Oracle® Communications

Cloud Native Core, OCI Adaptor Deployment Guide

Release 24.1.0
F92397-01
April 2024

Oracle Communications Cloud Native Core, OCI Adaptor Deployment Guide, Release 24.1.0

F92397-01

Contents

3 Deploying NFs in OCI

4 Troubleshooting Scenarios

Acronyms

The following table lists the acronyms and the terminologies used in the document:

Table Acronyms and Terminologies

Acronym	Description
APM	Application Performance Monitoring (of OCI, that receives traces)
CIDR	Classless Inter-Domain Routing
DRG	Dynamic Routing Gateway
EGW	Egress Gateway
IGW	Ingress Gateway
NF	A 3GPP-defined processing function in a network, which has defined functional behaviour and 3GPP-defined interfaces.
NF service	It is a functionality exposed by an NF through a service-based interface and consumed by other authorized NFs.
NF service instance	An identifiable instance of the NF service
NLB	Network Load Balancer
OCI	Oracle Cloud Infrastructure
OCI Adaptor	OCI Adaptor components conduit between CNC NFs and OCI's observability services.
OKE	Oracle Engine for Kubernetes
OTEL	OpenTelemetry
RDP	Remote Desktop Protocol
RPC	Remote Peering Connection
SSH	Secure Shell
Terraform	Infrastructure as Code (IaC) software tool to provision and manage infrastructure in any cloud.
VCN	Virtual Cloud Network
Private VM	A Private Virtual Machine is a virtual machine hosted on-premise within a private network or infrastructure, usually owned and managed by an organization.
Public VM	A Public Virtual Machine is a virtual machine hosted on a public cloud infrastructure, such as Amazon Web Services (AWS), Microsoft Azure, and so on.

What's New in This Guide

Release 24.1.0- F92397-01, April 2024

This is the initial release of the document.

1 Introduction

This document contains information about the functionalities of Cloud Native Core Oracle Cloud Infrastructure (OCI) Adaptor and how to use it to seamlessly deploy the CNC Network Functions (NFs) on OCI.

The information in this document allows the users to:

- Deploy the OCI Adaptor on the public cloud OCI platform.
- Deploy the NFs in OCI using OCI Adaptor.
- Collect logs, metrics, and traces of CNC NFs and share them with OCI PaaS (Platform as a Service) using OCI Adaptor.
- Oracle 5G NFs can now be integrated into the Oracle Cloud Infrastructure (OCI) using OCI Adaptor to leverage the existing Oracle Cloud infrastructure. OCI provides a single-point platform for all Oracle products. Deploying CNC NFs on OCI allows operational efficiency, reduced Total Cost of Ownership (TCO), better performance, resource utilization, and persistence.

Overview

To leverage the existing infrastructure of Oracle Cloud, Oracle 5G NFs can now be integrated into the Oracle Cloud Infrastructure (OCI) using OCI Adaptor to leverage the existing Oracle Cloud infrastructure. OCI provides a single-point platform for all Oracle products. Deploying CNC NFs on OCI allows operational efficiency, reduced Total Cost of Ownership (TCO), better performance, resource utilization, and persistence.

You must perform OCI deployment tasks in the same sequence as outlined in the following table:

Table 1-1 OCI Deployment Task Sequence

Task	Subtask	References
Creating OCI Infrastructure		Creating OCI Infrastructure
Creating OCI User Management		Creating OCI User Management
	Creating User Groups	Creating User Groups
	Creating the User	Creating the User
	Creating the Auth Token to be Used as Registry Password	Creating the Auth Token to be Used as Registry Password
Deploying OCI Adaptor		Deploying OCI Adaptor
Configuring OCI Observability and Management		Configuring OCI Observability and Management
	Configuring NF Metrics Dashboard on OCI	Configuring NF Metrics Dashboard on OCI
	Configuring NF Alerts on OCI	Configuring NF Alerts on OCI
Deploying CNC NFs		Deploying CNC NFs
Uninstalling OCI Adaptor		Uninstalling OCI Adaptor

Table 1-1 (Cont.) OCI Deployment Task Sequence

Task	Subtask	References
Deleting OCI Infrastructure		Deleting OCI Infrastructure

References

The following references provide additional information on product operations, maintenance, and support:

- *Oracle Cloud Infrastructure Documentation*
- *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide.*
- *Oracle Communications Cloud Native Configuration Console Installation, Upgrade, and Fault Recovery Guide.*
- *Oracle Communications Cloud Native Core, cnDBTier Installation, Upgrade, and Fault Recovery Guide.*
- *Oracle Communications Cloud Native Core, Service Communication Proxy Installation, Upgrade, and Fault Recovery Guide.*
- *Oracle Communications Cloud Native Core, Network Repository Function Installation, Upgrade, and Fault Recovery Guide.*
- *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide.*
- *Oracle Communications Cloud Native Configuration Console User Guide*
- *Reference Architecture for CNC deployment on OCI*

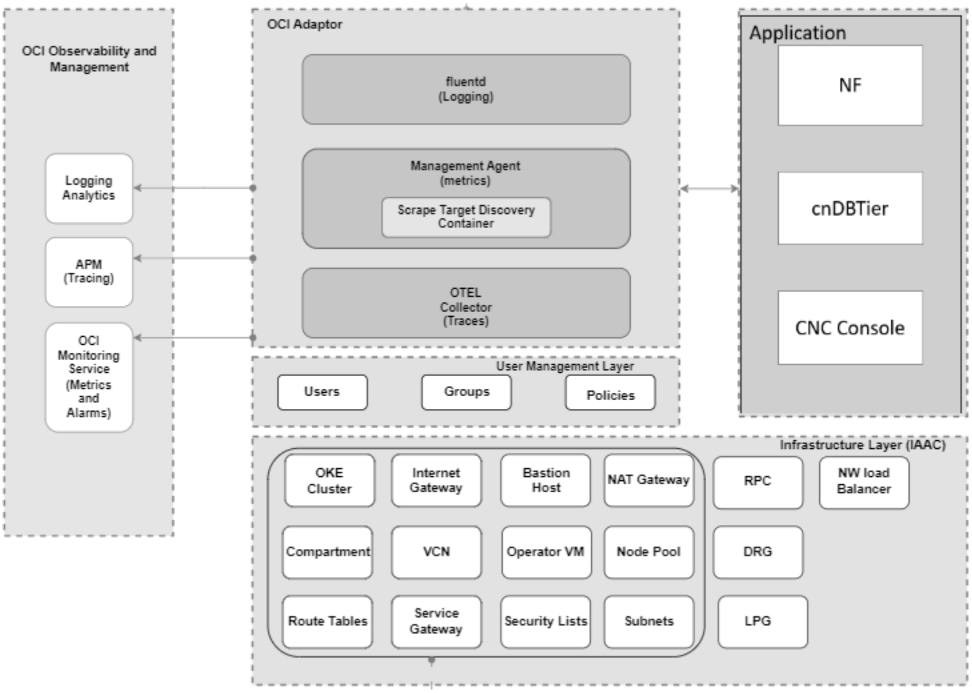
2 Deployment Model

The deployment model has the following four major components:

- OCI Infrastructure
- User Management
- OCI Adaptor
- OCI Observability and Management
- Application (Cloud Native Core Network Functions, cnDBTier, CNC Console)

The following diagram represents the deployment of Network Functions (NFs) on OCI:

Figure 2-1 Deployment of Network Functions (NFs) on OCI



Note

- OCI Adaptor and NF Applications (NFs, cnDBTier, and CNC Console) are deployed as a single OKE Cluster.
- The user must have OCI Tenancy and CNC NFs with OCI Adaptor to deploy the CNC NF on the OCI.
- The user can create a new compartment or use an existing compartment to deploy the OCI components.

Oracle Cloud Infrastructure

OCI provides high-performance computing capabilities (such as physical hardware instances) and storage capacity in a flexible overlay virtual network that is securely accessible from your on-premise network.

OCI infrastructure consists of Compartments, a Network Load balancer, a Bastion Host, a Dynamic Routing Gateway (DRG), a Remote Peering Connection (RPC), a Service Gateway, an Internet Gateway, and an OKE cluster, all in an OKE Virtual Cloud Network.

For creating the infrastructure layer, refer to Creating OCI Infrastructure section.

OKE Cluster

Oracle Kubernetes Engine (OKE) offers a carrier-grade container orchestration platform (based on Kubernetes) that supports the deployment requirements of cloud-native, containerized applications.

OKE is a fully managed and scalable service designed for deploying containerized applications in the cloud. It ensures high availability and reliability of cloud native applications. OKE utilizes Kubernetes to automate the deployment, scaling, and management of containerized applications.

OKE provides two types of clusters:

- **Basic cluster:** Basic clusters support all the core functionalities provided by Kubernetes and Container Engine for Kubernetes, but none of the enhanced features that Container Engine for Kubernetes provides. Basic clusters have a Service Level Objectives (SLOs) but not financially backed Service Level Agreements (SLAs).
- **Enhanced cluster:** Enhanced clusters support all the available features, such as virtual nodes, cluster add-on management, workload identity, and additional worker nodes per cluster. They also come with financially backed Service Level Agreements (SLAs).

CNC NFs can be deployed on basic or enhanced OKE clusters (through the Terraform scripts) with managed nodes. Since the NFs cannot be deployed on virtual nodes and virtual node pools, enhanced cluster is not required for CNC NF deployment. However, the user can always upgrade any basic cluster to an enhanced cluster but enhanced cluster cannot be downgraded.

For more details, refer to OKE Cluster and Enhanced Clusters and Basic Clusters.

Compartment

A compartment is a logical resource that helps achieve security isolation and controls resource access. CNC utilizes this resource to group CNC NF and its corresponding OCI resources and assigns users/groups with policies to access these resources, thus isolating access from the rest of the OCI Tenancy's users/groups.

For more details, refer to Managing Compartments.

Virtual Cloud Network

VCN is a networking service that helps create and manage a customizable and private network in the OCI cloud. A VCN service includes subnets, route tables, and security lists. CNC NF on OCI uses separate subnets for Bastion VM, Operator VM, Worker Nodes, and so on. Security lists define the Ingress and Egress rules for a subnet. Route tables assist in routing to a different network through gateways, such as sending CNC NF metrics data to the OCI Monitoring service.

For more details, refer to VCN Overview.

Network Load Balancer

The OCI Network Load Balancer (NLB) provides flexible automated traffic distribution from one entry point to multiple backend servers in your VCN. It operates at the connection level and load balances incoming client connections to healthy backend servers based on Layer 3/Layer 4 (IP protocol) data.

CNC uses the following LB annotations to route the incoming request to the appropriate backend. The annotations are:

- oci-network-load-balancer.oraclecloud.com/internal
- oci-network-load-balancer.oraclecloud.com/security-list-management-mode
- oci-network-load-balancer.oraclecloud.com/subnet
- oci.oraclecloud.com/load-balancer-type

For more details, refer to Network Load Balancer.

Dynamic Routing Gateway

A Dynamic Routing Gateway (DRG) acts as a virtual router, providing a path for traffic to flow between your on-premise networks and VCNs and can also route traffic between VCNs. CNC NF uses OCI DRG to send and receive messages to NFs residing on the customer's on-premise network and towards NFs residing in a different OCI Region. Each DRG attachment has an associated route table to route packets entering the DRG to their next hop.

For more details, refer to DRG documentation.

Bastion Host

Bastion Hosts allow authorized users to connect from specific IP addresses to target resources using SSH sessions. Connected users can interact with the target resource using any software or protocol supported by the SSH.

For more details, refer to Bastion Overview.

Operator VM

The Operator VM is a private VM that connects to the OKE Cluster. It allows you to perform user operations and run all `kubectl` commands. The user can log into the operator VM only through the Bastion Host, which is a public VM. The Operator VM uses OCI's instance principle mechanism to authenticate itself with the OKE cluster.

For more details, refer Instance Principals.

