

Oracle® Public Cloud Machine

Using Oracle Compute Cloud Service



Release 17.1.2

E62130-12

February 2020



Oracle Public Cloud Machine Using Oracle Compute Cloud Service, Release 17.1.2

E62130-12

Copyright © 2016, 2020, Oracle and/or its affiliates.

Primary Author: Salvador Hernandez

Contributing Authors: Ashish Thomas, Greg Verstraeten, Gavin Parish, Karen Wilson, Chris Broadbent, Jon Richards, Samuel Jackson, John Reinhart, and Leo Yuen.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	vii
Documentation Accessibility	vii
Related Documents	vii
Conventions	vii

1 Getting Started

About Oracle Compute Cloud Service	1-1
Before You Begin with Oracle Compute Cloud Service	1-2
Oracle Compute Cloud Service Terminology	1-3
About the Interfaces to Oracle Compute Cloud Service	1-4
Accessing the Self Service Portal in the UI	1-5
Accessing Oracle Compute Cloud Service Using the Command-Line Interface	1-5
Accessing Oracle Compute Cloud Service Using the REST API	1-5
Exploring the User Interface	1-6
Workflow for Using Oracle Compute Cloud Service	1-9
Understanding Oracle Compute Cloud Service User Roles	1-11
Getting Started with Tenant Administration on OPCM	1-11
Getting Started with Using Oracle Compute Cloud Service on OPCM	1-12

2 Creating and Managing Tenant Users

Creating a Tenant User	2-1
Creating a User Using the CLI or API	2-2
Updating a User Details	2-2
Updating a User Using the CLI or API	2-2

3 Enabling Secure Access to Instances Using SSH

About SSH Keys	3-1
Generating an SSH Key Pair	3-1
Adding an SSH Key	3-2

4 Managing Networks

About Networking in Oracle Compute Cloud Service	4-1
Creating an IPoIB Tenant vNet	4-2
Creating an IPoIB Tenant vNet Using the CLI or API	4-3
Updating an IPoIB Tenant vNet	4-4
Updating an IPoIB Tenant vNet Using the CLI or API	4-4
Reserving an IP Address	4-4
Reserving an IP Address Using the CLI or API	4-5
Deleting an IP Reservation	4-5
Deleting an IP Reservation Using the CLI or API	4-5
Deleting an IPoIB Tenant vNet	4-6
Deleting an IPoIB Tenant vNet Using the CLI or API	4-6

5 Working with Orchestrations

About Orchestrations	5-1
Orchestration Life Cycle	5-2
Workflow for Creating Instances Using Orchestrations	5-3
High Availability Policies	5-4
Discovering Orchestrations	5-5
Discovering Orchestrations Using the CLI or API	5-6
Starting an Orchestration	5-6
Starting an Orchestration Using the CLI or API	5-6
Shutting Down an Orchestration	5-7
Shutting Down an Orchestration Using the CLI or API	5-7
Restarting an Orchestration	5-7
Restarting an Orchestration Using the CLI or API	5-7
Stopping an Orchestration	5-8
Stopping an Orchestration Using the CLI or API	5-8
Updating an Orchestration	5-8
Updating an Orchestration Using the CLI or API	5-9
Deleting an Orchestration	5-9
Deleting an Orchestration Using the CLI or API	5-10

6 Managing Instances

About Instances	6-1
Instance Life Cycle	6-2
Creating Instances Using Orchestrations	6-3

Creating Instances Using a Simple Orchestration	6-3
Creating Instances Using Orchestrations with Custom Configurations	6-4
Creating Instances Using the CLI	6-8
Attaching a Storage Volume to an Instance	6-10
Attaching a Storage Volume to an Instance Using the CLI or API	6-10
Attaching a Storage Volume to an Instance on a 17.1.2 System	6-11
Detaching a Storage Volume from an Instance	6-11
Detaching a Storage Volume from an Instance Using the CLI or API	6-11
Creating a Snapshot of an Instance	6-12
Creating a Snapshot of an Instance Using the CLI or API	6-12
Shutting Down an Instance	6-12
Shutting Down an Instance Using the CLI or API	6-13
Restarting an Instance	6-13
Restarting an Instance Using the CLI or API	6-13
Updating Instances	6-14
Changing an Instance Shape	6-14
Changing an Instance Shape Using the CLI or API	6-14
Updating the Network Configuration of an Instance	6-14
Updating the Network Configuration of an Instance Using the CLI or API	6-15
Attaching a Network to an Instance	6-15
Attaching a Network to an Instance Using the CLI or API	6-16
Deleting an Instance	6-16
Accessing an Instance Using SSH	6-16
Accessing an Instance Using VNC	6-17
Accessing the Metadata of Instances	6-18
Upgrading Oracle Compute Cloud Service Instances	6-18
Configure the Yum Repository Server	6-19
Set Up the Patch Bundle	6-19
Patch Your Instances	6-21

7

Managing Storage Volumes

About Storage Volumes	7-1
About Snapshot of Storage Volumes	7-1
Creating a Storage Volume	7-2
Creating a Storage Volume Using the CLI or API	7-2
Creating a Storage Volume on the External Storage Appliance	7-2
Mapping Database Storage Volumes to Storage Properties	7-4
Creating a Storage Volume on the External Storage Appliance Using the CLI or API	7-7
Creating a Snapshot of a Storage Volume	7-8
Creating a Snapshot of a Storage Volume Using the CLI or API	7-8

Deleting a Storage Volume	7-9
Deleting a Storage Volume Using the CLI or API	7-9

8 Managing Templates

About Templates	8-1
Uploading a Private Template	8-2
Uploading a Private Template Using the CLI or API	8-2
Customizing an Image	8-3
Deploy and Customize an Instance	8-3
Create Your Custom Machine Image	8-4
Deploy Instances Using the Customized Machine Image	8-5
Deleting a Private Template	8-6
Deleting a Private Template Using the CLI or API	8-6

A Orchestration Templates

B Sample Orchestration

Preface

This guide explains how to provision and manage Oracle Compute Cloud Service instances, configure network and storage resources, add machine images, and manage SSH keys.

Audience

This document is intended for tenant administrators and tenant users of Oracle Compute Cloud Service.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information, see the following documents in Oracle Public Cloud Machine documentation library:

- *CLI Reference for Oracle Compute Cloud Service*
- *REST API Reference for Oracle Compute Cloud Service*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Getting Started

This section describes how to get started with Oracle Compute Cloud Service on Oracle Public Cloud Machine.

Topics:

- [About Oracle Compute Cloud Service](#)
- [Before You Begin with Oracle Compute Cloud Service](#)
- [Oracle Compute Cloud Service Terminology](#)
- [About the Interfaces to Oracle Compute Cloud Service](#)
- [Accessing the Self Service Portal in the UI](#)
- [Accessing Oracle Compute Cloud Service Using the Command-Line Interface](#)
- [Accessing Oracle Compute Cloud Service Using the REST API](#)
- [Exploring the User Interface](#)
- [Workflow for Using Oracle Compute Cloud Service](#)
- [Understanding Oracle Compute Cloud Service User Roles](#)
- [Getting Started with Tenant Administration on OPCM](#)
- [Getting Started with Using Oracle Compute Cloud Service on OPCM](#)

About Oracle Compute Cloud Service

Oracle Compute Cloud Service optimized for Oracle Public Cloud Machine (OPCM) provides Infrastructure as a Service (IaaS) and also acts as the base on which the other cloud services, such as Oracle Java Cloud Service and Oracle Integration Cloud Service are built. Using Oracle Compute Cloud Service, you can provision virtual machines with all the necessary storage and networking resources, manage and scale your virtual machine topology in the cloud, and migrate your Oracle and third-party applications to OPCM. Oracle Compute Cloud Service provides the infrastructure for users to run their applications.

Using Oracle Compute Cloud Service, you can do the following:

- **Migrate your applications from Oracle Cloud to a dedicated and secured site on premises**

You can migrate your cloud applications to virtual machines that you create on the secured, on-premises dedicated sites, and take advantage of the elastic compute, storage, and network capabilities that Oracle Compute Cloud Service provides.

- **Assign processor and memory resources from a range of resource profiles**

While creating Oracle Compute Cloud Service instances, you can assign CPU and memory resources by selecting from a wide range of resource profiles (called shapes), each of which is a carefully designed combination of processor and memory limits.

- **Automate your VM provisioning and management workflows**

You can define all the attributes for multiple, high availability enabled virtual machines of varying shapes and machine images in an orchestration. Using the web console, you can then create, remove, and reprovision all of the virtual machines and associated resources as required through the orchestration.

- **Create instances using Oracle-provided machine images**

You can use Oracle-provided machine images to quickly provision robust Oracle Linux virtual machines.

- **Provide a persistent boot disk for your instance**

When you create an instance, the instance is automatically set up to boot from the default persistent disk. This ensures that any changes that you make at the operating system-level persist when the instance is re-created.

- **Exercise fine-grained control over network traffic**

You can create and manage private network subnets.

- **Reserve and assign fixed public IP addresses**

For instances that require specific IP addresses, you can reserve and use a static IP address.

- **Attach high-capacity block storage to instances**

You can attach block storage to each of your instances for storing data and applications, by creating multiple persistent storage volumes and attaching them to the instances. Even after you delete instances, the data stored in the storage volumes remains intact until you delete the volumes.

- **Monitor and manage all of your resources through a unified interface**

You can access, administer, and use Oracle Compute Cloud Service through an easy-to-use graphical web console provided by Oracle Enterprise Manager. The console provides a single interface that you can use to monitor and manage all your Oracle Compute Cloud Service resources.

You can also access Oracle Compute Cloud Service and manage resources by using Oracle Compute CLI and REST API calls.

- **Ensure secure access to instances**

You can configure Oracle Compute Cloud Service instances to be accessed securely from remote hosts by using SSH.

Before You Begin with Oracle Compute Cloud Service

Before you begin using Oracle Compute Cloud Service:

- Understand the feature of the service. See [About Oracle Compute Cloud Service](#).
- Be familiar with the Oracle Compute Cloud Service terminology. See [Oracle Compute Cloud Service Terminology](#).
- Understand the user roles in Oracle Compute Cloud Service. See [Understanding Oracle Compute Cloud Service User Roles](#).

Oracle Compute Cloud Service Terminology

The following table lists the terminology used in Oracle Compute Cloud Service on OPCM.

Table 1-1 Terminology

Term	Description
Tenant	A container for system objects, such as users and virtual machine instances. A tenant can represent a whole organization or departments in an organization.
Tenant user	Users who are assigned this role consume the network, compute, and storage resources allocated to them by creating virtual machines that support their application requirements.
Orchestration	An orchestration is a JavaScript Object Notation (JSON) formatted file that groups one or more objects, such as instances and storage volumes, so that they can be managed as a unit. Within an orchestration, you can specify dependencies to control the sequence in which the objects are created and destroyed.
Launch plan	A launch plan is a JavaScript Object Notation (JSON) formatted orchestration section that defines the properties of one or more instances in Oracle Compute Cloud Service. The attributes in a launch plan include the instance label and name, the image and shape to be used for the instance, and so on.
Instance	An instance is a logical representation of a compute resource that has attributes defining network, storage, and compute resource (RAM, vCPU) requirements. When an instance is launched, a virtual machine is created and configured based on the instance attributes. During the life of a single instance, many virtual machines can be created and destroyed if the instance is shut down and restarted.
Shape	A shape is a resource profile that specifies the number of CPUs and the amount of memory to be allocated to an instance in Oracle Compute Cloud Service.
Machine images	A machine image is a template of a virtual hard disk of a specific size with an installed operating system. You use machine images to create virtual machine instances in Oracle Compute Cloud Service.
Image List	An image list is a collection of Oracle Compute Cloud Service machine images. Each machine image in an image list is identified by a unique entry number.
Storage Volume	A storage volume is a virtual disk that provides persistent block storage space for instances in Oracle Compute Cloud Service.
Volume snapshot	A point-in-time image of an existing storage volume.
Instance snapshot	A point-in-time state of a virtual machine's boot disk. The output of a snapshot is a machine image, which can be used as a template to create a new instance.
InfiniBand	InfiniBand is a networking communication standard used in high performance computing. Oracle's InfiniBand network solutions deliver the application isolation and elasticity required to consolidate and virtualize enterprise business applications and cloud computing infrastructures.

Table 1-1 (Cont.) Terminology

Term	Description
IP over InfiniBand (IPoIB)	IP over InfiniBand (IPoIB) enables applications running in separate virtual machines to communicate with each other over a private channel by using native InfiniBand protocols and without the overhead of Ethernet.
Ethernet over InfiniBand (EoIB)	Ethernet over InfiniBand (EoIB) enables applications on an InfiniBand fabric to communicate with the Ethernet networks external to OPCM.
Tenant networks	Networks that are assigned exclusively for use by instances in a tenant.
Tenant EoIB networks	Networks assigned for use by instances to communicate with external Ethernet networks.
Tenant IPoIB networks	Networks assigned for use to communicate across the instances in a tenant.
vEthernet	A layer 2 network that helps you isolate Oracle Compute Cloud Service instances created on OPCM. For example, you can create networks to isolate instances used by different departments in your company.
Tenant network vNet	A vNet represents an IP subnet on a given vEthernet. The subnet includes the CIDR and IP address range that are available for users to use for their instances.
vNet reservation	A vNet reservation is a public or private IP address that you can attach to an Oracle Compute Cloud Service instance.
Attributes	A collection of user defined key and value pairs that is made available to an instance through the metadata service. The attributes under <code>user_data</code> are made available through the metadata service. Instance metadata, including user-defined data is available at <code>http://192.0.0.192/latest/user-data</code> within an instance.
Relationships	A relationship is defined as part of an orchestration. There are two types of relationships, one between oplans and another between instances. An oplan relationship defines the dependencies between oplans in an orchestration. An instance relationship defines the affinity or anti-affinity rules for the placement of the instances. Oplan relationships are specified at the same level as oplans in an orchestration, and instance relationships are specified inside a launch plan.

About the Interfaces to Oracle Compute Cloud Service

The following table summarizes the key interfaces to Oracle Compute Cloud Service on OPCM.

Table 1-2 Interfaces to Oracle Compute Cloud Service

Type of Access	Description	More Information
Web console	Use the web console provided by Oracle Enterprise Manager Cloud Control to access, manage, and monitor Oracle Compute Cloud Service.	Exploring the User Interface

Table 1-2 (Cont.) Interfaces to Oracle Compute Cloud Service

Type of Access	Description	More Information
Command-line interface (CLI)	Use the <code>oracle-compute</code> CLI tool to manage and monitor Oracle Compute Cloud Service.	<i>CLI Reference for Oracle Compute Cloud Service</i>
REST API	Use REST API endpoints to manage Oracle Compute Cloud Service.	<i>REST API Reference for Oracle Compute Cloud Service</i>

Accessing the Self Service Portal in the UI

The Self Service Portal in the UI enables you to access the cloud services without IT intervention. You can create and manage Oracle Compute Cloud Service instances and the associated storage and networking resources from the Self Service Portal.

 **Note:**

For details on setting up and accessing Enterprise Manager Cloud Control for OPCM, contact your local Enterprise Manager administrator.

1. From the **Enterprise** menu, select **Cloud**, then select **Self Service Portal**.
2. On the right, click on the **stack** icon, and select **Infrastructure – Oracle Compute Cloud Services**.

Accessing Oracle Compute Cloud Service Using the Command-Line Interface

OPCM provides a comprehensive command-line interface (CLI) for using Oracle Compute Cloud Service and supports all the actions that you can perform using the REST API.

You can install the Oracle Compute CLI on a system provisioned with Oracle Linux 6.3 or later.

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

Accessing Oracle Compute Cloud Service Using the REST API

You can provision and manage Oracle Compute Cloud Service instances and the associated storage and networking resources by using the REST API.

Each REST API call maps to an HTTP request: getting an object (`GET`), adding an object (`POST`), updating an object (`PUT`), and deleting an object (`DELETE`). The HTTP response code indicates whether the request was successful. Each object for which

you can perform the GET, POST, PUT, and DELETE requests is identified uniquely by its URI.

Contact your tenant administrator for the REST API URI and login credentials for authentication.

