id* • `--display-name`: `workerlb-seclist`
One egress security rule: • Stateless: clear the box • Egress CIDR: 0.0.0.0/0 • IP Protocol: All protocols • Description: "Allow all outgoing traffic."	**One egress security rule:** `--egress-security-rules` • `isStateless`: `false` • `destination`: `0.0.0.0/0` • `destinationType`: `CIDR_BLOCK` • `protocol`: `all` • `description`: "Allow all outgoing traffic."
Two ingress security rules:	**Two ingress security rules:** `--ingress-security-rules`

Compute Web UI property	OCI CLI property

Ingress Rule 1
• Stateless: clear the box
• Ingress CIDR: *kube_client_cidr*
• IP Protocol: TCP
  • Destination Port Range: 80
• Description: "Allow inbound traffic for applications."
	Ingress Rule 1
	• `isStateless`: `false`
	• `source`: *kube_client_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `80`
	  • `min`: `80`
	• `description`: "Allow inbound traffic for applications."

Ingress Rule 2
• Stateless: clear the box
• Ingress CIDR: *kube_client_cidr*
• IP Protocol: TCP
  • Destination Port Range: 443
• Description: "Allow inbound traffic for applications."
	Ingress Rule 2
	• `isStateless`: `false`
	• `source`: *kube_client_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `443`
	  • `min`: `443`
	• `description`: "Allow inbound traffic for applications."

Create the Worker Load Balancer Subnet

Create a subnet. See Creating a Subnet on page 480. For Terraform input, see Example Terraform Scripts (VCN-Native Pod) on page 867.

For this example, use the following input to create the worker load balancer subnet. Use the OCID of the VCN that was created in Creating a VCN (VCN-Native Pod Networking) on page 879. Create the worker load balancer subnet in the same compartment where you created the VCN.

Create either a private or a public worker load balancer subnet. Create a public worker load balancer subnet to use with a public cluster. Create a private worker load balancer subnet to expose applications in a private cluster.

Create a Public Worker Load Balancer Subnet

Compute Web UI property	OCI CLI property
• Name: service-lb • CIDR Block: *workerlb_cidr* • Route Table: Select "public" from the list • Public Subnet: check the box • DNS Hostnames: Use DNS Hostnames in this Subnet: check the box • DNS Label: servicelb • Security Lists: Select "workerlb-seclist" and "Default Security List for oketest-vcn" from the list	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id* • `--display-name`: `service-lb` • `--cidr-block`: *workerlb_cidr* • `--dns-label`: `servicelb` • `--prohibit-public-ip-on-vnic`: `false` • `--route-table-id`: OCID of the "public" route table • `--security-list-ids`: OCIDs of the "workerlb-seclist" security list and the "Default Security List for oketest-vcn" security list

The difference in the following private subnet is the VCN private route table is used instead of the public route table.

Create a VCN Private Worker Load Balancer Subnet

Compute Web UI property	OCI CLI property
• Name: service-lb • CIDR Block: *workerlb_cidr* • Route Table: Select "vcn_private" from the list • Private Subnet: check the box • DNS Hostnames: Use DNS Hostnames in this Subnet: check the box • DNS Label: servicelb • Security Lists: Select "workerlb-seclist" and "Default Security List for oketest-vcn" from the list	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id* • `--display-name`: `service-lb` • `--cidr-block`: *workerlb_cidr* • `--dns-label`: `servicelb` • `--prohibit-public-ip-on-vnic`: `true` • `--route-table-id`: OCID of the "vcn_private" route table • `--security-list-ids`: OCIDs of the "workerlb-seclist" security list and the "Default Security List for oketest-vcn" security list

Creating a Control Plane Subnet (VCN-Native Pod)

Learn how to create a control plane subnet for VCN-Native Pod Networking On Private Cloud Appliance.

Create the following resources in the order listed:

1. Control plane security list
2. Control plane subnet

Create a Control Plane Security List

Create a security list. See Creating a Security List on page 496. For Terraform input, see Example Terraform Scripts (VCN-Native Pod) on page 867.

For this example, use the following input for the control plane subnet security list. The `kubernetes_api_port` is the port used to access the Kubernetes API: port 6443. See also Workload Cluster Network Ports (VCN-Native Pod) on page 866.

Compute Web UI property	OCI CLI property
• Name: kmi-seclist	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id* • `--display-name`: `kmi-seclist`

Compute Web UI property	OCI CLI property
One egress security rule: • Stateless: clear the box • Egress CIDR: 0.0.0.0/0 • IP Protocol: All protocols • Description: "Allow all outgoing traffic."	**One egress security rule:** `--egress-security-rules` • `isStateless`: `false` • `destination`: `0.0.0.0/0` • `destinationType`: `CIDR_BLOCK` • `protocol`: `all` • `description`: "Allow all outgoing traffic."
Eleven ingress security rules:	**Eleven ingress security rules:** `--ingress-security-rules`

Ingress Rule 1
• Stateless: clear the box
• Ingress CIDR: *`kube_client_cidr`*
• IP Protocol: TCP
  • Destination Port Range: *`kubernetes_api_port`*
• Description: "Allow clients to communicate with Kubernetes API."
	Ingress Rule 1
	• `isStateless`: `false`
	• `source`: *`kube_client_cidr`*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	  `destinationPortRange`
	  • `max`: *`kubernetes_api_port`*
	  • `min`: *`kubernetes_api_port`*
	• `description`: "Allow clients to communicate with Kubernetes API."

Ingress Rule 2
• Stateless: clear the box
• Ingress CIDR: *`kmilb_cidr`*
• IP Protocol: TCP
  • Destination Port Range: *`kubernetes_api_port`*
• Description: "Allow the load balancer to communicate with Kubernetes control plane APIs."
	Ingress Rule 2
	• `isStateless`: `false`
	• `source`: *`kmilb_cidr`*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	  `destinationPortRange`
	  • `max`: *`kubernetes_api_port`*
	  • `min`: *`kubernetes_api_port`*
	• `description`: "Allow the load balancer to communicate with Kubernetes control plane APIs."

Compute Web UI property	OCI CLI property

Ingress Rule 3
• Stateless: clear the box
• Ingress CIDR: *kmilb_cidr*
• IP Protocol: TCP
  • Destination Port Range: 12250
• Description: "Allow Kubernetes worker to Kubernetes API endpoint communication via the control plane load balancer."
	Ingress Rule 3
	• isStateless: false
	• source: *kmilb_cidr*
	• sourceType: CIDR_BLOCK
	• protocol: 6
	• tcpOptions
	  destinationPortRange
	  • max: 12250
	  • min: 12250
	• description: "Allow Kubernetes worker to Kubernetes API endpoint communication via the control plane load balancer."

Ingress Rule 4
• Stateless: clear the box
• Ingress CIDR: *worker_cidr*
• IP Protocol: TCP
  • Destination Port Range: *kubernetes_api_port*
• Description: "Allow worker nodes to access the Kubernetes API."
	Ingress Rule 4
	• isStateless: false
	• source: *worker_cidr*
	• sourceType: CIDR_BLOCK
	• protocol: 6
	• tcpOptions
	  destinationPortRange
	  • max: *kubernetes_api_port*
	  • min: *kubernetes_api_port*
	• description: "Allow worker nodes to access the Kubernetes API."

Ingress Rule 5
• Stateless: clear the box
• Ingress CIDR: *worker_cidr*
• IP Protocol: TCP
  • Destination Port Range: 12250
• Description: "Allow Kubernetes worker to Kubernetes API endpoint communication."
	Ingress Rule 5
	• isStateless: false
	• source: *worker_cidr*
	• sourceType: CIDR_BLOCK
	• protocol: 6
	• tcpOptions
	  destinationPortRange
	  • max: 12250
	  • min: 12250
	• description: "Allow Kubernetes worker to Kubernetes API endpoint communication."

Compute Web UI property	OCI CLI property

Ingress Rule 6
• Stateless: clear the box
• Ingress CIDR: *kmi_cidr*
• IP Protocol: TCP
  • Destination Port Range: *kubernetes_api_port*
• Description: "Allow the control plane to reach itself."
	Ingress Rule 6
	• isStateless: false
	• source: *kmi_cidr*
	• sourceType: CIDR_BLOCK
	• protocol: 6
	• tcpOptions
	  destinationPortRange
	  • max: *kubernetes_api_port*
	  • min: *kubernetes_api_port*
	• description: "Allow the control plane to reach itself."

Ingress Rule 7
• Stateless: clear the box
• Ingress CIDR: *kmi_cidr*
• IP Protocol: TCP
  • Destination Port Range: 2379-2381
• Description: "Allow the control plane to reach etcd services and metrics. Ports 2379 and 2380 are used by Kubernetes to communicate with the etcd server. Port 2381 is used by Kubernetes to collect metrics from etcd."
	Ingress Rule 7
	• isStateless: false
	• source: *kmi_cidr*
	• sourceType: CIDR_BLOCK
	• protocol: 6
	• tcpOptions
	  destinationPortRange
	  • max: 2381
	  • min: 2379
	• description: "Allow the control plane to reach etcd services and metrics. Ports 2379 and 2380 are used by Kubernetes to communicate with the etcd server. Port 2381 is used by Kubernetes to collect metrics from etcd."

Ingress Rule 8
• Stateless: clear the box
• Ingress CIDR: *kmi_cidr*
• IP Protocol: TCP
  • Destination Port Range: 10250
• Description: "Allow Kubernetes API endpoint to control plane node communication."
	Ingress Rule 8
	• isStateless: false
	• source: *kmi_cidr*
	• sourceType: CIDR_BLOCK
	• protocol: 6
	• tcpOptions
	  destinationPortRange
	  • max: 10250
	  • min: 10250
	• description: "Allow Kubernetes API endpoint to control plane node communication."

Compute Web UI property	OCI CLI property

Ingress Rule 9
• Stateless: clear the box
• Ingress CIDR: *kmi_cidr*
• IP Protocol: TCP
  • Destination Port Range: 10257-10260
• Description: "Allow inbound connection for Kubernetes components."
	Ingress Rule 9
	• `isStateless`: `false`
	• `source`: *kmi_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `10260`
	  • `min`: `10257`
	• `description`: "Allow inbound connection for Kubernetes components."

Ingress Rule 10
• Stateless: clear the box
• Ingress CIDR: *pod_cidr*
• IP Protocol: TCP
  • Destination Port Range: *kubernetes_api_port*
• Description: "Allow pods to communicate with Kubernetes APIs."
	Ingress Rule 10
	• `isStateless`: `false`
	• `source`: *pod_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: *kubernetes_api_port*
	  • `min`: *kubernetes_api_port*
	• `description`: "Allow pods to communicate with Kubernetes APIs."

Ingress Rule 11
• Stateless: clear the box
• Ingress CIDR: *pod_cidr*
• IP Protocol: TCP
  • Destination Port Range: 12250
• Description: "Allow Kubernetes pods to Kubernetes API endpoint communication."
	Ingress Rule 11
	• `isStateless`: `false`
	• `source`: *pod_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `12250`
	  • `min`: `12250`
	• `description`: "Allow Kubernetes pods to Kubernetes API endpoint communication."

Create the Control Plane Subnet

Create a subnet. See Creating a Subnet on page 480. For Terraform input, see Example Terraform Scripts (VCN-Native Pod) on page 867.

Use the following input to create the control plane subnet. Use the OCID of the VCN that was created in Creating a VCN (VCN-Native Pod Networking) on page 879. Create the control plane subnet in the same compartment where you created the VCN.

Create either a NAT private control plane subnet or a VCN private control plane subnet. Create a NAT private control plane subnet to communicate outside the VCN.

> **Important:**
>
> The name of this subnet must be exactly "control-plane".

Create a Data Center Private Control Plane Subnet

Compute Web UI property	OCI CLI property
• Name: control-plane • CIDR Block: *kmi_cidr* • Route Table: Select "nat_private" from the list • Private Subnet: check the box • DNS Hostnames: Use DNS Hostnames in this Subnet: check the box • DNS Label: kmi • Security Lists: Select "kmi-seclist" and "Default Security List for oketest-vcn" from the list	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id* • `--display-name`: `control-plane` • `--cidr-block`: *kmi_cidr* • `--dns-label`: `kmi` • `--prohibit-public-ip-on-vnic`: `true` • `--route-table-id`: OCID of the "nat_private" route table • `--security-list-ids`: OCIDs of the "kmi-seclist" security list and the "Default Security List for oketest-vcn" security list

The difference in the following private subnet is the VCN private route table is used instead of the NAT private route table.

Create a VCN Private Control Plane Subnet

Compute Web UI property	OCI CLI property
• Name: control-plane • CIDR Block: *kmi_cidr* • Route Table: Select "vcn_private" from the list • Private Subnet: check the box • DNS Hostnames: Use DNS Hostnames in this Subnet: check the box • DNS Label: kmi • Security Lists: Select "kmi-seclist" and "Default Security List for oketest-vcn" from the list	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id* • `--display-name`: `control-plane` • `--cidr-block`: *kmi_cidr* • `--dns-label`: `kmi` • `--prohibit-public-ip-on-vnic`: `true` • `--route-table-id`: OCID of the "vcn_private" route table • `--security-list-ids`: OCIDs of the "kmi-seclist" security list and the "Default Security List for oketest-vcn" security list

Creating a Control Plane Load Balancer Subnet (VCN-Native Pod)

Learn how to create a control plane load balancer subnet for VCN-Native Pod Networking On Private Cloud Appliance.

Create the following resources in the order listed:

1. Control plane load balancer security list
2. Control plane load balancer subnet

Create a Control Plane Load Balancer Security List

Create a security list. See Creating a Security List on page 496. For Terraform input, see Example Terraform Scripts (VCN-Native Pod) on page 867.

The control plane load balancer accepts traffic on port 6443, which is also called ***kubernetes_api_port*** in this guide. Adjust this security list to only accept connections from where you expect the network to run. Port 6443 must accept connections from the cluster control plane instances and worker instances.

For this example, use the following input for the control plane load balancer subnet security list.

Compute Web UI property	OCI CLI property
• Name: kmilb-seclist	• `--vcn-id`: `ocid1.vcn.`*`oke_vcn_id`* • `--display-name`: kmilb-seclist
One egress security rule: • Stateless: clear the box • Egress CIDR: 0.0.0.0/0 • IP Protocol: All protocols • Description: "Allow all outgoing traffic."	**One egress security rule:** `--egress-security-rules` • `isStateless`: `false` • `destination`: `0.0.0.0/0` • `destinationType`: `CIDR_BLOCK` • `protocol`: `all` • `description`: "Allow all outgoing traffic."
Eight ingress security rules:	**Eight ingress security rules:** `--ingress-security-rules`

Ingress Rule 1:
• Stateless: clear the box
• Ingress CIDR: *`kube_internal_cidr`*
This value is required. Do not change this CIDR value.
• IP Protocol: TCP
  • Destination Port Range: *`kubernetes_api_port`*
• Description: "Allow a Kubernetes container to communicate with Kubernetes APIs."
	Ingress Rule 1:
	• `isStateless`: `false`
	• `source`: *`kube_internal_cidr`*
	This value is required. Do not change this CIDR value.
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: *`kubernetes_api_port`*
	  • `min`: *`kubernetes_api_port`*
	• `description`: "Allow a Kubernetes container to communicate with Kubernetes APIs."

Ingress Rule 2:
• Stateless: clear the box
• Ingress CIDR: *`kube_client_cidr`*
• IP Protocol: TCP
  • Destination Port Range: *`kubernetes_api_port`*
• Description: "Allow clients to connect with the Kubernetes cluster."
	Ingress Rule 2:
	• `isStateless`: `false`
	• `source`: *`kube_client_cidr`*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: *`kubernetes_api_port`*
	  • `min`: *`kubernetes_api_port`*
	• `description`: "Allow clients to connect with the Kubernetes cluster."

Compute Web UI property	OCI CLI property

Ingress Rule 3:
• Stateless: clear the box
• Ingress CIDR: *kmi_cidr*
• IP Protocol: TCP
  • Destination Port Range: *kubernetes_api_port*
• Description: "Allow the control plane to reach itself via the load balancer."
	Ingress Rule 3:
	• `isStateless`: `false`
	• `source`: *kmi_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: *kubernetes_api_port*
	  • `min`: *kubernetes_api_port*
	• `description`: "Allow the control plane to reach itself via the load balancer."

Ingress Rule 4:
• Stateless: clear the box
• Ingress CIDR: *worker_cidr*
• IP Protocol: TCP
  • Destination Port Range: *kubernetes_api_port*
• Description: "Allow worker nodes to connect with the cluster via the control plane load balancer."
	Ingress Rule 4:
	• `isStateless`: `false`
	• `source`: *worker_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: *kubernetes_api_port*
	  • `min`: *kubernetes_api_port*
	• `description`: "Allow worker nodes to connect with the cluster via the control plane load balancer."

Ingress Rule 5:
• Stateless: clear the box
• Ingress CIDR: *worker_cidr*
• IP Protocol: TCP
  • Destination Port Range: `12250`
• Description: "Allow Kubernetes worker to Kubernetes API endpoint communication via the load balancer."
	Ingress Rule 5:
	• `isStateless`: `false`
	• `source`: *worker_cidr*
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `12250`
	  • `min`: `12250`
	• `description`: "Allow Kubernetes worker to Kubernetes API endpoint communication via the load balancer."

Compute Web UI property	OCI CLI property

Ingress Rule 6:
• Stateless: clear the box
• Ingress CIDR: *pod_cidr*
• IP Protocol: TCP
  • Destination Port Range: 12250
• Description: "Allow Kubernetes pods to Kubernetes API endpoint communication via the load balancer."
	Ingress Rule 6:
	• isStateless: false
	• source: *pod_cidr*
	• sourceType: CIDR_BLOCK
	• protocol: 6
	• tcpOptions
	  destinationPortRange
	  • max: 12250
	  • min: 12250
	• description: "Allow Kubernetes pods to Kubernetes API endpoint communication via the load balancer."

Ingress Rule 7: Private Endpoint
• Stateless: clear the box
• Ingress CIDR: *kmilb_cidr*
• IP Protocol: TCP
  • Destination Port Range: *kubernetes_api_port*
• Description: "Used to create a private control plane endpoint."
	Ingress Rule 7: Private Endpoint
	• isStateless: false
	• source: *kmilb_cidr*
	• sourceType: CIDR_BLOCK
	• protocol: 6
	• tcpOptions
	  destinationPortRange
	  • max: *kubernetes_api_port*
	  • min: *kubernetes_api_port*
	• description: "Used to create a private control plane endpoint."

Ingress Rule 8: Public Endpoint
• Stateless: clear the box
• Ingress CIDR: *public_ip_cidr*
• IP Protocol: TCP
  • Destination Port Range: *kubernetes_api_port*
• Description: "Used to access the control plane endpoint from the public CIDR. If you do not know what your public IP CIDR is, ask your Service Enclave administrator."
	Ingress Rule 8: Public Endpoint
	• isStateless: false
	• source: *public_ip_cidr*
	• sourceType: CIDR_BLOCK
	• protocol: 6
	• tcpOptions
	  destinationPortRange
	  • max: *kubernetes_api_port*
	  • min: *kubernetes_api_port*
	• description: "Used to access the control plane endpoint from the public CIDR. If you do not know what your public IP CIDR is, ask your Service Enclave administrator."

Create the Control Plane Load Balancer Subnet

Create a subnet. See Creating a Subnet on page 480. For Terraform input, see Example Terraform Scripts (VCN-Native Pod) on page 867.

For this example, use the following input to create the control plane load balancer subnet. Use the OCID of the VCN that was created in Creating a VCN (VCN-Native Pod Networking) on page 879. Create the control plane load balancer subnet in the same compartment where you created the VCN.

Create either a private or a public control plane load balancer subnet. Create a public control plane load balancer subnet to use with a public cluster. Create a private control plane load balancer subnet to use with a private cluster.