The Terraform scripts create a dynamic group, which allows the Operator VM to use the Instance Principal Mechanism to authenticate itself with the following matching rule and policy. For more information about the Terraform scripts, see Terraform Scripts for OCI Deployment.

```
Matching Rule
All {instance.compartment.id='COMPARTMENT_ID', tag.OPERATOR_VM's_TAG='true'}
Policy
Allow dynamic-group DYNAMIC_GROUP_NAME to manage cluster-family in
compartment COMPARTMENT_NAME
```

Node Pool

An OCI node pool is a group of nodes within a cluster that have the same configuration, such as a Linux image, number of OCPUs, RAM per node, boot volume, the applicable K8s version, and so on. CNC uses this node pool to create compute for the NFs. The advantage of this node pool is that a new compute is created with the same attributes when it is resized.

For more details, refer to OKE Node Pool.

Service Gateway

The service gateway provides access from a VCN to other OCI services, such as Oracle Cloud Infrastructure Object Storage. CNC uses Service Gateway to share metrics towards OCI Monitoring Services, share logs towards OCI Logging and Analytics and traces towards OCI APM data collector. The traffic from the VCN to the Oracle service does not use the internet. It uses the Oracle network fabric.

For more details, refer to Service Gateway.

User Management Layer

The user management layer allows you to create user groups and policies in a domain. There can be multiple identity domains in a tenancy:

- Default domain: Available by default and should only have limited number of admins who have administrative privileges on the tenancy level.
- Non-Default domains: All other application users.

The user needs to set up user groups in a domain.

There are three types of user groups:

- Tenancy Admin
- Compartment Admin

- Non-Admin Users

OCI Adaptor

The OCI Adaptor acts as a channel to transfer information between the application and OCI observability management. It assists CNC NFs in achieving observability and monitoring functions on the public cloud OCI platform.

The OCI Adaptor components include:

- Fluentd: It is a open source data collector software to collect log data.
- Management Agent, Scrape Target Discovery Container, and Metric Server are the components used to derive metrics.This component consists of the following sub-components:
  - OCI Management Agent
  - Scrape Target Discovery Container
  - Metric server
- OpenTelemetry (OTEL) Collector: OpenTelemetry (OTEL) Collector is the component to collect traces.

  To deploy the OCI Adaptor, see Deploying OCI Adaptor.

Fluentd

OCIs customized Fluentd is responsible for collecting the log data from the input sources, transforming the logs, and routing the log data to the OCI LA service.

For more details, refer to Fluentd.

A dynamic group is created, allowing Fluentd to upload the logs to the OCI LA service (part of the Observability Management layer) using the Terraform scripts based on the following matching rule and policy. For more information about the Terraform scripts, see Terraform Scripts for OCI Deployment.

```
Matching Rule:
ALL {instance.compartment.id = '<CNC NF's compartment_ocid>'}
Policy:
Allow dynamic-group DYNAMIC_GROUP_NAME to
{LOG_ANALYTICS_LOG_GROUP_UPLOAD_LOGS} in compartment COMPARTMENT_NAME
```

For more details, refer to Managing Dynamic Groups.

Management Agent, Scrape Target Discovery Container, and Metric Server

This section provides the information about the management agent, scrape target discovery container, and metric server. This is used for metrics.

Management Agent

The Management Agent is the central entity that collects metrics data from different services and sources that need to be monitored and uploads it into the OCI Monitoring Service over the REST interface.

For more details, refer to Management Agents.

A dynamic group is created that allows the management agent to upload metrics data to OCI's metric monitoring service using the Terraform scripts based on the following matching rule and policy. For more information about the Terraform scripts, see Terraform Scripts for OCI Deployment.

```
Matching Rule:
ALL {resource.type='managementagent',
resource.compartment.id='<AGENT_COMPARTMENT_OCID>'}
Policy:
allow dynamic-group DYNAMIC_GROUP_NAME to use metrics in compartment
<PROMETHEUS_METRIC_COMPARTMENT_NAME>
```

Metric Server

It is required to collect cAdvisor metrices. cAdvisor provides various insights into containers. CNC requires these details to monitor the health and performance of the containers running on a given node. For example, container_memory_usage_bytes, container_tasks_state.

OTEL Collector

OpenTelemetry (OTEL) is an observability framework which is used to capture and export metrics, logs, and traces. CNC applications use OpenTelemetry to create spans or traces. A trace is a collection of spans connected in a parent child relationship. The traces specify how APIs or requests are propagated through the microservices and other components. OpenTelemetry Collector is used to collect traces from the NFs and send them to OCI Application Performance Monitoring (APM).

For more information about APM, see Application Performance Monitoring (APM) (Tracing).

OCI Observability and Management

The OCI Observability and Management service provides dashboards and explorers for observing the metrics, logs, and traces of CNC Applications. These dashboards and explorers allow you to view, query, and study metrics, logs, and traces.

The Observability and Management module includes:

- OCI Monitoring Service (Metrics and Alarms)
- Application Performance Monitoring (APM) (Tracing)
- Logging Analytics

Integration of OCIs observability and management with NF applications is achieved using OCI Adaptor.

Logging Analytics

OKE cluster environment on a OCI tenancy comprises of three tiers from which logs are generated:

- Infrastructure tier comprising of worker nodes, networking resources.
- Kubernetes platform tier comprising of Kubelet, Core DNS and so on.

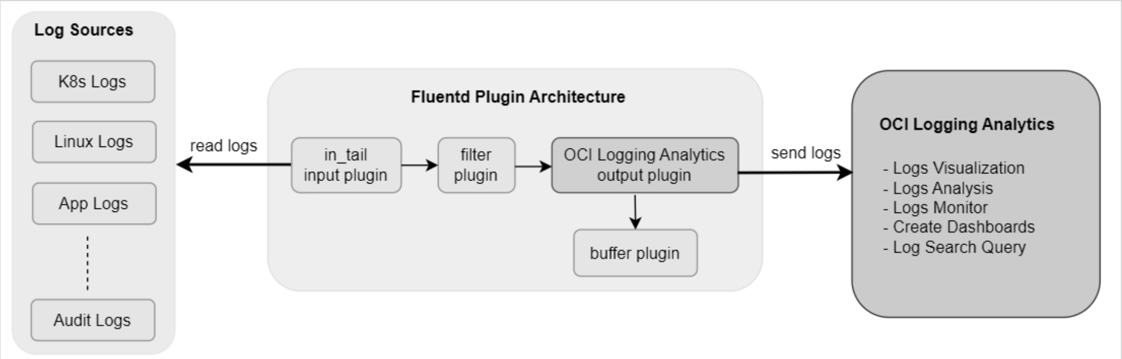
- Application tier comprising of applications and DB applications.

Oracle Cloud Infrastructure (OCI) provides two logging services, namely OCI Logging and OCI Logging Analytics.

Oracle Cloud Logging Analytics (LA), a cloud solution in OCI assists in indexing, aggregating, visualizing, searching and also monitoring all log data from your applications and system infrastructure.

OCI Logging Analytics supports integration or log ingestion by applications using either Management Agent or Fluentd. However, OCI recommends log ingestion using Fluentd.

Figure 2-2 Fluentd Plugin Architecture



The CNC shares logs towards OCI LA using Fluentd, the OCI LA is used to perform log analysis.

For more information, see the "Logging Analytics" section in *Oracle Cloud Infrastructure Documentation*.

Visualizing the Logs

The user can view the logs using the **Log Explorer** option in the Logging Analytics.

To access the Log Explorer:

1. Log in to **OCI Console**.
   For more information, see the "Signing In to the OCI Console" section in *Oracle Cloud Infrastructure Documentation*.
2. Open the navigation menu and click **Observability and Management**.
3. Under **Logging Analytics**, Click **Log Explorer**. The **Log Explorer** screen appears.
4. The user can customize the **Logs** view by choosing the **Visualizations** options.

For more information, see the "Visualize Data Using Charts and Controls" section in *Oracle Cloud Infrastructure Documentation*.

Application Performance Monitoring (APM) (Tracing)

The OCI Application Performance Monitoring (APM) is a service that provides deep visibility into the performance of applications and offers the ability to diagnose issues quickly. APM collects and processes the transaction instance trace data using Application Performance Monitoring data sources, open source tracers, or directly using API. It accepts OpenTelemetry spans and combines them into traces. CNC Applications microservices share spans or traces towards the OCI APM that help troubleshoot the application.

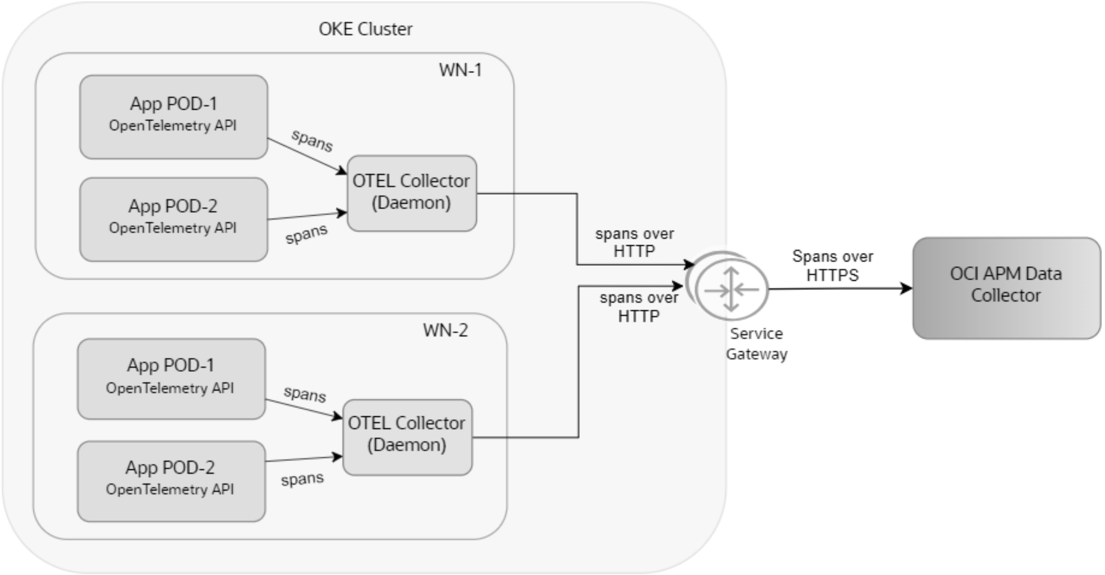
CNC Applications use OpenTelemetry to create spans or traces. A trace is a collection or list of spans connected in a parent or child relationship.

Traces specify how APIs or requests are propagated through the microservices and other components, assisting operations and developers in troubleshooting issues in a customer deployment.

OCI Application Performance Monitoring (APM) provides a comprehensive set of features to monitor applications and diagnose performance issues.

For more information, see the "Application Performance Monitoring" section in *Oracle Cloud Infrastructure Documentation*.

Figure 2-3 Trace Collection towards OCI APM Data Collector



Monitoring the Traces

The user can monitor the traces using the **Trace Explorer** option in the Application Performance Monitoring service.

To access the Trace Explorer:

1. Log in to **OCI Console**.
   For more information, see the "Signing In to the OCI Console" in *Oracle Cloud Infrastructure Documentation*.
2. Open the navigation menu and click **Observability and Management**.
3. Under **Application Performance Monitoring**, click **Trace Explorer**.
4. The **Tracing Explorer** screen appears.
5. The user can customize the view of traces view by choosing the different options available.