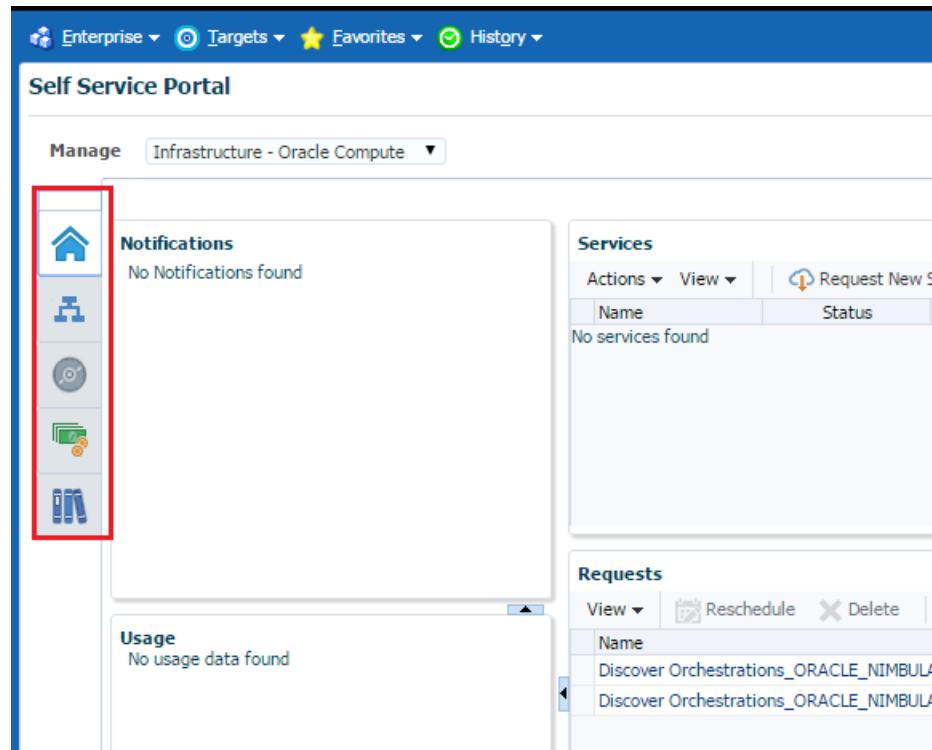
To access Oracle Compute Cloud Service by using the REST API, you must use the REST endpoint URI provided by your administrator. All requests require authentication, for which you send an authentication request. If the authentication request succeeds, a cookie is returned, which you must include in all future requests. The requesting client must always use the latest cookie it receives.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Exploring the User Interface

To log into the web console for accessing and monitoring Oracle Compute Cloud Service on OPCM, see [Accessing the Self Service Portal in the UI](#).

The Self Service Portal provides a dashboard to request and manage resources as shown in the following figure.



Home Page

The **Home** page displays the existing orchestrations and statuses. The page provides the options to discover, create, and manage orchestrations.

Self Service Portal

Manage Infrastructure - Oracle Compute

Services

Name	Status	State	Service Type	Resource Provider	Creation Date
/mytenant/public/myorch	Green	Ready	Orchestration	denp07	Dec 30, 2015

Requests

Name	Type	Submission Date	Begin Date
Discover Orchestrations_ORACLE_NIMBULA_ORC...	Discover Orchestrations	Jan 4, 2016	Jan 4, 2016
/mytenant/public/myorch_Snapshot Oracle VM In...	Snapshot Oracle VM Instance	Dec 30, 2015	Dec 30, 2015
/mytenant/public/myorch_Snapshot Oracle VM In...	Snapshot Oracle VM Instance	Dec 30, 2015	Dec 30, 2015
/mytenant/public/myorch_Snapshot Oracle VM In...	Snapshot Oracle VM Instance	Dec 30, 2015	Dec 30, 2015
denp07/mytenant/public/myorch_Start Virtual M...	Start Virtual Machine Instance	Dec 30, 2015	Dec 30, 2015
denp07/mytenant/public/myorch_Attach Network...	Attach Networks to Virtual Machine	Dec 30, 2015	Dec 30, 2015
Create Orchestration_ORACLE_NIMBULA_ORCHE...	Create Orchestration	Dec 30, 2015	Dec 30, 2015

Networks Page

The **Networks** page displays the tenant networks and service networks that are available for use, and IP addresses that were reserved in these networks.

Self Service Portal

Manage Infrastructure - Oracle Compute

Networks

* Oracle Compute Site | denp07

A Network Subnet contains a range of VNet IP addresses which can be reserved for Orchestration use.

View Tenant Networks Service Networks

Create **Edit** **Delete**

Subnet	Type	Description	CIDR	IP Ranges
/mytenant/public/ipoib_vnet1	ipoib		192.168.96.40/30	192.168.96.41-192.168.96.42
/mytenant/public/ipoib_vnet2	ipoib		192.168.96.48/29	192.168.96.49-192.168.96.54
/mytenant/public/ipoib_vnet3	ipoib		192.168.96.44/30	192.168.96.45-192.168.96.46

IP Reservation

Add **Delete**

IP Address	Name
192.168.96.50	/mytenant/public/ip_reserve1

Storage Page

The **Storage** page displays all the storage volumes in the tenant. The page provides options to create and manage storage volumes and snapshots.

Self Service Portal

Manage Infrastructure - Oracle Compute

Storage

* Oracle Compute Site: denp07

A storage volume object is a single volume that can be attached to an instance.

Name	Status	Size (GB)	Description
/mytenant/public/myorch_first_instance_mystvolume1	Online	1	

Library Page

The **Library** page displays all the templates and deployment plans saved for later use. The page provides options to upload and manage private templates, and view saved deployment plans.

Self Service Portal

Manage Infrastructure - Oracle Compute

Library

* Oracle Compute Site: denp07

Name	Attributes	Quota (i)	Uploaded Size (MB)	Total Size (MB)	Decompressed Size (MB)	Agent Configured
/oracle/public/ikoOel65wlb			359	359	5632	X
/oracle/public/linux5_12.2.1.0.0_64			1589	1589	12288	X
/oracle/public/linux6_12.2.1.0.0_64			2079	2079	18432	X

Saved Deployment Plans

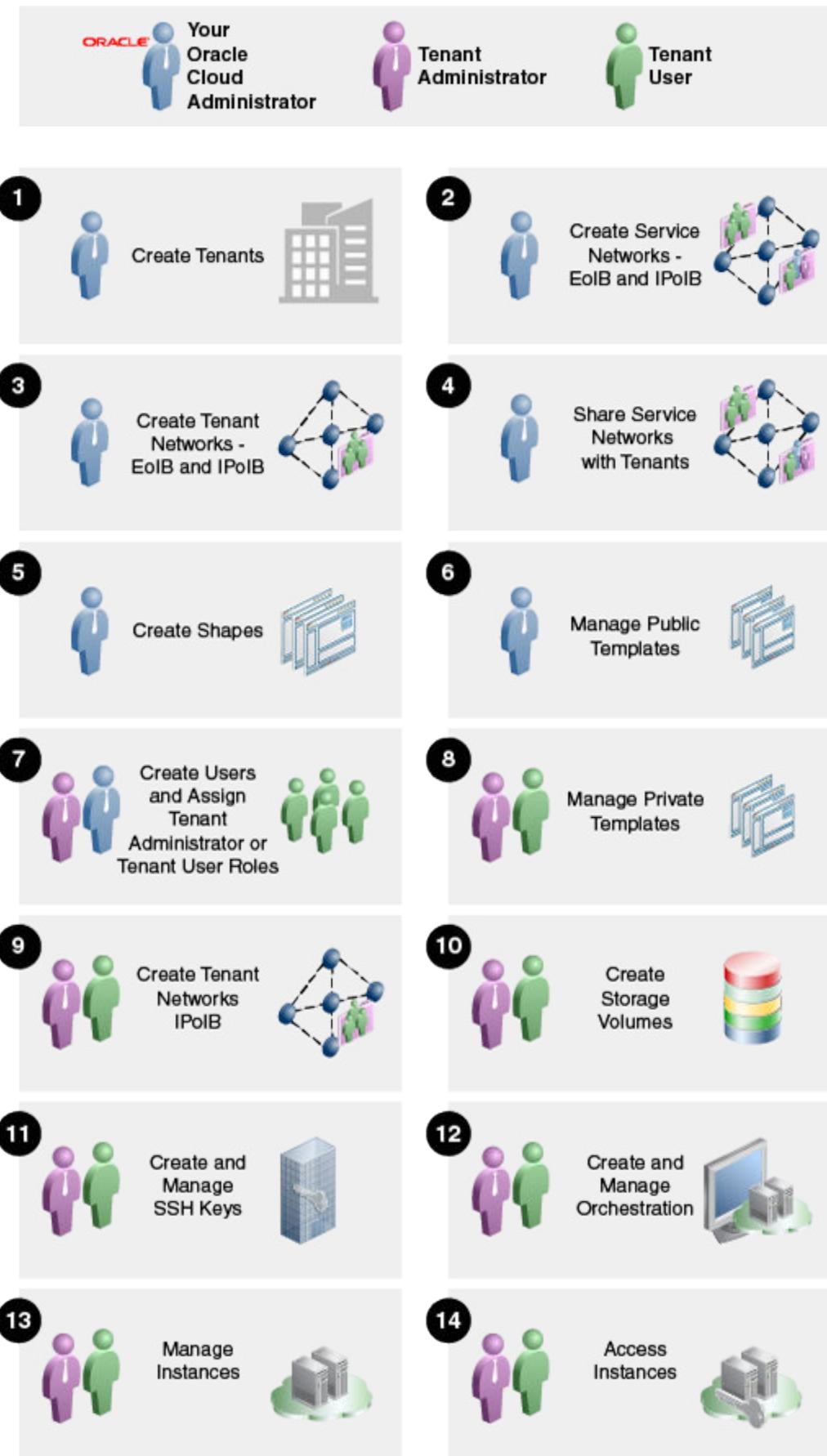
Delete the Deployment Plans which are saved for later deployment.

Name	Description	Owner	Created On
No data to display.			

Workflow for Using Oracle Compute Cloud Service

Oracle Compute Cloud Service provides you with Infrastructure as a Service solution that is designed for your business-critical workloads. Your Oracle Cloud administrator creates tenants, sets up networks, adds users to tenants, and designates users as tenant administrator or tenant user. Tenant users then create orchestrations. Each orchestration provisions one or more instances and links them to other existing cloud resources such as compute shapes, machine images, SSH keys, storage volumes, and networks.

A typical workflow to set up the cloud infrastructure, and use the Oracle Compute Cloud Service is described in the following illustration.



To perform the tasks described in each step, a user requires different roles:

- Steps 1 through 5 are performed by your Oracle Cloud Administrator
- Step 6 can be performed either by users designated as tenant administrators or by your Oracle Cloud Administrator
- Steps 7 through 14 are performed by tenant users or tenant administrators

See [Understanding Oracle Compute Cloud Service User Roles](#) for more information about user roles.

Understanding Oracle Compute Cloud Service User Roles

User roles abstract low level permissions to make it easy for the Oracle Cloud Administrator to delegate management functions. Each user is assigned a role. An Oracle Cloud Administrator creates users and assigns them the role of tenant administrator or tenant user. A tenant administrator can only create users with tenant user role.

The user roles required to use the Oracle Compute Cloud Service are tenant administrator and tenant user. The following table summarizes the permissions of these user roles.

Tasks	Tenant Administrator	Tenant User
Create and manage tenant users	Yes	No
Create and manage orchestrations, instances, private networks, storage volumes, snapshots, SSH keys, and private templates	Yes	Yes

About Tenant Users

A tenant user consumes the network, compute, and storage resources allocated to a tenant by creating cloud service instances and deploying applications. A tenant user name is represented as `/tenant_name/user_name` and the role as `/tenant_name/users`.

About Tenant Administrators

A tenant administrator has permissions and functions similar to a tenant user. A tenant administrator can also create and manage the tenant users in Oracle Compute Cloud Service. A tenant administrator user name is represented as `/tenant_name/user_name` and the role as `/tenant_name/admin`.

Getting Started with Tenant Administration on OPCM

A tenant administrator must have the following information to get started and use the Oracle Compute Cloud Service:

- URL for Enterprise Manager UI
- UI login credentials
- Oracle Compute Site name in the UI
- Oracle Compute Site login credentials

A tenant administrator can create tenant users who can access and use the Oracle Compute Cloud Service. See [Creating and Managing Tenant Users](#) for more information. A tenant administrator must provide the following details to a tenant user to get started with the Oracle Compute Cloud Service:

- URL for Enterprise Manager UI
- UI login credentials
- Oracle Compute Site name in the UI
- Oracle Compute Site login credentials

Contact your Oracle Cloud Administrator for more information.

Getting Started with Using Oracle Compute Cloud Service on OPCM

You must have the following information to get started and use the Oracle Compute Cloud Service:

- URL for Enterprise Manager UI
- Web UI login credential
- Oracle Compute Site name in the UI
- Oracle Compute Site login credential

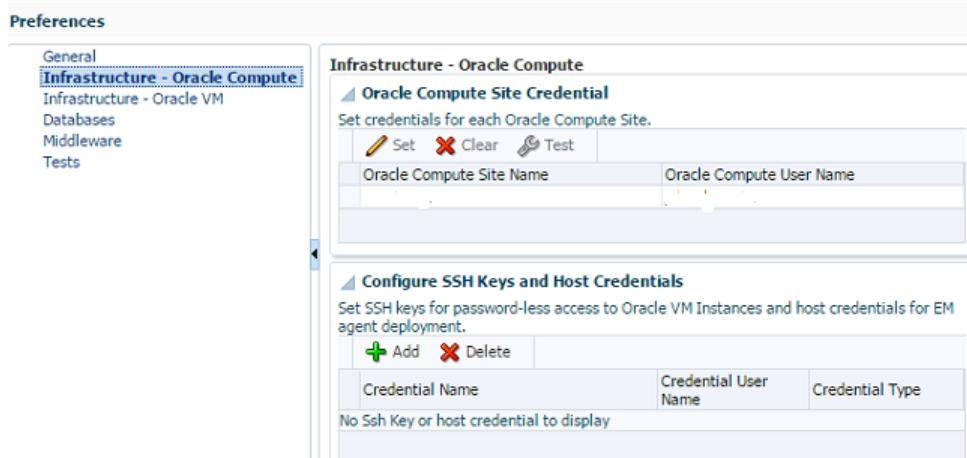
Contact your Tenant Administrator for more information.

Setting Up the Self Service Portal View

You must set your credentials for the Oracle Compute site in the UI. This is an one time setup that must be done to access the Self Service Portal and use the resources in the allocated site.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the top right corner of the Self Service Portal page, click **Preferences**.

The **Preferences** window is displayed.



3. In the left navigation pane, select **Infrastructure - Oracle Compute**.
4. Select the Oracle Compute Site name and click **Set**.
The **Setup Credential** window is displayed.
5. Enter the Oracle Compute Site login credentials provided to you as follows:
 - a. **Username**: Enter the user name in the format of `/tenant_name/user_name`.
For example, `/mytenant/user123`.
 - b. **Password**: Enter the password.
6. Click **OK**.
The user name is associated with the selected Oracle Compute Site.
7. Select the **Oracle Compute Site** and click **Test**.
The test must succeed, confirming the connection with the given credentials.
8. To set the preferences, click **Apply**.

Workflow to Create an Instance in Oracle Compute Cloud Service

Instances can be created or launched only through orchestrations in Oracle Compute Cloud Service. An orchestration can contain one or more plans where plans represent objects of a specific type, such as instances and storage volumes. Oracle Compute Cloud Service supports multiple workflows for creating compute, network, and storage resources. Within an orchestration you can specify dependencies to control the sequence in which the objects are created and destroyed. For example, if an instance uses a storage volume, then the storage volume must be created before the instance can be launched. If you create and attach a storage volume to an instance as part of the orchestration, then both the instance and storage volume are deleted when the orchestration is stopped. If you want to retain the storage volume of an instance, then shutdown the orchestration instead of stop to release the compute resources.

A typical workflow to create an instance is described in the following table. Use this workflow as a reference to get started with Oracle Compute Cloud Service on OPCM.