See Private Clusters on page 835 for information about using Local Peering Gateways to connect a private cluster to other instances on the Private Cloud Appliance and using Dynamic Routing Gateways to connect a private cluster to the on-premises IP address space. To create a private control plane load balancer subnet, specify one of the following route tables:

- vcn_private
- lpg_rt
- drg_rt

Create a Public Control Plane Load Balancer Subnet

Compute Web UI property	OCI CLI property
• Name: control-plane-endpoint • CIDR Block: *kmilb_cidr* • Route Table: Select "public" from the list • Public Subnet: check the box • DNS Hostnames: Use DNS Hostnames in this Subnet: check the box • DNS Label: kmilb • Security Lists: Select "kmilb-seclist" and "Default Security List for oketest-vcn" from the list	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id* • `--display-name`: `control-plane-endpoint` • `--cidr-block`: *kmilb_cidr* • `--dns-label`: `kmilb` • `--prohibit-public-ip-on-vnic`: `false` • `--route-table-id`: OCID of the "public" route table • `--security-list-ids`: OCIDs of the "kmilb-seclist" security list and the "Default Security List for oketest-vcn" security list

The difference in the following private subnet is the VCN private route table is used instead of the public route table. Depending on your needs, you could specify the LPG route table or the DRG route table instead.

Create a Private Control Plane Load Balancer Subnet

Compute Web UI property	OCI CLI property
• Name: control-plane-endpoint • CIDR Block: *kmilb_cidr* • Route Table: Select "vcn_private" from the list • Private Subnet: check the box • DNS Hostnames: Use DNS Hostnames in this Subnet: check the box • DNS Label: kmilb • Security Lists: Select "kmilb-seclist" and "Default Security List for oketest-vcn" from the list	• `--vcn-id`: `ocid1.vcn.`*oke_vcn_id* • `--display-name`: `control-plane-endpoint` • `--cidr-block`: *kmilb_cidr* • `--dns-label`: `kmilb` • `--prohibit-public-ip-on-vnic`: `true` • `--route-table-id`: OCID of the "vcn_private" route table • `--security-list-ids`: OCIDs of the "kmilb-seclist" security list and the "Default Security List for oketest-vcn" security list

Creating and Managing OKE Clusters

Learn how to create, update, and delete OKE clusters On Private Cloud Appliance. Be sure to carefully read the descriptions of the cluster parameters before you create the cluster.

You can create either a public cluster or a private cluster. See Public and Private Clusters on page 834 for the resources required for each.

> **Note:**
>
> You cannot create both public and private clusters in one VCN.

A cluster includes cluster management nodes. This topic describes how to recognize those management nodes in a list of instances.

This topic also describes how to create a Kubernetes configuration file. You need a Kubernetes configuration file for each OKE cluster that you work with. The Kubernetes configuration file enables you to access OKE clusters using the `kubectl` command and the Kubernetes Dashboard.

Creating an OKE Cluster

Learn how to create OKE clusters On Private Cloud Appliance.

If you create a public cluster, the Network Load Balancer and public IP address are created and assigned as part of cluster creation.

> **Important:**
>
> Before you can create a cluster, the following conditions must be met:
>
> - The OraclePCA-OKE.cluster_id defined tag must exist in the tenancy. See Create the OraclePCA-OKE.cluster_id Tag on page 827.
> - All fault domains must be healthy. A Roving Edge device has only 1 fault domain.
> - Each fault domain must have at least one healthy compute instance.
> - Sufficient resources must be available to create a cluster.
> - Ensure that no appliance upgrade is scheduled during the cluster create.

If notifications are configured for operations such as system upgrade, ensure you are on the list to be notified of such planned outages.

To create a node pool at the same time that you create the cluster, you must use the Compute Web UI.

To specify tags to be applied to all load balancers created by Kubernetes services, you must use the OCI CLI.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Containers**, then select **Kubernetes Clusters (OKE)**.
2. On the clusters list page, select the **Create Cluster** button.
3. In the **Create Cluster** dialog box, provide the following information:
   - **Name**: Enter a name for the new cluster. Avoid entering confidential information.
   - **Compartment**: Select the compartment in which to create the new cluster.
   - **Kubernetes Version**: Select the version of Kubernetes to run on the control plane nodes. Accept the default version or select a different version.

     If the Kubernetes version that you want to use isn't listed, use the OCI CLI or the API to create the cluster and specify the Kubernetes version.
   - **Tagging**: Add the following defined or free-form tags for the cluster resource.

     > **Note:**
     >
     > Don't specify values for the OraclePCA-OKE.cluster_id defined tag or for the ClusterResourceIdentifier free-form tag. These tag values are system-generated and only applied to nodes (instances), not to the cluster resource.

     Use OraclePCA defined tags to provide the following information for control plane nodes. If these tags aren't listed in the Compute Web UI Tagging menus, you must create them. See Creating OraclePCA Tags on page 828.

> **Important:**
>
> If you are using Private Cloud Appliance Release 3.0.2-b1081557, these defined tags are not recognized. You must use free-form tags to specify these values. In Private Cloud Appliance Release 3.0.2-b1185392 and later, the free-form tags are deprecated; use the defined tags described here for SSH key, number of control plane nodes, node shape, and node configuration in Private Cloud Appliance Release 3.0.2-b1185392 and later.

> **Note:**
>
> None of these values - SSH key, number of nodes, node shape, or node shape configuration - can be set or changed after the cluster is created. If you set these tags when you update the cluster, the new values are ignored.

- **Your public SSH key**.

  Specify `sshkey` for the tag key (`OraclePCA.sshkey`). Paste your public SSH key into the Value field.

  > **Important:**
  >
  > You can't add an SSH key after the cluster is created.

- **Number of nodes**.

  By default, the number of nodes in the control plane is 3. You can specify 1, 3, or 5 nodes. To specify the number of control plane nodes, specify `cp_node_count` for the tag key (`OraclePCA.cpNodeCount`), and select `1`, `3`, or `5` in the **Value** field.

- **Node shape**.

  For Private Cloud Appliance X10 systems, the shape of the control plane nodes is VM.PCAStandard.E5.Flex and you cannot change it. For all other Private Cloud Appliance systems, the default shape is VM.PCAStandard1.1, and you can specify a different shape.

  To use a different shape, specify cpNodeShape for the tag key (OraclePCA.cpNodeShape), and enter the name of the shape in the Value field. For descriptions of available shapes, see Compute Shapes.

- **Node shape configuration**.

  You can change the default shape configuration.

  To provide shape configuration information, specify `cp_node_shape_config` for the tag key (`OraclePCA.cpNodeShapeConfig`). You must specify the number of OCPUs (`ocpus`) you want. You can optionally specify the total amount of memory you want (`memoryInGBs`). The default value for gigabytes of memory is 16 times the number you specify for OCPUs.

  > **Note:**
  >
  > If the cluster will have 1-10 worker nodes, specify at least 16 GB memory. If the cluster will have 11-128 worker nodes, specify at least 2 OCPUs and 32 GB memory. You can't change the number of OCPUs or amount of memory when you update the cluster.

  In the **Value** field for the tag, enter the node shape configuration value as shown in the following examples.

  In the following example, the default amount of memory is be configured:

  ```
  {"ocpus":1}
  ```

  In the following example, the amount of memory is specified:

  ```
  {"ocpus":2, "memoryInGBs":48}
  ```

> **Note:**
>
> If you use Terraform to specify a complex value (a value that's a key/value pair), then you must escape the double quotation marks in the value as shown in the following example:
>
> ```
> "OraclePCA.cpNodeShapeConfig"="{\"ocpus\":2,
> \"memoryInGBs\":48}"
> ```

- **Add-ons**: This section shows a tile for each add-on that's available for this cluster. In the Create Cluster dialog box, all add-ons are Disabled. See Deploying the WebLogic Kubernetes Operator Add-on on page 920.

4. Select **Next**.
5. On the **Network** page in the **Create Cluster** dialog box, provide the following information:

   - **Network Type**. Specifies how pods running on nodes in the cluster communicate with each other, with the cluster's control plane nodes, with pods on other clusters, with other services (such as storage services), and with the internet.

     The **Flannel Overlay** network type encapsulates communication between pods in the Flannel Overlay network. The Flannel Overlay network is a simple private overlay virtual network that satisfies the requirements of the OKE networking model by attaching IP addresses to containers. The pods in the private overlay network are only accessible from other pods in the same cluster. For more description, see Creating Flannel Overlay Network Resources on page 836.
   - **VCN-Native Pod Networking** connects nodes in a Kubernetes cluster to pod subnets in the OKE VCN. As a result, pod IP addresses within the OKE VCN are directly routable from other VCNs that are connected (peered) to the OKE VCN, and from on-premises networks. For more description, see Creating VCN-Native Pod Network Resources on page 864.

     > **Note:**
     >
     > If you specify VCN-Native Pod Networking, then the VCN you specify must have a subnet named `pod`. See Creating VCN-Native Pod Network Resources on page 864.

   - **VCN**. Select the VCN that has the configuration of the "oke_vcn" VCN described in Creating a VCN (Flannel Overlay) on page 848 or Creating a VCN (VCN-Native Pod Networking) on page 879.
   - **Kubernetes Service LB Subnet**. The subnet that's configured to host the load balancer in a Kubernetes cluster. To create a public cluster, create and specify here the public version of the "service-lb" subnet described in Creating a Worker Load Balancer Subnet (Flannel Overlay) on page 856 or Creating a Worker Load Balancer Subnet (VCN-Native Pod) on page 893. To create a private cluster, create and specify here the private version of the "service-lb" subnet.
   - **Kubernetes API Endpoint Subnet**. The regional subnet in which to place the cluster endpoint. To create a public cluster, create and specify here the public version of the "control-plane-endpoint" subnet described in Creating an OKE Control Plane Subnet (Flannel Overlay) on page 858 or Creating a Control Plane Subnet (VCN-Native Pod) on page 895. To create a private cluster, create and specify here the private version of the "control-plane-endpoint" subnet.
   - **Kubernetes Service CIDR Block**. (Optional) The default value is 10.96.0.0/16.
   - **Pods CIDR Block**. (Optional) The default value is 10.244.0.0/16.
   - **Network Security Group**. If you check the box to enable network security groups, select **Add Network Security Group** and select an NSG from the drop-down list. You might need to change the compartment to find the NSG you want.
6. Select **Next**.
7. On the **Node Pool** page, select the Add Node Pool button to optionally add a node pool as part of creating this cluster. See Creating an OKE Worker Node Pool on page 932 to add node pools after the cluster is created.

   If you select the Add Node Pool button, enter the following information in the Add Node Pool section:

   - **Name**: The name of the new node pool. Avoid using confidential information.

- **Compartment**: The compartment in which to create the new node pool.
- **Node Count**: Enter the number of nodes you want in this node pool. The default is 0. The maximum number is 128 per cluster, which can be distributed across multiple node pools.
- See Creating an OKE Worker Node Pool on page 932 for information about Network Security Groups, Placement Configuration, Source Image, Shape, and Pod Communication.

8. Review your entries.

   If you created a node pool, you can edit or delete the node pool in this review.

9. Select **Submit**.

   The details page for the cluster is displayed. Scroll to the Resources section and select Work Requests to see the progress of the cluster creation. If you created a node pool, the NODEPOOL_CREATE work request might still be In Progress for a time after the cluster is Active and the CLUSTER_CREATE work request is Succeeded.

   Under **Resources**, select **Work Requests** to see the progress of the cluster creation.

   The cluster details page doesn't list OraclePCA tags on the Tags tab (and you can't filter a list of clusters by the values of OraclePCA tags). To review the settings of the OraclePCA tags, use the CLI.

   The cluster details page doesn't list the cluster control plane nodes. To view the control plane nodes, view the list of instances in the compartment where you created this cluster. Names of control plane nodes are in the following format:

   ```
   oke-ID1-control-plane-ID2
   ```

   - *ID1* - The first 32 characters after the *pca_name* in the cluster OCID.
   - *ID2* - A unique identifier added when the cluster has more than one control plane node.

   Search for the instances in the list whose names contain the *ID1* string from this cluster OCID.

What's Next:

1. Create a Kubernetes configuration file for the cluster. See Creating a Kubernetes Configuration File on page 912.
2. Deploy a Kubernetes Dashboard to manage the cluster and to manage and troubleshoot applications running in the cluster. On the https://kubernetes.io/ site, see Deploy and Access the Kubernetes Dashboard.
3. Create a node pool for the cluster. See Creating an OKE Worker Node Pool on page 932.
4. Create a backup for the workload cluster. For example, see Backing up an etcd cluster and Restoring up an etcd cluster in Operating etcd clusters for Kubernetes. Use the etcd backup to recover OKE clusters under disaster scenarios such as losing all control plane nodes. An etcd backup contains all OKE states and critical information. An etcd backup does not back up applications or other content on cluster nodes.

Using the CLI

Use the oci ce cluster create command and required parameters to create a new cluster.

```
oci ce cluster create --compartment-id coompartment_OCID --kubernetes-
version version-to-install --name cluster_name --vcn-id VCN_OCID [OPTIONS]
```

To install a cluster add-on, use the `cluster install-addon` command after you have created the cluster. See Deploying the WebLogic Kubernetes Operator Add-on on page 920.

Procedure

1. Get the information you need to run the command:

   - The OCID of the compartment where you want to create the cluster: `oci iam compartment list`
   - The name of the cluster. Avoid entering confidential information.
   - The Kubernetes version that you want to use. Use the following command to show a list of available Kubernetes versions:

     ```
     oci ce cluster-options get --cluster-option-id all
     ```

You might be able to list more Kubernetes versions by using the `compute image list` command and looking in the display name. In the following example, the Kubernetes version in the image is 1.29.9:

```
"display-name": "uln-pca-Oracle-Linux8-OKE-1.29.9-20250325.oci"
```

Another way to specify a version that's not listed is to use the OCID of an older cluster instead of the keyword `all` as the argument of the `--cluster-option-id` option to list the Kubernetes version used for that specified cluster:

```
oci ce cluster-options get --cluster-option-id cluster_OCID
```

If you're using Release 3.0.2-b1081557, the `cluster-options get` command isn't available. Use the `compute image list` command to get the Kubernetes version from the image display name.

- OCID of the virtual cloud network (VCN) in which you want to create the cluster. Specify the VCN that has the configuration of the "oke_vcn" VCN described in Creating a VCN (Flannel Overlay) on page 848 or Creating a VCN (VCN-Native Pod Networking) on page 879.
- OCID of the OKE service LB subnet. Specify the subnet that has a configuration such as the "service-lb" subnet described in Creating a Worker Load Balancer Subnet (Flannel Overlay) on page 856 or Creating a Worker Load Balancer Subnet (VCN-Native Pod) on page 893. Specify only one OKE service LB subnet.
- OCID of the Kubernetes API endpoint subnet. Specify the subnet that has configuration such as the "control-plane-endpoint" subnet described in Creating an OKE Control Plane Load Balancer Subnet (Flannel Overlay) on page 861 or Creating a Control Plane Load Balancer Subnet (VCN-Native Pod) on page 900. For a public cluster, follow the instructions to create the public version of the "service-lb" subnet. For a private cluster, create the private version of the "service-lb" subnet. Specify only one OKE Service LB subnet.
- OKE service CIDR block. (Optional) The default value is 10.96.0.0/16.
- Pods CIDR block. (Optional) The default value is 10.244.0.0/16.
- (Optional) The OCID of the Network Security Group to apply to the cluster endpoint. Don't specify more than one NSG. If you specify an NSG, use the following syntax:

  ```
  --endpoint-nsg-ids '["ocid1.networksecuritygroup.unique_ID"]'
  ```

- (Optional) Your public SSH key in RSA format. You can't add or update an SSH key after the cluster is created.
- The network type. (Optional) Specify either `OCI_VCN_IP_NATIVE` or `FLANNEL_OVERLAY` for the value of the `cniType` parameter in the argument for the `--cluster-pod-network-options` option. See the descriptions of Flannel Overlay and VCN-Native Pod Networking in the Console tab. If you don't specify the `--cluster-pod-network-options` option, `FLANNEL_OVERLAY` is used.

  ```
  --cluster-pod-network-options '[{"cniType": "OCI_VCN_IP_NATIVE"}]'
  ```

  Note:

  If you specify `OCI_VCN_IP_NATIVE`, then the VCN you specify must have a subnet named `pod`.

2. (Optional) Add free-form tags for the cluster resource by using the `--defined-tags` or `--freeform-tags` options.

   Note:

   Don't specify values for the OraclePCA-OKE.cluster_id defined tag or for the ClusterResourceIdentifier free-form tag. These tag values are system-generated and only applied to nodes (instances), not to the cluster resource.

   Define an argument for the `--defined-tags` option to provide the following information for control plane nodes. Specify OraclePCA as the tag namespace.

   Important:

   If you are using Private Cloud Appliance Release 3.0.2-b1081557, these defined tags are not recognized. You must use free-form tags to specify

these values. In Private Cloud Appliance Release 3.0.2-b1185392 and later, the free-form tags are deprecated; use the defined tags described here for SSH key, number of control plane nodes, node shape, and node configuration in Private Cloud Appliance Release 3.0.2-b1185392 and later.

Note:

None of these values - SSH key, number of nodes, node shape, or node shape configuration - can be set or changed after the cluster is created. If you set these tags when you update the cluster, the new values are ignored.

- Your public SSH key

  Specify `sshkey` for the tag key, and paste your public SSH key as the value.

  Important:

  You can't add an SSH key after the cluster is created.

- Number of nodes

  By default, the number of nodes in the control plane is 3. You can specify 1, 3, or 5 nodes. To specify the number of control plane nodes, specify `cpNodeCount` for the tag key, and enter `1`, `3`, or `5` for the value.

- Node shape

  For Private Cloud Appliance X10 systems, the shape of the control plane nodes is VM.PCAStandard.E5.Flex and you cannot change it. For all other Private Cloud Appliance systems, the default shape is VM.PCAStandard1.1, and you can specify a different shape. For descriptions of available shapes, see Compute Shapes.

  To use a different shape, specify cpNodeShape for the tag key (OraclePCA.cpNodeShape), and enter the name of the shape as the value. Use the following command to list the available shapes and their characteristics:

  ```
  oci compute shape list --compartment-id compartment_OCID
  ```

- Node shape configuration

  If you specify a shape that is not a flexible shape, do not specify a shape configuration. The number of OCPUs and amount of memory are set to the values shown for this shape in Compute Shapes.

  If you specify a flexible shape, you can change the default shape configuration. To provide shape configuration information, specify `cp_node_shape_config` for the tag key (`OraclePCA.cpNodeShapeConfig`). You must specify the number of OCPUs (`ocpus`) you want. You can optionally specify the total amount of memory you want (`memoryInGBs`). The default value for gigabytes of memory is 16 times the number you specify for OCPUs.

  Note:

  If the cluster will have 1-10 worker nodes, specify at least 16 GB memory. If the cluster will have 11-128 worker nodes, specify at least 2 OCPUs and 32 GB memory. You can't change the number of OCPUs or amount of memory when you update the cluster.

  Specify defined tags, either inline, or in a file in JSON format, such as the following example file:

  ```
  {
    "OraclePCA": {
      "sshkey": "ssh-rsa <remainder_of_key_text>",
      "cpNodeCount": 1,
      "cpNodeShape": "VM.PCAStandard1.Flex",
      "cpNodeShapeConfig": {
        "ocpus": 2,
        "memoryInGBs": 48
  ```

```
    }
  }
}
```

Use the following syntax to specify a file of tags. Specify the full path to the `.json` file unless the file is in the same directory where you're running the command.

```
--defined-tags file://cluster_tags.json
```

3. (Optional) You can use the `--service-lb-defined-tags` or `--service-lb-freeform-tags` options to specify tags to be applied to all load balancers created by Kubernetes services. Ensure that the applicable dynamic group includes the `use tag-namespaces` policy. See Exposing Containerized Applications on page 947.
4. Run the create cluster command.

   If the `--endpoint-subnet-id` that you specify is a public subnet, then a public endpoint is created, and the `--endpoint-public-ip-enabled` option must be set to `true`.

   If the `--endpoint-subnet-id` that you specify is a private subnet, then a private endpoint is created, and the `--endpoint-public-ip-enabled` option must be set to `false`.

   Example:

```
$ oci ce cluster create \
--compartment-id ocid1.compartment.unique_ID --kubernetes-version version
 \
--name "Native Cluster" --vcn-id ocid1.vcn.unique_ID \
--cluster-pod-network-options '{"cniType":"OCI_VCN_IP_NATIVE"}' \
--endpoint-subnet-id control-plane-endpoint_subnet_OCID \
--endpoint-public-ip-enabled false \
--service-lb-subnet-ids '["service-lb_subnet_OCID"]' \
--defined-tags '{"OraclePCA":{"sshkey":"ssh-rsa remainder_of_key_text"}}'
```

   The output from this `cluster create` command is the same as the output from the `cluster get` command.