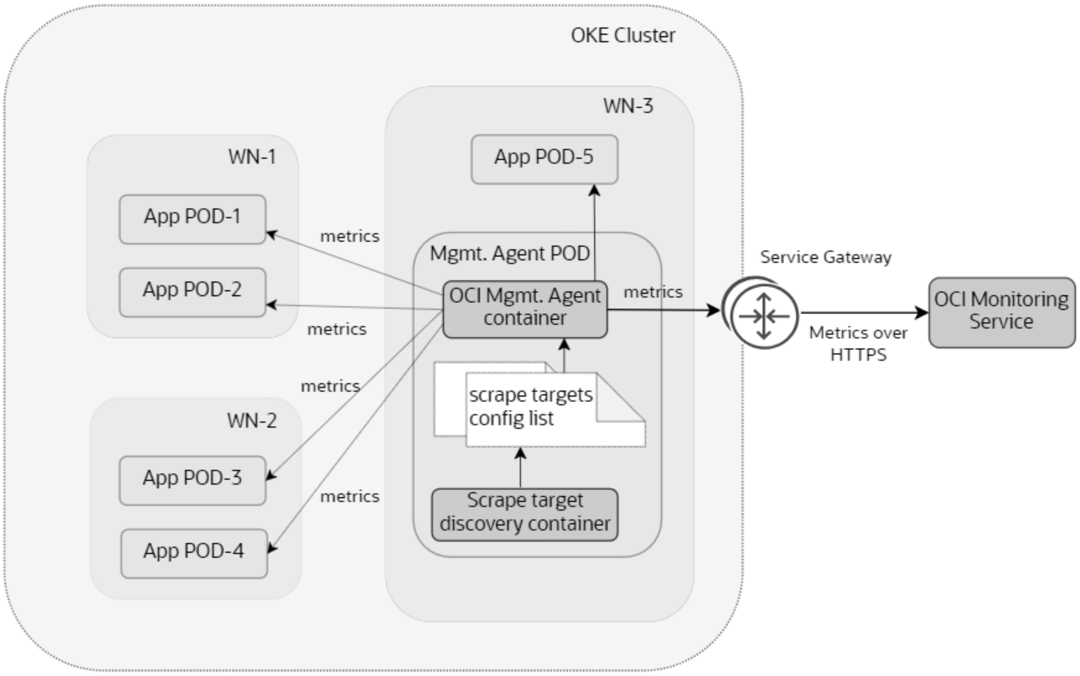
For more information, see the "Monitor Traces in Trace Explorer" section in *Oracle Cloud Infrastructure Documentation*.

OCI Monitoring Service (Metrics and Alarms)

Management Agent of OCI Adaptor fetches metrics from the scrape targets and publishes them towards the OCI Monitoring Service.

The following diagram briefly outlines the integration and use of Management Agent in fetching the metrics from the targets and publishing the same towards Monitoring Service.

Figure 2-4 OKE Cluster Monitoring with Management Agent



The Oracle Cloud Infrastructure (OCI) Monitoring service actively and passively monitors NF applications and cloud resources using the metrics and alarms features. CNC shares metrics with the OCI Monitoring service. These metrics help render various dashboards on OCI that eventually help monitor the CNC Applications. These metrics also assist in generating alerts for the customer. The management agent collects the metrics data from different services and sources that have to be monitored and uploads it to the OCI Monitoring Service.

For more information, see the "Monitoring" section in *Oracle Cloud Infrastructure Documentation*.

> ⓘ **Note**
>
> To view metrics and create the alarms, write the metrics queries using the Monitoring Query Language (MQL) syntax.
> For more information, see the "Monitoring Query Language (MQL) Reference" section in *Oracle Cloud Infrastructure Documentation*.

Viewing the Metrics

The user can view the metrics using the Metrics Explorer option in the OCI Monitoring Service.

To access the metrics explorer:

1. Log in to **OCI Console**.
   For more information, see "Signing In to the OCI Console" section in *Oracle Cloud Infrastructure Documentation*.
2. Open the navigation menu and click **Observability and Management**.
3. Under **Monitoring**, click **Metrics Explorer**. The **Metrics Explorer** screen appears.
4. The user can customize the metrics view by choosing the customizations options.

For more information, see the "Creating a Basic Query" section in *Oracle Cloud Infrastructure Documentation*.

Creating the Alarms or Alerts

The user can create the alarms using the OCI Monitoring Service.

For more information, see the "Configuring NF Alerts on OCI" section in *Oracle Communications Cloud Native Core, OCI Deployment Guide*.

Creating the Alarms

The user can create the alarms in the OCI Monitoring Service.

To create the alarms:

1. Log in to **OCI Console**.
   For more information, see the "Signing In to the OCI Console" section in *Oracle Cloud Infrastructure Documentation*.
2. Open the navigation menu and click **Observability and Management**.
3. Under **Monitoring**, click **Alarm Definitions**.
4. Click **Create Alarm**. The **Create Alarm** page opens in **Basic mode** (the default view).
   The following steps describe how to create an alarm in **Basic mode**:
   a. Enter a user-friendly name for the alarm. This name is sent as the title for notifications related to this alarm.
   b. For **Alarm severity**, select the perceived type of response required when the alarm is in the firing state.
   c. For **Alarm body**, provide user-readable notification content.
5. The user can customize the alarms by choosing the provided customizations options.

For more information, see the "Managing Alarms" section in *Oracle Cloud Infrastructure Documentation*.

Creating a Custom Dashboard on OCI

The user can create custom dashboards on OCI, such as the APM dashboard for visualizing the traces and the Logging Analytics dashboard for visualizing the logs and metrics.

For more information, see the "Create a Custom Dashboard", "Create a Query-based Widget Using Metrics" and "Create a Query-based Widget Using Traces" sections in *Oracle Cloud Infrastructure Documentation*.

The user can export a custom dashboard, widgets, and filters and import it to a different tenancy or region.

For more information, see the "Export and Import Dashboards" section in *Oracle Cloud Infrastructure Documentation*.

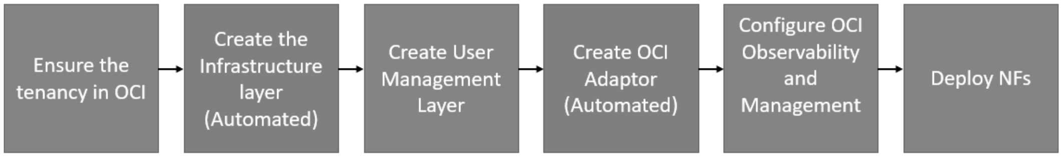
> **Note**
>
> The following are the available sample dashboards:
>
> - **Logging Dashboard**: Displays the Kubernetes Workloads, Kubernetes Cluster Summary, Kubernetes Nodes, Kubernetes Pods.
> - **Metric Dashboard**: This is a sample Kubernetes Monitoring dashboard. It is a part of the Logging Analytics dashboard.
> - **APM or Tracing Dashboard**: This is a sample APM or Tracing dashboard. It is part of the APM dashboard.
>
> By default, sample dashboards for logging, tracing, and metrics are created as part of OCI Adaptor terraform scripts. The user can refer to the sample dashboards to create personalized dashboards.
>
> The sample APM or Tracing dashboard is located as part of APM dashboards. Metric and Logging dashboards are located as part of Logging Analytics dashboards.

For more information, see the "OCI Monitoring Service" section in *Oracle Communications Cloud Native Core, OCI Adaptor User Guide*.

3 Deploying NFs in OCI

The following diagram represents the process for NF deployment in OCI:

Figure 3-1 Process for NF deployment in OCI


Ensure the tenancy in OCI
Create the Infrastructure layer (Automated)
Create User Management Layer
Create OCI Adaptor (Automated)
Configure OCI Observability and Management
Deploy NFs

The User must perform the given procedures to deploy the NF:

- Ensure the OCI tenancy is available.
- Create an infrastructure layer along with the components. To create an infrastructure layer, see Creating OCI Infrastructure.
- Create Admin and Non-Admin User Groups. To create User and User Groups, see Creating OCI User Management.
- Deploy the OCI Adaptor. The Adaptor acts as a channel to transfer information between the application and OCI observability management. To deploy the OCI Adaptor, see Deploying OCI Adaptor.
- Configure OCI Observability and Management. You can observe analytics and performance through OCI Observability and Management. To configure the OCI Observability and Management, see Configuring OCI Observability and Management.
- Deploy the CNC NFs.

Prerequisites

Before installing and configuring OCI Adaptor, ensure the following:

- The user has OCI tenancy.
- The user has **tenancy-admin** access.
- A **tenancy-admin** user runs the Terraform Stack, which is responsible for creating the infrastructure and deploying OCI Adaptors.

OCI Adaptor Resource Requirements

This section lists the resource requirements to install and run OCI Adaptor.

Table 3-1 Consolidated Resource Profile for OCI Adaptor Deployment

Resource Requirement	Service	
	Min	Max
CPU	500m	800m
Memory	1.5Gi	2.5Gi
Ephemeral Storage	400Mi	400Mi

The following table lists resource requirement for OCI Adaptor Services:

Table 3-2 Resource Profile for OCI Adaptor

Service Name	CPU		Memory (GB)		POD		Ephemeral Storage	
	Min	Max	Min	Max	Min	Max	Min	Max
oci-adaptor-opentelemetry-collector-agent	100m	100m	500Mi	500Mi	1	*	100Mi	100Mi
oci-onm-mgmt-agent-0	200m	500m	500Mi	1Gi	1	1	100Mi	100Mi
sepp-oci-adaptor-logan-8g9fl	100m	100m	250Mi	500Mi	1	*	100Mi	100Mi
metric-server	100m	100m	250Mi	500Mi	1	1	100Mi	100Mi
Total	500m	800m	1.5Gi	2.5Gi	4	NA	400Mi	400Mi

- * Both OTEL Collector and Fluentd logan are deployed as a daemonset. Thus, the maximum number of pods depends on the number of worker nodes in any cluster.

Terraform Scripts for OCI Deployment

Terraform is an Infrastructure as Code (IaC) tool that allows users to build, change, and version the cloud and on-premise resources safely and efficiently. Following are the Terraform Scripts (Infrastructure Automation Script) provided to automate the OCI deployment steps:

- **ocociadaptor_csar_<version>.zip**
  This package is in the standard CSAR format and contains the Terraform scripts to create the OCI infrastructure and deploy OCI Adaptors. It also includes OCI Adaptor images and Helm charts.

  The package is available to download on MOS. Users cannot upload the CSAR package directly to OCI's Resource Manager stack. Therefore, unzip the CSAR package to extract the Terraform scripts and then upload the scripts to the resource manager stack. Within the scripts directory, the following Terraform scripts are present:

  For more information on how to download, contact MOS.

  - **ocociadaptor_infra_create_<version>.zip**
    This package contains the Terraform scripts responsible for creating the infrastructure.

- **ocociadaptor_install_<version>.zip**
  This package contains the Terraform scripts, Helm charts, and Shell scripts to deploy the OCI Adaptors. Administrators can directly upload this package as the OCI's Resource Manager Stack and deploy the OCI Adaptors.

> **Note**
>
> The package is available for download at MOS.

Installation Sequence

This chapter provides information about deploying the NFs in the OCI environment.

Creating Identity Domain

This section explains how to create the identity domain.

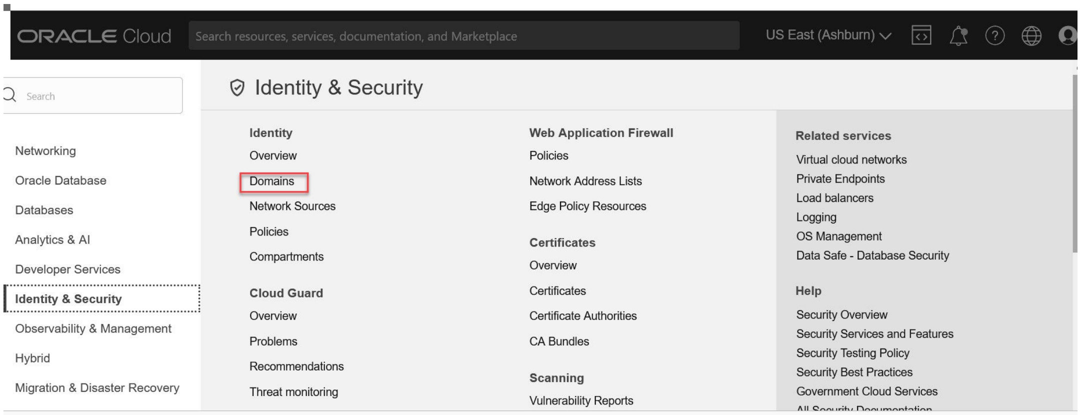
An identity domain is a container for managing users and roles, federating and provisioning users, secure application integration through *Oracle Single Sign-On* (SSO) configuration, and *SAML* or *OAuth*-based Identity Provider administration. It represents a user population in Oracle Cloud Infrastructure and its associated configurations and security settings.