Task	Description	More Information
Understand Oracle Compute Cloud Service terminology.	Learn about instances, images, shapes, orchestrations, private networks, storage volumes, snapshots, and so on.	Oracle Compute Cloud Service Terminology
Generate SSH key pairs.	Generate the SSH key pairs that you plan to use to access instances.	Generating an SSH Key Pair
Access the service.	Access the service through the Oracle Compute Cloud Service web console, Oracle Compute CLI, or RESTful API.	About the Interfaces to Oracle Compute Cloud Service
Create private networks.	Create private networks to connect multiple instances in a tenant.	Managing Networks
(Optional) Create storage volumes.	Provide storage for your instances by creating and attaching storage volumes.	Managing Storage Volumes

Task	Description	More Information
Launch instances.	Create and start the orchestration to launch the instances with the desired compute, storage, and network resources.	Managing Instances
Log in to instances.	Use SSH to access your instances securely.	Accessing an Instance Using SSH

Creating and Managing Tenant Users

This section describes how to create and manage tenant users.

Topics:

- [Creating a Tenant User](#)
- [Updating a User Details](#)

Creating a Tenant User

A tenant administrator creates tenant users in Oracle Compute Cloud Service on OPCM.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Self Service Portal**, click the **Users** tab.
3. To create a user, click **Create**.
The **Add User** window is displayed.
4. Enter the following information about the user:
 - a. Enter a name for the user.
 - b. Enter a full name that describes the user.
 - c. Enter the password for the user login.

Note:

The password for a user must meet the following requirements:

- Alphanumeric.
- At least 6 characters in length.
- At least 5 unique characters. For example, "azylaz" is not a valid password but "azylmz" is valid.
- Can contain underscore and dash.
- Must not contain a sequence of characters. For example, "abcde1" and "asdfgh" are not valid passwords.
- Must not contain a dictionary word.

- d. Enter the email address.
- e. Select the login to be enabled.

5. To create the user, click **Add**.

Creating a User Using the CLI or API

Use the following command to create a user:

```
oracle-compute add user username fullname email [--role] [--password]
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To create a user using the API, use the `POST https://api.oc.example.com/user/` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Updating a User Details

You can edit the user details such as password and email.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Self Service Portal**, click the **Users** tab.
3. Select the user that you want to update the details for.
4. To update the user details, click **Edit**.
The **Edit User** window is displayed.
5. Update the following information about the user:
 - a. Enter the full name that describes the user.
 - b. To update the password of the user, select **Change Password**.

Note:

The password for a user must meet the following requirements:

- Alphanumeric.
- At least 6 characters in length.
- At least 5 unique characters. For example, "azylaz" is not a valid password but "azylmz" is valid.
- Can contain underscore and dash.
- Must not contain a sequence of characters. For example, "abcde1" and "asdfgh" are not valid passwords.
- Must not contain a dictionary word.

- c. Enter the email address of the user.
- d. Select login to be enabled for the user.

6. To update the user details, click **Edit**.

Updating a User Using the CLI or API

Use the following command to update a user:

```
oracle-compute update user username [--fullname] [--email] [--password]
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To update a user using the API, use the `PUT https://api.oc.example.com/user/` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Enabling Secure Access to Instances Using SSH

This section describes how to access instances using SSH.

Topics:

- [About SSH Keys](#)
- [Generating an SSH Key Pair](#)
- [Adding an SSH Key](#)

About SSH Keys

You can log in securely to your Oracle Compute Cloud Service instances from a remote host by using a secure shell (SSH) connection.

SSH is a cryptographic network protocol that uses two keys, a public key and a private key, to provide secure communication between two computers. SSH uses port 22 by default.

Before creating instances, generate at least one SSH key pair and ensure that the private key is available on each host that you will use to access the instances. You can use any SSH utility to generate SSH keys and log in to your instances. For example, if you're logging in from a Windows host, then you can use PuTTY. If you are using a Linux host, then you can use OpenSSH.

You can associate a single SSH public key with multiple instances. You can select the SSH key file while creating an orchestration with a custom configuration.

Generating an SSH Key Pair

You must generate an SSH key pair, associate the public key with your instances, and use the private key to log in to the instances using SSH.

Generating an SSH Key Pair on UNIX and UNIX-Like Systems

Use the following procedure to generate an SSH key pair on UNIX and UNIX-like systems:

1. Run the `ssh-keygen` command.

You can use the `-t` option to specify the type of key to create. For example, to create an RSA key, run:

```
ssh-keygen -t rsa
```

You can use the `-b` option to specify the length (bit size) of the key, as shown in the following example:

```
ssh-keygen -b 2048 -t rsa
```

2. The command prompts you to enter the path to the file in which you want to save the key.

A default path and file name are suggested in parentheses, for example, `/home/user_name/.ssh/id_rsa`. To accept the default path and file name, press **Enter**. Otherwise, enter the required path and file name, and then press **Enter**.

3. The command prompts you to enter a passphrase.

The passphrase is not required, but you can use it to protect your private key against unauthorized use.

4. When prompted, enter the passphrase again to confirm it.

The command generates an SSH key pair consisting of a public key and a private key, and saves them in the specified path. The file name of the public key is created automatically by appending `.pub` to the name of the private key file. For example, if the file name of the SSH private key is `id_rsa`, then the file name of the public key is `id_rsa.pub`.

Make a note of the path and file names of the private and public keys. When you create an orchestration with a custom configuration, you must specify the SSH public key value. When you log in to an instance, you must provide the path to the corresponding SSH private key, and you must enter the passphrase when prompted.

Generating an SSH Key Pair on Windows

You can generate an SSH key pair on a Microsoft Windows machine by using an application such as PuTTY. See the tutorial, [Creating SSH Keys for Use with Oracle Cloud Services](#).

Adding an SSH Key

SSH keys are used for passwordless access to instances. Before creating any instance, you must generate an SSH key pair and upload the SSH public key that must be associated with the instance.

1. Log in to the UI and go to the **Self Service Portal**.
2. Click **Preferences**.
3. In the **Preferences** window, select **Infrastructure - Oracle Compute** in the left pane.
4. In the **Configure SSH Keys and Host Credentials** region, click **Add**.
5. In the **Configure SSH Keys and Host Credentials** window, provide the following details:
 - a. Select **SSH Key** as the Credential Type.
 - b. In the **Credential Name** field, enter the SSH key name.
 - c. Select the user name as **oracle** or **root**.
 - d. Select a credential value. If you select **Enter Key**, then you must specify the SSH public and private key pair required to access the VM instances. If you select **Key is in a File**, then click **Choose File** and select the public and private key from a file.
6. Click **OK**.

Adding an SSH Key Using the CLI or API

Use the following command to add an SSH public key:

```
oracle-compute add sshkey keyname keyfile [--enabled] [--osuser]
```

You can associate the SSH key with the operating system user; the SSH public key is appended to the users' authorized keys file `/home/<osuser>/ .ssh/authorized_keys`

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To add an SSH Key using the API, use the `POST https://api.oc.example.com/sshkey/` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Managing Networks

This section describes how to create and manage IPoIB tenant vNets, and manage IP addresses.

Topics:

- [About Networking in Oracle Compute Cloud Service](#)
- [Creating an IPoIB Tenant vNet](#)
- [Updating an IPoIB Tenant vNet](#)
- [Reserving an IP Address](#)
- [Deleting an IP Reservation](#)
- [Deleting an IPoIB Tenant vNet](#)

About Networking in Oracle Compute Cloud Service

All the client-side networks provided by Oracle Public Cloud Machine are either Ethernet over InfiniBand (EoIB) or IP over InfiniBand (IPoIB). An EoIB network connection is required for external communication, and an IPoIB network connection is required for internal communication.

There are two types of cloud networks that you can use for the instances:

- **Tenant networks**

Networks that are assigned exclusively for a specific tenant. Only the instances in that tenant can use this network.

- **Service networks**

Networks that provide access to shared services across multiple tenants. The instances using this network have either the Service Consumer or Service Provider role.

The Oracle Cloud Administrator creates and manages all the network connections for external access. Whereas, you can create and manage all the IPoIB tenant network connections for internal communication. When you want to connect to the external systems outside the cloud, use EoIB networks for the instances. For communicating with the systems inside the cloud, use IPoIB networks for the instances. You need to assign at least one network connection to launch an instance using orchestration.

IPoIB Tenant Networks

You can create and manage only the IPoIB tenant networks. An IPoIB tenant network must be created for communication among instances in the same tenant. EoIB tenant networks, and EoIB and IPoIB service networks are all created and managed by the Oracle Cloud Administrator.

You can create IPoIB networks either by specifying the IP subnet from which the IP addresses will be used for the instances to be launched, or specifying the global IP pool and the subnet is derived from the pool.

A global IP pool allows IPoIB networks to be created without specifying the IP CIDR or ranges. The IP subnet is allocated from a common pool of IP addresses. You can define the network size while creating the vNet, and the smallest subnet is derived from the global IP pool. A unique CIDR is assigned to the vNet and includes all the IP addresses represented by that CIDR. The IP subnet network size is determined by the following method:

1. Add 2 to the number of IP addresses requested while creating the IPoIB vNet.
2. The result is rounded to the nearest power of 2.

For example, if you requested 5 IP address, adding 2 extra addresses to it makes 7 and rounding to the nearest power of 2 is 8. The IP subnet is created with 8 IP addresses. The first and last addresses of a subnet are not allocated to instances as unicast addresses, therefore the balance (6 addresses) is assigned to the vNet for allocation.

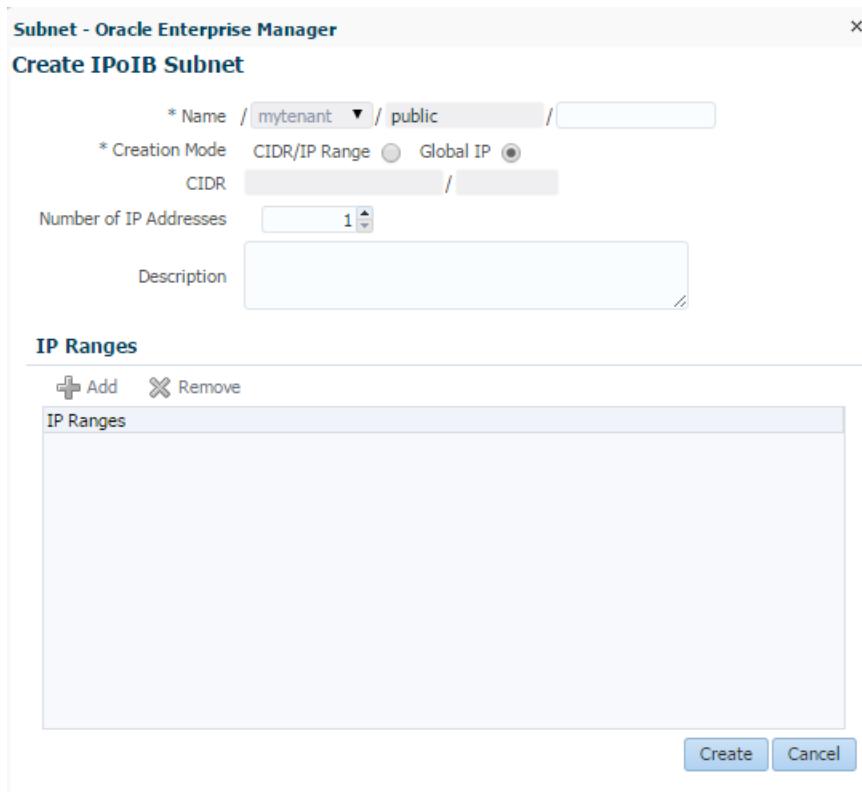
Creating an IPoIB Tenant vNet

A network subnet contains a range of IP addresses that can be used for an instance.

To create IPoIB subnets, specify the IP subnet details or the IP subnet is automatically derived from the global IP pool.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Self Service Portal**, click the **Networks** tab.
3. Select **Tenant Networks** and click **Create**.

The **Create IPoIB Subnet** window is displayed.



4. Enter a name for the subnet.
5. For **Creation Mode**, select **CIDR/IP Range** or **Global IP**.
 - a. If **CIDR/IP Range** is selected, then enter the CIDR for the IP address. Click **Add** in the **IP Ranges** section and enter the IP range.
 - b. If the IP subnet is derived from the global IP pool, then enter the number of IP addresses that are required from the IP pool.
6. (Optional) Enter a description for the IPoIB subnet.
7. To create the IPoIB subnet, click **Create**.

The IPoIB tenant vNet is created with the required number of IP addresses.

Creating an IPoIB Tenant vNet Using the CLI or API

Before you create a vNet, you must create the virtual Ethernet (vEthernet) to be used by vNet. Use the following command to create a vethernet:

```
oracle-compute add vethernet name description type
```

Use the following command to create an IPoIB vNet:

```
oracle-compute add vnet name vethernet [--cidr] [--ipranges] [--gateway] [--description] [--global_ip_num]
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To create an IPoIB tenant vNet using the API, use the `POST https://api.oc.example.com/vnet/` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Updating an IPoIB Tenant vNet

You can update the IPoIB tenant vNet details such as the CIDR and IP range.

The following table describes the conditions to update vNet details, such as IP CIDR, IP Range, and Global IP Pool.

IP CIDR	IP Range	Global IP Pool
No IP addresses reserved in the vNet or assigned to an instance.	Even if all the IP addresses in the range are reserved or in use, you can increase the number of IP addresses in the range.	No IP addresses reserved in the vNet or assigned to an instance.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Self Service Portal**, click the **Networks** tab.
3. Select **Tenant Networks** and choose the subnet that you want to modify.
4. Click **Edit**.
The **Update IPoIB Subnet** window is displayed.
5. For vNets created using CIDR, modify the CIDR and IP ranges. For vNets created using global IP pool, modify the number of IP addresses.
6. To save the changes, click **Edit**.

Updating an IPoIB Tenant vNet Using the CLI or API

Use the following command to update an IPoIB subnet:

```
oracle-compute update vnet name [--cidr] [--ipranges] [--gateway] [--description] [--global_ip_num]
```

If you precede the IP range by +, then the IP range that is specified is added to the current set of IP addresses. If you precede the IP range by -, then the IP range that is specified is deleted from the current set of IP addresses.

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To update an IPoIB tenant vNet using the API, use the `PUT https://api.oc.example.com/vnet/name` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Reserving an IP Address

You can reserve an IP address from the available tenant networks in a tenant. The reserved IP address can be a public or private IP address that you want to assign to an instance. The specified IP address must not be in use by an existing instance, and must be within the IP range of the vNet.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Self Service Portal**, click the **Networks** tab.

3. Select **Tenant Networks** and choose the network from which you want to reserve the IP address.
4. Select a network from the list and click **Add** in the IP Reservation section.

The **Create IP Reservation** window is displayed.

5. Enter a name for the IP reservation.
6. Enter the IP address that you want to reserve.
7. Click **OK**.

The selected IP address in the subnet is reserved.

Reserving an IP Address Using the CLI or API

Use the following command to reserve an IP address in a network associated with a tenant:

```
oracle-compute add vnetreservation name vnet --ip
```

If you did not specify the IP address in the command, then an IP address from the set of available IP addresses assigned to the vnet is automatically reserved.

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To reserve an IP address using the API, use the `POST https://api.oc.example.com/vnetreservation/` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Deleting an IP Reservation

When you do not need the IP reservation, you can delete it. Ensure that no instance is using the reserved IP address before you delete it.

If the IP reservation was created by using an orchestration, then you can delete an IP reservation by stopping the orchestration. See [Stopping an Orchestration](#).

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Self Service Portal**, click the **Networks** tab.
3. Select **Tenant Networks** and select a network from the list.
4. In the **IP Reservation** section, select the reserved IP address.
5. Click **Delete** and confirm the delete action.

Deleting an IP Reservation Using the CLI or API

Use the following command to delete an IP reservation in a network associated with a tenant:

```
oracle-compute delete vnetreservation name
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To delete an IP reservation using the API, use the `DELETE https://api.oc.example.com/vnetreservation/name` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Deleting an IPoIB Tenant vNet

You can delete an IPoIB tenant vNet if it is not attached to any running instances and there are no reserved IP addresses in the vNet.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Self Service Portal**, click the **Networks** tab.
3. Select **Tenant Networks** and select the IP subnet that you want to delete.
4. Click **Delete** and confirm the delete action.