   Use the `work-request get` command to check the status of the create operation. The work request OCID is in `created-by-work-request-id` in the `metadata` section of the `cluster create` output.

```
$ oci ce work-request get --work-request-id workrequest_OCID
```

   To identify the control plane nodes for this cluster, list instances in the compartment where you created the cluster. Names of control plane nodes are in the following format:

```
oke-ID1-control-plane-ID2
```

   - *ID1* - The first 32 characters after the *pca_name* in the cluster OCID.
   - *ID2* - A unique identifier added when the cluster has more than one control plane node.

   Search for the instances in the list whose names contain the *ID1* string from this cluster OCID.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

What's Next:

1. Create a Kubernetes configuration file for the cluster. See Creating a Kubernetes Configuration File on page 912.
2. Deploy a Kubernetes Dashboard to manage the cluster and to manage and troubleshoot applications running in the cluster. On the https://kubernetes.io/ site, see Deploy and Access the Kubernetes Dashboard.
3. Create a node pool for the cluster. See Creating an OKE Worker Node Pool on page 932.
4. Create a backup for the workload cluster. For example, see Backing up an etcd cluster and Restoring up an etcd cluster in Operating etcd clusters for Kubernetes. Use the etcd backup to recover OKE clusters under disaster

scenarios such as losing all control plane nodes. An etcd backup contains all OKE states and critical information. An etcd backup does not back up applications or other content on cluster nodes.

Using the API

Use the CreateCluster operation to create a new cluster.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

What's Next:

1. Create a Kubernetes configuration file for the cluster. See Creating a Kubernetes Configuration File on page 912.
2. Deploy a Kubernetes Dashboard to manage the cluster and to manage and troubleshoot applications running in the cluster. On the https://kubernetes.io/ site, see Deploy and Access the Kubernetes Dashboard.
3. Create a node pool for the cluster. See Creating an OKE Worker Node Pool on page 932.
4. Create a backup for the workload cluster. For example, see Backing up an etcd cluster and Restoring up an etcd cluster in Operating etcd clusters for Kubernetes. Use the etcd backup to recover OKE clusters under disaster scenarios such as losing all control plane nodes. An etcd backup contains all OKE states and critical information. An etcd backup does not back up applications or other content on cluster nodes.

Creating a Kubernetes Configuration File

On Private Cloud Appliance, you can set up a Kubernetes configuration file for each OKE cluster that you work with. Your Kubernetes configuration file enables you to access OKE clusters using the `kubectl` command and the Kubernetes Dashboard.

Kubernetes configuration files organize information about clusters, users, namespaces, and authentication mechanisms. You can define contexts to easily switch between clusters and namespaces. The `kubectl` tool uses Kubernetes configuration files to find the information it needs to select a cluster and communicate with the API server of a cluster.

Installing the Kubernetes Command Line Tool

Install and configure the Kubernetes command line tool `kubectl`. The `kubectl` tool enables you to perform operations on OKE clusters such as deploy applications, inspect and manage cluster resources, and view logs.

To install `kubectl`, see Kubernetes Install Tools. The `kubectl` version must be within one minor version of the OKE cluster Kubernetes version. For example, a v1.29 client can communicate with v1.28, v1.29, and v1.30 control planes. See Supported Versions of Kubernetes on page 822S.

For more information, including a complete list of `kubectl` operations, see the https://kubernetes.io/docs/reference/kubectl/ reference page.

Create a Kubernetes Configuration File

Use the CLI to create your Kubernetes configuration file.

> **Tip:**
>
> The Quick Start button on a cluster details page in the Compute Web UI shows how to create a Kubernetes configuration file, and provides the OCID of the cluster.

1. Get the OCID of the cluster: `oci ce cluster list`
2. Run the oci ce cluster create-kubeconfig command as described to create the configuration file:

   The `--cluster-id` option is required.

   The default value of the `--file` option is `~/.kube/config`. If you already have a file at the specified location and you want to replace it, use the `--overwrite` option. To maintain more than one configuration file, select a different file by using the `KUBECONFIG` environment variable or the `--kubeconfig` option.

The value of the `--kube-endpoint` option must be `PUBLIC_ENDPOINT`.

If you don't specify the `--profile` option, the current value of your `OCI_CLI_PROFILE` environment variable is used. Best practice is to specify this value.

If provided, the value of the `--token-version` option must be 2.0.0.

Example:

Use the following command to configure a Kubernetes configuration file for the specified cluster using the public endpoint:

```
$ oci ce cluster create-kubeconfig --cluster-id ocid1.cluster.unique_ID \
--file $HOME/.kube/config --kube-endpoint PUBLIC_ENDPOINT --
profile profile-name
New config written to the Kubeconfig file /home/username/.kube/config
```

A Kubernetes configuration file includes an OCI CLI command that dynamically generates an authentication token and inserts it when you run a `kubectl` command. By default, the OCI CLI command in the Kubernetes configuration file uses your current OCI CLI profile when generating an authentication token. If you have defined multiple profiles in your OCI CLI configuration file, use one of the following methods to specify which profile to use when generating the authentication token. The value of *profile-name* is the name of the profile in your OCI CLI configuration file.

- Ensure that your `OCI_CLI_PROFILE` environment variable is set to the profile for the tenancy where the `ocid1.cluster.unique_ID` resides. This setting is ignored if one of the following methods was used to specify the profile for this cluster in the Kubernetes configuration file.
- Specify the `--profile` option on the `create-kubeconfig` command line as shown in the preceding example command.
- Edit the generated configuration file as shown in the following example.

```
user:
  exec:
    apiVersion: client.authentication.k8s.io/v1beta1
    args:
    - ce
    - cluster
    - generate-token
    - --cluster-id
    - cluster ocid
    - --profile
    - profile-name
    command: oci
    env: []
```

Use the following command to set your `KUBECONFIG` environment variable to the Kubernetes configuration file that you created or updated in the preceding command:

```
$ export KUBECONFIG=$HOME/.kube/config
```

The following command shows the content of your new YAML configuration file:

```
$ kubectl config view
```

If you run the command again with a different cluster OCID, the new information is merged with the existing information. The following message is displayed:

```
Existing Kubeconfig file found at /home/username/.kube/config and new
 config merged into it
```

Verify Your Cluster Access

Before you run `kubectl` commands, enure that your `OCI_CLI_PROFILE` environment variable is set to the name of the profile that is defined in your OCI configuration file:

```
$ export OCI_CLI_PROFILE=profile-name
```

Run the following command to confirm that you can access your cluster:

```
$ kubectl cluster-info
```

Every Kubernetes namespace contains at least one ServiceAccount: the default ServiceAccount for that namespace, which is named `default`. If you don't specify a ServiceAccount when you create a Pod, the OKE service automatically assigns the ServiceAccount named `default` in that namespace.

An application running inside a Pod can access the Kubernetes API using automatically mounted service account credentials.

Create a Kubernetes Dashboard

The dashboard helps you manage the cluster and manage and troubleshoot applications running in the cluster.

On the Kubernetes site, see Deploy and Access the Kubernetes Dashboard.

What's Next:

Creating an OKE Worker Node Pool on page 932

Updating an OKE Cluster

On Private Cloud Appliance, when you update an OKE cluster, you can change the cluster name, Kubernetes version, and tags.

Best practice is to keep your clusters upgraded so that they're always running versions of Kubernetes that are currently supported by OKE. See the instructions in the following procedures to determine whether a newer supported version of Kubernetes is available.

> **Note:**
>
> If you set or modify any of the following tags, the new values are ignored: SSH key (OraclePCA.sshkey), number of nodes (OraclePCA.cpNodeCount), node shape (OraclePCA.cpNodeShape), or node shape configuration (OraclePCA.cpNodeShapeConfig). These values can be set only when you create the cluster.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Containers**, then select **Kubernetes Clusters (OKE)**.
2. In the clusters list, if a Kubernetes version update is available, an exclamation point icon is displayed next to the Kubernetes Version number. To upgrade to a newer version, select the Actions menu (⋮) and select **Upgrade Available**. Select a new version from the drop-down menu.

   Alternatively, on the cluster details page, select **Upgrade Available** next to the Kubernetes Version number, or select **Upgrade** at the top of the page.
3. Select the name of the cluster that you want to update.
4. At the top of the cluster details page, select **Edit**.

   Don't specify values for the OraclePCA-OKE.cluster_id defined tag or for the ClusterResourceIdentifier free-form tag. These tag values are system-generated and only applied to nodes (instances), not to the cluster resource.
5. When you're finished making changes, select **Save Changes**.

Using the CLI

Use the oci ce cluster update command and required parameters to update the details of a cluster.

```
oci ce cluster update --cluster-id cluster_OCID [OPTIONS]
```

1. Get the OCID of the cluster that you want to update: `oci ce cluster list`
2. Check whether a newer version of Kubernetes is available.

   Run the get cluster command: `oci ce cluster get`

   If the value of `available-kubernetes-upgrades` isn't the empty set, specify one of the listed versions as the `--kubernetes-version` in the update cluster command.
3. Run the update cluster command.

   If you specify the `--defined-tags` or `--freeform-tags` options, don't specify values for the OraclePCA-OKE.cluster_id defined tag or for the ClusterResourceIdentifier free-form tag. These tag values are system-generated and only applied to nodes (instances), not to the cluster resource.

   Example:

   ```
   $ oci ce cluster update --cluster-id ocid1.cluster.unique_ID \
   --kubernetes-version newer_kubernetes_version --name new_cluster_name
   ```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateCluster operation to update the details of a cluster.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting an OKE Cluster

Learn how to delete OKE clusters On Private Cloud Appliance.

On Private Cloud Appliance, deleting an OKE cluster deletes the cluster control plane nodes, worker nodes, and node pools. Other cluster resources such as VCNs, internet gateways, NAT gateways, route tables, security lists, load balancers, and block volumes aren't deleted when you delete the cluster. Those resources must be deleted separately.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Containers**, then select **Kubernetes Clusters (OKE)**.
2. For the cluster that you want to delete, select the Actions menu (⋮), and select **Delete**.
3. Confirm that you want to delete the cluster.

   Enter the cluster name, and select **Delete**.

Using the CLI

Use the oci ce cluster delete command and required parameters to delete a cluster.

```
oci ce cluster delete --cluster-id cluster_OCID [OPTIONS]
```

Procedure

1. Get the OCID of the cluster that you want to delete: `oci ce cluster list`
2. Run the delete cluster command.

   Example:

   ```
   $ oci ce cluster delete --cluster-id ocid1.cluster.unique_ID --force
   ```

The `--force` option performs the deletion without prompting for confirmation.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteCluster operation to delete a cluster.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Managing OKE Cluster Add-ons

On Private Cloud Appliance, cluster add-ons are components that you can choose to deploy on a Kubernetes cluster. Cluster add-ons extend core Kubernetes functionality and improve cluster manageability and performance. This section describes how to install and manage the following supported add-ons.

- The WebLogic Kubernetes Operator add-on, which supports running WebLogic Server and Fusion Middleware appliance domains on Kubernetes. For more information about the WebLogic Kubernetes Operator, see the public documentation at https://github.com/oracle/weblogic-kubernetes-operator.
- The Database Operator for Kubernetes add-on (OraOperator) helps developers, database administrators, DevOps, and GitOps teams reduce the time and complexity of deploying and managing Oracle databases. For more information, see the public documentation at https://github.com/oracle/oracle-database-operator/tree/main. For guidance on how to deploy and configure the add-on, we recommend this solution paper: Deploying Oracle Database Operator Cluster Add-on for Oracle Kubernetes Engine (OKE).
- The NVIDIA GPU Plugin add-on is a convenient way to manage the NVIDIA Device Plugin for Kubernetes. The NVIDIA Device Plugin for Kubernetes is the NVIDIA implementation of the Kubernetes device plug-in framework for exposing the number of NVIDIA GPUs on each worker node, and tracking the health of those GPUs. For more information about the NVIDIA Device Plugin for Kubernetes, see https://github.com/NVIDIA/k8s-device-plugin.
- The optional Certificate Manager add-on, also known as cert-manager, adds certificates and certificate issuers to Kubernetes clusters as resource types. The Certificate Manager also simplifies the process of obtaining, using, and renewing those certificates. For more information, see https://github.com/cert-manager/cert-manager.

Add-on Prerequisites

Review the following requirements before you install the add-ons.

Oracle Database Operator add-on

- Database Operator can only be enabled on an existing cluster.
- The Certificate Manager add-on must be installed, enabled, and in the ACTIVE state before you can use the Database Operator add-on. See Install an Add-on for an Existing Cluster on page 917.

Nvidia GPU Plugin add-on

- For both the cluster and node pool, Kubernetes version 1.29.14 or higher is required
- Enable the Nvidia GPU plugin *before* creating a GPU node pool.
- Once a node pool is created as either GPU or non-GPU, its type cannot be switched.
- Use the appropriate required base image for your use case.

Base Image	Use Case
`nvcr.io/nvidia/cuda:12.9.0-runtime-ubi9` or equal as a runtime environment	Deploying of a prebuilt AI application

Base Image	Use Case
`nvcr.io/nvidia/cuda:12.9.0-devel-ubi9` or equal as a development environment	Developing, compiling, or training AI with native CUDA/C++ code.

- All worker nodes come with default block volume size of 50 GB however, GPU applications when used with the required base image might run out of disk space. Ensure to use Persistent Volume Claim in the GPU applications.
- To avoid unintended scheduling, the GPU nodes are tainted by default. Without this taint any pod could be scheduled onto a GPU node, even if it doesn't need a GPU. This means only pods that explicitly request GPU resources and tolerate the taint are be scheduled on GPU nodes.
- To ensure that the Nvidia device-plugin pods are scheduled and run only on nodes that possess NVIDIA GPU hardware, GPU nodes are labeled by default. The OKE controller labels GPU nodes with the following node label `"nvidia.com/gpu": "true"`.

WebLogic Kubernetes Operator add-on

See Deploying the WebLogic Kubernetes Operator Add-on on page 920.

Install an Add-on When You Create a Cluster

To install an add-on when you create a cluster, you must use the Compute Web UI.

Review the Add-on Prerequisites on page 916 before you begin.

Compute Web UI

1. In the Compute Web UI navigation menu, select **Containers**, then select **Kubernetes Clusters (OKE)**.
2. On the clusters list page, select the **Create Cluster** button.
3. On the bottom of the first page of the **Create Cluster** dialog box, the **Add-ons** section shows the available cluster add-ons. In the **Create Cluster** dialog box, all add-ons are **Disabled**.

   > **Note:**
   >
   > Oracle Database Operator add-on can only be enabled on an existing cluster.

4. Select the add-on you want to deploy and configure.
   a. **Enable**: Select the checkbox for the add-on you want to deploy and enable for this cluster.
   b. **Add-on version updates**: Select how you want the version of the add-on to be updated as newer versions of the add-on become available and as newer versions of Kubernetes are supported for OKE. Select either **Automatic Updates** or select a **Version**. See descriptions of these options in Version Updates for Add-ons on page 925.

      If you select **Choose a Version**, then you must select a version from the list.
   c. **Configurations**: Select **Add configuration** to select a configuration option and specify a value.

      Select **Add configuration** to set another configuration parameter.

Install an Add-on for an Existing Cluster

Before you install the WebLogic Kubernetes Operator add-on on an existing cluster, check the cluster's certificate authority.

Outside Certificates

To install the WebLogic Kubernetes Operator add-on on an existing cluster that is using a certificate that's not the certificate that's specific to the Private Cloud Appliance, perform the following steps on the cluster where you want to install the add-on:

1. Perform certificate rotation. See Updating the Certificate Authority Bundle on page 830.

2. Perform any updates to node pool configuration that are required, such as boot volume size change or shape changes, for example.
3. Cycle worker nodes as described in Node Cycling an OKE Node Pool on page 943.
4. Enable or install the WebLogic Kubernetes Operator add-on as described in this procedure.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Containers**, then select **Kubernetes Clusters (OKE)**.
2. In the clusters list, select the name of the cluster in which you want to install the add-on.
3. On the cluster details page, under **Resources**, select **Add-ons**.
4. In the add-ons list, for the **WLS Operator** add-on, select the Actions menu (⋮), and select **Edit**. On the **WLS Operator** dialog box, select the **Enable Add-on WLS Operator** checkbox to do one of the following:
   - Deploy and enable the WebLogic Kubernetes Operator add-on if the add-on hasn't been enabled on this cluster before.
   - Enable the WebLogic Kubernetes Operator add-on if the add-on was previously deployed for this cluster but is currently disabled.
5. Configure the add-on.
   a. **Add-on version updates**: Select the method you want to use to update the version of the add-on as newer versions of the add-on become available and as newer versions of Kubernetes are supported for OKE: either **Automatic Updates** or **Choose a Version**. See descriptions of these options in Version Updates for Add-ons on page 925.

      If you select **Choose a Version**, then you must select a version from the list.
   b. **Add-on configuration**: Select **Add configuration** to select a configuration option and specify a value. See the descriptions in Configuration Parameters for Add-ons on page 928.

      To set another configuration parameter, select **Add configuration**.
6. Select **Save Changes**.

Using the CLI

Use the oci ce cluster install-addon command and required parameters to install the add-on.

```
oci ce cluster install-addon --cluster-id cluster_OCID --addon-
name addon_name [OPTIONS]
```

Procedure

1. Get the OCID of the cluster for which you want to install an add-on: `oci ce cluster list`
2. (Optional) Construct an argument for the `--configurations` option.

   Use the `--configurations` option to specify one or more key/value pairs in JSON format to pass as arguments to the cluster add-on.

   For descriptions of the configuration parameters, see Configuration Parameters for Add-ons on page 928.

   The inline syntax is shown in the example in the next step of this procedure. You might find it easier to use a file:

   ```
   --configurations file://./weblogic-cfg.json
   ```

   The format and content of the configuration file is given by the following command:

   ```
   $ oci ce cluster install-addon --generate-param-json-input configurations
   [
     {
       "key": "string",
       "value": "string"
     },
     {
   ```

```
    "key": "string",
    "value": "string"
  }
]
```

In the following example, both `requests` and `limits` are specified because the memory limit is lower than the default memory request. If a limit is less than the corresponding request, the deployment fails.

Double quotation marks within a value must be escaped with a single backslash.

```
[
    {
        "key": "weblogic-operator.ContainerResources",
        "value": "{
            \"requests\": {
                \"cpu\": \"250m\",
                \"memory\": \"150Mi\"
            },
            \"limits\": {
                \"cpu\": \"500m\",
                \"memory\": \"200Mi\"
            }
        }"
    },
    {
        "key": "weblogic-operator-webhook.ContainerResources",
        "value": "{
            \"limits\": {
                \"cpu\": \"150m\",
                \"memory\": \"200Mi\"
            }
        }"
    },
    {
        "key": "numOfReplicas",
        "value": "1"
    }
]
```

3. Run the install add-on command.

   Syntax:

   ```
   $ oci ce cluster install-addon --cluster-id cluster_OCID \
   --addon-name addon_name
   ```

   Example:

   If you specify a version, you're selecting the `Stay on the specific version` option for updating the add-on version, described in Version Updates for Add-ons on page 925. If you set the version to null, or you omit the `--version-parameterconflict` option, you're selecting the default behavior `Automatically update the add-on`.

   Note that the version string must begin with a `v`.

   Enclose the configurations argument in single quotation marks so that you don't need to escape every double quotation mark in the argument value.

   ```
   $ oci ce cluster install-addon --cluster-id ocid1.cluster.unique_ID \
   --addon-name WeblogicKubernetesOperator --version-parameterconflict
    "v4.2.13" \
   --configurations '[{"key": "weblogic-operator.ContainerResources",
    "value": "{\"limits\": {\"cpu\": \"500m\", \"memory\": \"512Mi\"}}"}, \
   ```

```
{"key": "weblogic-operator-webhook.ContainerResources", "value":
 "{\"limits\": {\"cpu\": \"150m\", \"memory\": \"200Mi\"}}"}]'

{
 "opc-work-request-id": "ocid1.cccworkrequest.unique_ID"
}
```

Using the API

Use the InstallAddon operation to install an add-on.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deploying the WebLogic Kubernetes Operator Add-on

On Private Cloud Appliance, you can enable the WebLogic Kubernetes Operator add-on when you create a cluster or for an existing cluster.

> **Note:**
>
> To bring the WebLogic Server to the running state, create additional rules in separate WebLogic Server security lists for the control plane and worker subnets, and for the pod subnet if you're using VCN-Native Pod Networking. See Specify Ports Required by WebLogic Server on page 921.