> **Note**
>
> You can use the default domain or create a new domain (recommended).

The following are the steps to create the identity domain:

1. Log in to the **OCI Console**.
   For more information, see the "Signing In to the OCI Console" section in *Oracle Cloud Infrastructure Documentation*.
2. Open the navigation menu and select **Identity and Security**. The **Identity and Security** page appears.
3. Under **Identity**, select **Domains**. The **Domains** page appears.

Figure 3-2 Identity Domain



4. Click **Create Domain** on the right pane.
5. On the **Create Domain** page, assign a name to the domain and enter a description.
6. Select **Free** in Domain Type.
7. Enter the details of the Identity Domain **Administrator** and select the **Compartment**.
8. Click **Create Domain**.

Creating OCI Infrastructure

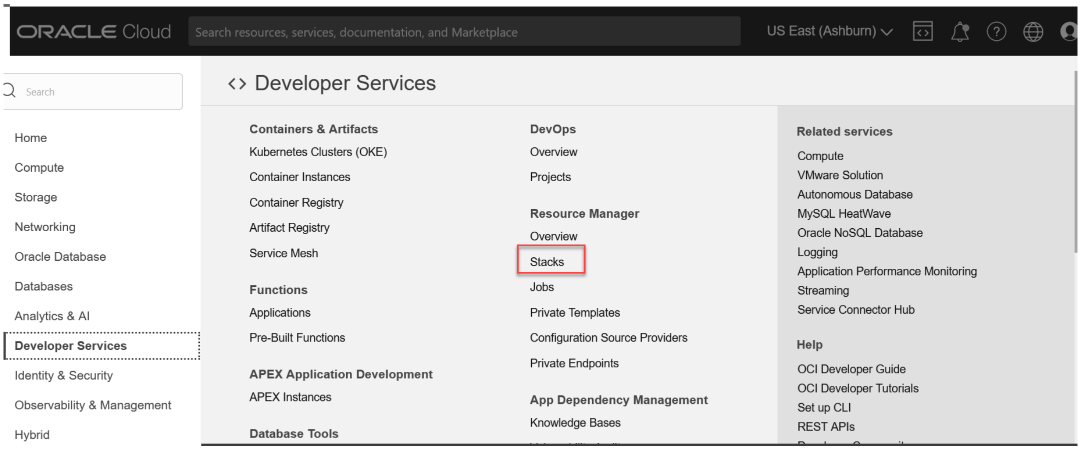
This section describes how to create the OCI infrastructure.

> **Note**
>
> - This procedure creates one OKE Cluster, and all the necessary platform components required to run OCI Adaptor and CNC NFs.
> - To create the Oracle Cloud Infrastructure (OCI), it is recommended that first-level subcompartments be created in the root compartment. The creation of OCI infrastructure at the second-level subcompartment is not supported.
> - Use the terraform tool to set up the necessary infrastructure components, including the OKE Cluster, Bastion Service, Compartment,CLI Server, and Virtual Cloud Network (VCN). Run the appropriate version-specific script `(ocociAdaptor_infra_create_<version>.zip)`.
> - The terraform script does not create the Network Load Balancers (NLB), Dynamic Routing Gateways (DRG), and Remote Peering Connections (RPC). You must create the NLB, DRGs, and RPCs manually.

The following are the steps to create OCI Infrastructure:

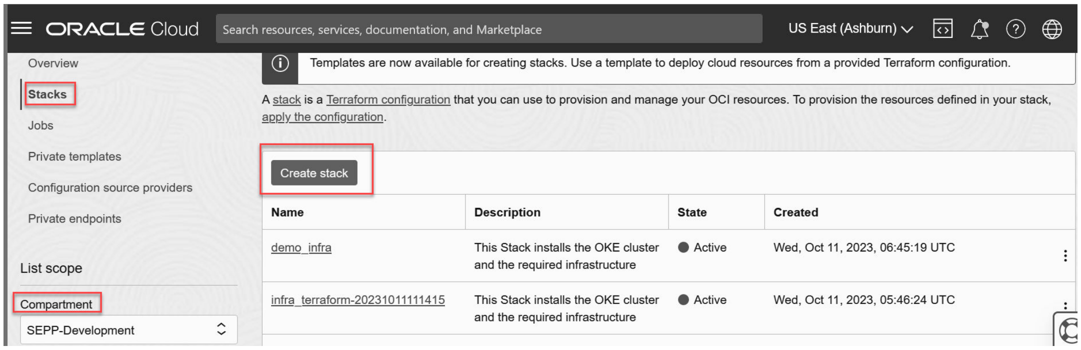
1. Log in to the **OCI Console.**
   For more information, see the "Signing In to the OCI Console" section in *Oracle Cloud Infrastructure Documentation*.
2. Open the navigation menu and select **Developer Services**. The **Developer Services** window appears in the right pane.
3. In **Developer Services**, select **Resource Manager**.
4. Under **Resource Manager**, select **Stacks**.

Figure 3-3 Create Stack



5. In the stack window, select **Compartment**.
6. Click **Create stack** on the right pane.

Figure 3-4 Create Stack



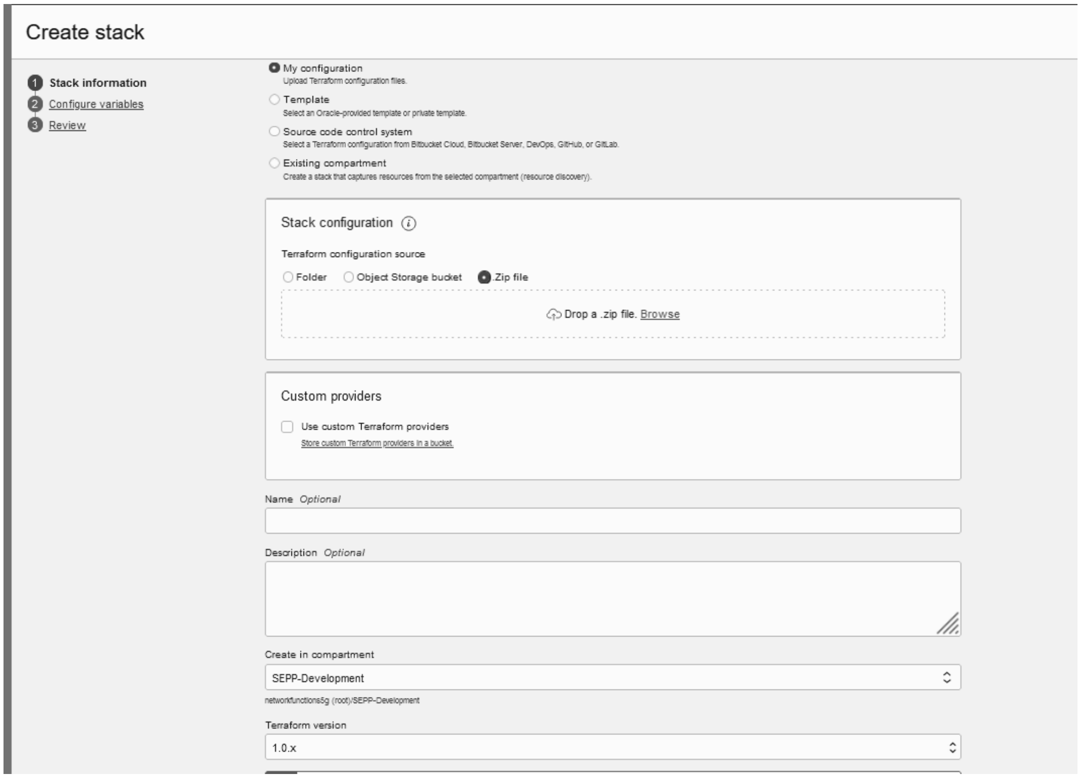
Templates are now available for creating stacks. Use a template to deploy cloud resources from a provided Terraform configuration.

A stack is a Terraform configuration that you can use to provision and manage your OCI resources. To provision the resources defined in your stack, apply the configuration.

Name	Description	State	Created
demo_infra	This Stack installs the OKE cluster and the required infrastructure	● Active	Wed, Oct 11, 2023, 06:45:19 UTC
infra_terraform-20231011111415	This Stack installs the OKE cluster and the required infrastructure	● Active	Wed, Oct 11, 2023, 05:46:24 UTC

7. Click the default **My configuration** radio button.
8. Under **Stack configuration**, select the **.Zip file** radio button and upload the **ocociadaptor_infra_create_<version>.zip** file.

Figure 3-5 Stack Configuration



Create stack

1 Stack information
2 Configure variables
3 Review

● My configuration
Upload Terraform configuration files.
○ Template
Select an Oracle-provided template or private template.
○ Source code control system
Select a Terraform configuration from Bitbucket Cloud, Bitbucket Server, DevOps, GitHub, or GitLab.
○ Existing compartment
Create a stack that captures resources from the selected compartment (resource discovery).

Stack configuration

Terraform configuration source
○ Folder ○ Object Storage bucket ● Zip file

Drop a .zip file. Browse

Custom providers

□ Use custom Terraform providers
Store custom Terraform providers in a bucket.

Name *Optional*

Description *Optional*

Create in compartment
SEPP-Development
networkfunctions5g (root)/SEPP-Development

Terraform version
1.0.x

9. Enter the **Name** and **Description** and select the **compartment**.
10. Click **Next**. The **Edit stack** screen appears.
11. Enter the required inputs to create the infrastructure layer components and click **Save** and **Run apply**.
    The inputs required are as follows:
    a. **Create Stack**
        i. **Name**: Enter the name of the Stack.
        ii. **Description**: Provide a description for your stack.
        iii. **Compartment**: Specify the compartment name.
        iv. **Terraform** (Select the latest one).
    b. **Tenancy Configuration**
        i. **Identity Domain Name**: Enter the domain name.
        ii. **Tenancy Home Region Identifier**: Home Region Identifier of the tenancy. Eg - us-ashburn-1.
        For more information, see the "Regions and Availability Domains" section in *Oracle Cloud Infrastructure Documentation*.
        iii. **Enclosing Compartment ID**: Specify the ID of the parent compartment.
        iv. **Compartment Name**: Create a new compartment or select an existing one.
        v. **Compartment tag Namespace**: Enter an alphanumeric string to tag your instance.
        vi. **Identity Domain URL**: Enter the Identity Domain URL where the dynamic groups are to be created. For more information, see the Getting Identity Domain URL.

c. **VCN Configuration**

   i. **VCN Name**: Enter the name of your Virtual Cloud Network (VCN).

   ii. **CIDR Block**: Provide the CIDR (Classless Inter-Domain Routing) block for your VCN.

d. **Cluster Configuration**

   i. **Cluster Name**: Enter the name of your OKE Cluster.

   ii. **Kubernetes Version**: Specify the version of Kubernetes you are using.

   iii. **Node Pool Size**: Set the size of your Node Pool.

   iv. **Node Pool Shape**: Choose the shape of your Node Pool.

   v. **Node Pool Image**: Select the image for your Node Pool.

   vi. **OCPUs**: Define the number of Oracle CPUs.

   vii. **Memory (GB)**: Input the memory capacity.

   viii. **Node Pool Boot Volume Size**: Set the boot volume size for the Node Pool.

   ix. **Public Key**: Enter the public key.

e. **CLI Server Configuration**

   i. **Public Key**: Enter the public key.

   ii. **Private Key**: Enter the private key.

12. After entering the values, user needs to click next and then click save changes button.

13. Plan and apply the terraform stack and it will create the required infrastructure.

> **Note**
>
> The `Run apply` option is provided along with saving the stack itself, but it is recommended to first execute plan on your stack and then apply it.

> **Note**
>
> Important:
>
> - To use the OKE Cluster created as a part of the OCI infrastructure and to deploy the OCI Adaptor, install the following components in the CLI Server:
>   - `kubectl`
>   - `Helm`
> - Select the kubectl version based on the Kubernetes version installed in the OKE cluster.

Creating OCI User Management

This section describes how to create the users. User and Groups must be created manually. The users can be created and grouped together.

The `Tenancy Admin` is responsible for creating `Compartment Admins` and other users and user groups.