Deleting an IPoIB Tenant vNet Using the CLI or API

Use the following command to delete an IPoIB vNet:

```
oracle-compute delete vnet name
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To delete an IPoIB vNet using the API, use the `DELETE https://api.oc.example.com/vnet/name` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Working with Orchestras

This section describes how to manage orchestrations.

Topics:

- [About Orchestrations](#)
- [Workflow for Creating Instances Using Orchestrations](#)
- [High Availability Policies](#)
- [Discovering Orchestrations](#)
- [Starting an Orchestration](#)
- [Shutting Down an Orchestration](#)
- [Restarting an Instance](#)
- [Stopping an Orchestration](#)
- [Updating an Orchestration](#)
- [Deleting an Orchestration](#)

About Orchestrations

An orchestration is a container that groups one or more objects, such as instances and storage volumes, to manage them as a unit. Within an orchestration, you can specify dependencies to control the sequence in which the objects are created and destroyed. For example, if an instance uses a storage volume, then the storage volume must be created before the instance can be launched, and the instance must be stopped before the storage volume can be destroyed.

Orchestrations are specified in JavaScript Object Notation (JSON) format and contain one or more plans where each plan represents objects of a specific type, such as instances and storage volumes. Each plan defines the properties of one or more objects. For example, a launch plan can define one or more instances including templates, networks, storage, and resource requirements.

To build an orchestration file in JSON format, see [Orchestration Templates](#) for more information about orchestration templates.

Life Cycle of an Orchestration

To use an orchestration, it must be added to the system. After it is added to the system, the orchestration can be started, shut down, restarted, stopped, or deleted. Starting an orchestration creates the objects defined by the orchestration. Stopping it destroys all the objects created by the orchestration. After an orchestration has been stopped, it can be started again and new objects are created.

Deleting an orchestration removes the definition from the system. Orchestrations can only be deleted if they are stopped.

Shutting down an orchestration shuts down all the instances defined by the orchestration without releasing any of the network or storage resources allocated when the instance was created. Restarting an orchestration restarts all the instances defined by the orchestration, reusing the network and storage resources allocated when the instance was created. This is equivalent to powering off and powering on the instances.

See [Orchestration Life Cycle](#) for more information about orchestration status.

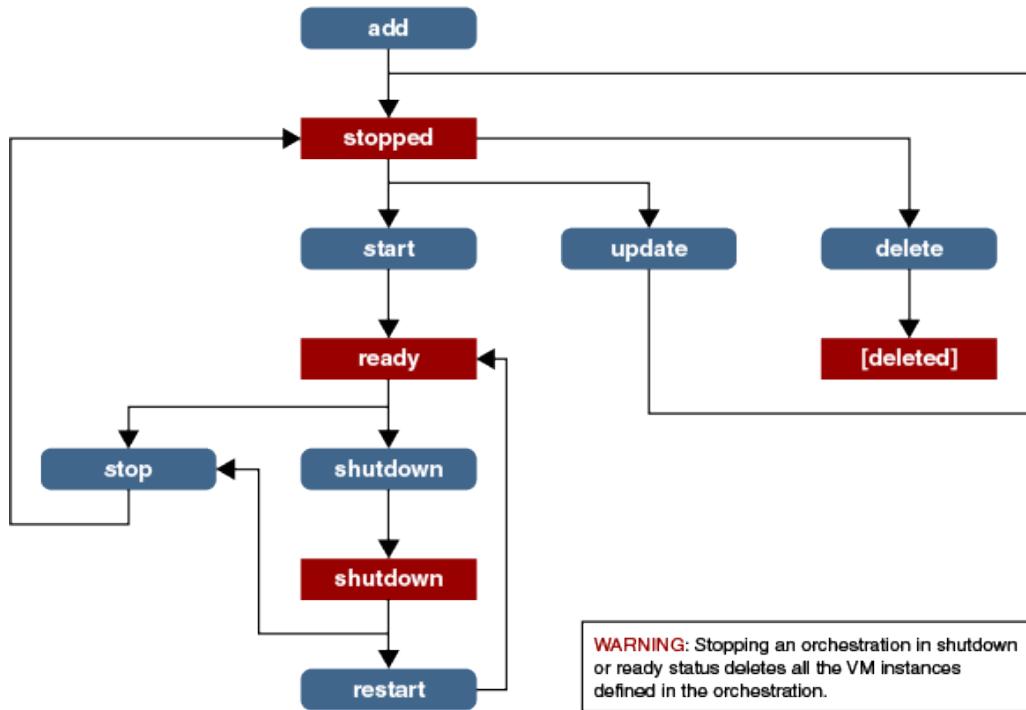
When an Oracle Cloud Administrator does a back up of orchestrations and instances in OPCM, the orchestrations that you created are backed up only if they meet the following requirements:

- The orchestration must contain the definition of one or more instances. The orchestration cannot define other IaaS objects without defining an instance.
- The orchestration must not create IaaS objects that are used by instances of other orchestrations.

Orchestration Life Cycle

An orchestration can be in terminal and transitional statuses.

The orchestration statuses are shown in the following figure:



Terminal Statuses

A terminal status is reached upon completion of an operation such as start or stop. The terminal statuses of an orchestration after performing an operation on it are:

- Ready: The orchestration is in a ready status after starting (creating an orchestration) or restarting a stopped or a shutdown orchestration. All the objects

described by the orchestration are created successfully, and all the instances are running.

- Shut_down: All the instances are gracefully shut down. Shutting down each instance individually is equivalent to shutting down the orchestration as a whole.
- Stopped: The orchestration exists in the system, but none of the objects described in the orchestration are created. All the instances in the orchestration are stopped, and the resources are released.
- Error: When a failure occurs during starting or restarting the orchestration.
- Warning: When at least one instance is not running either as a result of explicitly issuing a command to change the status or the result of a failure.

Transitional Statuses

A transitional status indicates that an operation is in progress on the orchestration. The transitional statuses for an orchestration are:

- Starting: When starting a new or stopped orchestration
- Shutting_down: When shutting down an orchestration in ready status
- Stopping: When stopping an orchestration in ready, error, or shut_down status
- Restarting: When restarting an orchestration in shut_down or error status

Workflow for Creating Instances Using Orchestrations

An orchestration defines the attributes and interdependencies of a collection of compute, networking, and storage resources in Oracle Compute Cloud Service. Orchestrations are specified in JSON format and are used to launch the instances in Oracle Compute Cloud Service.

Note:

For information on using the Oracle Compute CLI to manage orchestrations, see [Managing Orchestrations](#)

To use an orchestration to create instances and manage the resources:

1. Build your orchestration.

The web console helps you build your orchestration where you can configure one or more launch plans. Each launch plan can define one or more instances including templates, networks, storage, and other resource requirements. You can save the orchestration as a deployment plan and use it as a template to create other orchestrations quickly. The web console also provides the view of the orchestration in JSON format that you can copy and edit using a text editor.

2. Start the orchestration immediately or schedule to start and end the orchestration at a specified date and time.

Starting the orchestration creates the defined objects in the orchestration.

3. Shut down the orchestration to gracefully shut down all the instances in the orchestration.

All the resources associated with the instances are preserved.

4. Restart the shutdown orchestrations to move the instances in to a running state.
5. Stop the orchestration to delete the defined objects in the orchestration.

High Availability Policies

The high availability (HA) policy for an orchestration affects how the orchestration is managed by the system.

Values for the Policy

The high availability policy can take one of the following values:

- Active: Enables HA for instances. Monitors the state of the VM and automatically restarts the VM if it stops unexpectedly due to one of the following causes:
 - Node power outage
 - Node network failure
 - Hypervisor failing for any reason

 **Note:**

The instance might be restarted on a different node.

 **Note:**

- If an instance fails to launch successfully or it does not reach the running state, then the orchestration reflects the error state and the instance cannot be restarted. To restart the orchestration, first stop it, resolve the error, and then start the orchestration again.
- When a compute node goes down or the network connectivity is lost, the instance is in unreachable state and it is not restarted. If the network connectivity is restored, the compute node is up and the VM is still running, then the instance state transitions to running. If the compute node goes down and the VM is not running, then the instance goes to error state and it is restarted on other compute node.
- When the compute node is unreachable, the instances with HA policy set to active are not restarted automatically.
- An instance will not be automatically restarted unless it is verified that the corresponding virtual machine is not running. Check with your cloud administrator to verify that the virtual machine of the instance is not running. A cloud administrator can forcefully shut down the instance and restart it.
- There are no agents running in the instance reporting on the instance's health. Things like kernel panics must be handled by the instance itself.

- **Monitor:** Monitors the state of the instance and notifies administrators if the instance stops, fails, or becomes unreachable. The components are not restarted. This state is set only for objects of type `launchplan` and `storagevolume`. Only instances and storage volumes can be monitored. Monitoring certain components in an orchestration allows an aggregation of status for the set of monitored components. For example, if you want to integrate the aggregate status of a set of instances, create them in a single launch plan within an orchestration and set the `ha_policy` on the launch plan to monitor. You can then query the status of the orchestration, and if any of the instances are not in a running state, it will be reflected in the orchestration status.

 **Note:**

Launch plans are always monitored by default regardless of the `ha_policy` element setup in the orchestration.

If you do not want the components to be restarted or monitored, do not include the `ha_policy` element for the components configuration defined in the orchestration.

Discovering Orchestrations

You can discover the orchestrations that are not discovered and displayed in the UI, and promote them as instances. For example, if you used the CLI to create and start

an orchestration, then the orchestration does not display automatically in the UI. Instead, you must discover the orchestration from the UI to view it.

Orchestrations in ready state are discovered and displayed.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Self Service Portal**, click the **Home** tab.
3. From the **Actions** menu, select **Discover** and then **Orchestration**.
4. To confirm the action to discover the orchestrations, click **Submit**.

Discovering Orchestrations Using the CLI or API

Use the following command to discover orchestrations in a tenant:

```
oracle-compute discover orchestration container
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To discover orchestrations in a tenant using the API, use the `GET https://api.oc.example.com/orchestration/container` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Starting an Orchestration

When you start an orchestration, the objects defined in it are created. You start an orchestration that is in ready or stopped state.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Self Service Portal**, click the **Home** tab.
3. Select the orchestration that you want to start.
4. From the **Actions** menu, select **Start**.
5. To confirm the start action, click **Yes**.

When all the objects defined in the orchestration are created, the orchestration state changes to Ready. When an orchestration cannot create an object defined in it, the state of the orchestration changes to Error. Identify and fix the problem in the orchestration and start the orchestration again.

Starting an Orchestration Using the CLI or API

Use the following command to start a new or stopped orchestration:

```
oracle-compute start orchestration name
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To start an orchestration using the API, use the `PUT https://api.oc.example.com/orchestration/name?action=START` method. You must first add an orchestration using the `POST https://api.oc.example.com/orchestration/` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Shutting Down an Orchestration

When you shut down an orchestration, all the instances are gracefully shut down and their network configuration, storage volumes, and disks metadata associated with the instance are preserved. You can restart a shutdown orchestration to move instances to a running state.

You can shut down an orchestration in the following states:

- Ready
- Error
- Unreachable

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Self Service Portal**, click the **Home** tab.
3. Select the orchestration that you want to shut down.
4. From the **Actions** menu, click **Shutdown**.

All the instances in the orchestration are shut down and their configurations are preserved.

Shutting Down an Orchestration Using the CLI or API

Use the following command to shut down an orchestration:

```
oracle-compute shutdown orchestration name
```

An Oracle Cloud Administrator can shut down an orchestration in the Unreachable state using the `force` option in the command only after it is verified that the virtual machine of the instance is not running.

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To shut down an orchestration using the API, use the `PUT https://api.oc.example.com/orchestration/name?action=SHUTDOWN` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Restarting an Orchestration

You restart an orchestration that is in `Shut_down` or `Error` status.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Self Service Portal**, click the **Home** tab.
3. Select the orchestration that must be restarted in the **Services** section.
4. From the **Actions** menu, select **Start**.
5. Confirm the start action.

Restarting an Orchestration Using the CLI or API

Use the following command to restart a shutdown orchestration:

```
oracle-compute restart orchestration name
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To restart an orchestration using the API, use the `PUT https://api.oc.example.com/orchestration/name?action=RESTART` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Stopping an Orchestration

When you **stop** an orchestration, all the instances and other resources provisioned by the orchestration are **permanently deleted**.

Only the resources that are created by an orchestration are deleted. Orchestration must be in `Ready` or `Shut_down` status to be stopped.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Self Service Portal**, click the **Home** tab.
3. Select the orchestration that you want to stop.
4. From the **Actions** menu, click **Stop**.
5. To confirm the stop action, click **Yes**.

Stopping an Orchestration Using the CLI or API

Use the following command to stop an orchestration:

```
oracle-compute stop orchestration name [--confirm]
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To stop an orchestration using the API, use the `PUT https://api.oc.example.com/orchestration/name?action=STOP` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Updating an Orchestration

You can modify the configuration parameters for the instances in an orchestration. To update an orchestration, you must stop the orchestration.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Self Service Portal**, click the **Home** tab.
3. In the **Services** section, click the orchestration that you want to update.
The Oracle Compute Orchestration page is displayed.
4. In the **Oracle Compute Orchestration** menu, click **Update**.
The **Orchestration: Deployment Configuration** wizard is displayed.
5. In the Request Orchestration: General step, you can edit the orchestration description and schedule.
Click **Next**.

6. In the Request Orchestration: Deployment Configuration step, you can edit the existing launch plans and instances configuration, and you can also add more launch plans and instances.

See [Creating Instances Using Orchestration with Custom Configurations](#) for more information about the launch plan and instance configuration parameters. Click **Next**.

7. Review the configuration parameters for the orchestration, and click **Submit** to submit the update orchestration request.

Updating an Orchestration Using the CLI or API

Download a recent copy of the orchestration in JSON format to a file, and then make the necessary changes to the file.

To update an orchestration, the orchestration must be in the `Stopped` state. You can download the details of a specific orchestration using the following command:

```
oracle-compute download orchestration name [--filename]
```

Edit the orchestration file and replace the orchestration file using the following command:

```
oracle-compute update orchestration name [--filename] [-o plan_label] [--ha_policy]
```

For more information, see [CLI Reference for Oracle Compute Cloud Service](#).

To update an orchestration using the API, use the `PUT https://api.oc.example.com/orchestration/name` method.

For more information, see [REST API Reference for Oracle Compute Cloud Service](#).

Deleting an Orchestration

You must stop an orchestration before deleting it.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Self Service Portal**, click the **Home** tab.
3. Select an orchestration that you want to delete.

 **Note:**

Stop the orchestration before you proceed to delete it. See [Stopping an Orchestration](#).

4. Click **Delete**.
5. Select whether you want to delete the orchestration immediately or schedule the date and time to delete it.
6. Click **OK**.

Deleting an Orchestration Using the CLI or API

Use the following command to delete an orchestration in `Stopped` state:

```
oracle-compute delete orchestration name
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To delete an orchestration using the API, use the `DELETE https://api.oc.example.com/orchestration/name` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Managing Instances

This section describes how to create instances using orchestrations, and manage instances.

Topics:

- [About Instances](#)
- [Creating Instances Using Orchestrations](#)
- [Attaching a Storage Volume to an Instance](#)
- [Detaching a Storage Volume from an Instance](#)
- [Creating a Snapshot of an Instance](#)
- [Shutting Down an Instance](#)
- [Restarting an Instance](#)
- [Updating Instances](#)
- [Deleting an Instance](#)
- [Accessing an Instance Using SSH](#)
- [Accessing an Instance Using VNC](#)
- [Accessing the Metadata of Instances](#)

About Instances

An instance is a logical representation of a compute resource that has attributes that defines network, storage RAM, and vCPU requirements.

When an instance is launched, a virtual machine is created and configured based on the instance attributes. During the life of a single instance, many virtual machines are created and destroyed if the instance is shut down and restarted. Storage and network resources allocated when the instance is launched are released only when the instance is deleted. Storage and network resources are not released when the instance is shut down. This enables an instance to be shut down and restarted any number of times, similar to powering off and powering on a physical machine. Each time an instance is shut down the corresponding virtual machine is destroyed and the compute resources it was using (RAM, vCPU) are also released. When the instance is restarted, a new virtual machine is created and configured to use the existing storage and network resources, and new compute resources (RAM, vCPU) are allocated.