Add-on installation remains in Accepted state and waits until the cluster is in the Active state.

After the cluster is in the Active state, the WebLogic Kubernetes Operator is in Needs Attention state until a node pool is created for the cluster.

When a node-pool has been created for the cluster, the add-on is reconciled, and the add-on is in Ready state unless some other problem exists. See Add-on Reconciliation on page 925.

> **Note:**
>
> Enabling the WebLogic Kubernetes Operator add-on on a VCN-Native Pod Networking cluster requires an entry for 169.254.169.254 in `crio-noproxy` node metadata for the nodepools where the add-on pods might be scheduled. See "Proxy settings" in the OCI CLI procedure in Creating an OKE Worker Node Pool on page 932.

> **Note:**
>
> The WebLogic Kubernetes Operator add-on is not supported on GPU-only node pools.

- Step 1: Check the Cluster's Certificate Authority on page 920
- Step 2: Install the Add-on on page 921
- Step 3: Specify Ports Required by WebLogic Server on page 921
- Step 4: Add-on Reconciliation on page 925

Before you install the WebLogic Kubernetes Operator add-on on an existing cluster, check the cluster's certificate authority.

Check the Cluster's Certificate Authority

To install the WebLogic Kubernetes Operator add-on on an existing cluster that is using a certificate that's not the certificate that's specific to the Private Cloud Appliance, perform the following steps on the cluster where you want to install the add-on:

1. Perform certificate rotation. See Updating the Certificate Authority Bundle on page 830.

2. Perform any updates to node pool configuration that are required, such as boot volume size change or shape changes, for example.
3. Cycle worker nodes as described in Node Cycling an OKE Node Pool on page 943.
4. Enable or install the WebLogic Kubernetes Operator add-on as described in this procedure.

Install the Add-on

See Install an Add-on When You Create a Cluster on page 917 or Install an Add-on for an Existing Cluster on page 917.

Specify Ports Required by WebLogic Server

This section describes additional security rules needed to specify ports that are required to bring the WebLogic Server to the running state.

Create additional rules in separate WebLogic Server security lists for the control plane and worker subnets, and for the pod subnet if you are using VCN-Native Pod Networking.

The following rules are for the control plane subnet. These rules are used for both Flannel Overlay networking and VCN-Native Pod Networking.

WebLogic Server Security Rules for the Control Plane Subnet

Compute Web UI property	OCI CLI property

Ingress Rule 1
• Stateless: uncheck the box
• Ingress CIDR: ***kmi_cidr***
• IP Protocol: TCP
  • Destination Port Range: 8084
• Description: "This service port is the default for the WebLogic Server Console and is used to manage WebLogic Server domains."
	Ingress Rule 1
	• `isStateless`: `false`
	• `source`: ***kmi_cidr***
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `8084`
	  • `min`: `8084`
	• `description`: "This service port is the default for the WebLogic Server Console and is used to manage WebLogic Server domains."

Ingress Rule 2
• Stateless: uncheck the box
• Ingress CIDR: ***worker_cidr***
• IP Protocol: UDP
  • Destination Port Range: 8472
• Description: "WebLogic Server administration."
	Ingress Rule 2
	• `isStateless`: `false`
	• `source`: ***worker_cidr***
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `17`
	• `udpOptions`
	`destinationPortRange`
	  • `max`: `8472`
	  • `min`: `8472`
	• `description`: "WebLogic Server administration."

The following rules are for the worker subnet. These rules are used for both Flannel Overlay networking and VCN-Native Pod Networking.

WebLogic Server Security Rules for the Worker Subnet

Compute Web UI property	OCI CLI property

Ingress Rule 1
• Stateless: uncheck the box
• Ingress CIDR: ***kmi_cidr***
• IP Protocol: TCP
  • Destination Port Range: 8084
• Description: "This service port is the default for the WebLogic Server Console and is used to manage WebLogic Server domains."
	Ingress Rule 1
	• `isStateless`: `false`
	• `source`: ***kmi_cidr***
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `8084`
	  • `min`: `8084`
	• `description`: "This service port is the default for the WebLogic Server Console and is used to manage WebLogic Server domains."

Ingress Rule 2
• Stateless: uncheck the box
• Ingress CIDR: ***kmi_cidr***
• IP Protocol: UDP
  • Destination Port Range: 8472
• Description: "WebLogic Server administration."
	Ingress Rule 2
	• `isStateless`: `false`
	• `source`: ***kmi_cidr***
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `17`
	• `udpOptions`
	`destinationPortRange`
	  • `max`: `8472`
	  • `min`: `8472`
	• `description`: "WebLogic Server administration."

Ingress Rule 3
• Stateless: uncheck the box
• Ingress CIDR: ***worker_cidr***
• IP Protocol: UDP
  • Destination Port Range: 7001-9000
• Description: "These ports are used by WebLogic Server."
	Ingress Rule 3
	• `isStateless`: `false`
	• `source`: ***worker_cidr***
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `17`
	• `udpOptions`
	`destinationPortRange`
	  • `max`: `9000`
	  • `min`: `7001`
	• `description`: "These ports are used by WebLogic Server."

The following rules are for the pod subnet. These rules are used for VCN-Native Pod Networking.

WebLogic Server Security Rules for the Pod Subnet

Compute Web UI property	OCI CLI property

Ingress Rule 1
• Stateless: uncheck the box
• Ingress CIDR: ***kmi_cidr***
• IP Protocol: TCP
  • Destination Port Range: 8084
• Description: "This service port is the default for the WebLogic Server Console and is used to manage WebLogic Server domains."
	Ingress Rule 1
	• `isStateless`: `false`
	• `source`: ***kmi_cidr***
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `6`
	• `tcpOptions`
	`destinationPortRange`
	  • `max`: `8084`
	  • `min`: `8084`
	• `description`: "This service port is the default for the WebLogic Server Console and is used to manage WebLogic Server domains."

Ingress Rule 2
• Stateless: uncheck the box
• Ingress CIDR: ***worker_cidr***
• IP Protocol: UDP
  • Destination Port Range: 8472
• Description: "WebLogic Server administration."
	Ingress Rule 2
	• `isStateless`: `false`
	• `source`: ***worker_cidr***
	• `sourceType`: `CIDR_BLOCK`
	• `protocol`: `17`
	• `udpOptions`
	`destinationPortRange`
	  • `max`: `8472`
	  • `min`: `8472`
	• `description`: "WebLogic Server administration."

Viewing OKE Cluster Add-ons

On Private Cloud Appliance, you can list all add-ons for a cluster and view more information about a specified cluster add-on.

To list versions of an add-on that are supported on a specific version of Kubernetes, use the OCI CLI, or use the **Edit Add-on** option in the Compute Web UI.

Using the Compute Web UI

1. On the dashboard, select **Containers / View Kubernetes Clusters (OKE)**.
2. In the clusters list, select the name of the cluster for which you want to view add-ons.
3. On the cluster details page, under **Resources**, select **Add-ons**.

   The list table shows the name of the cluster add-on, whether automatic updates are enabled, the lifecycle state of the add-on (for example, Active, Updating, Disabled), and the current version of the add-on.
4. To show more information, select the Actions menu (⋮), and select **Edit Add-on**.

   On the dialog box, you can view the current add-on configuration and change the configuration. You can enable the add-on, select either **Automatically Update** or **Choose a version**, and specify configuration parameters.

Using the CLI

Use the oci ce cluster list-addons command and required parameters to view add-ons.

```
oci ce cluster list-addons --cluster-id <cluster_OCID>
```

Use the oci ce cluster get-addon command and required parameters to view details for an add-on.

```
oci ce cluster get-addon --cluster-id <cluster_OCID> --addon-
name <addon_name>
```

Procedure

1. Get the OCID of the cluster for which you want to view add-ons: `oci ce cluster list`
2. List all add-ons in the specified cluster.

   ```
   $ oci ce cluster list-addons --cluster-id ocid1.cluster.unique_ID
   ```

   For each add-on that is available to the specified cluster, the output shows the name of the add-on, the current installed version, and the lifecycle state.
3. For more information about a specific add-on, run the get add-on command.

   The output is the same as for the `list-addons` command except that `get-addon` also shows the configuration values.

   If the add-on lifecycle state is NEEDS_ATTENTION, see Add-on Reconciliation on page 925.

   Example:

   ```
   $ oci ce cluster get-addon --cluster-id ocid1.cluster.unique_ID \
   --addon-name WeblogicKubernetesOperator
   {
     "data": {
       "addon-error": {
           "code": null,
           "message": null,
           "status": null
       },
       "configurations": [
         {
           "key": "numOfReplicas",
           "value": "0"
         },
         {
           "key": "weblogic-operator.ContainerResources",
           "value": "{'limits': {'cpu': '500m', 'memory': '200Mi'}}"
         },
         {
           "key": "weblogic-operator-webhook.ContainerResources",
           "value": "{'limits': {'cpu': '200m', 'memory': '300Mi'}}"
         }
       ],
       "current-installed-version": "v4.2.13",
       "lifecycle-state": "ACTIVE",
       "name": "WeblogicKubernetesOperator",
       "time-created": "2025-02-26T01:41:52.020696+00:00",
       "version": null
     },
     "etag": "5f3eef22-eb32-5c2c-774c-c7a98836a13a"
   }
   ```

Using the API

Use the ListAddons operation to list add-ons.

Use the GetAddon operation to get add-on details.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Version Updates for Add-ons

When you enable a cluster add-on On Private Cloud Appliance, specify how you want to update the add-on version when new versions become available.

(Default) Automatic Updates

If possible, this option automatically updates the add-on when new versions become available. This is the default behavior.

The newest version of the add-on that supports the Kubernetes version that is specified for the cluster is deployed when you install the add-on. When a newer version of the add-on is released, the add-on is automatically updated if the new add-on version is compatible with the versions of Kubernetes that are supported by OKE at that time and the version of Kubernetes that the cluster is running.

Best practice is to keep your clusters upgraded so that they are always running versions of Kubernetes that are listed as currently supported by OKE. See *Supported Versions of Kuberrnetes* in Kubernetes Engine (OKE) On Private Cloud Appliance on page 821 and Updating an OKE Cluster on page 914.

Choose a Version

This option keeps the add-on on the specific version that you select until you change it.

If you specify that you want to choose the version of the add-on to deploy, the version you choose is enabled. Ensure that the add-on version is compatible with the Kubernetes version that you have selected for the cluster or that is already running on the cluster.

When you use the Compute Web UI, you select the version from a list. All versions on the list are compatible with the Kubernetes version that you have selected for the cluster or that is already running on the cluster.

When you use the OCI CLI, use the following `addon-option list` commands to get the information you need before you run the `cluster install-addon` command.

List available versions of all cluster add-ons that are supported on the specified Kubernetes version.

```
$ oci ce addon-option list --kubernetes-version v1.29.9
```

List available versions of the specified cluster add-on that are supported on the specified Kubernetes version.

```
$ oci ce addon-option list --kubernetes-version v1.29.9 --addon-name
 WeblogicKubernetesOperator
```

Add-on Reconciliation

On Private Cloud Appliance, the OKE service includes a reconciliation process that periodically evaluates the state of the add-on and updates the add-on if necessary.

> **Note:**
>
> You should not install, configure, update, or delete add-ons manually. Use the OKE service installation, configuration, update, and delete interfaces.

The reconciliation process behaves differently depending on the state of the add-on.

Active State

For add-ons in ACTIVE state, the reconciliation process runs every twelve hours. If resources have been manually deleted, the process detects the change and attempts recovery. Full recovery is not guaranteed.

If full recovery is successful, the add-on state returns to ACTIVE.

If recovery is partial or fails, the add-on state changes to NEEDS_ATTENTION.

Needs Attention State

For add-ons in NEEDS_ATTENTION state, the reconciliation process runs every few minutes, not every twelve hours, while the add-on remains in the NEEDS_ATTENTION state. The process checks whether all deployments associated with the add-on are ready, whether all pods are healthy. The interval between reconciliation process runs is a little longer each time.

Some issues, such as unschedulable nodes, might resolve during the reconciliation process. Other issues, such as configuration problems, require user intervention to fix. If the add-on remains in NEEDS_ATTENTION state after the reconciliation process has run, try to identify the issues. Check the K8s_app application in Grafana or check the state of the add-on operator manually.

Recovery actions that might be appropriate for a user to take include the following:

- Ensure that node pools in the cluster have at least one node available. If no nodes are available, the add-on pods can't be scheduled, the add-on can't be deployed.
- Ensure the configuration values and other settings are correct.
- Update the add-on as needed. Updating the add-on will trigger another reconciliation process.
- Disable and reinstall the add-on.

If the add-on recovers, the add-on is transitioned back to ACTIVE state.

If the add-on remains unhealthy, the system schedules the next check.

The reconciliation process continues to run for approximately 12.5 hours if the add-on remains in NEEDS_ATTENTION state. After the reconciliation process stops running, the add-on remains in NEEDS_ATTENTION state indefinitely. After approximately 30 minutes, the work request moves to FAILED state.

The following example shows an add-on in NEEDS_ATTENTION state.

```
$ oci ce cluster get-addon --cluster-id ocid1.cluster.unique_ID --addon-name
 WeblogicKubernetesOperator
{
  "data": {
    "addon-error": {
      "code": "409",
      "message": "Incorrect state for CR",
      "status": "IncorrectState"
    },
    "configurations": [
      {
        "key": "numOfReplicas",
        "value": "0"
      },
      {
        "key": "weblogic-operator.ContainerResources",
        "value": "{'limits': {'cpu': '500m', 'memory': '200Mi'}}"
      },
      {
        "key": "weblogic-operator-webhook.ContainerResources",
        "value": "{'limits': {'cpu': '200m', 'memory': '300Mi'}}"
      }
    ],
    "current-installed-version": "",
    "lifecycle-state": "NEEDS_ATTENTION",
```

```
    "name": "WeblogicKubernetesOperator",
    "time-created": "2025-02-26T01:41:52.020696+00:00",
    "version": null
  },
  "etag": "5f3eef22-eb32-5c2c-774c-c7a98836a13a"
}
```

Updating Cluster Add-ons

On Private Cloud Appliance, you can configure the method for updating the cluster add-on version, and you can update the number of replicas and the resource usage limits.

The add-on update remains in Accepted state and waits until the cluster is in the Active state.

Using the Compute Web UI

1. On the dashboard, select **Containers / View Kubernetes Clusters (OKE)**.
2. In the clusters list, select the name of the cluster for which you want to update an add-on.
3. On the cluster details page, Under **Resources**, select **Add-ons**.
4. In the add-ons list, locate the add-on you want to modify, select the Actions menu (⋮), and select **Edit Add-on**.
5. In the dialog box that opens, configure the add-on:
   a. **Add-on version updates**: Select how you want the version of the add-on to be updated as newer versions of the add-on become available and as newer versions of Kubernetes are supported for OKE: either **Automatic Updates** or **Choose a Version**. See descriptions of these options in Version Updates for Add-ons on page 925.

      If you select **Choose a Version**, you must select a version from the list.
   b. **Add-on configuration**: Optionally select a configuration parameter and a value for that parameter.

      To set another configuration parameter, select the Add configuration button.
6. Select the Save Changes button in the dialog.

Using the CLI

Use the oci ce cluster update-addon command and required parameters to update addon details for a cluster.

```
oci ce cluster update-addon --addon-name <addon_name> --cluster-
id <cluster_OCID> [OPTIONS]
```

Procedure

1. Get the OCID of the cluster for which you want to update an add-on: `oci ce cluster list`
2. Run the update add-on command.

   Example:

   ```
   $ oci ce cluster update-addon --cluster-id ocid1.cluster.unique_ID \
   --addon-name WeblogicKubernetesOperator \
   --configurations file://./weblogic-cfg.json --force

   {
    "opc-work-request-id": "ocid1.cccworkrequest.unique_ID"
   }
   ```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateAddon operation to update addon details for a cluster.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Disabling and Removing OKE Cluster Add-ons

On Private Cloud Appliance, when you disable an add-on, it's disabled and deleted from the cluster. If you later enable the add-on, the add-on is reinstalled.

The add-on must be in the Active state to be deleted.

Using the Compute Web UI

1. On the dashboard, select **Containers / View Kubernetes Clusters (OKE)**.
2. In the clusters list, select the name of the cluster for which you want to delete the add-on.
3. On the cluster details page, under **Resources**, select **Add-ons**.
4. In the add-ons list, select the Actions menu (⋮), next to the appropriate add-on, then select **Edit**.
5. In the edit dialog, clear **Enable Add-on** to disable and remove the selected add-on for this cluster.
6. Select **Save Changes**.

Using the CLI

Use the oci ce cluster disable-addon command and required parameters to disable and remove an add-on.

```
oci ce cluster disable-addon --addon-name <addon_name> --cluster-
id <cluster_OCID> [OPTIONS]
```

Procedure

1. Get the OCID of the cluster that you want to delete: `oci ce cluster list`
2. Run the disable add-on command.

   Example:

   ```
   $ oci ce cluster disable-addon --cluster-id ocid1.cluster.unique_ID --
   addon-name WeblogicKubernetesOperator
   ```

   The add-on is disabled and removed from the cluster:

   Note that the `--is-remove-existing-add-on` option is ignored.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DisableAddon operation to disable and remove an add-on.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Configuration Parameters for Add-ons

This topic describes the configuration parameters that are available On Private Cloud Appliance for cluster add-ons.

WebLogic Kubernetes Operator add-on parameters

When you enable the WebLogic Kubernetes Operator cluster add-on, you can pass the following key/value pairs as arguments.

> **Note:**
>
> For weblogic-operator container resources and weblogic-operator-webhook container resources, if you set a limit without specifying a request, and the limit is less than the default request, the deployment fails.

Use the values of the weblogic-operator container resources parameter and the weblogic-operator-webhook container resources parameter to determine the maximum number of replicas you can specify. The values of these parameters

could be the default values shown in the table below or different values that you requested when you enabled the WebLogic Kubernetes Operator.

Parameter Name **Compute Web UI** **OCI CLI**	**Description**
numOfReplicas `numOfReplicas`	(Required) The integer number of replicas of the add-on deployment.
weblogic-operator container resources `weblogic-operator.ContainerResources`	(Optional) These are resource values for the main WebLogic Operator container. The resource quantities that the add-on containers request, and the resource usage limits that the add-on containers cannot exceed. See Resource Management for Pods and Containers in the Kubernetes documentation. If you do not specify a request, the default request values are: • cpu: 250m • memory: 512Mi If you set a usage limit, you must set a limit equal to or greater than these default resource request values or the request values that you specified.
weblogic-operator-webhook container resources `weblogic-operator-webhook.ContainerResources`	(Optional) These are resource values for the webhook container used by the operator. The resource quantities that the add-on containers request, and the resource usage limits that the add-on containers cannot exceed. If you do not specify a request, the default request values are: • cpu: 100m • memory: 100Mi If you set a usage limit, you must set a limit equal to or greater than these default resource request values or the request values that you specified.

Example

A worker node with 6 Gb RAM and 2 OCPUs could accommodate 12 pods if you don't count other cluster-related or custom pods. Each WebLogic Server pod could require 250m/768Mi (cpu/memory), and Flannel Overlay pods could require 100m/50Mi. Best practice is not to exceed 70% of CPU/memory usage per node. Considering only weblogic-operator pods, 8 would be ideal to leave room for system daemons or fluctuating workloads.

Because you also must allocate weblogic-operator-webhook pods, a better configuration would be to schedule a maximum of 6 replicas per weblogic-operator pod per node, leaving room for a maximum of 8 weblogic-operator-webhook pods,

In this example (a worker node with 6 Gb RAM and 2 OCPUs), 6 replicas per node (12 for a 2 worker node cluster) is the best configuration.

This calculation of maximum number of replicas varies for each case, depending on the WebLogic requirements, the size of the node pools, and the shape configuration (CPU and memory) for each node pool.

Oracle Database Operator

When you enable the Oracle Database Operator for Kubernetes cluster add-on, you can pass the following key/value pairs as arguments.