> **Note**
>
> Now the infrastructure creation terraform can be executed by both `Tenancy Admin` and `Compartment Admins`.

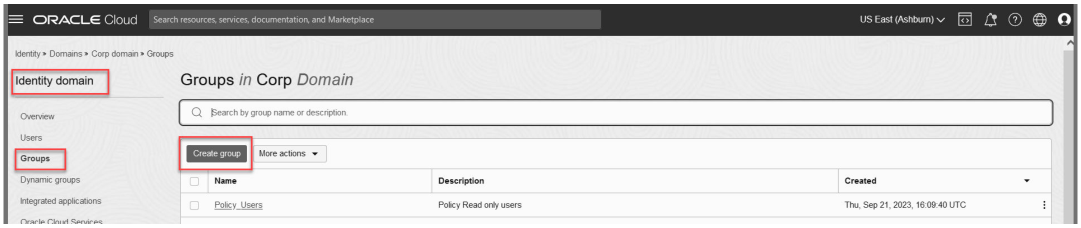
To enable `Compartment Admin`, create infrastructure using provided terraform. Following polices needs to applied on the `Compartment Admins User Groups`.

- Allow group `<DOMAIN_NAME>/<COMPARTMENT_ADMIN_USER_GROUP>` to manage all-resources in compartment `<COMPARTMENT_NAME>`.
- Allow group `<DOMAIN_NAME>/<COMPARTMENT_ADMIN_USER_GROUP>` to manage dynamic-groups in tenancy where `target.resource.domain.name=<DOMAIN_NAME>`.

Creating User Groups

This section describes the steps to create the user groups.

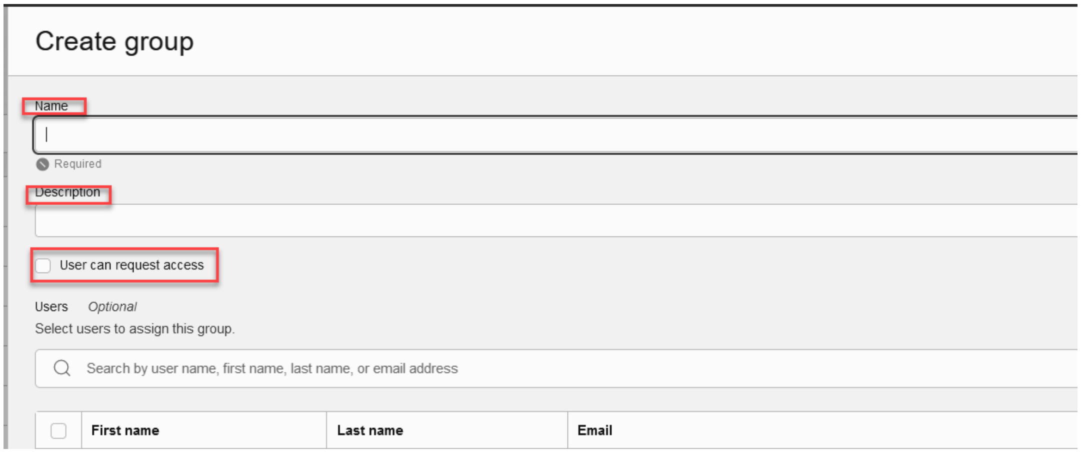
1. Log in to the **OCI Console**.
   For more information, see the "Signing In to the OCI Console" section in *Oracle Cloud Infrastructure Documentation*.
2. Open the navigation menu and click **Identity and Security**. Under **Identity**, click **Domains**.
3. Select the **Identity Domain** that you want to work in. Change the compartment to find the required domain. Then, click **Groups**.
4. Click **Create Group**.



Groups *in* Corp *Domain*

	Name	Description	Created
□	Policy_Users	Policy Read only users	Thu, Sep 21, 2023, 16:09:40 UTC

In the **Name** and **Description** fields on the **Create Group** window, enter the **Name** and **Description** about the group.



Create group
Name
Required
Description
□ User can request access
Users Optional
Select users to assign this group.
Search by user name, first name, last name, or email address
□ First name
Last name
Email

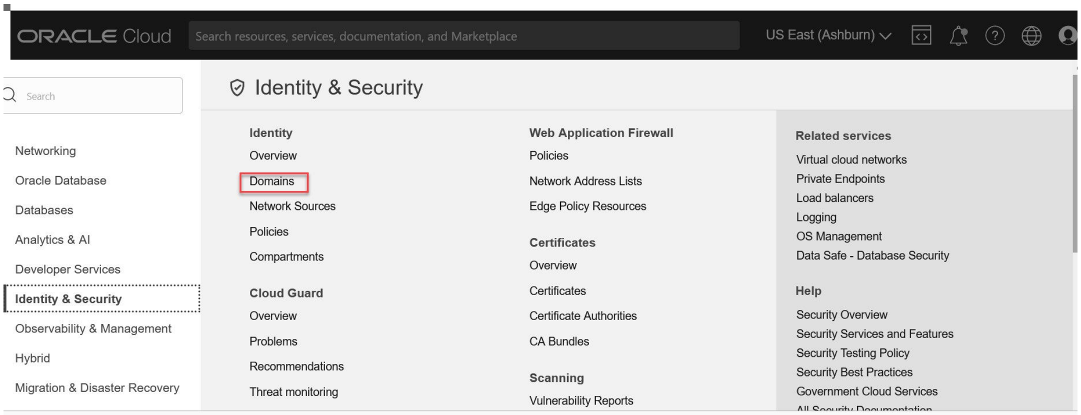
5. To allow users to request access to this group, select **User can request access**.
6. To add users to the group, select the check box for each user that you want to add to the group.
7. Click **Create**.

The **User Group** is created.

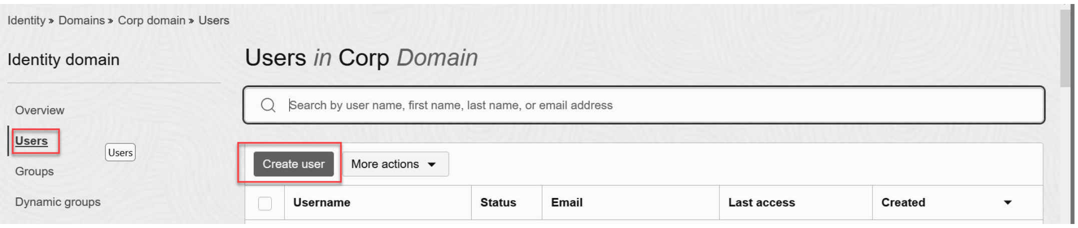
Creating the User

The following are the steps to create a user account for a user in an OCI IAM identity domain:

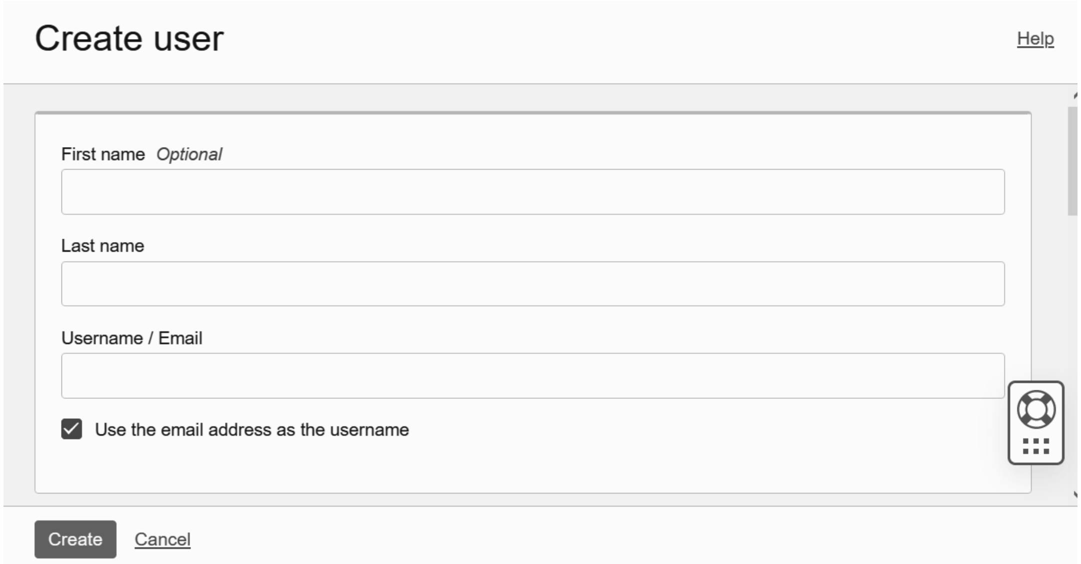
1. Open the navigation menu and click **Identity and Security**. Under **Identity**, click **Domains**.



2. Select the **Identity domain** that you want to work in. Change the compartment to find the required domain. Then, click **Groups**.



3. Click **Create user**.



☑ Use the email address as the username

4. In the **First name** and **Last name** fields, enter the user’s name.
5. To sign in using email address, follow the following steps:
   a. In the **Username or Email** field, enter the email address for the user account.
   b. Leave the **Use the email address as the username** check box selected.
6. Alternatively, to sign in with the username, follow the steps below:
   a. In the **Username or Email** field, enter the username.
   b. Clear the **Use the email address as the username** check box.
      The following characters are allowed in the **Username or Email** field:
      i. a-z
      ii. A-Z
      iii. 0-9
      iv. Special characters! @ # $ % ^ & * ( ) _ + = - { } [ ] | \ : " ' ; < > ? / . ,
      v. Blank spaces
   c. In the **Email** field, enter the email address.
7. To assign the user to a group, select the check box for each group that you want to assign to the user account.
8. Click **Create**.

   The user account is created.

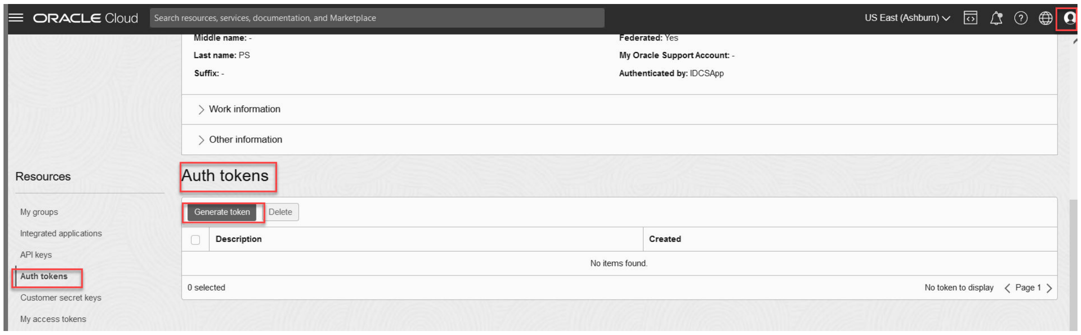
Creating the Auth Token to be Used as Registry Password

This section describes how to create the Auth token to be used as registry password.

1. Log in to the **OCI console**.
   For more information, see "Signing In to the OCI Console" section in *Oracle Cloud Infrastructure Documentation*.
2. Click **User** on the top right corner.
3. Click **My profile**.
4. Click **Auth tokens**.

5. Click **Generate token**.

Figure 3-6 Create Auth Token



Middle name: -
Last name: PS
Suffix: -
Federated: Yes
My Oracle Support Account: -
Authenticated by: IDCSApp

□	Description	Created
No items found.		

0 selected

6. Enter the **Description**.
7. Click **Generate token**.

The token is created.

> **Note**
>
> Copy the generated token. Ensure to preserve that as that will not be displayed again.

Deploying OCI Adaptor

This section describes how to deploy the OCI Adaptor.

Create the OCI Adaptor components, which are the Fluentd, Management Agent (Management agent, Scrape Target Discovery Container, and Metrics Server), and OTEL Collector. This step is automated using the Terraform`oci_adaptor_install_<version>.zip`.

Following are the steps to create the components of the OCI Adaptor:

1. Move all the tar files in the file's directory of `ocociadaptor_csar_<version>.zip` file to operator instance at ***/home/<OPERATOR_USER>/oci_adaptor***. Create the path if it does not exist.