The identity of the instance is also preserved across shutdown or restart boundaries. Instances can only be created or launched through orchestrations. A single orchestration can contain one or more instances, and relationships can be defined among the instances to control their relative placement on the physical hardware. For example, instances that need to share resources or communicate frequently need to be placed on the same node. However, if an application must be highly available, then each instance must be placed on a different node. Once an instance is running, it can

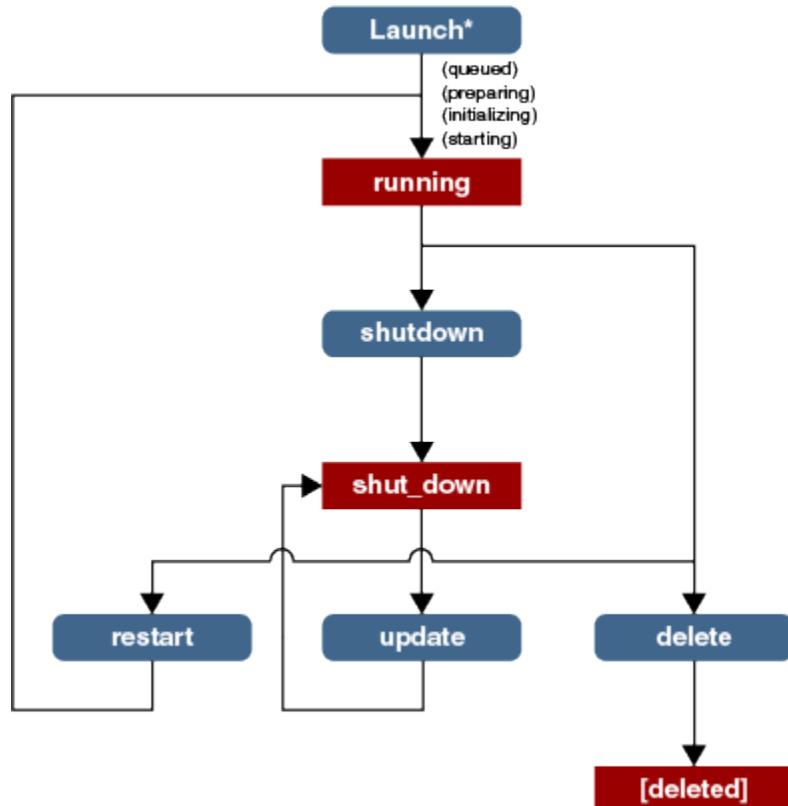
be shut down and restarted individually. If an instance is shut down outside the context of the owning orchestration, then the orchestration registers a warning state indicating that an instance it expects to be running was explicitly shut down by a user. If an instance terminates unexpectedly, then the orchestration registers an error state.

When defining an instance, you can specify the following attributes:

- Number of CPUs and RAM in the form of shapes
- EoIB network for external access
- IPoIB network for internal communication
- Service networks
- Domain Naming System (DNS)
- NIS
- Storage volumes
- SSH keys for passwordless access
- Attributes

Instance Life Cycle

An instance lifecycle is shown in the following figure:



An Oracle Compute Cloud Service instance can have one of the following states:

- **Terminal states**
 - running: The instance is successfully launched.
 - shut_down : A running instance is gracefully shut down. Memory and vCPU resources are released. Storage, networking configuration, and the tenant's quota are preserved. Storage resources are detached, but not deleted.
 - error: The instance failed. For example, when you create an instance by starting its orchestration, if some of the resources required to create the instance are not available, then the status of the instance changes to error.
- **Transitional states**
 - starting: When starting a new instance or restarting an instance in the shut_down or error state. The associated image has been fetched, network and security has been set up, and the hypervisor has been instructed to launch the instance.
 - shutting_down: When the shut down of an running instance is in progress.
 - stopping: When stopping an instance in running or error status.
 - paused: When creating a snapshot of a running instance.
 - unreachable: The compute services cannot determine the state of a virtual machine when the node on which the instance was running has shut down or is currently rebooting, one of the control plane services is not accessible due to a network error, the node on which the virtual machine was running has shut down or is currently rebooting, or one of the control plane services is not running. After the connectivity is restored automatically, the state of the instance is updated. If the virtual machine is running, then the instance state returns to running state. Otherwise, the instance state is set to error.

Creating Instances Using Orchestrations

Enterprise Manager UI provides two options to create orchestrations. A simple orchestration or an orchestration with custom configuration to create a set of instances. You can also use the CLI to add and start an orchestration.

Topics:

- [Creating Instances Using a Simple Orchestration](#)
- [Creating Instances Using Orchestrations with Custom Configurations](#)
- [Creating Instances Using the CLI](#)

Creating Instances Using a Simple Orchestration

You can use the simple orchestration option from the UI when you want to create one or more instances with the default configuration and minimum input. The simple orchestration option is available only from the UI.

Prerequisites

To create a simple orchestration, you must have the following:

- Template

OPCM provides two machine images for Oracle Linux 5.11 and 6.5. The images are optimized for OPCM and include the essential packages that are necessary to get started using the instance that you create in Oracle Compute Cloud Service.

- Network configuration

Specify one or more network interfaces to be attached to the instance. At least one network interface must be configured. See [Managing Networks](#).

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Self Service Portal**, click the **Home** tab.
3. In the **Services** section, click **Request New Service**.

The **Request New Service** window is displayed.

4. Click **Select** in the **Simple Orchestration** offering.

The **Request Simple Orchestration** page is displayed.

5. Select a site from the Oracle Compute Site list.
6. In the **Name** field, enter the name of the orchestration.
7. Select a template from the Oracle Compute Template list.
8. Enter the number of instances to create.
9. Select the **High Availability (HA)** policy. The policy can be Active or Monitor. If HA is not required, then select **No High Availability**.
10. Select a shape from the list.

A shape is a resource profile that specifies the number of CPUs and the amount of memory to be allocated to an instance in Oracle Compute Cloud Service.

11. In the **Network Configuration** section, click **Add** to add a network. You can select either a tenant network or service network to be assigned to the instance. If a reserved IP is available, then you can choose an IP address for static assignment to the instance.
12. Select to start the orchestration immediately or schedule when to start and end the orchestration.
13. Click **Submit**.

Creating Instances Using Orchestrations with Custom Configurations

You can create orchestrations with custom configurations for multitier application deployment.

Prerequisites

For instances to be launched using the orchestration, you must define the following elements for the orchestration:

- SSH key
- Network configuration.
- Storage volumes

You can also save the orchestration as a deployment plan for creating other orchestrations.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Self Service Portal**, click the **Home** tab.
3. In the **Services** section, click **Request New Service**.
4. Click **Select** for creating an orchestration to create a set of instances with custom configurations for multitier applications.

The Request Orchestration wizard is displayed.

5. In the Request Orchestration: General step, enter the following details:
 - Select the Oracle Compute Site.
 - In the **Name** field, enter a name for the orchestration.
 - (Optional) Provide a description for the orchestration.
 - Select **None** if you want to specify the orchestration elements in the wizard or you can select a saved JSON file for the orchestration or a deployment plan to create the orchestration.
 - Select to start the orchestration immediately or schedule when to start and end the orchestration.

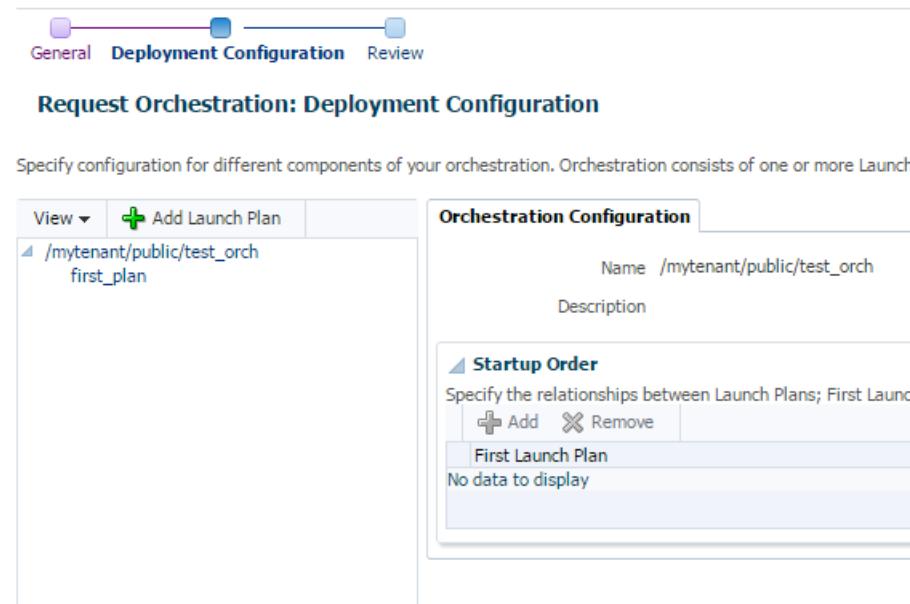
Click **Next**.

6. In the Request Orchestration: Deployment Configuration step, click **Add Launch Plan** to define the configuration settings for one or more instances.

Specify the configuration for the different elements of your orchestration using one or more launch plans. A launch plan defines the configuration settings for one or more instances. An orchestration consists of one or more launch plans and their startup order.

7. Enter a name for the launch plan and click **Add**.

The launch plan name you entered appears in the launch panel.



8. Click the newly created launch plan.

The **Launch Configuration** and **Common Settings** tabs are displayed in the **Deployment Configuration** page.

9. In the **Launch Configuration** tab, enter the following details:
 - a. Specify the high availability policy for an orchestration, which affects how the orchestration is managed by the system. You can select **Active**, **Monitor**, or **No High Availability**.
 - b. To deploy the launch plan on different compute nodes, select **Deploy on Different Nodes**. Otherwise, the launch plans may be deployed on the same nodes.
10. In the **Common Settings** tab, enter the following details:
 - a. Oracle Compute Template: Click the Search icon and select a template from the list.
 - b. Shape: Select a shape from the drop-down list.
 - c. Configure EM Agent: Select this option to configure the EM agent on the instances. This option is disabled if the tenant does not have access to the EoIB OMS and IPoIB instance storage networks that are required to deploy the management agent on the instances, and the selected template is not configured for the EM agent.
 - d. Credentials: You can set the SSH key credentials for passwordless access to the instances. You can use any existing SSH key available in the tenant.
 - Select from Compute Site: Select any existing SSH keys from the list.
 - Enter Key: If you generated an SSH key pair, then enter the SSH public key.
 - Key is in a File: If you stored the SSH key in a file, then click **Choose File** and select the public key stored in a file.
 - e. Network Configuration: Click **Add** to select one or more network interfaces that will be assigned to the instances. The networks can be either a tenant or service network.

 **Note:**

- If you have selected the **Configure EM Agent** option, you must add at least one network to your orchestration for it to successfully boot up.
- If you have not selected the **Configure EM Agent** option, you must add at least two networks to your orchestration for it to successfully boot up.

- f. (Optional) Domain Name System (DNS) Configuration: Enter the DNS servers and search domains for the selected network.
- g. (Optional) Network Information System (NIS) Configuration: Enter the NIS Domain name and add the NIS servers list.
- h. (Optional) Storage Configuration: You can create and attach storage volumes to the instances. Click **Create** and enter the details of the storage volume such as name, size, and number of volumes.

- (Optional) Advance Configuration: You can specify additional attributes and tags to the instances. For attributes, click **Add**. In the **Add Attributes** window, specify the key and value for the attribute. In the **Tags** field, you can enter tags to identify the instance deployment request.

11. After you have defined the launch plan, you can add one or more instances that can use this launch plan. In the left navigation pane, click **Add Oracle VM Instance**. In the **Add VM Instance** window, specify the name and number of the VM instances to be created. If you are adding more than one instance, then the name you specify will be used as a prefix to generate a unique name for each VM instance. Define configuration settings at the instance level and override the settings defined at the launch plan level.

You can select each instance and override the common launch plan settings.

12. Click **Next**.

The **Request Orchestration: Review** page is displayed.

13. To save the orchestration as a deployment plan that can be used for other orchestrations, click **Save as Deployment Plan**. You can also view the JSON format of the orchestration configuration. You can copy this orchestration to a file and save it for later use.

For more information about orchestration templates, see [Orchestration Templates](#).

14. Review the configuration parameters for creating the orchestration and click **Submit** to submit the orchestration request.

The orchestration is started, and it launches the instances as configured.

Creating Instances Using the CLI

For using the CLI, the name and configuration parameters of an orchestration are stored in a file in JavaScript Object Notation (JSON) format.

See [Orchestration Templates](#) for more information about orchestration format.

An example of working with an orchestration to add, start, monitor, and stop the instantiation of an instance is shown here. The orchestration in this example is stored in a file named `myorch.json`.

A sample format of the file:

```
{  
  "name": "/mytenant/public/single_instance",  
  "description": "Orchestration containing a single instance",  
  "oplans": [  
    {  
      "obj_type": "launchplan",  
      "label": "single_instance_plan",  
      "objects": [  
        {  
          "instances": [  
            {  
              "name": "/mytenant/public/single_instance",  
              "label": "single_instance",  
              "shape": "oc3",  
              "imagelist": "/oracle/public/linux6_16.1.1_64",  
              "networking": {  
                "net0": {  
                  "vnet": "/mytenant/public/ipoib-vnet"  
                }  
              }  
            }  
          ]  
        }  
      ]  
    }  
  ]  
}
```

Use the following steps to add and start the orchestration:

1. Add the orchestration to the system.

```
oracle-compute add orchestration myorch.json
```

When you add the orchestration, the state of the orchestration is stopped.

2. Start the orchestration.

```
oracle-compute start orchestration /mytenant/public/single_instance -f json
```

```
{
  "list": [
    {
      "status": "starting",
      "info": {},
      "warnings": {}
    },
    "account": "/mytenant/default",
    "description": "Orchestration containing a single instance",
    "relationships": [],
    "uri": "https://api/orchestration/mytenant/public/single_instance",
    "oplans": [
      {
        "status": "starting",
        "info": {},
        "warnings": {}
      },
      "obj_type": "launchplan",
      "ha_policy": "",
      "label": "single_instance_plan",
      "objects": [
        {
          "instances": [
            {
              "networking": {
                "net0": {
                  "vnet": "/mytenant/public/myvnet"
                }
              },
              "name": "/mytenant/public/single_instance",
              "uri": null,
              "label": "single_instance",
              "shape": "oc3",
              "imagelist": "/oracle/public/linux6_16.1.1_64"
            }
          ]
        }
      ]
    ],
    "status_timestamp": "2015-11-30T08:07:30Z",
    "name": "/mytenant/public/single_instance"
  }
}
```

```
]  
}
```

3. Watch the orchestration progress.

```
oracle-compute get orchestration /mytenant/public/single_instance -  
Fname,status
```

The orchestration moves to the Ready status if no errors are reported. The orchestration is in full operation.

The instance is launched and moves to the running state.

Attaching a Storage Volume to an Instance

When you attach a storage volume to an instance, the volume remains attached until the instance or storage attachment is deleted.

1. Log in to the UI and go to the **Self Service Portal**.

If you are performing this task in a 17.1.2 system see also [Attaching a Storage Volume to an Instance on X5](#).

2. In the **Home** tab, click an orchestration in the **Services** section.

All the instances in the orchestration are displayed.

3. In the **Name** column, click an instance for which you want to attach the storage volume.

4. In the instance details page, click **Attach** in the **Storage** section.

5. In the **Attach Storage** window, select a storage volume that you want to attach to the instance.

6. Enter the index value for the storage volume. The index value must be between 1 and 10.

The index number that you specify here determines the device name. The volume attached at index 1 is named /dev/vda or /dev/sda (depending on the use of VirtIO drivers by the instance), and the volume at index 2 is /dev/[sv]db and so on.

7. Click **Attach**.

Attaching a Storage Volume to an Instance Using the CLI or API

Use the following command to attach a storage volume to an instance:

```
oracle-compute add storageattachment instance_name storage_volume_name  
index
```

The index value must be between 1 and 10.

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To attach a storage volume to an instance using the API, use the POST <https://api.oc.example.com/storage/attachment> method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Attaching a Storage Volume to an Instance on a 17.1.2 System

If you are attaching a storage volume to an instance on an 17.1.2 system, perform the following best practices:

1. When you create a new instance, the customer must define the necessary storage attachments in the orchestration (attach during creation). This is preferable to attaching volumes after the instance is running, defined as a hot plug. see the example for *Single Instance with Storage Attachment* at [Sample Orchestrations](#).
2. If an instance has hot plugged storage attachments, then the customer should keep a copy of the instance definition using the command `oracle-compute get instance <instance-name> -fjson`. The customer can use this saved copy to look up the hot plugged storage attachments and reattach them in case they get detached, such as in the event of a power outage. If a power outage already happened, the hot plug information can't be retrieved using the command.