Parameter Name **Compute Web UI** **OCI CLI**	**Description**
numOfReplicas `numOfReplicas`	(Required) The integer number of replicas of the add-on deployment.
manager container resources `manager.ContainerResources`	(Optional) You can specify the resource quantities that the add-on containers request, and set resource usage limits that the add-on containers cannot exceed. Use JSON format in plain text or Base64 encoded. If you do not specify a request, the default request values are: **Limits** • cpu: 400m • memory: 400Mi **Requests** • cpu: 400m • memory: 400Mi Example: Create add-on containers that request 100 milllicores of CPU, and 100 mebibytes of memory. Limit add-on containers to 500 milllicores of CPU, and 200 mebibytes of memory. {"limits": {"cpu": "500m", "memory": "200Mi" }, "requests": {"cpu": "100m", "memory": "100Mi"}}

Nvidia GPU Plugin add-on parameters

When you enable the Nvidia GPU Plugin cluster add-on, you can pass the following key/value pairs as arguments.

Parameter Name Compute Web UI OCI CLI	Description

nvidia-gpu-device-plugin container resources
`nvidia-gpu-device-plugin container resources`
	You can specify the resource quantities that the add-on containers request, and set resource usage limits that the add-on containers cannot exceed.
	`{"limits": {"cpu": "500m", "memory": "200Mi" },`
	`"requests": {"cpu": "100m", "memory": "100Mi"}}`
	Create add-on containers that request 100 milllicores of CPU, and 100 mebibytes of memory. Limit add-on containers to 500 milllicores of CPU, and 200 mebibytes of memory.

Certificate Manager add-on parameters

When you enable the Certificate Manager cluster add-on, you can pass the following key/value pairs as arguments.

Parameter Name Compute Web UI OCI CLI	Description
numOfReplicas `numOfReplicas`	(Required) The integer number of replicas of the add-on deployment.

cert-manager-controller container resources
`cert-manager-controller.ContainerResources`
	(Optional) You can specify the resource quantities that the add-on containers request, and set resource usage limits that the add-on containers cannot exceed.
	`{"limits": {"cpu": "500m", "memory": "200Mi" },`
	`"requests": {"cpu": "100m", "memory": "100Mi"}}`
	Create add-on containers that request 100 milllicores of CPU, and 100 mebibytes of memory. Limit add-on containers to 500 milllicores of CPU, and 200 mebibytes of memory.

cert-manager-cainjector container resources
`cert-manager-cainjector.ContainerResources`
	(Optional) You can specify the resource quantities that the add-on containers request, and set resource usage limits that the add-on containers cannot exceed.
	`{"limits": {"cpu": "500m", "memory": "200Mi" },`
	`"requests": {"cpu": "100m", "memory": "100Mi"}}`
	Create add-on containers that request 100 milllicores of CPU, and 100 mebibytes of memory. Limit add-on containers to 500 milllicores of CPU, and 200 mebibytes of memory.

Parameter Name Compute Web UI OCI CLI	Description

weblogic-operator-webhook container resources
`weblogic-operator-webhook.ContainerResources`
	(Optional) You can specify the resource quantities that the add-on containers request, and set resource usage limits that the add-on containers cannot exceed.
	`{"limits": {"cpu": "500m", "memory": "200Mi" },`
	`"requests": {"cpu": "100m", "memory": "100Mi"}}`
	Create add-on containers that request 100 milllicores of CPU, and 100 mebibytes of memory. Limit add-on containers to 500 milllicores of CPU, and 200 mebibytes of memory.

Creating and Managing OKE Worker Node Pools

Learn how to create, update, and delete node pools for an OKE cluster On Private Cloud Appliance. Be sure to carefully read the descriptions of the node pool parameters before you create the node pool.

The following topics describe how to manage Kubernetes worker node pools in Private Cloud Appliance:

- Creating an OKE Worker Node Pool on page 932
- Updating an OKE Node Pool on page 939
- Deleting an OKE Node Pool Node on page 945
- Deleting an OKE Node Pool on page 946

This topic also describes how to recognize node pool nodes in a list of all instances in a tenancy, and how to delete a single node from a node pool.

Creating an OKE Worker Node Pool

Learn how to create OKE worker node pools On Private Cloud Appliance for a workload cluster.

Nodes are Private Cloud Appliance compute instances. When you create a worker node pool, you specify the number of nodes to create and other parameters that define instances.

> **Note:**
>
> You can't customize the OKE `cloud-init` scripts.

To configure proxy settings, use the OCI CLI or API to set the proxy in node metadata. If the cluster is using VCN-Native Pod Networking, add 169.254.169.254 to the noproxy setting.

Custom CA support is available only by using the CLI (`oci ce node-pool create` and `oci ce node-pool update`).

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Containers**, then select **Kubernetes Clusters (OKE)**.

   If the cluster to which you want to attach a node pool is not listed, select a different compartment from the compartment menu at the top of the page.
2. Select the name of the cluster to which you want to add a node pool.
3. On the cluster details page, scroll to the **Resources** section and select **Node Pools**.
4. On the **Node Pools** list, select the **Add Node Pool** button.
5. In the **Add Node Pool** dialog box, provide the following information:

   - **Name**: The name of the new node pool. Avoid entering confidential information.

- **Compartment**: The compartment in which to create the new node pool.
- **Node Pool Options**: In the **Node Count** field, enter the number of nodes you want in this node pool. The default is 0. The maximum number is 128 per cluster, which can be distributed across multiple node pools.
- **Network Security Group**: If you check the box to enable network security groups, select the **Add Network Security Group** and select an NSG from the drop-down list. You might need to change the compartment to find the NSG you want. The primary VNIC from the worker subnet will be attached to this NSG.
- **Placement configuration**
  - **Worker Node Subnet**: Select a subnet that has configuration such as the "worker" subnet described in Creating a Worker Subnet (Flannel Overlay) on page 852 or Creating a Worker Subnet (VCN-Native Pod) on page 887. For a public cluster, create the NAT private version of the "worker" subnet. For a private cluster, create the VCN-only private version of the "worker" subnet. Select only one subnet. The subnet must have rules set to communicate with the control plane endpoint. The subnet must use a private route table and must have a security list like the worker-seclist security list described in Creating a Worker Subnet (Flannel Overlay) on page 852 or Creating a Worker Subnet (VCN-Native Pod) on page 887.
  - **Fault Domain**: Select a fault domain, or select **Automatically select the best fault domain**, which is the default option. A Roving Edge device has only 1 fault domain.
- **Source Image**: Select an image.

  a. Select the **Platform Image Source Type**.

  b. Select an image from the list.

  The image list has columns Operating System, OS Version, and Kubernetes Version. You can use the drop-down menu arrow to the right of the OS Version or Kubernetes Version to select a different version. If more than one image has the same Kubernetes version, select the newest image, according to the date in the image name.

  If the image that you want to use is not listed, use the CLI procedure and specify the OCID of the image. To get the OCID of the image you want, use the `ce node-pool get` command for a node pool where you used this image before.

  > **Note:**
  >
  > The image that you specify must not have a Kubernetes version that's newer than the Kubernetes version that you specified when you created the cluster. The Kubernetes Version for the cluster is in a column of the cluster list table.

- **Shape**: Select a shape for the worker nodes. For descriptions of available shapes, see Compute Shapes. If you're using the GPU Operator cluster add on, you can use the supported GPU shapes.

  If you select a shape that is not a flexible shape, the amount of memory and number of OCPUs are displayed. These numbers match the numbers shown for this shape in the table in Compute Shapes.

  If you select a flexible shape, then you must specify the number of OCPUs you want. You can optionally specify the total amount of memory you want. The default value for gigabytes of memory is 16 times the number you specify for OCPUs. Click inside each value field to see the minimum and maximum allowed values.

  > **Note:**
  >
  > Allocate at least 2 OCPUs and 32 GB memory for every 10 running pods. You might need to allocate more resources, depending on the workloads that are planned. See Resource Management for Pods and Containers.

- **Boot Volume**: (Optional) Check the box to specify a custom boot volume size.

  Boot volume size (GB): The default boot volume size for the selected image is shown. To specify a larger size, enter a value from 50 to 16384 in gigabytes (50 GB to 16 TB) or use the increment and decrement arrows.

If you specify a custom boot volume size, you need to extend the partition to take advantage of the larger size. Oracle Linux platform images include the `oci-utils` package. Use the `oci-growfs` command from that package to extend the root partition and then grow the file system. See oci-growfs.

- **Pod Communication** (VCN-Native Pod Networking clusters only)

  Pod Communication Subnet: Select a subnet that has configuration like the "pod" subnet described in Creating a Pod Subnet (VCN-Native Pod) on page 884.

  Number of Pods per node: The maximum number of pods that you want to run on a single worker node in a node pool. The default value is 31. You can enter a number from 1 to 110. The number of VNICs allowed by the shape you specify (see "Shape" above) limits this maximum pods number. See Node Shapes and Number of Pods on page 864. To conserve the pod subnet's address space, reduce the maximum number of pods you want to run on a single worker node. This reduces the number of IP addresses that are pre-allocated in the pod subnet.

  If you check the box to Use Security Rules in Network Security Group (NSG), select the Add Network Security Group button and select an NSG from the drop-down list. You might need to change the compartment to find the NSG you want. Secondary VNICs from the pod subnet will be attached to this NSG.
- **Cordon and Drain**: (Optional) Enter the number of minutes of eviction grace duration, or use the arrows to decrease or increase the number of minutes of eviction grace duration. The maximum value and default value is 60 minutes.

  - Private Cloud Appliance Release 3.0.2-b1261765. Specify an integer from 0 to 60. If you enter 0, the value will be converted to 0.333 because 20 seconds is the minimum eviction grace duration. If you then select the up arrow, the value will change to 1.
  - Private Cloud Appliance Release 3.0.2-b1185392. Specify an integer from 1 to 60.

  You cannot clear "Force terminate after grace period." Nodes are deleted after their pods are evicted or at the end of the eviction grace duration, even if not all pods are evicted.

  For descriptions of cordon and drain and eviction grace duration, select the CLI tab on this page, and see Node and node pool deletion settings.
- **SSH Key**: The public SSH key for the worker nodes. Either upload the public key file or copy and paste the content of the file.
- **Kubernetes Labels**: Select the Add Kubernetes Label button and enter a key name and value. You can use these labels to target pods for scheduling on specific nodes or groups of nodes. For a description and example, select the CLI tab on this page, and see Labels.
- **Node Pool Tags**: Defined or free-form tags for the node pool resource.

  > **Note:**
  >
  > Don't specify values for the OraclePCA-OKE.cluster_id defined tag or for the ClusterResourceIdentifier free-form tag. These tag values are system-generated and only applied to nodes (instances), not to the node pool resource.
- **Node Tags**: Defined or free-form tags that are applied to every node in the node pool.

  > **Important:**
  >
  > Don't specify values for the OraclePCA-OKE.cluster_id defined tag or for the ClusterResourceIdentifier free-form tag. These tag values are system-generated.

6. Select the **Add Node Pool** button in the dialog box.

   The details page for the node pool is displayed. Scroll to the **Resources** section and select **Work Requests** to see the progress of the node pool creation and see nodes being added to the Nodes list. The work request status is Accepted until the cluster is in either Active state or Failed state.

To identify these nodes in a list of instances, note that the names of these nodes are in the format `oke-ID`, where `ID` is the first 32 characters after the `pca_name` in the node pool OCID. Search for the instances in the list whose names contain the `ID` string from this node pool OCID.

What's Next:

1. Configure any registries or repositories that the worker nodes need. Ensure you have access to a self-managed public or intranet container registry to use with the OKE service and your application images.
2. Create a service to expose containerized applications outside the Private Cloud Appliance. See Exposing Containerized Applications on page 947.
3. Create persistent storage for applications to use. See Adding Storage for Containerized Applications on page 949.

Using the CLI

Use the oci ce node-pool create command and required parameters to create a new node pool.

```
oci ce node-pool create --cluster-id cluster_OCID --compartment-
id compartment_OCID --name pool_name --node-shape node_shape_name [OPTIONS]
```

1. Get the information you need to run the command.
   - The OCID of the compartment where you want to create the node pool: `oci iam compartment list`
   - The OCID of the cluster for this node pool: `oci ce cluster list`
   - The name of the node pool. Avoid entering confidential information.
   - The placement configuration for the nodes, including the worker subnet OCID and fault domain. For additional description, select the Console tab on this page, and see Placement configuration. Use the following command to show the content and format of this option:

     ```
     $ oci ce node-pool create --generate-param-json-input placement-configs
     ```

     Use the following command to list fault domains: `oci iam fault-domain list`. Don't specify more than one fault domain or more than one subnet in the placement configuration. To allow the system to select the best fault domains, don't specify any fault domain. A Roving Edge device has only 1 fault domain.
   - (VCN-Native Pod Networking clusters only) The OCID of the pod subnet. See Creating a Pod Subnet (VCN-Native Pod) on page 884. For additional description, select the Console tab on this page, and see Pod Communication. Use the `--pod-subnet-ids` option. Although the `--pod-subnet-ids` option value is an array, you can specify only one pod subnet OCID.

     The maximum number of pods that you want to run on a single worker node in a node pool. Use the `--max-pods-per-node` option. The default value is 31. You can enter a number from 1 to 110. The number of VNICs allowed by the shape you specify (see The name of the shape) limits this maximum pods number. See Node Shapes and Number of Pods on page 864. To conserve the pod subnet's address space, reduce the maximum number of pods you want to run on a single worker node. This reduces the number of IP addresses that are pre-allocated in the pod subnet.

     (Optional) The OCID of the Network Security Group to use for the pods in this node pool. Use the `--pod-nsg-ids` option. You can specify up to five NSGs.
   - The OCID of the image to use for the nodes in this node pool.

     Use the following command to get the OCID of the image that you want to use:

     ```
     $ oci compute image list --compartment-id compartment_OCID
     ```

     If the image that you want to use isn't listed, you can get the OCID of the image from the output of the `ce node-pool get` command for a node pool where you used this image before.

> **Note:**
>
> The image that you specify must have `-OKE-` in its `display-name` and must not have a Kubernetes version that's newer than the Kubernetes version that you specified when you created the cluster.

The Kubernetes version for the cluster is shown in `cluster list` output. The Kubernetes version for the image is shown in the `display-name` property in `image list` output. The Kubernetes version of the following image is 1.29.9.

```
"display-name": "uln-pca-Oracle-Linux8-OKE-1.29.9-20250325.oci"
```

If more than one image has the same Kubernetes version, select the newest image, according to the date in the image name.

Don't specify the `--kubernetes-version` option in the `node-pool create` command.

You can specify a custom boot volume size in gigabytes. The default boot volume size is 50 GB. To specify a custom boot volume size, use the `--node-source-details` option to specify both the boot volume size and the image. You can't specify both `--node-image-id` and `--node-source-details`. Use the following command to show the content and format of the node source details option.

```
$ oci ce node-pool create --generate-param-json-input node-source-
details
```

If you specify a custom boot volume size, you need to extend the partition to take advantage of the larger size. Oracle Linux platform images include the `oci-utils` package. Use the `oci-growfs` command from that package to extend the root partition and then grow the file system. See oci-growfs.

- The name of the shape of the worker nodes in this node pool. If you're using the GPU Operator cluster add on, you can use the supported GPU shapes. For all other Private Cloud Appliance systems, the default shape is VM.PCAStandard1.1, and you can specify a different shape. For descriptions of available shapes, see Compute Shapes.

  If you specify a flexible shape, then you must also specify the shape configuration, as shown in the following example. You must provide a value for `ocpus`. The `memoryInGBs` property is optional; the default value in gigabytes is 16 times the number of `ocpus`.

  ```
  --node-shape-config '{"ocpus": 32, "memoryInGBs": 512}'
  ```

  > **Note:**
  >
  > Allocate at least 2 OCPUs and 32 GB memory for every 10 running pods. You might need to allocate more resources, depending on the workloads that are planned. See Resource Management for Pods and Containers.

  If you specify a shape that is not a flexible shape, do not specify `--node-shape-config`. The number of OCPUs and amount of memory are set to the values shown for this shape in "Standard Shapes" in Compute Shapes.
- (Optional) The OCID of the Network Security Group to use for the nodes in this node pool. Use the `--nsg-ids` option. Do not specify more than one NSG.
- (Optional) Labels. Setting labels on nodes enables you to target pods for scheduling on specific nodes or groups of nodes. Use this functionality to ensure that specific pods only run on nodes with certain isolation, security, or regulatory properties.

  Use the `--initial-node-labels` option to add labels to the nodes. Labels are a list of key/value pairs to add to nodes after they join the Kubernetes cluster. See Metadata Key Restrictions on page 636 for more information about metadata limits.

  The following is an example label to apply to the nodes in the node pool:

```
--initial-node-labels '[{"key":"disktype","value":"ssd"}]
```

An easy way to select nodes based on their labels is to use `nodeSelector` in the pod configuration. Kubernetes only schedules the pod onto nodes that have each of the labels that are specified in the `nodeSelector` section.

The following example excerpt from a pod configuration specifies that pods that use this configuration must be run on nodes that have the `ssd` disk type label:

```
nodeSelector:
  disktype: ssd
```

- (Optional) Node metadata. Use the `--node-metadata` option to attach custom user data to nodes. See the following proxy settings item for a specific example.

  See Metadata Key Restrictions on page 636 for information about metadata limits. The maximum size of node metadata is 32,000 bytes.

- (Optional) Proxy settings. If your network requires proxy settings to enable worker nodes to reach outside registries or repositories, for example, create an argument for the `--node-metadata` option.

  In the `--node-metadata` option argument, provide values for `crio-proxy` and `crio-noproxy` as shown in the following example file argument:

  ```
  {
    "crio-proxy": "http://your_proxy.your_domain_name:your_port",
    "crio-noproxy":
   "localhost,127.0.0.1,your_domain_name,ocir.io,Kubernetes_cidr,pods_cidr"
  }
  ```

  If the cluster is using VCN-Native Pod Networking, add 169.254.169.254 to the noproxy setting, as in the following example:

  ```
  "crio-noproxy":
   "localhost,127.0.0.1,your_domain_name,ocir.io,Kubernetes_cidr,pods_cidr,169.254.16
  ```

- (Optional) Node and node pool deletion settings. You can specify how to handle node deletion when you delete a node pool, delete a specified node, decrement the size of the node pool, or change the node pool nodes placement configuration. These node deletion parameters can also be set or changed when you update the node pool, delete a specified node, or delete the node pool.

  To specify node pool deletion settings, create an argument for the `--node-eviction-node-pool-settings` option. You can specify the eviction grace duration (`evictionGraceDuration`) for nodes. Nodes are always deleted after their pods are evicted or at the end of the eviction grace duration.

  - Eviction grace duration. This value specifies the amount of time to allow to cordon and drain worker nodes.

    A node that's cordoned can't have new pods placed on it. Existing pods on that node aren't affected.

    When a node is drained, each pod's containers terminate gracefully and perform any necessary cleanup.

    The eviction grace duration value is expressed in ISO 8601 format: for example, PT45S, PT20M, or PT39M21S. The default value and the maximum value are 60 minutes (PT60M). The minimum value is 20 seconds (PT20S). OKE always tries to drain nodes for at least 20 seconds.
  - Force delete. Nodes are always deleted after their pods are evicted or at the end of the eviction grace duration. After the default or specified eviction grace duration, the node is deleted, even if one or more pod containers aren't completely drained.

The following shows an example argument for the `--node-eviction-node-pool-settings` option. If you include the `isForceDeleteAfterGraceDuration` property, then its value must be `true`. Nodes are always deleted after their pods are evicted or at the end of the eviction grace duration.

```
--node-eviction-node-pool-settings '{"evictionGraceDuration": "PT30M",
 "isForceDeleteAfterGraceDuration": true}'
```

> **Note:**
>
> If you use Terraform and you specify `node_eviction_node_pool_settings`, then you must explicitly set `is_force_delete_after_grace_duration` to `true`, even though true is the default value. The `is_force_delete_after_grace_duration` property setting isn't optional if you're using Terraform.

- (Optional) Tags. Add defined or free-form tags for the node pool resource by using the `--defined-tags` or `--freeform-tags` options. Do not specify values for the OraclePCA-OKE.cluster_id defined tag or for the ClusterResourceIdentifier free-form tag. These tag values are system-generated and only applied to nodes (instances), not to the node pool resource.

  To add defined or free-form tags to all nodes in the node pool, use the `--node-defined-tags` and `--node-freeform-tags` options.

  > **Important:**
  >
  > Do not specify values for the OraclePCA-OKE.cluster_id defined tag or for the ClusterResourceIdentifier free-form tag. These tag values are system-generated.