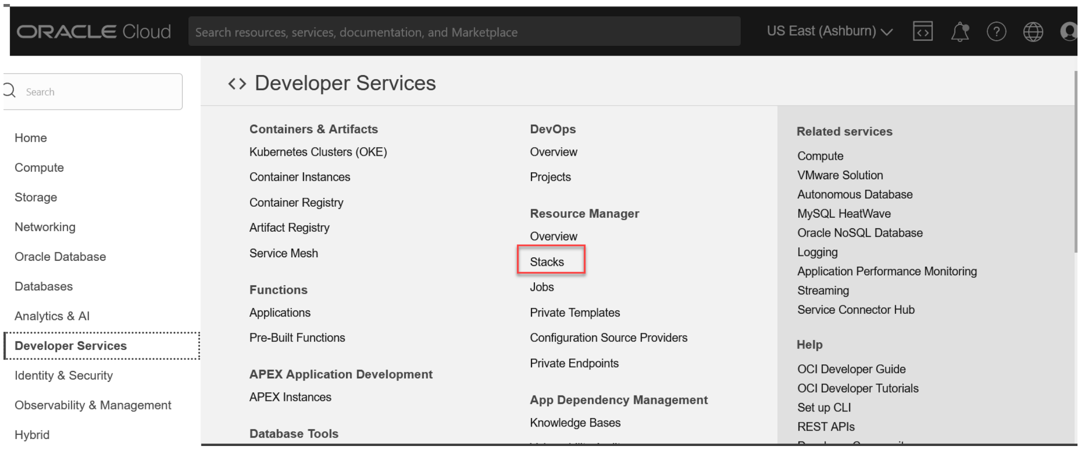
   > **Note**
   >
   > Since the operator instance is only accessible from Bastion Host, the user needs to copy the tar files to the Bastion Host first, and then from the Bastion Host to the operator instance. The private key to access Bastion Host and operator instance will be the same.

2. Log in to the **OCI Console.**

   > **Note**
   >
   > For more details about logging in to the OCI, refer to Signing In to the OCI Console.

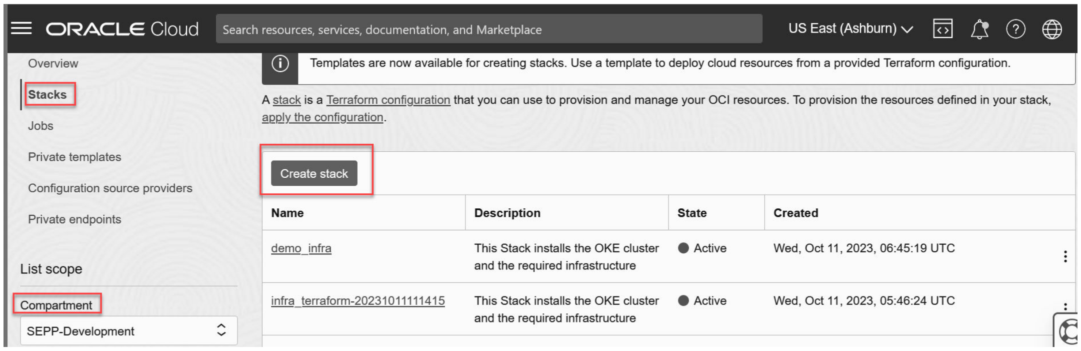
3. Open the navigation menu and select **Developer Services**. The **Developer Services** window appears in the right pane.
4. Under the **Developer Services**, select **Resource Manager**.
5. Under **Resource Manager**, select **Stacks**. The **Stacks** window appears.

Figure 3-7 Developer Services



6. Click **Create Stack**.

Figure 3-8 Create Stack

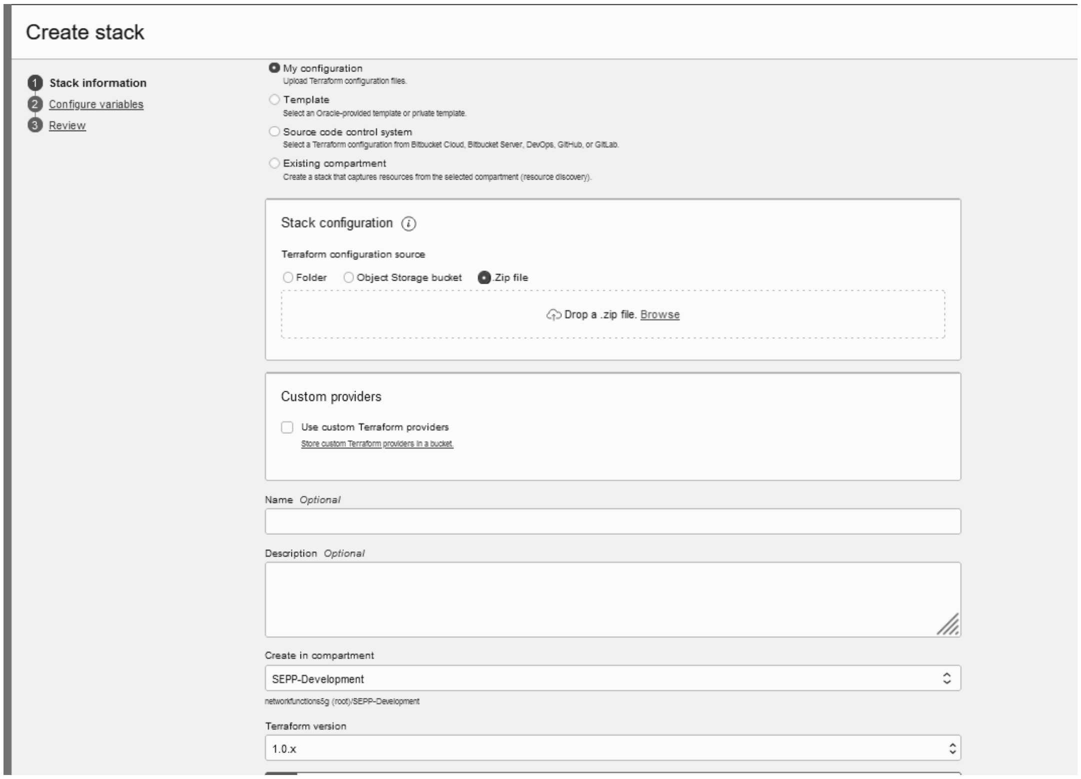


Templates are now available for creating stacks. Use a template to deploy cloud resources from a provided Terraform configuration.

A stack is a Terraform configuration that you can use to provision and manage your OCI resources. To provision the resources defined in your stack, apply the configuration.

Name	Description	State	Created
demo_infra	This Stack installs the OKE cluster and the required infrastructure	● Active	Wed, Oct 11, 2023, 06:45:19 UTC
infra_terraform-20231011111415	This Stack installs the OKE cluster and the required infrastructure	● Active	Wed, Oct 11, 2023, 05:46:24 UTC

7. Select the default **My Configuration** radio button.
8. Under **Stack configuration**, select the .zip file radio button and upload the `oci_adaptor_install_<version>.zip` file.



Create stack

1 Stack information
2 Configure variables
3 Review

● My configuration
Upload Terraform configuration files.

○ Template
Select an Oracle-provided template or private template.

○ Source code control system
Select a Terraform configuration from Bitbucket Cloud, Bitbucket Server, DevOps, GitHub, or GitLab.

○ Existing compartment
Create a stack that captures resources from the selected compartment (resource discovery).

Stack configuration

Terraform configuration source

○ Folder ○ Object Storage bucket ● .Zip file

Drop a .zip file. Browse

Custom providers

□ Use custom Terraform providers
Store custom Terraform providers in a bucket.

Name *Optional*

Description *Optional*

Create in compartment: SEPP-Development
networkfunctions5g (root)/SEPP-Development

Terraform version: 1.0.x

9. Enter the **Name** and **Description** and select the **compartment**.
10. Select the latest Terraform version from the **Terraform version** drop-down.
11. Click **Next**. The **Edit Stack** screen appears.
12. Enter the required inputs to create the infrastructure layer components and click **Save** and **Run Apply.**
13. Verify that the OCI Adaptors have been installed in the namespace provided.
    The required inputs are:
    - **Tenancy Configuration**
      - **Identity Domain Name -** Name of the domain.
      - **Compartment Name -** Name of the Compartment.
    - **Identifiers**
      - **NF Name -** Name of the NF.
      - **Unique Identifier -** Unique identifier for each NF in the timestamp format.
    - **User Groups**
      - **Admin Group Name -** The admin group name.
      - **Non-Admin Group Name -** The non-admin group name.
    - **Cluster Configuration**
      - **Cluster Name -** Name of the OKE Cluster.
      - **Cluster OCID -** All the cluster IDs in the compartment will be displayed in this drop-down.
    - **Baston Configuration**
      - **IP address -** IP address of bastion Instance.

    - **IP address for the operator instance** - IP address for the operator instance.
    - **Private Key** - Private key to login into Bastion. Enter the Key in Base64 encoded format.
- **OCIR Registry Configuration**
    - **Registry Name**- The name registry name.
    - **Registry Username**- The name registry username.
    - **Registry Password**- The password registry username.
- **OCI Adaptors Generic Configuration**
    - **OCI Adaptor Namespace**
- **Management Agent/Metrics Configuration**
    - **Max Management Agent Install Count** - User Configurable (The upper limit is 1000).
    - **Management Agent Key Expire Time** - The time in the specified format.
    - **NF Metric Path** - The metric path will be populated.
    - **Metric Namespace** - The metrics namespace.

The OCI Adaptor component is created.

> ⓘ **Note**
>
> The **Plan** option is provided along with the **Run Apply**. The (optional) **Plan** step provides a view of the steps that are going to be performed while creating the stack. This is recommended.

Configuring OCI Observability and Management

This section describes how to configure OCI Observability and Management.

> ⓘ **Note**
>
> By default, sample dashboards for Logging, tracing, and metrics are created as part of OCI Adaptor Terraform scripts. The user can refer to the sample dashboards to create personalized dashboards.
> The following are the available sample dashboards:
>
> - **Logging Dashboard:** Displays the Kubernetes Workloads, Kubernetes Cluster Summary, Kubernetes Nodes, Kubernetes Pods.
> - **Metric Dashboard:** This is a sample Kubernetes Monitoring dashboard. It is a part of the Logging Analytics dashboard.
> - **APM/Tracing Dashboard:** This is a sample APM/Tracing dashboard. It is part of the APM dashboard.
>
> The sample APM/Tracing dashboard is located as part of APM dashboards. Metric and Logging dashboards are located as part of Logging Analytics dashboards.

Configuring NF Metrics Dashboard on OCI

This section describes about the steps to upload the NF specific json file (Example: `<NF>_oci_dashboard_<version>.json`) file on OCI Logging Analytics Dashboard Service. As OCI doesn't support Grafana, OCI uses the Logging Analytics Dashboard Service for visualizing the metrics and logs.

Follow the steps below:

1. Log in to **OCI Console**.

   > **Note**
   >
   > For more information, see the "Signing In to the OCI Console" section in *Oracle Cloud Infrastructure Documentation*.

2. Open the navigation menu and click **Observability and Management**.
3. Under **Logging Analytics**, Click **Dashboards**. The **Dashboards** page appears.
4. Choose the **Compartment** in the left pane.
5. Click **Import dashboards**.
6. Select and upload the `<NF>_oci_dashboard_<version>.json` file. Customize the following three parameters of JSON file before uploading it:
   a. COMPARTMENT_ID: The OCID of the compartment.
   b. METRIC_NAMESPACE: The metrics namespace that the user provided while deploying OCI Adaptor.
   c. K8_NAMESPACE: Kubernetes namespace where SEPP is deployed.
7. **Import dashboard** page appears. Click **Import** button on the page. Users can view the imported dashboard and the metrics on the dashboard.

For more information, see the "NF specific metrics" section in *NF-specific User Guides*.

Configuring NF Alerts on OCI

The following procedure describes how to configure the NF alerts for OCI. OCI supports metric expressions written in MQL (Metric Query Language) and thus requires a new NF alert file to configure alerts in the OCI observability platform.