Detaching a Storage Volume from an Instance

You can detach the storage volume from an instance. If the instance is in the `running` state, then you must first unmount the storage volume in the instance. Log in to the instance, identify the storage volume, and unmount the storage volume using the command `umount path-to-disk-mount-point`.

 **Note:**

You cannot delete a storage volume that was attached while creating an instance.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Home** tab, click an orchestration in the **Services** section.
All the instances in the orchestration are displayed.
3. In the **Name** column, click an instance from which you want to detach a storage volume.
4. In the instance details page, select the volume and click **Detach** in the **Storage** section.
5. Confirm the detach action.

Detaching a Storage Volume from an Instance Using the CLI or API

Use the following command to detach a storage volume from an instance:

```
oracle-compute delete storageattachment attachment_name
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To detach a storage volume from an instance using the API, use the `DELETE https://api.oc.example.com/storage/attachment/name` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Creating a Snapshot of an Instance

When you take a snapshot of an instance, a copy of the instance's boot disk at that time is taken. The result of an instance snapshot is a machine image. You can then create new instances from the machine images generated from the snapshots.

For example, if you created an instance and configured the environment for a specific purpose, then you can take a snapshot of that instance. You can then use the snapshot to replicate the same environment. When you use that snapshot to create instances, the instance network configuration is updated based on the new instance's setting.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Home** tab, click an orchestration in the **Services** section.
All the instances in the orchestration are displayed.
3. In the **Name** column, click an instance for which you want to take a snapshot.
4. In the instance details page, click **Create** in the **Snapshot** section.
5. Enter a name and description for the snapshot and click **OK**.

The snapshot of the instance is stored as a template in the UI.

Creating a Snapshot of an Instance Using the CLI or API

Use the following command to take a snapshot of an instance:

```
oracle-compute add snapshot instance [--name] [--machineimage] [--quota]
```

When you specify the machine image name in the option, the machine image is generated from the snapshot request.

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To take a snapshot of an instance using the API, use the `POST https://api.oc.example.com/snapshot/` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Shutting Down an Instance

When you shut down an instance, the instance is gracefully shut down and its network configuration, storage volumes, and disk metadata associated with the instance are preserved. Shutting down an instance is necessary for maintenance or for updating the configuration of an instance, such as changing the shape. You can shut down only a running instance.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Self Service Portal**, click the **Home** tab.
3. Click the orchestration name in which the instance is running.
All the instances in the orchestration are displayed with their status.
4. Select the instance that you want to shut down and click **Shutdown**.

5. Click **Shutdown** to confirm the action.

The selected instance is shut down and its configuration is preserved.

Shutting Down an Instance Using the CLI or API

Use the following command to shut down a running instance:

```
oracle-compute shutdown instance name
```

You can also shut down the instance in the `error` state if it has been in the `running` state at least once. You can use the `force` option in the command when the instance is in the `unreachable` state, the associated virtual machine is not running, and the virtual machine will not recover on its own. You can force the shut down of the instance and then restart it.

```
oracle-compute shutdown instance name --force
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To shut down an instance using the API, use the `PUT http://api.oc.example.com/instance/name?action=SHUTDOWN` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Restarting an Instance

You restart an instance that is in `shut_down` or `error` status.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Home** tab, click an orchestration in the **Services** section.
All the instances in the orchestration are displayed.
3. Select the instance that is in `shut_down` or `error` status
4. To start the instance, click **Start**.
5. Confirm the start action.

Restarting an Instance Using the CLI or API

Use the following command to restart an instance that is in `shut_down` or `error` state:

```
oracle-compute restart instance name
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To restart an instance using the API, use the `PUT https://api.oc.example.com/instance/name?action=RESTART` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Updating Instances

This section describes how to scale up and scale down an instance of its CPU and memory resources.

Topics:

- [Changing an Instance Shape](#)
- [Updating the Network Configuration of an Instance](#)
- [Attaching a Network to an Instance](#)

Changing an Instance Shape

You can change the shape of an instance to change the vCPU and memory configuration of an instance. You must shut down the instance, change the shape, and then restart the instance.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Home** tab, select and click an orchestration.
All the instances in the orchestration are displayed.
3. In the **Name** column, click an instance for which you want to change the shape.
4. In the instance details page, click the **Edit** icon for **Shape** in the **Configuration** section.
5. Select the desired shape from the list and click **OK**.
6. Start the instance to apply the new shape.

Changing an Instance Shape Using the CLI or API

Use the following command to change the shape of an instance:

```
oracle-compute update instance name [--shape]
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To change the shape of an instance using the API, use the `PUT https://api.oc.example.com/instance/name` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Updating the Network Configuration of an Instance

You must shut down an instance before updating the network information. The changes are applied after restarting the instance.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Home** tab, click an orchestration in the **Services** section.
All the instances in the orchestration are displayed.
3. In the **Name** column, click an instance for which you want to update the network configuration.

4. In the instance details page, click **Edit** in the **Network** section.

The **Edit Network** window is displayed. All the network connections attached to the instance are listed.

5. In the **Edit Networks** window, you can edit one or more network interfaces of the instance.

You can modify the IP address, hostname, and the following parameters of an EoIB network:

- Default Gateway: The selected network will be configured as the default gateway network.
- Symmetric Routing: If this option is selected, the reply packets are routed through the same network interface on which the vNet IP packets were received.
- Provider Role: If this option is selected, the instance becomes the service provider and provides shared services through the selected service network.

6. Click **Submit**.

Updating the Network Configuration of an Instance Using the CLI or API

Use the following command to update the network configuration of an instance. The networking option of the command uses the same syntax of the networking object type of the orchestration JSON file.

```
oracle-compute update instance name [--shape] [--networking] [--nis] [--tag] [--delete_tag]
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To update the network configuration of an instance using the API, use the `PUT https://api.oc.example.com/instance/name` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Attaching a Network to an Instance

You can attach a tenant or service network to an instance. The IPoIB network type facilitates the internal communication, and the EoIB network type facilitates the external communication.

Note:

If you are updating an instance hosting a Database Cloud Service RAC database, you must shut down the instance before attaching a network.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Home** tab, click an orchestration in the **Services** section.
All the instances in the orchestration are displayed.
3. In the **Name** column, click an instance for which you want to attach the network.
4. In the instance details page, click **Attach** in the **Network** section.

5. In the **Select Network** window, select one or more tenant or service networks that you want to attach to the instance.
6. Click **Attach**.

Attaching a Network to an Instance Using the CLI or API

Use the following command to attach a network to an instance:

```
oracle-compute update instance name [--networking]
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To attach a network to an instance using the API, use the `PUT https://api.oc.example.com/instance/name` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Deleting an Instance

You can delete an instance only if it is in `running`, `shut_down` or `error` state.

Before deleting an instance, consider the following options:

- If you want to free the compute resources, then shut down the instance. See [Shutting Down an Instance](#).
- If you want to delete all the instances in an orchestration, then stop the orchestration. See [Stopping an Orchestration](#).
- If you want to delete only one of the instances in an orchestration, then use the CLI or REST API to delete the instance.

When an instance is deleted, the associated orchestration transitions to the `warning` status. You need to stop and start this orchestration if you want to associate it with a new instance.

Using the CLI or REST API

Use the following command to delete an instance in an orchestration:

```
oracle-compute delete instance name
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To delete an instance using the API, use the `DELETE https://api.oc.example.com/instance/name` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Accessing an Instance Using SSH

You can use SSH to log in to your instance.

Prerequisites

- The SSH public key must be associated with the instance. See [Enabling Secure Access to Instances Using SSH](#).

- The SSH private key corresponding to the public key associated with the instance must be available on the host from which you want to SSH to the instance.
- The IP address of the instance that is in the running state. The IP address must be accessible from your remote host.

Using SSH

Get the IP address of the instance using the following command:

```
oracle-compute list instance container -Fip,label,state
```

For example, list the instances in the container /mytenant/public and filter the result for the IP address, instance label, and state:

```
oracle-compute list instance /mytenant/public -Fip,label,state
```

In the filtered output, identify the IP address of the instance that you want to log in.

You can log in to your instance as the default `root` user by using the following command:

```
ssh root@ip_address -i private_key.
```

Accessing an Instance Using VNC

In release 16.1.2 and onwards, you can access instances using VNC. After you log in to the instance, you can perform standard operations on the operating system.

Before you can access your instance using VNC, your Oracle Cloud administrator must enable VNC access. Contact your Oracle Cloud Administrator to verify if VNC access has been enabled.

1. Log in to the UI and go to the **Self Service Portal**.
2. Click the **Home** tab.
3. In the **Services** section, click the orchestration that contains the instance. All the instances in the orchestration are displayed.
4. Under the **VNC URL** column, click the URL for the instance that you want to access.

A Java Network Launching Protocol (JNLP) file is downloaded.

5. Open the JNLP file to start the VNC console.

Note:

If an “Application Blocked by Java Security” error message is displayed, you must add the displayed address to the Exception Site List. For more information on editing the Exception Site List, see the following link:
https://www.java.com/en/download/faq/exception_sitelist.xml

Accessing the Metadata of Instances

Two types of metadata are stored within your instances: user-defined instance attributes that you can define explicitly while creating instances, and predefined instance metadata fields that are stored by default for all instances.

You can refer to the user-defined instance attributes defined as instance attributes in the structure of an orchestration JSON file at [Orchestration Templates](#). You can also specify the attributes when you upload a private template. The attributes specified in the template are passed on to the instance. See [Uploading a Private Template](#).

You can retrieve an instance's metadata at 192.0.0.192. After you log in to an instance, use the following commands to retrieve the user-defined instance attributes, predefined instance metadata, and generic attributes:

```
curl http://192.0.0.192/latest/user-data/attribute_key_name  
curl http://192.0.0.192/latest/meta-data/attribute_key_name  
curl http://192.0.0.192/latest/attributes/attribute_key_name
```

Upgrading Oracle Compute Cloud Service Instances

This section describes how to upgrade your Oracle Compute Cloud Service instances.

This procedure is intended for upgrading the following instances:

- Your Oracle Compute Cloud Service instances
- Oracle Compute Cloud Service instances that host your PaaS instances

Oracle Cloud Ops handles the upgrades to the infrastructure and control services of your Oracle Public Cloud Machine (OPCM). However, Oracle Cloud Ops does not patch your user-created IaaS and PaaS instances. After Oracle Cloud Ops completes an upgrade of your machine, you must upgrade the Oracle Compute Cloud instances for all your user-created IaaS and those hosting your PaaS instances as described in this section.

Note:

The patches described in this section patch the kernel and drivers of an instance.

- For your Oracle Compute Cloud Service instances, you must patch the other packages in your instances using Oracle Public Yum. The Yum exclusion list is already part of the Yum configuration of the image. Users must not override the exclusion list either by using a different configuration file or by using `--disableexcludes` option
- For your PaaS instances, you must also patch the PaaS software using the associated service console, CLI, or API.

Topics

- [Configure the Yum Repository Server](#)
- [Set Up the Patch Bundle](#)
- [Patch Your Instances](#)

Configure the Yum Repository Server

Before patching Oracle Compute Cloud Service instances, you must create a local repository server. This server should be hosted on an Oracle Linux server.

To create the server, do the following:

1. Log in to the server on which you want to host the repository server.
2. Install the `httpd` package:

```
sudo yum install httpd
```

3. Configure the `/etc/httpd/conf/httpd.conf` file:
 - a. Edit the `/etc/httpd/conf/httpd.conf` file:

```
vi /etc/httpd/conf/httpd.conf
```

3. Configure the `/etc/httpd/conf/httpd.conf` file:
 - b. In the `<Directory />` section of the file, ensure that `Options FollowSymLinks` is present.

Example:

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>
```

Set Up the Patch Bundle

To set up the patch bundle on the repository server, do the following:

1. Download the patch bundle from My Oracle Support.
 - a. Navigate to My Oracle Support at:
<https://support.oracle.com>
 - b. Log in with your My Oracle Support account.
 - c. Click the **Patches & Updates** tab.
 - d. Under the Patch Search section, click **Product or Family (Advanced)**.
 - e. In the **Product** field, enter Oracle Public Cloud Machine.
 - f. In the **Release** field, select the release that you are currently running.

For example, if you were upgraded from release 16.3.1 to 17.1.2, select Oracle Public Cloud Machine 17.1.2.0.0.

- g. Click the **Search** button.

The patch bundle is typically called Patchset For Oracle-Compute Instance OS Release - X.Y.Z, where X.Y.Z denotes the version number.

- h. Click the patch bundle that is displayed.
- i. Click the **Download** button.
- j. To start the download, click the filename of the patch.

2. On your local repository server, switch to the `root` user.
3. Create a `patchrepo` directory in the `DocumentRoot` directory of the server. By default, the `DocumentRoot` is `/var/www/html`:

```
mkdir /var/www/html/patchrepo
```

4. Navigate to the `patchrepo` directory

```
cd /var/www/html/patchrepo
```

5. Copy the patch bundle from the machine on which you downloaded the bundle to this location:

```
scp Address_Of_Machine:Full_Path_to_Bundle /var/www/html/patchrepo
```

6. Unzip the downloaded bundle:

```
unzip FileNameofBundle
```

Example:

```
unzip p25468544_1712_Linux-x86-64.zip
```

The `.tgz`patch bundle and the `readme` file are extracted.

7. Extract the `.tgz` bundle:

```
tar --no-same-owner -xvf FileNameofTgzBundle
```

Example:

```
tar --no-same-owner -xvf p25468544_1712-64-Linux-x64.tgz
```

The following directories are created:

- `additional_packages` (17.1.2 bundle only)
- `scripts` (17.1.2 bundle only)
- `template`
- `updates`

8. Edit the template/exalogic.repo

```
vi template/exalogic.repo
```

9. If the file has the following lines, delete them:

```
[exalogic]
name = exalogic
baseurl = http://<http-server>:<port><DocumentRoot/SubDirectory>/base/
active/$releasever/$basearch
gpgcheck = 0
enabled = 1
```

10. Under [exalogic-updates], replace the following:

- Replace <http-server> with the address of the server.
- Replace <port> to 80 if you are using the default port.
- Replace <DocumentRoot/SubDirectory> with /patchrepo

Example:

```
[exalogic-updates]
name = exalogic-updates
baseurl = http://myhost.mydomain:80/patchrepo/updates/
active/$releasever/$basearch
gpgcheck = 0
enabled = 1
[root@http-server template]#
```

11. Start the httpd server:

```
service httpd start
```

12. Verify that the httpd server is running by accessing the patchrepo location in a browser.

Patch Your Instances

Describes how to patch your instances to the required version.

- [Patch Your 16.1.2 or 16.3.1 Instances to 17.1.2](#)
- [Patch Your 16.1.2 Instances to 16.3.1](#)

Patch Your 16.1.2 or 16.3.1 Instances to 17.1.2

1. Log in to the instance you want to patch as the `root` user.
2. Navigate to the `/etc/yum.repos.d` directory

```
cd /etc/yum.repos.d
```

3. Download the `exalogic.repo` file from the repository server to your instance:

```
wget http://http-repo-server:port/patchrepo/template/exalogic.repo
```

4. Identify the version of Oracle Linux of your guest instance:

```
lsb_release -rs | cut -f1 -d.
```

- If 6 appears, the instance is an Oracle Linux 6 instance.
- If 5 appears, the instance is an Oracle Linux 5 instance.

5. Based on the OS of your guest instance, patch the instance as follows:

- For Oracle Linux 6 instances:

- a. Download the following scripts using `wget`:

```
wget http://http-repo-server:port/patchrepo/scripts/  
update_instance.sh  
wget http://http-repo-server:port/patchrepo/scripts/  
remove_old_kernel.sh
```

- b. Add execute permissions to the downloaded scripts:

```
chmod +x update_instance.sh remove_old_kernel.sh
```

- c. Update the instance:

```
./update_instance.sh
```

- d. Shut down the instance as described in [Shutting Down an Instance](#)

- e. Restart the instance as described in [Restarting an Instance](#)

- f. Wait for the instance to be accessible.

- g. Log in to the instance as the root user.