- (Optional) Custom CA Certificate Bundle. You can specify a custom CA certificate bundle to be used for verifying TLS connections to a private container registry. Use the `custom-ca-bundle-cert`, `registry-host`, and `registry-port` metadata parameters with the `--node-metadata` option.

  In the `--node-metadata` option argument, you must provide values for `custom-ca-bundle-cert`:

  ```
  --node-metadata '{"custom-ca-bundle-cert":"<base64-encoded-cert-
  content>"}'
  ```

  If you configured a private registry with a custom CA bundle, ensure nodes are newly provisioned or cycled (scaled down and back up or rolling restart) to apply the updated certificate to all worker nodes.

2. Run the create node pool command.

   Example:

   See the Console procedure for information about the options shown in this example, and other options such as `--node-boot-volume-size-in-gbs` and `nsg-ids`. The `--pod-subnet-ids` option is only applicable if the cluster uses VCN-Native Pod Networking.

   ```
   $ oci ce node-pool create \
   --cluster-id ocid1.cluster.unique_ID --compartment-id
    ocid1.compartment.unique_ID \
   --name node_pool_name --node-shape shape_name --node-image-id
    ocid1.image.unique_ID \
   --placement-configs
    '[{"availabilityDomain":"AD-1","subnetId":"ocid1.subnet.unique_ID"}]' \
   --pod-subnet-ids '["ocid1.subnet.unique_ID"]' --size 10 --ssh-public-key
    "public_key_text"
   ```

   The output from this `node-pool create` command is the same as the output from the `node-pool get` command. The cluster OCID is shown, and a brief summary of each node is shown. For more information about a node, use the `compute instance get` command with the OCID of the node.

Use the `work-request get` command to check the status of the node pool create operation. The work request OCID is in `created-by-work-request-id` in the `metadata` section of the `cluster get` output.

```
$ oci ce work-request get --work-request-id workrequest_OCID
```

The work request status will be `ACCEPTED` until the cluster is in either Active state or Failed state.

To identify these nodes in a list of instances, note that the names of these nodes are in the format `oke-ID`, where `ID` is the first 32 characters after the name in the node pool OCID. Search for the instances in the list whose names contain the `ID` string from this node pool OCID.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

What's Next:

1. Configure any registries or repositories that the worker nodes need. Ensure you have access to a self-managed public or intranet container registry to use with the OKE service and your application images.
2. Create a service to expose containerized applications outside the Private Cloud Appliance. See Exposing Containerized Applications on page 947.
3. Create persistent storage for applications to use. See Adding Storage for Containerized Applications on page 949.

Using the API

Use the CreateNodePool operation to create a new node pool.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Next Steps:

1. Configure any registries or repositories that the worker nodes need. Ensure you have access to a self-managed public or intranet container registry to use with the OKE service and your application images.
2. Create a service to expose containerized applications outside the Private Cloud Appliance. See Exposing Containerized Applications on page 947.
3. Create persistent storage for applications to use. See Adding Storage for Containerized Applications on page 949.

Updating an OKE Node Pool

On Private Cloud Appliance, you can update any configuration that you can set when you create a node pool except for the compartment where nodes are created.

When you update node properties, by default existing nodes aren't updated. The updated values only apply to new nodes that are created. New nodes are created when you increase the node count.

To replace existing nodes with new nodes that use these updated settings, see Node Cycling an OKE Node Pool on page 943.

If new worker nodes are added, perform the following steps:

1. Configure any registries or repositories that the worker nodes need. Ensure you have access to a self-managed public or intranet container registry to use with the OKE service and your application images.
2. Create a service to expose containerized applications outside the Private Cloud Appliance. See Exposing Containerized Applications on page 947.
3. Create persistent storage for applications to use. See Adding Storage for Containerized Applications on page 949.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Containers**, then select **Kubernetes Clusters (OKE)**.
2. Select the name of the cluster that contains the node pool that you want to update.
3. On the cluster details page, scroll to the **Resources** section and select **Node Pools**.

4. For the node pool that you want to update in the **Node Pools** list, select the Actions menu (⋮), then select **Edit**.

   The **Edit Node Pool** dialog box opens. You can change any configuration except the compartment where new nodes are created. See Creating an OKE Worker Node Pool on page 932 for property descriptions. The updated configuration only applies to new nodes that are created.

   Cordon and Drain: Enter the number of minutes of eviction grace duration, or use the arrows to decrease or increase the number of minutes of eviction grace duration. The maximum value and default value is 60 minutes.

   - Private Cloud Appliance Release 3.0.2-b1261765. You can specify an integer from 0 to 60. If you enter 0, the value will be converted to 0.333 because 20 seconds is the minimum eviction grace duration. The field will show a decimal value if the existing value was set in ISO 8601 format and included a seconds value. For example, an existing value of PT45S will show as 0.45, PT20M will show as 20, and PT39M21S will show as 39.35.
   - Private Cloud Appliance Release 3.0.2-b1185392. Specify an integer from 1 to 60. If the existing value was set in ISO 8601 format and includes a seconds value, that value will display as the next higher integer number of minutes. The seconds value will still be used, even though it does not display.

   You can't clear **Force terminate after grace period**. For descriptions of cordon and drain and eviction grace duration, go to Creating an OKE Worker Node Pool on page 932 and see Node and node pool deletion settings on the CLI tab.

   > **Note:**
   >
   > Don't specify values for the OraclePCA-OKE.cluster_id defined tag or for the ClusterResourceIdentifier free-form tag. These tag values are system-generated and only applied to nodes (instances), not to the node pool resource.

5. When you're finished making changes, select **Save Changes**.

   The details page for the node pool is displayed. In addition to **Node Pool Information** and **Tags** tabs, the node pool details page has a **Placement Configuration** tab.

   The updated configuration only applies to new nodes that are created by this procedure or in the future.

   To replace existing nodes with new nodes that use these updated settings, see Node Cycling an OKE Node Pool on page 943.

What's Next:

If you make changes that add new worker nodes, consider these next steps:

1. Configure any registries or repositories that the worker nodes need. Ensure you have access to a self-managed public or intranet container registry to use with the OKE service and your application images.
2. Create a service to expose containerized applications outside the Private Cloud Appliance. See Exposing Containerized Applications on page 947.
3. Create persistent storage for applications to use. See Adding Storage for Containerized Applications on page 949.

Using the OCI CLI

Use the oci ce node-pool update command and required parameters to update a node pool.

```
oci ce node-pool update --node-pool-id <node-pool_OCID> [OPTIONS]
```

1. Get the information you need to run the command.
   - The OCID of the node pool that you want to update: `oci ce node-pool list`
   - (Optional) Node and node pool deletion settings. Use the `--node-eviction-node-pool-settings` option or the `--override-eviction-grace-duration` option to set the eviction grace duration for nodes. Nodes are always deleted after their pods are evicted or at the end of the eviction grace duration. See the description in Creating an OKE Worker Node Pool on page 932.

- (Optional) Labels. To add labels to new nodes, use the `--initial-node-labels` option. Labels on existing nodes cannot be changed by using the `--initial-node-labels` option. Labels on existing nodes can be modified using `kubectl`. For more information about node labels, see Creating an OKE Worker Node Pool on page 932.
- (Optional) Tags. Add, change, or delete defined or free-form tags for the node pool resource by using the `--defined-tags` and `--freeform-tags` options. Do not specify values for the OraclePCA-OKE.cluster_id defined tag or for the ClusterResourceIdentifier free-form tag. These tag values are system-generated and only applied to nodes (instances), not to the node pool resource.

  To add tags to nodes that are newly added to the node pool, use the `--node-defined-tags` and `--node-freeform-tags` options.
- (Optional) Custom CA Certificate Bundle. You can specify a custom CA certificate bundle to be used for verifying TLS connections to a private container registry. Use the `custom-ca-bundle-cert`, `registry-host`, and `registry-port` metadata parameters with the `--node-metadata` option.

  In the `--node-metadata` option argument, you must provide values for `custom-ca-bundle-cert` and `registry-host`, and optionally you can provide a value for `registry-port` as shown in the following example file argument:

  ```
  --node-metadata custom-ca-bundle-cert=<base64-encoded-cert-content>,registry-host=<registry.example.com>,registry-port=5000
  ```

  If you configured a private registry with a custom CA bundle, ensure nodes are newly provisioned or cycled (scaled down and back up or rolling restart) to apply the updated certificate to all worker nodes.

2. (Optional) Create an argument for the `--node-pool-cycling-details` option, and use that option to apply these updates to all of the nodes in the node pool.

   Without the `--node-pool-cycling-details` option, the updated configuration specified in this `node-pool update` command only applies to new nodes that are created by this command or in the future, as described at the beginning of this topic.

   To replace existing nodes with new nodes that use these updated settings, specify the `--node-pool-cycling-details` option as described in Node Cycling an OKE Node Pool on page 943.
3. Run the update node pool command.

   ```
   $ oci ce node-pool update --node-pool-id ocid1.nodepool.unique_ID \
   <new_configuration_settings>
   ```

What's Next:

If you make changes that add new worker nodes, consider these next steps:

1. Configure any registries or repositories that the worker nodes need. Ensure you have access to a self-managed public or intranet container registry to use with the OKE service and your application images.
2. Create a service to expose containerized applications outside the Private Cloud Appliance. See Exposing Containerized Applications on page 947.
3. Create persistent storage for applications to use. See Adding Storage for Containerized Applications on page 949.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the UpdateNodePool operation to update a node pool.

If you make changes that add new worker nodes, consider these next steps:

1. Configure any registries or repositories that the worker nodes need. Ensure you have access to a self-managed public or intranet container registry to use with the OKE service and your application images.
2. Create a service to expose containerized applications outside the Private Cloud Appliance. See Exposing Containerized Applications on page 947.

3. Create persistent storage for applications to use. See Adding Storage for Containerized Applications on page 949.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Using Node Doctor to Troubleshoot Worker Node Issues

On Private Cloud Appliance, Node Doctor is a script that is included in the latest OKE images.

> **Note:**
>
> If Private Cloud Appliance has worker nodes that were created before Node Doctor was included in OKE images, you can use node cycling to update older worker node images. See Node Cycling an OKE Node Pool on page 943.

If a cluster has a worker node that's in a state other than Active or Running, use the Node Doctor utility to troubleshoot the issues.

Node Doctor scans a worker node and reports the health status of the node. Node Doctor can do the following tasks:

- Identify potential problem areas and provide references to information to help you address those problem areas. See Print Troubleshooting Information on page 943.
- Collect node system information into a support bundle if you need help from Oracle Support. See Create a Support Bundle on page 943.

> **Important:**
>
> Use Node Doctor only on worker nodes. Because Node Doctor is installed on OKE images, Node Doctor is also available on cluster control plane nodes. Don't use Node Doctor on control plane nodes.

Node Doctor was first delivered in Private Cloud Appliance Release 3.0.2-b1325160. If your node pools were created on that release or later, then you can proceed with the instructions in this topic. If your worker node image is from an earlier release, then that node does not have access to Node Doctor. Note that if your Private Cloud Appliance is running a release that includes Node Doctor, then you could use node cycling to update older worker node images. See Node Cycling an OKE Node Pool on page 943.

Connect to the Worker Node Using SSH

Perform the following steps to connect to the worker node that you want to troubleshoot.

1. Ensure that you have a private and public SSH key pair.

   You must have the private key that goes with the public key that was added to the node when the node was created.
2. Get the node username. OKE images have the initial username `opc` configured.
3. Get the IP address of the worker node that you need to troubleshoot.

   The IP address is on the Networking tab of the node details page in the Compute Web UI.

   - If the node has a public IP address, use the public IP address.
   - If the node is on a private IP, then connect to the node through the bastion host.

     If a bastion host isn't available, see Creating a Bastion.
4. Enter the following command at a shell prompt on your local system (public IP address) or on the bastion host (private IP address):

   ```
   ssh -i private_key_file username@ip-address
   ```

   - *private_key_file*. The full path and name of the file that contains the private SSH key that goes with the public key that was added to the node when the node was created.

- *username*. The default username for the node. This value probably is `opc`.
- *ip-address*. The node IP address that you got in the previous step.

5. Ensure that you have execute permissions for the following script. You run the script later.

```
ls -l /usr/local/bin/node-doctor.sh
-rwxr-xr-x 1 user1 user1 6288 Dec  5  2024 usr/local/bin/node-doctor.sh
```

Print Troubleshooting Information

While logged in to the worker node as described in Connect to the Worker Node Using SSH on page 942, enter the following command to print information that identifies potential problem areas:

```
$ sudo /usr/local/bin/node-doctor.sh --check
```

Use the following command to see more options:

```
$ sudo /usr/local/bin/node-doctor.sh --help
```

Create a Support Bundle

If you can't resolve the issue, use the following command to create a support bundle with relevant information for Oracle Support:

```
$ sudo /usr/local/bin/node-doctor.sh --generate
```

The support bundle is in the `/tmp` directory as `oke-support-bundle-`*`dateTtime`*`.tar`.

> **Note:**
>
> Monitor the `/tmp` directory to ensure that it doesn't fill up. Remove old files using the `rm` command, for example.

See the following resources for information about submitting a Support Request and uploading a bundle:

- Creating a Support Request
- Quick User Guide to Upload Files to My Oracle Support - MOS (Doc ID 1588459.1)
- How to Upload Files to Oracle Support (Doc ID 1547088.2)

Node Cycling an OKE Node Pool

On Private Cloud Appliance, when you update a node pool, only new nodes that are added during this update or that are added later receive the updates. To replace existing nodes with new nodes that use updated settings, enable the node cycling option.

Node cycling performs an in-place update of all existing nodes in the node pool to the latest specified configuration. New nodes are created, workloads moved onto them from existing nodes, current node pool updates applied, and the original nodes terminated.

You can set the maximum number of nodes that are starting or terminating at any particular time.

- **Maximum surge**. The maximum number of new nodes that can be starting at any time during this update operation. Set this value to avoid adding too many new nodes before existing nodes are terminated, which could incur excessive cost. The default value is 1. The maximum value is 5.
- **Maximum unavailable**. The maximum number of existing nodes that can be terminating at any time during this update operation. Set this value to ensure that enough nodes remain to handle the workload. The default value is 0. The maximum value is 7.

One of these values must be greater than 0.

Both of these values can be set to either a number (from 0 to the configured number of nodes in the node pool, but not greater than the maximum cited above) or a percentage (from 0% to 100%, but not a percentage that would result in a number greater than the maximum cited above). These values can be a maximum of four characters.

Both of these values must be a number of nodes, or both of these values must be a percentage. You cannot specify one of these values as a number of nodes and the other value as a percentage.

If you set either of these properties to a percent value that exceeds the maximum allowed number of nodes, the error message tells you the maximum allowed percent value for this node pool.

> **Note:**
>
> If the node cycling operation fails (for example, the operation times out), try re-running the operation. You might need to run the node cycling operation multiple times if the system is loaded and running at scale.

Using the Compute Web UI

Follow the procedure in Updating an OKE Node Pool on page 939 to update the node pool configuration.

1. On the node pool details page, select the Cycle Nodes button.
2. In the Cycle Nodes dialog, enter values for the Maximum Surge and Maximum Unavailable properties.

   See the rules at the beginning of this topic.
3. Select the Cycle Nodes button in the dialog to start the node pool update operation.

   To monitor the progress of the update operation, view the status of the associated work request.

Using the OCI CLI

Use the `--node-pool-cycling-details` option of the oci ce node-pool update command to update all existing nodes in a node pool.

```
oci ce node-pool update [OPTIONS]
```

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Procedure

1. Construct a command to update the node pool configuration as described in Updating an OKE Node Pool on page 939 under the CLI tab.
2. In that same command (not later) include the `--node-pool-cycling-details` option.

   In addition to setting `maximumUnavailable` and `maximumSurge`, enable node cycling by setting `isNodeCyclingEnabled` to `true`. By default, `isNodeCyclingEnabled` is `false`, and node cycling is not performed if you set only `maximumUnavailable` or `maximumSurge` and do not set `isNodeCyclingEnabled` to `true`.

   ```
   $ oci ce node-pool update --node-pool-id ocid1.nodepool.unique_ID \
   new_configuration_settings \
   --node-pool-cycling-details
    '{"isNodeCyclingEnabled":true,"maximumUnavailable":"value","maximumSurge":"value"}'
   ```

   See the beginning of this topic for the possible values.

   In the following example, the image is updated for all nodes in the node pool:

   ```
   $ oci ce node-pool update --node-pool-id ocid1.nodepool.unique_ID \
   --node-source-details
    '{"imageId":"ocid1.image.unique_ID","sourceType":"IMAGE"}' \
   --node-pool-cycling-details
    '{"isNodeCyclingEnabled":true,"maximumUnavailable":"5%","maximumSurge":"5%"}'
   ```

   To monitor the progress of the update operation, view the status of the associated work request.

Find the work request OCID:

```
oci ce work-request list --compartment-id ocid1.compartment.unique_ID \
--resource-id ocid1.nodepool.unique_ID
```

Show the current state of the work request:

```
oci ce work-request get --work-request-id ocid1.workrequest.unique_ID
```

Using the API

Use the UpdateNodePool operation to update a node pool. To update all existing nodes in the node pool, use the nodePoolCyclingDetails attributes: isNodeCyclingEnabled, maximumSurge, and maximumUnavailable.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting an OKE Node Pool Node

Learn how to explicitly delete an OKE worker node On Private Cloud Appliance.

Worker nodes are also deleted when you update a node pool to scale down the node pool. See Updating an OKE Node Pool on page 939.

> **Caution:**
>
> Deleting a worker node permanently deletes the node. You can't recover a deleted worker node.

When you delete a node, by default a new node is created to satisfy the node count set for the pool. To override this behavior, select the option to decrease node pool size.

> **Important:**
>
> Don't use the `kubectl delete node` command to terminate worker nodes in an OKE cluster. The `kubectl delete node` command removes the worker node from the cluster's etcd key-value store, but the command doesn't terminate the underlying compute instance.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Containers**, then select **Kubernetes Clusters**.
2. Select the name of the cluster that contains the node that you want to delete.
3. On the cluster details page, scroll to the **Resources** section and select **Node Pools**.
4. Select the name of the node pool that contains the node that you want to delete.
5. On the node pool details page, scroll to the **Resources** section and select **Nodes**.
6. For the node that you want to delete, select the Actions menu (⋮) and select **Delete**.
7. Confirm the deletion.
   a. If you don't want a new node to be automatically created to replace the deleted node, select **Decrease node pool size**.
   b. Check the box to override the eviction grace duration in the cordon and drain settings for the node. See the description of this field in Updating an OKE Node Pool on page 939.

      Use the arrows to decrease or increase the number of minutes of eviction grace duration.

      You can't clear "Force terminate after grace period." The node is deleted after its pods are evicted or at the end of the eviction grace duration, even if not all pods are evicted.

      For descriptions of cordon and drain and eviction grace duration, go to Creating an OKE Worker Node Pool on page 932, select the CLI tab, and see *Node and node pool deletion settings*.

c. Select **Delete**.

Using the CLI

Use the oci ce node-pool delete-node command and required parameters to delete a node pool node.

```
oci ce node-pool delete-node --node-id <node_OCID> --node-pool-id <node-pool_OCID> [OPTIONS]
```

1. Get the information you need to run the command.
   - OCID of the node pool: `oci ce node-pool list`
   - OCID of the node: `oci ce node-pool list`
2. Run the delete node pool node command.

   If you don't want a new node to be automatically created to replace the deleted node, specify the `--is-decrement-size` option.

   Example:

   ```
   $ oci ce node-pool delete-node --node-pool-id ocid1.nodepool.unique_ID \
   --node-id ocid1.instance.unique_ID --is-decrement-size true --force
   ```

   You can use the `--override-eviction-grace-duration` option to set a new value for `evictionGraceDuration` for this node deletion. See the description of `--node-eviction-node-pool-settings` in Creating an OKE Worker Node Pool on page 932. For `node-pool delete-node`, this new eviction grace duration value only applies to the node being deleted.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteNode operation to delete a node pool node.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Deleting an OKE Node Pool

Learn how to delete an OKE node pool On Private Cloud Appliance.

> **Caution:**
>
> Deleting a node pool permanently deletes the node pool. You can't recover a deleted node pool.

Using the Compute Web UI

1. In the Compute Web UI navigation menu, select **Containers**, then select **Kubernetes Clusters**.
2. Select the name of the cluster that contains the node pool that you want to delete.
3. On the cluster details page, scroll to the **Resources** section and select **Node Pools**.
4. For the node pool that you want to delete, select the Actions menu (⋮) and select **Delete**.
5. Confirm the deletion.
   a. Enter the name of the node pool to confirm that you want to delete the node pool.
   b. Check the box to override the eviction grace duration in the cordon and drain settings for the nodes in the pool.