The following are the steps:

1. Run the following command to extract the NF specific alert .zip file:

   ```
   unzip <nf>_oci_alertrules_<version>.zip
   ```

   Example:

   ```
   unzip ocAdaptor_oci_alertrules_<version>.zip
   ```

   Depending on the NF, either one or both of the following folders are available in the zip file:

   - `<NF>_oci`

- `<NF>_oci_resources`

For example:

- In SEPP, the following folders are available:
  - `ocsepp_oci`
  - `ocsepp_oci_resources`
- In cnDBTier, the following folder is available:
  - `ocsepp_oci`

> **Note**
>
> The zip file is available in the Scripts folder of the NF CSAR package.

2. Open the `<NF>_oci` folder and look for the `notifications.tf` file.
3. Open the `notifications.tf` file and update the `endpoint` parameter with the email ID of the user.
4. Open the `<NF>_oci_resources` folder, in the `notifications.tf` file, update the parameter `endpoint` with the email id of the user.
5. Log in to the **OCI Console**.
   For more information, see the "Signing In to the OCI Console" section in *Oracle Cloud Infrastructure Documentation*.
6. Open the navigation menu and select **Developer Services**. The **Developer Services** window appears in the right pane.
7. Under the **Developer Services**, select **Resource Manager**.
8. Under **Resource Manager**, select **Stacks**. The **Stacks** window appears.
9. Click **Create stack**.
10. Select the default **My configuration** radio button.
11. Under **Stack configuration**, select the folder radio button and upload the `<NF>_oci` folder.
12. Enter the **Name** and **Description** and select the **compartment**.
13. Select the latest terraform version from the **Terraform version** drop-down.
14. Click **Next**. The **Edit stack** screen appears.
15. Enter the required inputs to create the SEPP alerts or alarms.
16. Click **Save** and **Run apply.**
17. Verify that the alarms are created in the Alarm Definitions screen (**OCI Console> Observability and Management> Monitoring>Alarm definitions**) provided.
    The required inputs are:
    - **Alarms Configuration**
      a. **Compartment name**: Choose the compartment's name from the drop-down list.
      b. **Metric namespace**: Metric namespace that the user provided while deploying OCI Adaptor.
      c. **Topic name**: This is a user-configurable name. It can contain a maximum of 256 characters. Only alphanumeric characters plus hyphens (-) and underscores (_) are allowed.

   d. **Message format**: Keep it as **ONS_OPTIMIZED** (This is pre-populated).
   e. **Alarm is_enabled**: Keep it as **True** (This is pre-populated).

18. Repeat the steps 6 to 17 for uploading the `<NF>_oci_resources` folder. Here **Metric namespace** will be pre-populated.
    For more information, see the "NF specific alerts" section in *NF User Guides*.

Deploying CNC Applications

CNC NF deployment on OCI is a manual process performed on the CLI Server.

- On-premise deployment
  – It refers to deploying CNC NF and its components on the customer's private data centre. The deployment components include `CNC NF`, `cnDBTier`, and `CNC Console`, deployed on the underlying CNE platform.
- OCI deployment
  – It refers to the deployment of CNC NF and its components on the customer's tenancy in the public cloud OCI. The deployment components include `CNC NF`, `cnDBTier` and `CNC Console` deployed on OCI platform.

For more information, see the "Deploying CNC Applications" section in *CNC NF-specific installation guides*.

Uninstalling OCI Adaptor

This section provides information about uninstalling the OCI Adaptor.

Prerequisite:

The user must purge (delete) the logs manually.

To delete the logs, go to **Observability and Management** > **Logging Analytics** > **Administration** > **Storage**. Click **Purge Logs** and then, click **Purge**.

Uninstalling the OCI Adaptor

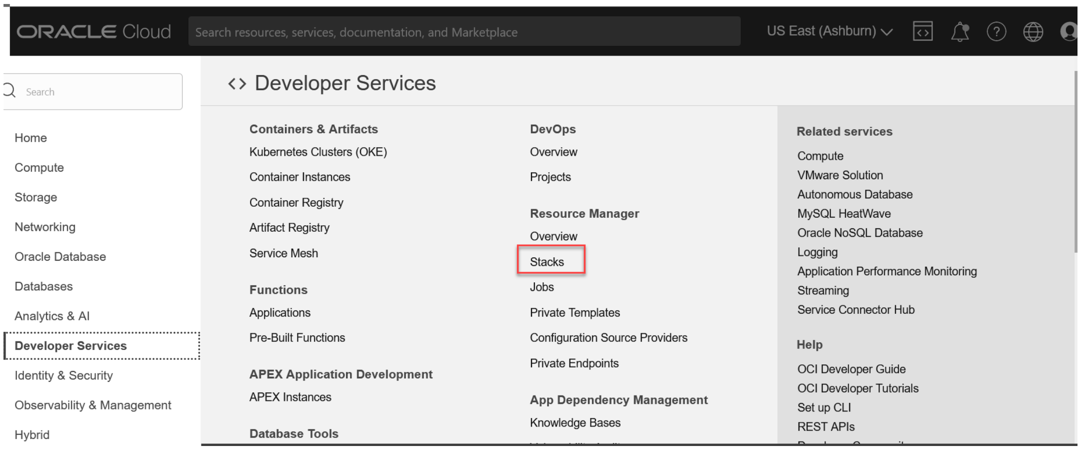
To uninstall OCI Adaptor, perform the following procedure:

1. Log in to the **OCI Console**.

   > **Note**
   >
   > For more details about logging in to the OCI, refer to Signing In to the OCI Console.

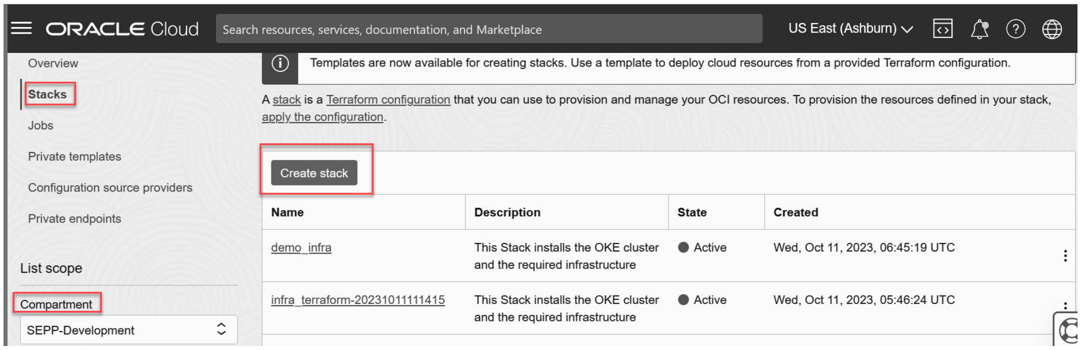
2. Open the navigation menu and select **Developer Services**. The **Developer Services** window appears in the right pane.
3. Under the **Developer Services**, select **Resource Manager**.
4. Under **Resource Manager**, select **Stacks**. The **Stacks** window appears.

Figure 3-9 Developer Services



5. Select Compartment from the **Compartment** drop-down list.

Figure 3-10 Compartment



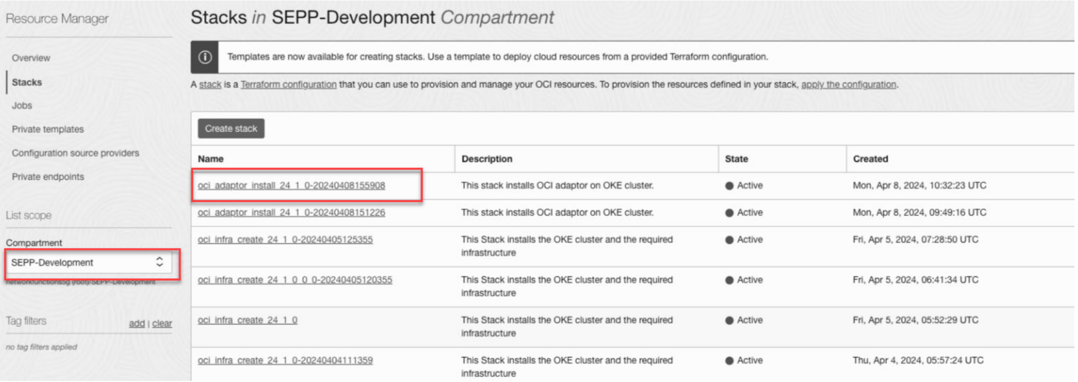
Templates are now available for creating stacks. Use a template to deploy cloud resources from a provided Terraform configuration.

A stack is a Terraform configuration that you can use to provision and manage your OCI resources. To provision the resources defined in your stack, apply the configuration.

Name	Description	State	Created
demo_infra	This Stack installs the OKE cluster and the required infrastructure	● Active	Wed, Oct 11, 2023, 06:45:19 UTC
infra_terraform-20231011111415	This Stack installs the OKE cluster and the required infrastructure	● Active	Wed, Oct 11, 2023, 05:46:24 UTC

6. Select the stack created during OCI Adaptor deployment.

Figure 3-11 Select Stack



Stacks *in* SEPP-Development *Compartment*

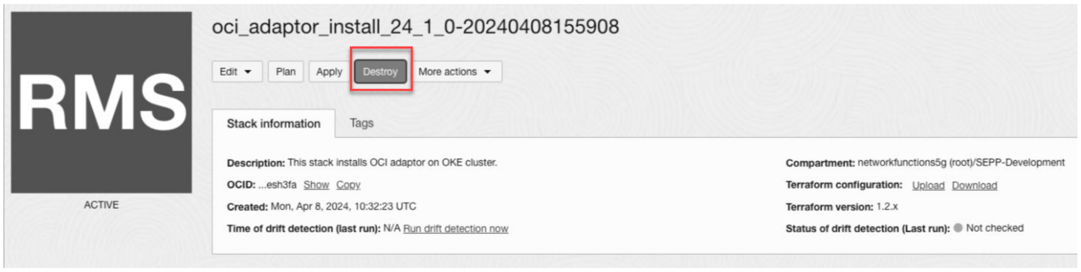
Templates are now available for creating stacks. Use a template to deploy cloud resources from a provided Terraform configuration.

A stack is a Terraform configuration that you can use to provision and manage your OCI resources. To provision the resources defined in your stack, apply the configuration.

Name	Description	State	Created
oci_adaptor_install_24_1_0-20240408155908	This stack installs OCI adaptor on OKE cluster.	● Active	Mon, Apr 8, 2024, 10:32:23 UTC
oci_adaptor_install_24_1_0-20240408151226	This stack installs OCI adaptor on OKE cluster.	● Active	Mon, Apr 8, 2024, 09:49:16 UTC
oci_infra_create_24_1_0-20240405125355	This Stack installs the OKE cluster and the required infrastructure	● Active	Fri, Apr 5, 2024, 07:28:50 UTC
oci_infra_create_24_1_0_0_0-20240405120355	This Stack installs the OKE cluster and the required infrastructure	● Active	Fri, Apr 5, 2024, 06:41:34 UTC
oci_infra_create_24_1_0	This Stack installs the OKE cluster and the required infrastructure	● Active	Fri, Apr 5, 2024, 05:52:29 UTC
oci_infra_create_24_1_0-20240404111359	This Stack installs the OKE cluster and the required infrastructure	● Active	Thu, Apr 4, 2024, 05:57:24 UTC

7. Click **Destroy**.

Figure 3-12   Destroy Stack



oci_adaptor_install_24_1_0-20240408155908

Description: This stack installs OCI adaptor on OKE cluster.
OCID: ...esh3fa Show Copy
Created: Mon, Apr 8, 2024, 10:32:23 UTC
Time of drift detection (last run): N/A Run drift detection now

Compartment: networkfunctions5g (root)/SEPP-Development
Terraform configuration: Upload Download
Terraform version: 1.2.x
Status of drift detection (Last run): ● Not checked

8. The confirmation page appears. Click **Destroy** to uninstall the OCI Adaptor.

> **Note**
>
> The sample dashboards created automatically during OCI Adaptor installation are not deleted during OCI Adaptor uninstallation. It is recommended that you delete them manually.