- h. Run the `remove_old_kernel.sh` script:

```
./remove_old_kernel.sh
```

- For Oracle Linux 5 instances:

- a. Download the following script using `wget`:

```
wget http://http-repo-server:port/patchrepo/scripts/  
update_instance.sh
```

- b. Add execute permissions to the downloaded script:

```
chmod +x update_instance.sh
```

- c. Update the instance using the script:

```
./update_instance.sh
```

d. Download the kernel header package using `wget`:

```
wget http://http-repo-server:port/patchrepo/additional_packages/  
kernel-headers-2.6.18-417.0.0.0.1.el5.x86_64.rpm
```

e. Install the kernel header package:

```
rpm -ivh --force kernel-  
headers-2.6.18-417.0.0.0.1.el5.x86_64.rpm
```

Patch Your 16.1.2 Instances to 16.3.1

1. Log in to the instance you want to patch as the `root` user.
2. Navigate to the `/etc/yum.repos.d` directory

```
cd /etc/yum.repos.d
```

3. Download the `exalogic.repo` file from the repository server to your instance:

```
wget http://http-repo-server:port/patchrepo/template/exalogic.repo
```

4. Remove cached packages or headers:

```
yum clean all
```

5. Run the `yum update` command to upgrade your instance:

```
yum update --disableexcludes=main -y
```

6. Verify that the instance has been updated to the new version by running the `imageinfo` command:

In this example, the instance was patched from 16.1.2 to 16.3.1.

```
imageinfo
```

```
Oracle Public Cloud Machine 16.3.1
```

```
Image version : 16.3.1
```

```
...
```

Managing Storage Volumes

This section describes how to create and manage storage volumes, and take snapshots of storage volumes.

Topics:

- [About Storage Volumes](#)
- [Creating a Storage Volume](#)
- [Creating a Storage Volume on the External Storage Appliance](#)
- [Creating a Snapshot of a Storage Volume](#)
- [Deleting a Storage Volume](#)

About Storage Volumes

A storage volume is a virtual disk that provides persistent block storage space for instances in Oracle Compute Cloud Service. You can use storage volumes to store data and applications.

You can attach one or more storage volumes to an instance either while creating the instance or later when the instance is running. After creating an instance, you can easily scale up or scale down the storage resource for the instance by attaching or detaching storage volumes. However, you cannot detach a storage volume that was attached during instance creation. Note that, when a storage volume is detached from an instance or when the instance is deleted, data stored on the storage volume is not lost.

You can also take snapshots of the storage volumes at any time. See [About Snapshot of Storage Volumes](#).

About Snapshot of Storage Volumes

A snapshot is a copy made of the storage volume at any instant. A snapshot of a storage volume is another storage volume.

When you want to attach the storage volume to an running instance, you can back up the original volume by taking the snapshot of it.

When a storage volume is attached to a running instance and is running out of space. You can take a snapshot of the storage volume with an increased size of the volume. Then, you can attach the new storage volume and detach the old storage volume.

Creating a Storage Volume

You can create empty storage volumes and attach them to instances. You can even create storage volumes from existing storage volumes.

 **Note:**

You can only run single node Oracle databases on Oracle Compute Cloud Service on OPCM using customer-owned externally connected storage over 10 GbE. The internal block storage of OPCM cannot be used for databases running on Oracle Compute Cloud Service instances.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Storage** tab, click **Create**.
3. Enter the name, size (GB), and description for the volume.

 **Note:**

Do not select a template for the storage volume.

4. Click **OK**.

Creating a Storage Volume Using the CLI or API

Use the following command to create a storage volume:

```
oracle-compute add storagevolume name size properties [--description] [--source_storagevolume_name] [--imagelist] [--imagelist_entry] [--tags] [--quota]
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To create a storage volume using the API, use the `POST https://api.oc.example.com/storage/volume/` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Creating a Storage Volume on the External Storage Appliance

If you have a subscription to the Database Cloud Service (DBCS), a tenant user can create storage volumes on the external storage appliance.

An external storage appliance is included in the DBCS subscription of Oracle Public Cloud Machine. You can create storage volumes on this external storage appliance to attach to Oracle Compute Cloud Service instances.

Note:

You can run all the commands in this procedure as a tenant user.

Note:

You can only run single node Oracle databases on Oracle Compute Cloud Service on OPCM using customer-owned externally connected storage over 10 GbE. The internal block storage of OPCM cannot be used for databases running on Oracle Compute Cloud Service instances.

1. Verify that the machine has the required storage properties on the external storage appliance:

- a. Verify that the `latency` storage property exists by running the `list property` command:

```
oracle-compute list property storage /oracle/public/storage/latency
```

Sample output:

uri	name	description
https://api/property/storage/oracle/public/storage/		
latency	/oracle/public/storage/latency	Latency
	Optimized	

- b. Verify that the `throughput` storage property exists by running the `list property` command:

```
oracle-compute list property storage /oracle/public/storage/throughput
```

Sample output:

uri	name	description
https://api/property/storage/oracle/public/storage/		
throughput	/oracle/public/storage/throughput	Throughput
	Optimized	

2. In the orchestration JSON section that specifies the storage volume, set the `properties` field to `/oracle/public/storage/latency` or `/oracle/public/storage/throughput`.

Example:

```
{  
  "objects": [  
    {  
      "name": "/mytenant/public/data",  
      "properties": [  
        "/oracle/public/storage/throughput"]  
    }  
  ]  
}
```

```

        ],
        "size": "50G",
        "bootable": "False"
    }
],
"label": "data",
"obj_type": "storage/volume"
}

```

In this orchestration, you are creating a storage volume of size 50 GB called /mytenant/public/data on the external storage appliance with throughput write-bias.

3. Add the orchestration to your tenant using the `add orchestration` command:

```
oracle-compute add orchestration filename
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

4. Start the orchestration using the `start orchestration` command:

```
oracle-compute start orchestration OrchestrationName
```

The *OrchestrationName* should be the hierachal name of the orchestration specified in the JSON file.

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

Mapping Database Storage Volumes to Storage Properties

If you are deploying a legacy database instance on an Oracle Compute Cloud Service instance, Oracle recommends that you follow this mapping when creating storage volumes.

The following table lists the storage volumes created for a DBCS instance and the storage property that is associated with each volume. Oracle recommends that you similarly map the storage volumes for your instance.

Storage Volume	Description	Storage Property
boot	Boot disk that hosts the Operating System.	/oracle/public/storage/latency
bits	Hosts the Oracle database software.	/oracle/public/storage/latency
redo	Hosts the redo log.	/oracle/public/storage/latency
data	Used for the files of the database.	/oracle/public/storage/throughput
fra	Used for flash recovery.	/oracle/public/storage/throughput

Sample JSON files for a Database Deployment:

In the following samples, we will use two orchestrations to deploy a database:

- Resources orchestration: Deploys the storage volumes and instance for the database.

- Launch plan orchestration: Links and launches the objects.

Sample Resources Orchestration

```
{  
  "name": "/tenant1/public/sample1-resources",  
  "oplans": [  
    {  
      "objects": [  
        {  
          "imagelist_entry": "1",  
          "name": "/tenant1/public/sample1-boot",  
          "imagelist": "/oracle/public/linux6_16.3.1_64",  
          "properties": [  
            "/oracle/public/storage/default"  
          ],  
          "size": "32GB",  
          "bootable": "True"  
        }  
      ],  
      "label": "sample1-boot",  
      "obj_type": "storage/volume"  
    },  
    {  
      "objects": [  
        {  
          "name": "/tenant1/public/sample1-data",  
          "properties": [  
            "/oracle/public/storage/throughput"  
          ],  
          "size": "25G",  
          "bootable": "False"  
        }  
      ],  
      "label": "sample1-data",  
      "obj_type": "storage/volume"  
    },  
    {  
      "objects": [  
        {  
          "name": "/tenant1/public/sample1-fra",  
          "properties": [  
            "/oracle/public/storage/throughput"  
          ],  
          "size": "10G",  
          "bootable": "False"  
        }  
      ],  
      "label": "sample1-fra",  
      "obj_type": "storage/volume"  
    },  
    {  
      "objects": [  
        {  
          "name": "/tenant1/public/sample1-bits",  
          "properties": [  
            "/oracle/public/storage/throughput"  
          ],  
          "size": "10G",  
          "bootable": "False"  
        }  
      ],  
      "label": "sample1-bits",  
      "obj_type": "storage/volume"  
    }  
  ]  
}
```

```

"properties": [
    "/oracle/public/storage/latency"
],
"size": "30G",
"bootable": "False"
}
],
"label": "sample1-bits",
"obj_type": "storage/volume"
},
{
"objects": [
{
"name": "/tenant1/public/sample1-redo",
"properties": [
"/oracle/public/storage/latency"
],
"size": "10G",
"bootable": "False"
}
],
"label": "sample1-redo",
"obj_type": "storage/volume"
}
],
"description": "Resources orch for instance sample1"
}

```

Sample Launch Plan JSON

```

{
"relationships": [],
"name": "/tenant1/public/sample1-launch",
"oplans": [
{
"ha_policy": "monitor",
"objects": [
{
"instances": [
{
"name": "/tenant1/public/sample1_instname",
"virtio": "false",
"storage_attachments": [
{
"index": 1,
"volume": "/tenant1/public/sample1-boot"
},
{
"index": 2,
"volume": "/tenant1/public/sample1-bits"
},
{
"index": 3,
"volume": "/tenant1/public/sample1-data"
}
]
}
]
}
]
}

```

```
        },
        {
            "index": 4,
            "volume": "/tenant1/public/sample1-fra"
        },
        {
            "index": 5,
            "volume": "/tenant1/public/sample1-redo"
        }
    ],
    "tags": [],
    "boot_order": [
        1
    ],
    "shape": "oc3",
    "label": "sample1",
    "networking": {
        "net0": {
            "vnet": "/tenant1/public/oeib-network",
            "options": {
                "default_gateway": "true",
                "hostname": "sample1",
                "dns": {
                    "search_domains": "example.com",
                    "servers": "10.10.10.7,10.10.10.8"
                }
            }
        }
    }
],
    "relationships": []
}
],
    "label": "sample1-launch",
    "obj_type": "launchplan"
}
],
"description": "Orchestration for instance sample1"
}
```

Creating a Storage Volume on the External Storage Appliance Using the CLI or API

You can create a storage volume on the external storage volume using the CLI by specifying one of the following storage properties:

- To create a storage volume on the latency pool of the external storage appliance, use the following command:

```
oracle-compute add storagevolume name size /oracle/public/storage/  
latency [--description] [--source_storagevolume_name] [--imagelist]  
[--imagelist_entry] [--tags] [--quota]
```

- To create a storage volume on the throughput pool of the external storage appliance, use the following command:

```
oracle-compute add storagevolume name size /oracle/public/storage/  
throughput [--description] [--source_storagevolume_name] [--imagelist]  
[--imagelist_entry] [--tags] [--quota]
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To create a storage volume using the API, use the `POST https://api.oc.example.com/storage/volume/` method. In the `properties` parameter, specify either `/oracle/public/storage/latency` or `/oracle/public/storage/throughput`.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Creating a Snapshot of a Storage Volume

You can make a copy of a storage volume at any instant. A snapshot of a storage volume is another storage volume.

Source Storage Volume Properties

If the existing storage volume is a bootable storage volume, then the new storage volume is also a boot volume. The new storage volume is an exact copy of the source storage volume, that is independent of the source. If the source storage volume is attached to a running instance, then you must unmount or freeze the disk buffers of the storage volume before taking a snapshot:

```
umount path-to-disk-mount-point
```

or

```
sync  
fsfreeze -f path-to-disk-mount-point
```

After creating the new volume, log in to the instance and enable the original file system. If the file system is unmounted, mount the file system. If the file system is frozen, then unfreeze it using the following command:

```
fsfreeze -u path-to-disk-mount-point
```

To create a snapshot of the storage volume:

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Storage** tab, select a storage volume that you want to take a snapshot of.
3. Click **Snapshot**.
4. Enter a name and description for the volume snapshot.
5. (Optional) Modify the size of the volume.
6. Click **OK**.

Creating a Snapshot of a Storage Volume Using the CLI or API

Use the following command to create a snapshot of a storage volume:

```
oracle-compute add storagevolume name size properties [--  
source_storagevolume_name] source_volume_name
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To create a snapshot of a storage volume using the API, use the `POST https://api.oc.example.com/storage/volume/` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Deleting a Storage Volume

If you delete a storage volume, then all the data and applications that were saved on that storage volume are lost.

You can delete the storage volumes that are not attached to any instances. To detach a storage volume from an instance, see [Detaching a Storage Volume from an Instance](#).

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Storage** tab, select the storage volume that you want to delete.
3. Click **Delete**.
4. Confirm the delete action.

Deleting a Storage Volume Using the CLI or API

Use the following command to delete a storage volume:

```
oracle-compute delete storagevolume name
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

To delete a storage volume using the API, use the `DELETE https://api.oc.example.com/storage/volume/name` method.

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Managing Templates

This section describes how to upload private templates and manage the templates.

Topics:

- [About Templates](#)
- [Uploading a Private Template](#)
- [Customizing an Image](#)
- [Deleting a Private Template](#)

About Templates

A template is a machine image defined in an image list.

A machine image is a copy of virtual hard disk with an installed operating system. You use a machine image to launch a virtual machine instance in Oracle Compute Cloud Service. The image must be a whole disk image (including a partition table and kernel) stored in a `tar` archive and compressed with `gzip`.

An image list is a collection of Oracle Compute Cloud Service machine images. Each machine image in an image list is identified by a unique entry number. You can group similar machine images in an image list for easy administration and usage.

When you upload a machine image using the web console, an image list is automatically created by using the name that you specified for the image. The new machine image becomes the default (and only) entry in the image list.

Oracle Public Cloud Machine provides two default Oracle Linux images, releases 5.11 and 6.5. These images are optimized for OPCM and include the essential packages that are necessary to get started using the instance that you create in Oracle Compute Cloud Service.

Note:

You cannot use any templates that are not based or developed from the OPCM default images.

Public and Private Templates

The templates that are available for all the tenants in the cloud are referred to as public templates. Oracle Cloud Administrator uploads the public templates and are available under `/oracle/public`.

The templates are available only for you and other tenant users in the tenant are referred to as private templates. You and other tenant users in the tenant upload the private templates.

Uploading a Private Template

When you upload the machine image from the web console, an image list is automatically created using the name that you specified for the image. The new machine image becomes the default (and only) entry in the image list.

Ensure that the `.tar.gz` file is available on the local disk or software library of the Enterprise Manager UI.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Self Service Portal**, click the **Library** tab.
3. Click **Upload**.
4. Enter a name for the template.
5. (Optional) Provide attributes as comma-separated key value pairs.
Attributes are user-defined parameters passed to an instance using this template when it is launched.
6. Select the template file to upload from your local system or from the Software Library.
7. Click **Upload**.

Uploading a Private Template Using the CLI or API

Uploading a private template using the CLI requires you to add the machine image, create an image list, and add an image list entry.

Use the following commands to upload a private template:

1. Add a machine image.

```
oracle-compute add machineimage name file [--attributes] [--quota]
```

2. Add an image list.

```
oracle-compute add imagelist name description [--default]
```

3. Add an image list entry.