      Use the arrows to decrease or increase the number of minutes of eviction grace duration. See the description of this field in Updating an OKE Node Pool on page 939.

      You can't clear **Force terminate after grace period**. Nodes are deleted after their pods are evicted or at the end of the eviction grace duration, even if not all pods are evicted.

For descriptions of cordon and drain and eviction grace duration, go to Creating an OKE Worker Node Pool on page 932, select the CLI tab, and see *Node and node pool deletion settings*.

c. Select **Delete**.

Using the CLI

Use the oci ce node-pool delete command and required parameters to delete a node pool.

```
oci ce node-pool delete --node-pool-id <node-pool_OCID> [OPTIONS]
```

1. Get the OCID of the node pool that you want to delete: `oci ce node-pool list`
2. Run the delete node pool command. You can use the optional `--force option` to delete the node pool without prompting for confirmation.

   Example:

   ```
   $ oci ce node-pool delete --node-pool-id ocid1.nodepool.unique_ID --force
   ```

   You can use the `--override-eviction-grace-duration` option to set a new value for `evictionGraceDuration` for this node pool deletion. See the description of `--node-eviction-node-pool-settings` in Creating an OKE Worker Node Pool on page 932.

For a complete list of CLI commands, flags, and options, see the Command Line Reference.

Using the API

Use the DeleteNodePool operation to delete a node pool.

For information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Exposing Containerized Applications

On Private Cloud Appliance, you can expose an application deployment so that worker node applications can be reached from outside the appliance.

Do the following to expose worker node applications outside the appliance:

- Create an external load balancer.
- Update ingress and egress rules as necessary to support the port requirements of the containerized applications. For example, if any application uses TCP port 3000, then an ingress rule needs to be added with port 3000.

Create an External Load Balancer

An external load balancer is a Service of type LoadBalancer. The service provides load balancing for an application that has multiple running instances.

If you use the `--service-lb-defined-tags` or `--service-lb-flexible-tags` options to specify tags to be applied to external load balancers, then ensure that the applicable dynamic group includes the following policy. See Create a Cluster Dynamic Group and Policies on page 824.

```
allow dynamic-group dynamic-group-name to use tag-namespaces in
 compartment compartment-name
```

Ensure that the load balancer shape parameter has one of the following values:

- `400Mbps`
- `flexible` – Requires that you also provide `flex-min` and `flex-max` annotations.

You might need to edit the application deployment file to modify the load balancer shape value. See Specifying Alternative Load Balancer Shapes and Specifying Flexible Load Balancer Shapes for more information and examples of how to set these values.

To create a service load balancer on a private cluster (a cluster with a private worker load balancer subnet), then use the following annotation in the external load balancer template:

```
service.beta.kubernetes.io/oci-load-balancer-internal: "true"
```

Use the following command to create the external load balancer:

```
# kubectl create -f expose_lb
```

The following is the content of the `expose_lb` file:

```
apiVersion: v1
kind: Service
metadata:
  name: my-nginx-svc
  labels:
    app: nginx
  annotations:
    oci.oraclecloud.com/load-balancer-type: "lb"
    service.beta.kubernetes.io/oci-load-balancer-shape: "400Mbps"
spec:
  type: LoadBalancer
  ports:
   - port: 80
  selector:
    app: nginx
```

The following command shows more information about this external load balancer. The LoadBalancer Ingress IP address is the IP address that's used to reach node applications from outside the Private Cloud Appliance. In the Compute Web UI, the LoadBalancer Ingress IP address is shown under the heading "IP Address" at the bottom of the first column of the load balancer details page, followed by the label "(Public)."

```
# kubectl describe svc my-nginx-svc
Name:                     my-nginx-svc
Namespace:                default
Labels:                   app=nginx
Annotations:              oci.oraclecloud.com/load-balancer-type: lb
                          service.beta.kubernetes.io/oci-load-balancer-
shape: 400Mbps
Selector:                 app=nginx
Type:                     LoadBalancer
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       IP_address
IPs:                      IP_address
LoadBalancer Ingress:     Load_Balancer_IP_address
Port:                     <unset> 80/TCP
TargetPort:               80/TCP
NodePort:                 <unset> 32145/TCP
Endpoints:                IP_address:port, IP_address+1:port, IP_address
+2:port
Session Affinity:         None
External Traffic Policy:  Cluster
Events:
  Type    Reason                Age    From                Message
  ----
  Normal  EnsuringLoadBalancer  7m48s  service-controller  Ensuring load
 balancer
  Normal  EnsuredLoadBalancer   6m40s  service-controller  Ensured load
 balancer
```

For descriptions of traffic policies, see Preserving the client source IP.

Use the following command to list IP addresses and ports for the external load balancer:

```
# kubectl get svc
NAME          TYPE          CLUSTER-IP  EXTERNAL-IP                PORT(S)
    AGE
kubernetes    ClusterIP     IP_address  <none>                     443/TCP
    6h17m
my-nginx-svc  LoadBalancer  IP_address
         Load_Balancer_IP_address  80:32145/TCP  5h5m
```

What's Next:

Adding Storage for Containerized Applications on page 949

Adding Storage for Containerized Applications

On Private Cloud Appliance, you can add persistent storage for use by applications on an OKE cluster node. Storage created in a container's root file system is deleted when you delete the container. For more durable storage for containerized applications, configure persistent volumes to store data outside of containers.

A persistent volume (PV) is storage that enables data to remain intact when the containers to which the storage is connected are terminated.

A PV is a resource in the cluster. A persistent volume claim (PVC) is a request for a PV resource. A PVC is a storage request that is met by binding the PVC to a PV. A PVC provides an abstraction layer to the underlying storage.

You can provision PVCs using the following methods:

- **Block volumes**. Attach volumes from the Private Cloud Appliance Block Volume service. The volumes are connected to clusters created by OKE using a CSI (Container Storage Interface) volume plugin deployed on the clusters.
  - To provision a regular block volume, see Creating Persistent Block Volume Storage on page 949.
  - To provision a high performance block volume, see Creating Persistent High Performance Block Volume Storage on page 951.

  For information about block volumes On Private Cloud Appliance, see Block Volume Storage on page 712.
- **File systems**. Mount file systems from the Private Cloud Appliance File Storage service. The File Storage service file systems are mounted inside containers running on clusters created by OKE using a CSI volume plugin deployed on the clusters.
  - Provision a PVC on a new file system using the CSI volume plugin. Create a storage class and a PVC. The CSI volume plugin dynamically creates both a new File Storage service file system and a new persistent volume backed by the new file system. See Creating Persistent File System Storage Using the CSI Volume Plugin on page 952.
  - Provision a PVC on an existing file system. Create a file system, mount target, PV, and PVC. See Creating Persistent File System Storage Using an Existing File System on page 954.

  For information about file systems on the Private Cloud Appliance, see File Storage on page 754.

Creating Persistent Block Volume Storage

The Private Cloud Appliance Block Volume service provides persistent, durable, and high-performance block storage that you can use to store data outside of containers.

This procedure automatically creates the requested `oci-bv` storage class; you don't need to create it. This procedure starts with using the `kubectl` command to create the persistent volume claim.

1. Create a persistent volume claim, specifying the storage class name `oci-bv`.

   ```
   $ kubectl create -f csi-bvs-pvc.yaml
   ```

The following is the content of the `csi-bvs-pvc.yaml` file:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mynginxclaim
spec:
  storageClassName: "oci-bv"
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 50Gi
```

The persistent volume claim name in the `metadata` section is user-specified. You can have more than one persistent volume claim on a persistent volume.

For the value of `accessModes`, specify `ReadWriteOnce`. Don't use `ReadWriteMany`.

The value of the `storage` property must be at least 50 gigabytes.

2. Run the following command to verify that the PVC has been created:

```
$ kubectl get pvc
NAME           STATUS   VOLUME   CAPACITY   ACCESSMODES   STORAGECLASS
 AGE
mynginxclaim   Pending                                     oci-bv
 4m
```

The PVC has a status of `Pending` because the `oci-bv` storage class definition includes the following:

```
volumeBindingMode: WaitForFirstConsumer
```

3. Use the PVC when creating other objects, such as pods.

For example, you could create a new pod from the following pod definition, which instructs the system to use the `mynginxclaim` PVC as the `nginx` volume, which is mounted by the pod at `/data`:

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
    - name: nginx
      image: nginx:latest
      ports:
        - name: http
          containerPort: 80
      volumeMounts:
        - name: data
          mountPath: /usr/share/nginx/html
  volumes:
    - name: data
      persistentVolumeClaim:
        claimName: mynginxclaim
```

Run the following command to verify that the PVC has been bound to a new PV:

```
$ kubectl get pvc
NAME           STATUS   VOLUME          CAPACITY   ACCESSMODES
 STORAGECLASS   AGE
```

```
mynginxclaim   Bound    csi-unique_ID   50Gi       RWO            oci-bv
```

Run the following command to verify that the pod is using the new PVC:

```
$ kubectl describe pod nginx
```

Creating Persistent High Performance Block Volume Storage

Learn to create a high performance block volume as persistent storage On Private Cloud Appliance.

If you don't need a high performance block volume, use the instructions in Creating Persistent Block Volume Storage on page 949.

1. Create a high performance block volume using the CSI plugin specified by the `oci-bv-high` storage class definition (`provisioner: blockvolume.csi.oraclecloud.com`).

   ```
   $ kubectl create -f csi-bvs-high.yaml
   ```

   The following is the content of the `csi-bvs-high.yaml` file:

   ```
   apiVersion: storage.k8s.io/v1
   kind: StorageClass
   metadata:
     name: oci-bv-high
   provisioner: blockvolume.csi.oraclecloud.com
   parameters:
     vpusPerGB: "20"
     attachment-type: "paravirtualized"
   volumeBindingMode: WaitForFirstConsumer
   allowVolumeExpansion: true
   reclaimPolicy: Delete
   ```

2. Create a persistent volume claim, specifying the storage class name `oci-bv-high`.

   ```
   $ kubectl create -f csi-bvs-high-pvc.yaml
   ```

   The following is the content of the `csi-bvs-high-pvc.yaml` file:

   ```
   apiVersion: v1
   kind: PersistentVolumeClaim
   metadata:
     name: mynginxclaim-high
   spec:
     storageClassName: "oci-bv-high"
     accessModes:
       - ReadWriteOnce
     resources:
       requests:
         storage: 50Gi
   ```

   The persistent volume claim name in the `metadata` section is user-specified. You can have more than one persistent volume claim on a persistent volume.

   For the value of `accessModes`, specify `ReadWriteOnce`. Do not use `ReadWriteMany`.

   The value of the `storage` property must be at least 50 gigabytes.

3. Run the following command to verify that the PVC has been created:

   ```
   $ kubectl get pvc
   NAME                STATUS   VOLUME   CAPACITY   ACCESSMODES
    STORAGECLASS   AGE
   ```

```
mynginxclaim-high   Pending                                      oci-bv-high
    4m
```

The PVC has a status of Pending because the `oci-bv-high` storage class definition includes the following:

```
volumeBindingMode: WaitForFirstConsumer
```

4. Use the PVC when creating other objects, such as pods.

   For example, you could create a new pod from the following pod definition, which instructs the system to use the `mynginxclaim-high` PVC as the `nginx` volume, which is mounted by the pod at `/data`:

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-high
spec:
  containers:
    - name: nginx
      image: nginx:latest
      ports:
        - name: http
          containerPort: 80
      volumeMounts:
        - name: data
          mountPath: /usr/share/nginx/html
  volumes:
    - name: data
      persistentVolumeClaim:
        claimName: mynginxclaim-high
```

   Run the following command to verify that the PVC has been bound to a new PV:

```
$ kubectl get pvc
NAME                STATUS   VOLUME          CAPACITY   ACCESSMODES
 STORAGECLASS   AGE
mynginxclaim-high   Bound    csi-unique_ID   50Gi       RWO           oci-
bv-high
```

   Run the following command to verify that the pod is using the new PVC:

```
$ kubectl describe pod nginx-high
```

Creating Persistent File System Storage Using the CSI Volume Plugin

On Private Cloud Appliance, you can provision a PVC on a new file system using the CSI volume plugin. Use the `kubectl` command to create the storage class and persistent volume claim. The CSI volume plugin provisions the PVC on a new file system.

You can have only one mount target and one file system per VCN. You can have multiple storage classes, persistent volumes, and persistent volume claims per cluster. All storage classes, persistent volumes, and persistent volume claims in a cluster share one NFS.

1. Create a new storage class that uses the `fss.csi.oraclecloud.com` provisioner.

```
$ kubectl create -f sc.yaml
```

   The following is the content of the `sc.yaml` manifest file:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: fss-dyn-storage
```

```
provisioner: fss.csi.oraclecloud.com
parameters:
  availabilityDomain: AD-1
  compartmentOcid: ocid1.compartment.unique_ID
  mountTargetSubnetOcid: ocid1.subnet.unique_ID
  exportPath: AUTOSELECT
  exportOptions: "[{\"source\":\"0.0.0.0/0\",\"requirePrivilegedSourcePort
\":false,\"access\":\"READ_WRITE\",\"identitySquash\":\"NONE\"}]"
  encryptInTransit: "false"
```

- The name for the new storage class is `fss-dyn-storage`.
- Either `mountTargetSubnetOcid` or `mountTargetOcid` is required. The value of `mountTargetSubnetOcid` is the OCID of the subnet where you want the CSI plugin to create a mount target. The value of `mountTargetOcid` is the OCID of an existing mount target. If you specify both `mountTargetSubnetOcid` and `mountTargetOcid`, `mountTargetOcid` is used and `mountTargetSubnetOcid` is ignored.

  To ensure that the mount target can be reached from worker nodes, specify the subnet that has configuration like the "worker" subnet described in Creating a Worker Subnet (Flannel Overlay) on page 852 or Creating a Worker Subnet (VCN-Native Pod) on page 887 or create the mount target on the subnet that has configuration like the worker subnet. Ensure that TCP port 2049 to the NFS server is open on that subnet.
- The `compartmentOcid` is optional. This value is the OCID of the compartment where the new file system (and the new mount target, if `mountTargetSubnetOcid` is specified) will be created. The default value is the same compartment as the cluster.
- You must specify `AUTOSELECT` as the value for `exportPath`.
- The `exportOptions` value is the NFS export options entry within the file system export that defines the access granted to NFS clients when they connect to a mount target. The `source` can be a single IP address or CIDR block range. This value is a set of parameters in JSON format.
- The value of `encryptInTransit` specifies whether to encrypt data in transit.

2. Create a PVC to be provisioned by the new file system in the File Storage service.

   ```
   $ kubectl create -f fss-dyn-claim.yaml
   ```

   The following is the content of the `fss-dyn-claim.yaml` manifest file:

   ```
   apiVersion: v1
   kind: PersistentVolumeClaim
   metadata:
     name: fss-dynamic-claim
   spec:
     accessModes:
     - ReadWriteMany
     storageClassName: "fss-dyn-storage"
     resources:
       requests:
         storage: 50Gi
   ```

3. Verify that the PVC has been bound to the new persistent volume.

   ```
   $ kubectl get pvc
   NAME              STATUS VOLUME
    CAPACITY ACCESS MODES STORAGECLASS    AGE
   fss-dynamic-claim Bound  csi-fss-f6823a66-8b6f-4c42-9d1f-d25723e69257 50Gi
        RWX          fss-dyn-storage 6m47s
   ```

4. Use the new PVC when you create objects such as pods.

   The following is an example object creation:

   ```
   $ kubectl create nginx.yaml
   ```

The following is the content of the `nginx.yaml` file. See the `claimName` on the last line:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx_image_url
        ports:
        - name: http
          containerPort: 80
        volumeMounts:
        - name: persistent-storage
          mountPath: /usr/share/nginx/html
      volumes:
      - name: persistent-storage
        persistentVolumeClaim:
          claimName: fss-dynamic-claim
```

Verify that the object is created and deployed:

```
$ kubectl get deploy
NAME             READY UP-TO-DATE AVAILABLE AGE
nginx-deployment 3/3   3          0         104s
```

Creating Persistent File System Storage Using an Existing File System

On Private Cloud Appliance, you can provision a PVC on an existing file system.

This procedure creates a mount target, file system, and file system export. Then uses the `kubectl` command to create the storage class, persistent volume, and persistent volume claim.

1. Create a mount target.

   For instructions, see Creating a Mount Target on page 763.

   > **Important:**
   >
   > To ensure that the mount target can be reached from worker nodes, create the mount target on the subnet that has the worker subnet described in Creating OKE Network Resources on page 833. Ensure that TCP port 2049 to the NFS server is open on that subnet.

   Note the export set OCID and mount target OCID. The export set OCID is required to create the file system export, and the mount target OCID is required to create the storage class in later steps.

   You can have only one mount target per VCN.
2. Create a file system.

   For instructions, see Creating a File System on page 766.

You can create only one file system per VCN. You can have multiple storage classes, persistent volumes, and persistent volume claims per cluster, and they all share one NFS.

3. Create a file system export to associate the mount target with the file system.

   For instructions, see Creating an Export for a File System on page 769.

   - Specify the export set OCID from the output from creating the mount target.
   - Specify the longest CIDR (smallest network) in the CIDR range that you specified when you created the "worker" subnet as described in Creating a Worker Subnet (Flannel Overlay) on page 852 or Creating a Worker Subnet (VCN-Native Pod) on page 887.

   Note the export path and the mount target IP address.

4. Create a storage class, specifying the mount target OCID from the output of the create mount target step.

```
$ kubectl create -f sc.yaml
```

   The following is the content of the `sc.yaml` file:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: pca-fss
provisioner: fss.csi.oraclecloud.com
parameters:
  mntTargetId: ocid1.mounttarget.unique_ID
```

   The values of the `apiVersion` and `provisioner` properties are standard. The value of the storage class name in the metadata section is user-specified. You can create more than one storage class per mount target, and the storage class name is used in the following steps to create a persistent volume and persistent volume claim.