Deleting OCI Infrastructure

This section provides information about deleting the OCI infrastructure that was created for deploying the OCI Adaptor and NFs.

Prerequisite:

The user must uninstall the OCI Adaptor.

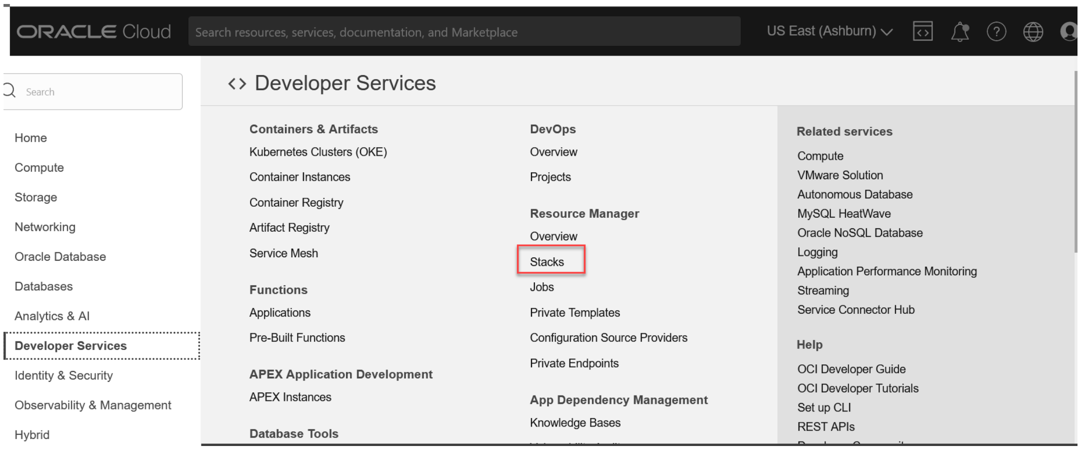
For more information, see the "Uninstalling OCI Adaptor" section in *Oracle Communications Cloud Native Core, OCI Adaptor User Guide*.

Deleting the OCI Infrastructure

To delete the OCI Infrastructure, perform the following procedure:

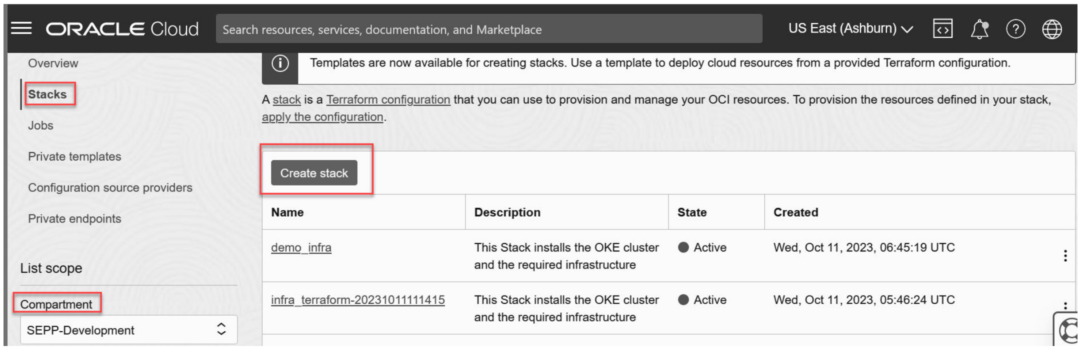
1. Log in to the **OCI Console**.
   For more information, see the "Signing In to the OCI Console" section in *Oracle Cloud Infrastructure Documentation*.
2. Open the navigation menu and select **Developer Services**. The **Developer Services** window appears in the right pane.
3. Under the **Developer Services**, select **Resource Manager**.
4. Under **Resource Manager**, select **Stacks**. The **Stacks** window appears.

Figure 3-13 Developer Services



5. Select **Compartment** from the **Compartment** drop-down list.

Figure 3-14 Compartment



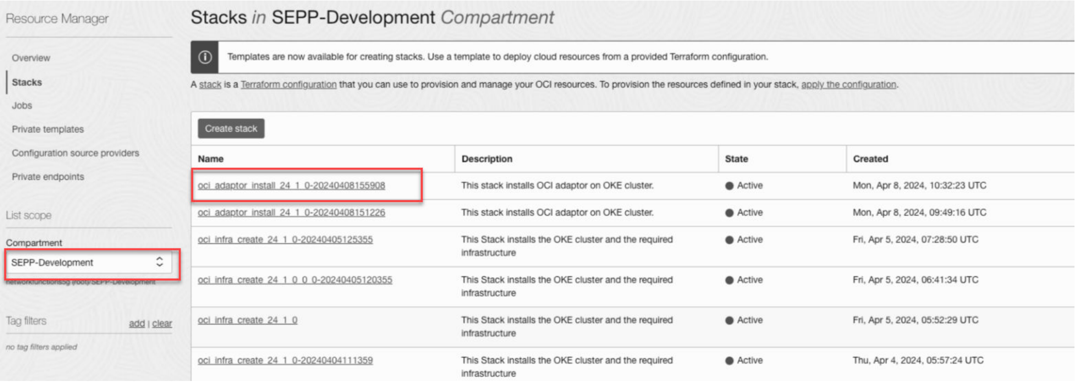
Templates are now available for creating stacks. Use a template to deploy cloud resources from a provided Terraform configuration.

A stack is a Terraform configuration that you can use to provision and manage your OCI resources. To provision the resources defined in your stack, apply the configuration.

Name	Description	State	Created
demo_infra	This Stack installs the OKE cluster and the required infrastructure	● Active	Wed, Oct 11, 2023, 06:45:19 UTC
infra_terraform-20231011111415	This Stack installs the OKE cluster and the required infrastructure	● Active	Wed, Oct 11, 2023, 05:46:24 UTC

6. Select the **stack** which was created for creating the OCI Infrastructure.

Figure 3-15 Select Stack



Stacks *in* SEPP-Development *Compartment*

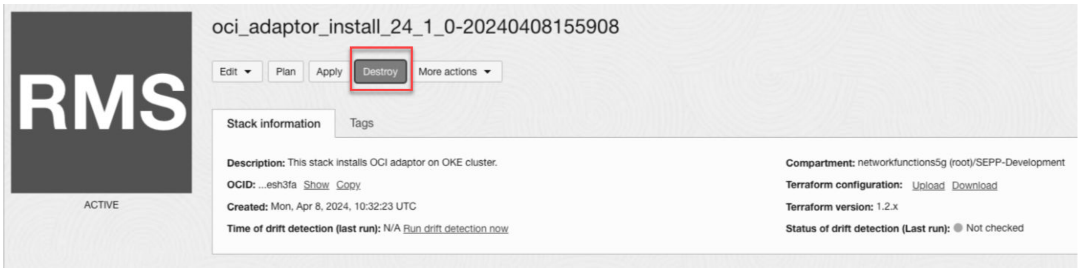
Templates are now available for creating stacks. Use a template to deploy cloud resources from a provided Terraform configuration.

A stack is a Terraform configuration that you can use to provision and manage your OCI resources. To provision the resources defined in your stack, apply the configuration.

Name	Description	State	Created
oci_adaptor_install_24_1_0-20240408155908	This stack installs OCI adaptor on OKE cluster.	● Active	Mon, Apr 8, 2024, 10:32:23 UTC
oci_adaptor_install_24_1_0-20240408151226	This stack installs OCI adaptor on OKE cluster.	● Active	Mon, Apr 8, 2024, 09:49:16 UTC
oci_infra_create_24_1_0-20240405125355	This Stack installs the OKE cluster and the required infrastructure	● Active	Fri, Apr 5, 2024, 07:28:50 UTC
oci_infra_create_24_1_0_0_0-20240405120355	This Stack installs the OKE cluster and the required infrastructure	● Active	Fri, Apr 5, 2024, 06:41:34 UTC
oci_infra_create_24_1_0	This Stack installs the OKE cluster and the required infrastructure	● Active	Fri, Apr 5, 2024, 05:52:29 UTC
oci_infra_create_24_1_0-20240404111359	This Stack installs the OKE cluster and the required infrastructure	● Active	Thu, Apr 4, 2024, 05:57:24 UTC

7. Click **Destroy**.

Figure 3-16   Destroy Stack



oci_adaptor_install_24_1_0-20240408155908

Description: This stack installs OCI adaptor on OKE cluster.

OCID: ...esh3fa Show Copy

Created: Mon, Apr 8, 2024, 10:32:23 UTC

Time of drift detection (last run): N/A Run drift detection now

Compartment: networkfunctions5g (root)/SEPP-Development

Terraform configuration: Upload Download

Terraform version: 1.2.x

Status of drift detection (Last run): ● Not checked

8. The confirmation page appears. Click **Destroy** to delete the OCI Infrastructure.

> ⓘ **Note**
>
> Deleting the OCI Infrastructure will not delete the compartment. User can delete it manually.

4 Troubleshooting Scenarios

This section provides information on troubleshooting the common errors encountered during the deployment and creation of the OCI Adaptor infrastructure.

- **Problem**: The resource limit exceeded and the following error displayed: `"Error: 400-LimitExceeded, The following service limits were exceeded: <OCI Resource name>. Request a service limit increase from the service limits page in the console"`.

  Solution: The user must request for the additional resources.

- **Problem**: The user encounters the following errors while running the Terraform: `"Failed to read ssh private key: no key found"` or `"Failed to read ssh public key: no key found"`

  Solution: Verify the format of the public and private keys and ensure there are no blank spaces either at the beginning or at the end of the keys.

- **Problem**: The user gets the following error: `"NotAuthorizedOrNotFound, Authorization failed or requested resource not found"`

  Solution: Verify the following:

  - The user is a tenency Admin.
  - To create the OCI infrastructure, it is recommended that first-level subcompartments be created in the root compartment. The creation of OCI infrastructure at the second-level subcompartment is not supported.

- **Problem**: At present, management agent has a non-configurable scraping interval of five minutes. By default all the metrics gets populated after five minutes.

  Solution:

  Follow the below steps to update the scraping interval to one minute:

  - Run the following command to enter the management agent's pod:

    ```
    kubectl exec -it <Management-agent pod name> bash -n oci-adaptor
    ```

  - Open the following file:

    ```
    /opt/oracle/mgmt_agent/agent_inst/config/destinations/OCI/services/
    Agent/1/types/PrometheusEmitter.json
    ```

  - Update the key **interval** associated with **metricStream** block to one minute, and save and close using `:wq` command.

    ```
    "metricStream":{
        "metricEndpoint":"TelemetryStreaming",
        "streamDescriptor":{
            "collectorId":"Prometheus",
            "credentialName":"RestCreds",
            "properties":[
    ```

```
           {  "name":"url", "scope":"INSTANCE", "content":"url"},
           {  "name":"streamNamespace","scope":"INSTANCE",
"content":"namespace"},
           {  "name":"compartmentId", "scope": "INSTANCE",
"content":"compartmentId","optional":true},
           {  "name":"additionalDimensions", "scope": "INSTANCE",
"content":"metricDimensions","optional":true},
           {  "name":"allowMetrics", "scope": "INSTANCE",
"content":"allowMetrics","optional":true},
           {  "name":"read-timeout","scope": "INSTANCE","content":"read-
timeout","optional":true},
           {  "name":"connection-timeout","scope":
"INSTANCE","content":"connection-timeout","optional":true},
           {  "name":"proxy-url","scope": "INSTANCE","content":"proxy-
url","optional":true},

{  "name":"dimensionFilterFilePath","scope":"INSTANCE","content":
"dimensionFilterFilePath","optional":true},
           {  "name":"dimensionFilter","scope":"INSTANCE","content":
"dimensionFilter","optional":true}
        ]
    },
    "schedule": {
    "interval":1,
     "timeUnit":"Min"
   }
}
```

- Exit from management agent pod.
- Restart the management agent pod by deleting the existing management agent pod.

ⓘ **Note**

Updating the scraping interval using the above procedure updates the metrics data only for the Metric Namespace specified by the user while deploying the OCI Adaptor.