   Use the `get sc` subcommand to view information about the new storage class:

```
$ kubectl get sc
```

5. Create a persistent volume, specifying the storage class name, the export path, and the mount target IP address.

   The storage class name is in the metadata in the `sc.yaml` file in the preceding step. The export path and the mount target IP address are output from the create file system export step. See Step 3 on page 955.

```
$ kubectl create -f pv.yaml
```

   The following is the content of the `pv.yaml` file:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: fss-pv
spec:
  storageClassName: pca-fss
  capacity:
    storage: 200Gi
  accessModes:
    - ReadWriteMany
  mountOptions:
    - nosuid
  nfs:
    server: mount_target_IP_address
    path: "/export/unique_ID"
    readOnly: false
```

The persistent volume name in the `metadata` section is user-specified. You can have more than one persistent volume in a storage class.

In the `nfs` section, the `server` value is the mount target IP address, and the `path` value is the export path.

Use the `get pv` subcommand to view information about the new persistent volume:

```
$ kubectl get pv
NAME    CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM
 STORAGECLASS  REASON  AGE
fss-pv  200Gi     RWX           Retain          Bound   default/fss-pvc
 pca-fss               20h
```

6. Create a persistent volume claim, specifying the persistent volume name and the storage class name.

   The persistent volume name and storage class name are in the output of the `get pv` command.

   Wait for the PVC status to be Bound before using this storage.

   ```
   kubectl create -f pvc.yaml
   ```

   The following is the content of the `pvc.yaml` file:

   ```
   apiVersion: v1
   kind: PersistentVolumeClaim
   metadata:
     name: fss-pvc
   spec:
     storageClassName: pca-fss
     accessModes:
       - ReadWriteMany
     resources:
       requests:
         storage: 200Gi
     volumeName: fss-pv
   ```

   The persistent volume claim name in the `metadata` section is user-specified. You can have more than one persistent volume claim on a persistent volume.

   The value of the `accessModes` property must be `ReadWriteMany`.

   The value of the `storage` property must be at least 50 gigabytes.

   Run the following command to view information about the new persistent volume claim:

   ```
   $ kubectl get pvc
   NAME      STATUS   VOLUME   CAPACITY   ACCESSMODES   STORAGECLASS   AGE
   fss-pvc   Bound    fss-pv   200Gi      RWX           pca-fss        2h
   ```

7. Use the PVC when creating other objects, such as pods.

   For example, you could create a new pod from the following pod definition, which instructs the system to use the `fss-pvc` PVC as the `nginx` volume, which is mounted by the pod at `/persistent-storage`:

   ```
   apiVersion: v1
   kind: Pod
   metadata:
     name: fss-dynamic-app
   spec:
     containers:
       - name: nginx
         image: nginx:latest
         ports:
           - name: http
   ```

```
          containerPort: 80
      volumeMounts:
        - name: persistent-storage
          mountPath: /usr/share/nginx/html
  volumes:
  - name: persistent-storage
    persistentVolumeClaim:
      claimName: fss-pvc
```

Run the following command to verify that the pod is using the new PVC:

```
$ kubectl describe pod fss-dynamic-app
```

Using a Persistent Volume

On Private Cloud Appliance, to use this persistent storage, create a Kubernetes deployment and assign a persistent volume claim as shown in the following sections.

- Using Block Volume Storage on page 957
- Using File System Storage on page 958
- Verify the New Storage Asset on page 958

Using Block Volume Storage

The following example uses block volume storage:

```
$ kubectl create -f nginx-deploy.yaml
```

The following output is the content of the `nginx-deploy.yaml` file.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-bv-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx-bv
  template:
    metadata:
      labels:
        app: nginx-bv
    spec:
      containers:
      - name: nginx
        image: available_internal_registry/nginx:latest
        volumeMounts:
        - mountPath: /usr/share/nginx/
          name: data
        ports:
        - containerPort: 80
          name: http
          protocol: TCP
      volumes:
      - name: data
        persistentVolumeClaim:
          claimName: mynginxclaim
```

Using File System Storage

The following example uses file system storage:

```
$ kubectl create -f nginx-deploy.yaml
```

The following output is the content of the `nginx-deploy.yaml` file.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-fss-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx-fss
  template:
    metadata:
      labels:
        app: nginx-fss
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        volumeMounts:
        - mountPath: /usr/share/nginx/
          name: data
        ports:
        - containerPort: 80
          name: http
          protocol: TCP
      volumes:
      - name: data
        persistentVolumeClaim:
          claimName: fss-pvc
```

Verify the New Storage Asset

Use the `get pod` subcommand to show the names of the replicas in the pod:

```
$ kubectl get pod
nginx-deployment-55ff88b668-2k8rt 1/1 Running 0 4m54s
nginx-deployment-55ff88b668-79c2t 1/1 Running 0 4m54s
nginx-deployment-55ff88b668-qpdfd 1/1 Running 0 4m54s
```

Log in to the pod and use the Linux `df` command to show that the application replicas are using the `persistentVolumeClaim` storage. The `Filesystem` column in the `df` output shows the mount target IP address and the file system export path.

```
$ kubectl exec -it nginx-deployment-55ff88b668-2k8rt -- df -h /usr/share/
nginx/html
Filesystem
        Size Used Avail Use% Mounted on
xxx.xx.xxx.xxx:/
export/4fsderwh09ufyf84ei1lh3q2x8ou86pq5vcbx3aeeo060xxxxxxxxxxxxxxx 67T  0
  67T   0%   /usr/share/nginx/html
```

Deleting a Persistent Volume

Learn how to delete a persistent volume (PV), or retain a PV after all associated persistent volume claims (PVCs) are deleted on Private Cloud Appliance.

To delete a PVC, see Deleting a Persistent Volume Claim on page 959.

For file system storage, the default behavior is to retain the PV when all associated PVCs are deleted.

For block volume storage, the default behavior is to delete the PV when all associated PVCs are deleted. You might prefer to retain the PV after all associated PVCs are deleted, for example, if the volume contains critical data. See Retaining a Persistent Volume on page 960.

If a PV is retained, you can optionally delete the PV later.

Deleting a Persistent Volume Claim

On Private Cloud Appliance, to delete a PVC, first delete all pods that are using that PVC. If you attempt to delete the PVC while a pod is still using the PVC, the PVC will be stuck in Terminating state and will not be deleted. When all the pods that are using that PVC are deleted, the PVC is deleted.

1. List all pods that are using the PVC.

   Ensure that you have JQ command line utilities installed to query JSON objects.

   Use the following command to list pods across all the namespaces that are associated with the PVC that you want to delete.

   ```
   $ kubectl get pods --all-namespaces -o=json | jq -c '.items[]
    | {name: .metadata.name, namespace: .metadata.namespace,
    claimName: .spec | select(has("volumes")).volumes[] |
    select(has("persistentVolumeClaim")).persistentVolumeClaim.claimName} |
    select(.claimName != null)'

   {"name":"pod1_name","namespace":"namespace1_name","claimName":"claim1_name"}
   {"name":"pod2_name","namespace":"namespace1_name","claimName":"claim1_name"}
   {"name":"pod3_name","namespace":"namespace2_name","claimName":"claim2_name"}
   ```

   To list pods only in the current namespace, use the same command as the preceding command except omit the `--all-namespaces` option.
2. Delete all pods that are using the PVC.

   Use the pod names reported by the `kubectl get pods` command that are associated with the `claimName` that you want to delete.

   ```
   $ kubectl delete pod pod1_name pod2_name
   ```
3. Delete the PVC.

   ```
   $ kubectl delete pvc claim1_name
   ```
4. (Optional) Delete the PV.

   If the Persistent Volume Reclaim Policy is Delete, the PV is automatically deleted when all PVCs that are associated with this PV are deleted.

   To list all PVCs, use the `kubectl get pvc` command.

   If the Persistent Volume Reclaim Policy is Retain, you can use the following command to delete the PV:

   ```
   $ kubectl delete pv pv_name
   ```

Retaining a Persistent Volume

Learn how to retain a persistent volume (PV) On Private Cloud Appliance.

Rather than delete a PV, you might prefer to retain the PV after all associated PVCs are deleted, for example if the volume contains critical data. See Changing the Reclaim Policy of a Persistent Volume on page 960 for instructions to change the reclaim policy of the PV so that the PV will be retained after all associated PVCs are deleted.

If the Persistent Volume Reclaim Policy is Delete, the PV is automatically deleted when all PVCs that are associated with this PV are deleted. To prevent this behavior, specify the Retain policy. With the Retain policy, the PV isn't deleted but is released of its claim. See Recovering the Data from a Released Persistent Volume on page 960 for instructions to recover the data.

If you decide you want to delete the PV even though it was retained, or you want to delete the PV after you have recovered the data, use the following command:

```
$ kubectl delete pv pv_name
```

Changing the Reclaim Policy of a Persistent Volume

1. List the PVs in the cluster.

```
$ kubectl get pv
NAME    CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM
 STORAGECLASS  REASON  AGE
fss-pv  200Gi     RWX           Delete          Bound   default/fss-pvc
 pca-fss              20h
```

2. Change the reclaim policy of the PV.

```
$ kubectl patch pv fss-pv -p '{"spec":
{"persistentVolumeReclaimPolicy":"Retain"}}'
```

3. Verify the reclaim policy change.

   The `RECLAIM POLICY` column should now say `Retain`.

```
$ kubectl get pv
```

Recovering the Data from a Released Persistent Volume

The PV isn't available for another claim after the PV has been released of its previous claim because the previous claimant's data is still on the volume. Recover the data and then re-create the PV using the same storage to make a new claim on that storage.

1. Delete the PV.

```
$ kubectl delete pv pv_name
```

   The associated block volume or file system still exists after the PV is deleted.
2. Manually recover and clean up the data on the block volume or file system.
3. (Optional) Manually delete the block volume or file system.

   To reuse the same block volume or file system, create a new PV with the same storage asset definition.

Troubleshooting Private Cloud Appliance

Identify and address common issues that can occur while working with Private Cloud Appliance.

Private Cloud Appliance Known Issues

These known issues have been identified in Private Cloud Appliance.

- Upgrade might impact Load Balancer as a Service functionality on page 961
- GPU Drivers Not Included in Oracle Linux Platform Images on page 962
- After an API key is created or changed, the initial CLI command might fail on page 964
- The oci ccc get infrastructure and oci ccc infrastructure update CLI commands return null for the provisioning_pin value on page 964
- When using the ccc infrastructure list CLI command with the --compartment-id-in-subtree true option, no results are returned on page 965
- Output from oci iam user get doesn't list user capabilities on page 966

Upgrade might impact Load Balancer as a Service functionality

The first upgrade after May 2025, changes the Load Balancer as a Service (LBaaS) to a new background implementation. As a result, a few features are either different or no longer available. An existing configuration that's no longer supported in the new implementation, can have a negative impact on the software upgrade.

Details

Before the first upgrade after May 2025, if you use the Load Balancer service, review the following issues, and act if needed:

- **Response body regex parsing** (Bug: 37629014)

  If you have a Load Balancer configured with regular expression (regex) parsing of backend responses for health status information, that won't work after the upgrade. Health status reporting is limited to response codes.

  Workaround: Unconfigure the optional regex setting (`--response-body-regex`) for the response from the backend servers.
- **Cipher suites** (Bug: 37461876)

  In the new Load Balancer implementation, weaker cipher suites have been removed. Going forward, SSL/TLS connections can be secured with these cipher suites:

  ```
  AES128-GCM-SHA256, AES256-GCM-SHA384,
  ECDHE-ECDSA-AES128-GCM-SHA256, ECDHE-ECDSA-AES256-GCM-SHA384,
  ECDHE-RSA-AES128-GCM-SHA256, ECDHE-RSA-AES256-GCM-SHA384,
  AES128-SHA, AES256-SHA, DES-CBC3-SHA,
  ECDHE-ECDSA-AES128-SHA, ECDHE-ECDSA-AES256-SHA,
  ECDHE-RSA-AES128-SHA, ECDHE-RSA-AES256-SHA,
  PSK-AES128-CBC-SHA, PSK-AES256-CBC-SHA
  ```

  Workaround: Ensure that your Load Balancer configuration is using the latest cipher suites. If necessary, change the existing Load Balancer configurations.
- **Cookie-based session persistence** (Bug: 37473362)

  For existing load balancers, session persistence between clients and backend servers can be enabled using either application cookies or load balancer cookies. These are no longer supported after upgrade.

  Workaround: Unconfigure cookie-based session persistence. Or, load balancer cookies can be preserved on the condition that the load balancing policy is set to IP hash before the upgrade.
- **Server order preference**

The SSL parameter to prioritize server ciphers over client ciphers isn't supported.

For more information, see Load Balancer as a Service on page 550.

GPU Drivers Not Included in Oracle Linux Platform Images

The Oracle Linux 8 or Oracle Linux 9 platform images provided with Private Cloud Appliance don't include GPU drivers. If you create a GPU instance, you must manually install the GPU drivers.

Details

If a Private Cloud Appliance installation includes compute nodes with GPUs, you can access them by selecting a dedicated shape. The GPU shapes can be selected for compute instances based on an Oracle Linux 8 or Oracle Linux 9 platform image. The current image versions don't include GPU drivers. The instance OS detects the allocated GPUs, but to use them, you need the CUDA Toolkit from the NVIDIA developer site to install the required drivers.

> **Note:**
>
> The large download and local repository installation need a large amount of disk space. The default 50GB boot volume is insufficient on Oracle Linux 9 and only just large enough on Oracle Linux 8. We highly recommend increasing the boot volume size to at least 60GB, and extend the file system accordingly.

Workaround

After creating the instance, log in to the instance and install the CUDA Toolkit. Follow the instructions for your version of Oracle Linux.

Installing GPU Drivers in an Oracle Linux 9 Instance

1. From the command line of the instance, download and install the CUDA Toolkit rpm for your OS.

```
$ wget https://developer.download.nvidia.com/compute/
cuda/12.8.0/local_installers/cuda-repo-rhel9-12-8-
local-12.8.0_570.86.10-1.x86_64.rpm
$ sudo rpm -i cuda-repo-rhel9-12-8-
local-12.8.0_570.86.10-1.x86_64.rpm
$ sudo dnf clean all
$ sudo dnf install cuda-toolkit-12-8
```

2. Enable the Oracle Linux 9 EPEL yum repository. Install the `dkms` package.

```
$ sudo yum-config-manager --enable ol9_developer_EPEL
$ sudo dnf install dkms
```

3. Install the GPU drivers.

```
$ sudo dnf install cuda-12-8
```

4. Verify the installation with the NVIDIA System Management Interface.

```
$ nvidia-smi
+-----------------------------------------------------------------------------------------
+
| NVIDIA-SMI 570.86.10              Driver Version: 570.86.10
 CUDA Version: 12.8     |
|-----------------------------------------+------------------------
+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A |
 Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage |
 GPU-Util  Compute M. |
```

```
|                                         |                        |
               MIG M. |
|=========================================+========================
+======================|
|   0  NVIDIA L40S                    Off |   00000000:00:05.0 Off |
                    0 |
| N/A   26C    P8             23W /  350W |       1MiB /  46068MiB |
     0%      Default |
|                                         |                        |
                  N/A |
+-----------------------------------------+------------------------
+----------------------+

+-----------------------------------------------------------------------------
+
| Processes:
                       |
|  GPU   GI   CI              PID   Type   Process name
          GPU Memory |
|        ID   ID
          Usage      |
|
=============================================================================
|  No running processes found
                       |
+-----------------------------------------------------------------------------
+
```

Installing GPU Drivers in an Oracle Linux 8 Instance

1. From the command line of the instance, download and install the CUDA Toolkit rpm for your OS.

```
$ wget https://developer.download.nvidia.com/compute/
cuda/12.8.0/local_installers/cuda-repo-rhel8-12-8-
local-12.8.0_570.86.10-1.x86_64.rpm
$ sudo rpm -i cuda-repo-rhel8-12-8-
local-12.8.0_570.86.10-1.x86_64.rpm
$ sudo dnf clean all
$ sudo dnf install cuda-toolkit-12-8
```

2. Enable the Oracle Linux 8 EPEL yum repository. Install the `dkms` package.

```
$ sudo yum-config-manager --enable ol8_developer_EPEL
$ sudo dnf install dkms
```

3. Install the GPU drivers.

```
$ sudo dnf install cuda-12-8
```

4. Install the NVIDIA kernel module.

```
$ sudo scl enable gcc-toolset-13 bash
# dkms install nvidia-open -v 570.86.10
```

If this `make` error appears while the kernel module is built, you can safely ignore it.

```
Cleaning build area...(bad exit status: 2)
Failed command:
make -C /lib/modules/5.15.0-206.153.7.el8uek.x86_64/build M=/var/
lib/dkms/nvidia-open/570.86.10/build clean
```

5. Verify the installation with the NVIDIA System Management Interface.

```
# nvidia-smi
+-----------------------------------------------------------------------------
+
| NVIDIA-SMI 570.86.10              Driver Version: 570.86.10
 CUDA Version: 12.8     |
|-----------------------------------------+------------------------
+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A |
 Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage |
 GPU-Util  Compute M. |
|                                         |                        |
               MIG M. |
|=========================================+========================
+======================|
|   0  NVIDIA L40S                    Off |   00000000:00:05.0 Off |
                    0 |
| N/A   26C    P8             23W /  350W |       1MiB /  46068MiB |
      0%      Default |
|                                         |                        |
                   N/A |
+-----------------------------------------+------------------------
+----------------------+

+-----------------------------------------------------------------------------
+
| Processes:
                       |
|  GPU   GI   CI              PID   Type   Process name
          GPU Memory |
|        ID   ID
          Usage      |
|
=============================================================================
|  No running processes found
                       |
+-----------------------------------------------------------------------------
+
```

After an API key is created or changed, the initial CLI command might fail

Details

When an API key is added or changed for a user, the first CLI command with the new or changed key might fail.

Workaround

Wait a few minutes for the new key to synchronize on the appliance, then retry the CLI command.

The `oci ccc get infrastructure` and `oci ccc infrastructure update` CLI commands return `null` for the `provisioning_pin` value

Details

When you create an infrastructure, a PIN is generated and displayed in the output.

However, if you use the `oci ccc get infrastructure` command right after creating or updating the infrastructure, the PIN might not be returned.

This happens because the PIN isn't available to the `get` command for up to 5 minutes after creation.

Example output:

```
{
  "compartment_id": "ocid1.compartment.oc1..uniqueID",
. . .
  },
  "display_name": "C3ResourcePrincipal_infra",
  "freeform_tags": {},
  "id": "ocid1.cccinfrastructure.uniqueID",
  "lifecycle_details": null,
  "lifecycle_state": "ACTIVE",
  "provisioning_fingerprint": null,
  "provisioning_pin": null,
  "rack_inventory": {
    "capacity_storage_tray_count": null,
    "compute_node_count": null,
    "management_node_count": null,
    "performance_storage_tray_count": null,
    "serial_number": null
. . .

}
```

Workaround

Obtain the PIN from the `create` command, or wait 5 minutes to retrieve the PIN using the `get` command.

For more information, see the ccc infrastructure CLI Reference page.

When using the `ccc infrastructure list` CLI command with the `--compartment-id-in-subtree true` option, no results are returned

Details

You get an empty list even though there are items in the subtree.

Example:

```
oci ccc infrastructure list --profile user1 --compartment-id-in-subtree
 true -c ocid1.tenancy.oc1..uniqueID
{
  "data": {
    "items": []
  }
}
```

Workaround

Instead of using the `--compartment-id-in-subtree` option, query each compartment directly using the `-compartment` option.

Example:

```
oci ccc infrastructure list --profile user1 -compartment
 ocid1.tenancy.oc1..uniqueID
{
  "data": {
    "items": [ list of compartment details ]
  }
}
```

For more information, see the ccc infrastructure CLI Reference page.

Output from `oci iam user get` doesn't list user capabilities

Details

The output from `oci iam user get` differs between Oracle Cloud Infrastructure (OCI) and Private Cloud Appliance. The Private Cloud Appliance output shows `null` for `capabilites` and omits the list of capabilities, as shown in the following table.

OCI Output	Private Cloud Appliance Output

`oci iam user get --user-id ocid1.user.oc1..uniqueID`
`{`
`"data": {`
`"capabilities": {`
`"can-use-api-keys": true,`
`"can-use-auth-tokens": true,`
`"can-use-console-password": true,`
`"can-use-customer-secret-keys": true,`
`"can-use-o-auth2-client-credentials": true,`
`"can-use-smtp-credentials": true`
`},`
`"compartment-id": "ocid1.tenancy.oc1..uniqueID",`
`"defined-tags": {`
`"Oracle-Recommended-Tags": {`
`"ResourceType": "group",`
`"UtilExempt": "minrequired"`
`}`
`},`
`"description": "user-1",`
`"email": null,`
`"email-verified": false,`
`"external-identifier": null,`
`"freeform-tags": {},`
`"id": "ocid1.user.oc1..uniqueID",`
`"identity-provider-id": null,`
`"inactive-status": null,`
`"is-mfa-activated": false,`
`"last-successful-login-time": "2024-02-08T10:25:44.036000+00:00",`
`"lifecycle-state": "ACTIVE",`
`"name": "user-1",`
`"previous-successful-login-time": null,`
`"time-created": "2024-02-08T09:12:35.256000+00:00"`
`},`
`"etag": "60f0527b3bbd0f40f137d4149d131fbf77eb44ab"`
`}`
	`oci iam user get --user-id ocid1.user.oc1..uniqueID`
	`{`
	`"data": {`
	`"capabilities": null,`
	`"compartment-id": "ocid1.tenancy.oc1..uniqueID",`
	`"defined-tags": {`
	`"Oracle-Recommended-Tags": {`
	`"ResourceType": "group",`
	`"UtilExempt": "minrequired"`
	`}`
	`},`
	`"description": "user-1",`
	`"email": null,`
	`"email-verified": null,`
	`"external-identifier": null,`
	`"freeform-tags": {},`
	`"id": "ocid1.user.oc1..uniqueID",`
	`"identity-provider-id": null,`
	`"inactive-status": null,`
	`"is-mfa-activated": null,`
	`"last-successful-login-time": null,`
	`"lifecycle-state": "ACTIVE",`
	`"name": "user-1",`
	`"previous-successful-login-time": null,`
	`"time-created": "2023-02-08T09:12:35.256000+00:00"`
	`},`
	`"etag": "bee44237-6d70-4691-b7f9-a98fbb332b12"`

Workaround

To see the list of capabilities, run the `oci iam user get` command in your OCI tenancy.

Index

C

cooling requirements 62

F

floor load requirements 58

V

ventilation requirements 62