```
oracle-compute add imagelistentry imagelist name machineimages version [--attributes]
```

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

The following table lists the API method for uploading a template.

Action	Method and URI
Add a machine image by creating and uploading a machine image	POST <code>http://api.oc.example.com/machineimage/</code> POST <code>http://api.oc.example.com/machineimage/</code> <i>ImageLocation</i>
Add an image list	POST <code>https://api.oc.example.com/imagelist/</code>
Add an image list entry	POST <code>https://api.oc.example.com/imagelist/</code> <i>name</i> <code>/entry/</code>

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

Customizing an Image

You can create new custom machine images based on the default machine images.

Oracle Compute Cloud Service provides two default standard templates or machine images based on Oracle Linux 5 and Oracle Linux 6. For example, on OPCM version 17.1.2, these images are registered as:

- /oracle/public/linux5_17.1.2_64
- /oracle/public/linux6_17.1.2_64

These default machine images include required packages and device drivers specific to OPCM. All guest instances are instantiated from these images or versions of these images.

For Oracle Compute Cloud Service, users can customize and upload new custom boot images based on one of the default machine images.

Note:

You cannot customize Platform as a Service (PaaS) machine images. For example, you cannot use the Java Cloud Service or Database Cloud Service images to create your own custom images.

Alternatively to can create a machine image for each type of instance deployment, you can use the `opc-init` package to set up the customization to be performed automatically when an instance starts. For details, see [Automating Instance Initialization Using opc-init](#)

Topics

- [Deploy and Customize an Instance](#)
- [Create Your Custom Machine Image](#)
- [Deploy Instances Using the Customized Machine Image](#)

Deploy and Customize an Instance

Describes how to deploy and customize an instance using the default machine image.

To clone an instance using an instance snapshot, first create an instance using an appropriate machine image. This instance must use a non-persistent boot volume. When your instance is running, customize your instance as required – such as by adding users, incorporating DevOps procedures to automate installation and configuration of applications once the instance boots up. These changes are stored on your non-persistent boot disk.

1. Deploy and instance using one of the default machine images. For more information, see [Creating Instances Using Orchestrations](#)
2. Log in to the instance.

3. Customize the instances to meet your requirements.

For example, you can do the following:

- Install RPMs.
- Define users.
- Customize the Operating System (kernel parameters, LVM, SELinux).
- Set up the swap space.

For setting up the swap space, the default machine images come with a swap of 4 GB. The volume group, `VolGroup00`, that manages the boot disk (18 GB) contains three logical volumes:

- `LogVol00` for root (11 GB)
- `LogVol01` for swap (4 GB)
- `LogVol02` for the Enterprise Manager agent (2 GB)

The Volume group has 764 MB of free space that can be used to increase the swap space. If you require more swap space, you can do the following:

- If you don't want to use the EM agent auto-deployment feature, re-purpose `LogVol02` for swap
- Create and attach an additional storage volume to be used as swap. This storage volume cannot be part of the machine image and every instance deployment would need a new storage volume.

Create Your Custom Machine Image

When you are done customizing your instance, to use the instance as a template to create other instances, create an instance snapshot. Instance snapshots capture the current state of your boot disk and create a corresponding machine image. These instances will contain all the configuration and customization that you'd done on the original instance when you took the snapshot.

1. Create an instance snapshot using the Oracle Compute CLI with the `--machineimage` option.

```
oracle-compute add snapshot instance_name --machineimage image_name
```

Example:

```
oracle-compute add snapshot /mytenant/public/af2c7a90-36ee-4276-a567-665913f1f4dc --machineimage /mytenant/public/mycustomimage
```

Note:

The result of an instance snapshot is a machine image. Any networking, host name, and SSH key information are not saved to the machine image. For more information, see [Creating a Snapshot of an Instance](#)

2. Add the image to the `imagelist`:

```
oracle-compute add imagelist image_name description
```

Example:

```
oracle-compute add imagelist /mytenant/public/mycustomimage  
"imagelist for custom image"
```

3. Create an `imagelistentry`:

```
oracle-compute add imagelistentry image_name imagelistentry_name version
```

```
oracle-compute add imagelistentry /mytenant/public/mycustomimage /  
mytenant/public/mycustomimage 1
```

You can delete the instance and use your custom machine image to deploy more instances.

 **Note:**

By default, this image is only available to the specified tenant. To make it accessible to all tenants on the machine, you can submit a Change Request to Oracle Cloud Ops using My Oracle Support.

Deploy Instances Using the Customized Machine Image

You can deploy instances using your custom machine image.

For more information on deploying instances, see [Creating Instances Using Orchestrations](#)

In the orchestration, ensure that you specify your customized machine image for the `imagelist` parameter:

```
{  
  "name": "/mytenant/public/single_instance",  
  "oplans": [  
    {  
      "obj_type": "launchplan",  
      "label": "single_instance_plan",  
      "objects": [  
        {  
          "instances": [  
            {  
              "name": "/mytenant/public/single_instance",  
              "label": "single_instance",  
              "shape": "oc3",  
              "imagelist": "/mytenant/public/mycustomimage",  
              "networking": {  
                "net0": {  
                  "type": "bridge",  
                  "label": "net0",  
                  "bridge": "br0",  
                  "mac": "00:0C:29:00:00:01",  
                  "ip": "192.168.1.100",  
                  "netmask": "255.255.255.0",  
                  "gw": "192.168.1.1",  
                  "dns": "8.8.8.8",  
                  "mtu": 1500  
                }  
              }  
            }  
          ]  
        }  
      ]  
    }  
  ]  
}
```

Deleting a Private Template

You can delete the templates that you uploaded to the tenant.

Ensure that the template is not used in any orchestrations.

1. Log in to the UI and go to the **Self Service Portal**.
2. In the **Self Service Portal**, click the **Library** tab.
3. Select the private template that you want to delete.
4. Click **Delete**.
5. Confirm the delete action.

Deleting a Private Template Using the CLI or API

Deleting a private template using the CLI includes deleting the machine image and the corresponding image list and image list entry.

Use the following commands to delete a private template:

```
oracle-compute delete machineimage name
oracle-compute delete imagelistentry imagelist name version
oracle-compute delete imagelist name
```



Note:

If you delete the `imagelist` before deleting the `imagelistentry`, the `imagelistentry` is also deleted as part of the `imagelist`.

For more information, see *CLI Reference for Oracle Compute Cloud Service*.

The following table lists the API methods to delete a private template.

Action	API Method
Delete a machine image.	DELETE <code>https://api.oc.example.com/machineimage/{name}</code>
Delete an image list entry.	DELETE <code>https://api.oc.example.com/imagelistentry/{name}/entry/{version}</code>

Action	API Method
Delete an image list.	DELETE https://api.oc.example.com/imagelist/name

For more information, see *REST API Reference for Oracle Compute Cloud Service*.

A

Orchestration Templates

An orchestration is stored in a file in JSON format.

The following sample JSON file shows the high-level structure of an orchestration. The primary structure is called an object plan (**oplan**). In a single orchestration, you can include up to 10 oplans. Oplans contain objects and can have dependencies declared through relationships.

Note:

- The orchestration templates use placeholder text for object names, labels, and other user-specific values. When you use these templates to build your orchestration, remember to replace placeholder values with values specific to your environment.
- In the JSON file, a comma is required between each element of an orchestration.

```
{  
  "description": "someDescriptionHere",  
  "name": "/<tenant_name>/public/<orchestration_name>",  
  "oplans": [  
    {  
      "ha_policy": "active",  
      "label": "someText",  
      "obj_type": "objectType",  
      "objects": [  
        {  
          "definitions of objects specific to this object type  
        }  
      ]  
    },  
    {  
      "label": "someText",  
      "obj_type": "objectType",  
      "objects": [  
        {  
          "definitions of objects specific to this object type  
        }  
      ]  
    },  
    .  
    .  
    .  
    . up to 10 oplans  
    .
```

```

        ]
    }

```

 **Note:**

An object is used with an orchestration to encapsulate the various orchestratable system components. For example, for a launch plan, the obj_type is launchplan. The fields in the objects section vary with the object type.

The following table describes the elements in the orchestration.

Table A-1 Orchestration Elements

Object Identifier	Description	Permitted Elements or Value
description	A description of the orchestration	Text that describes the orchestration.
name	Name of the orchestration. It consists of three components and begins with /tenant_name/public/ Example: /mytenant/public/single_instance	Text that provides the name of the orchestration.
oplans	An object plan is the primary structure of the orchestration for defining objects in an orchestration.	A list of oplans that have elements for defining different objects in an orchestration: ha_policy, label, obj_type, and objects.
label	User-specified name for an object. Labels are used to specify relationships and appear in error messages at the orchestration level to identify an oplan that is in error.	Text that describes the object.
relationships	A container structure to specify the sequence in which the components in an orchestration start. In the relationships section of the orchestration, specify the labels of the oplans and the relationship type.	A JSON structured element that describes the relationship.

The following table describes the elements in the oplans.

Table A-2 Oplan Elements

Object Identifier	Description	Permitted Values or Elements
ha_policy	An optional element to specify a high availability policy for an object type (obj_type). If you do not want high availability or monitoring, do not include the ha_policy element. Launch plans are monitored by default.	active: Only set for object type launchplan. monitor: Only set for object types launchplan and storage/volume. none: To disable HA policy or for using with objects that not supported.
obj_type	A type of object to be defined.	launchplan, storage/volume, and vnetreservation.
objects	A container structure for defining objects for an object type.	launchplan: instances, relationships. storage/ volume: name, size, properties. ip/reservation: vnetreservation.

Unless they were created beforehand, the network, storage, or other objects referred to in an orchestration are created when the orchestration starts, and they are destroyed when the orchestration stops.

The orchestration structure and the schema for its elements are described in detail in the following sections.

Orchestration Structure

```
{
  "name": <string>,
  "description": <string>,
  "oplans": [ <oplan>, ... ]
  "relationships": [ <oplan_relationship>, ... ] (optional)
}
```

The following table describes the elements of the orchestration.

Fields	Description
name	Name of the orchestration plan
description	Description of the orchestration plan
oplans	List of oplans
relationships	A list of relationship specifications to be satisfied on this orchestration

Orchestration Template for Oplans

An object plan, or oplan, is a top-level orchestration attribute. Within an object plan, you can specify various object types and define one or more objects for each object type.

```
{
  "label": <string>
  , "obj_type": "launchplan" | "storage/volume",
  "objects": [ <objtype_launchplan>, ... ] | [ <objtype_storagevolume>, ... ]
  "ha_policy": "monitor" | "active", (optional)
}
```

The following table describes the object plan attributes.

Fields	Description
ha_policy	Determines the HA behavior of an oplan. Specifying active applies only to launch plans and if an instance terminates unexpectedly, then the instance is automatically restarted. Specifying monitor applies to all object types: the state of all the objects is monitored so changes in state are reflected in the orchestration. If ha_policy is not specified, then storage volumes are not monitored. Launch plan are monitored by default and omission of this property does not have any effect on objects of type launchplan.
label	Description of the oplan.
obj_type	Type of the object.
objects	List of all objects of type obj_type.

Oplan Relationship

```
{
  "oplan": <string>,
  "to_oplan": <string>,
  "type": "depends",
}
```

The following table describes the attributes of oplan relationship.

Fields	Description
oplan	Label of oplan1
to_oplan	Label of an oplan on which oplan1 depends
type	Type of the relationship. It must be depends.

Orchestration Template for Object Type: Launch Plan

Use this object type if you want to define one or more instances. In an orchestration, instance is an attribute of the launchplan object type.

```
{
  "instances": [ <instance>, ... ]
```

```

  "relationships": [ <instance_relationship>, ... ] (optional)
}

```

The following table describes the attributes for `launchplan`.

Fields	Description
<code>instances</code>	List of instances defined by the launch plan.
<code>relationships</code>	List of relationships that exist between the instances in the launch plan and the instances that are currently running.

Orchestration Template for Object Type: Storage/Volume

Use this object type to create storage volumes that you want to attach to your instances.

```

{
  "name": <string>,
  "properties": [ <string>, ... ]
  "size": <string>,
  "description": <string>, (optional)
  "source_storagevolume_name": <string>, (optional)
  "imagelist": <string>, (optional)
  "imagelist_entry": <number>, (optional)
  "tags": [ <string>, ... ] (optional)
  "quota": <string>, (optional)
}

```

The following table describes the attributes for `storage/volume`.

Fields	Description
<code>name</code>	Name of the storage volume
<code>properties</code>	A list of the storage properties that are associated with this storage volume.
<code>size</code>	The size of this storage volume measured in the number of bytes or multiples of bytes
<code>description</code>	Description of the storage volume.
<code>source_storagevolume_name</code>	Name of existing storage volume to copy from onto this volume when created. Cannot be used when specifying <code>imagelist</code> option.
<code>imagelist</code>	Name of the machine image to extract onto this volume when created. Cannot be used when specifying the <code>source_storagevolume_name</code> option.
<code>imagelist_entry</code>	Specific image list entry version to extract.
<code>tags</code>	A list of strings that will tag the storage volume.
<code>quota</code>	The quota against which this volume will be created.

Instance Attributes

In an orchestration, `instance` is an attribute of the `launchplan` object type. If any of the objects referred to in `instance` attributes are defined in the same orchestration as the

instance, you must specify a relationship between each object and the instance launch plan.

```
{
  "shape": <string>,
  "networking": <string>,
  "imagelist": <string>,
  "label": <string>, (optional but recommended)
  "name": <string>, (optional but recommended)
  "storage_attachments": [ <storage_attachment>, ...] (optional)
  "tags": [ <string>, ...] (optional)
  "attributes": <userdata>, (optional)
  "nis": <nis>, (optional)
  "sshkeys": [ <string>, ...] (optional)
  "entry": <number>, (optional)
  "quota": <string>, (optional)
}
```

The following table describes the attributes for instances.

Fields	Description
shape	Defines the size or resource requirements (memory/CPUs) for the instance. Must correspond to one of the shapes defined in the site configuration.
networking	Mapping of the device name to network specifiers for virtual NICs to be attached to the instance.
imagelist	Name of the image list to be launched.
label	A label assigned by the user, specifically for defining inter-instance relationships. Optional but recommended. Also used in some error and status messages. If not specified, a label will be automatically assigned.
name	Name of the instance generated by the server. It is optional but recommended. Specifying a name prefix makes it easier to find and identify the instances when using CLI commands, such as <code>list instance</code> . Users can specify a name prefix and a system generated UUID will be appended to the prefix. Example: <code>tenant_name/public/instance_name</code> results in the instance name <code>tenant_name/public/instance_name/uuid</code> . If the name is not specified, the name will be <code>tenant_name/public/uuid</code> .
storage_attachments	List of dictionaries that contain storage attachment information.
tags	A list of strings that will tag the instance.
attributes	A collection of user-defined key and value pairs that is made available to an instance through the metadata service.
placement_requirements	Not applicable.
nis	Network Information Service configuration.
sshkeys	The SSH key that is installed on the instance.
entry	Optional image list entry number. The default will be used if one is not specified.
quota	The quota against which this instance will be run.

Relationships Attributes

In an orchestration, `relationships` is an attribute of the `launchplan` object type.

```
{
  "instances": [ <string>, ... ]
  "type": "same_node" | "different_node",
}
```

The following table describes the attributes for `relationships`.

Fields	Description
<code>instances</code>	List of instance labels. Can also use <code>instance: running instance name</code> to create a relationship between an instance in the <code>launchplan</code> and an instance that is already running.
<code>type</code>	Type of relationship between the specified instances.

nis Attributes

```
{
  "domain": <string>,
  "servers": [ <nis_server>, ... ]
}
```

The following table describes the `nis` attributes.

Fields	Description
<code>domain</code>	The NIS domain name.
<code>servers</code>	List of NIS server entries.

nis_server Attributes

```
{
  "name": <string>,
  "ip": <string>,
}
```

The following table describes the attributes for `nis_server`.

Fields	Description
<code>name</code>	The name of the NIS server.
<code>ip</code>	The IP address of the NIS server.

userdata Attributes

```
{
  "userdata": { <string>: <string>, ... }
}
```

The following table describes the attributes for `userdata`.

Fields	Description
<code>userdata</code>	A collection of user-defined key and value pairs that will be made available to an instance through the metadata service.

Networking Attributes

```
{
  <string>: <networkblock>,
  ...
}
```

The following table describes the attributes for `networking`.

Fields	Description
<code>string</code>	A unique label for the network described by <code>networkblock</code> . A short string is used such as <code>net0</code> . Network entries are processed in sorted order using this key and not the order they appear in the document.

storage_attachment Attributes

```
{
  "volume": <string>,
  "index": <number>,
}
```

The following table describes the attributes for `storage_attachment`.

Fields	Description
<code>volume</code>	Name of the storage volume to which the instance will be attached.
<code>index</code>	The integer index used when generating the device name, for example, 1 in <code>/dev/sda1</code> .

networkblock Attributes

```
{
  "vnet" | "vnetreservation": <string>,
  "options": <networkblock_options>, (optional)
}
```

The following table describes the attributes for `networkblock`.

Fields	Description
vnet	The full vNet name. Only one vNet or vNet reservation can be specified.
options	The set of options for the networking configuration of the instance. When more than one network contain options, the first block to specify a particular option is used and the others are ignored.

networkblock_options Attributes

```
{
  "dns": <networkblock_dns>, (optional)
  "hostname": <string>, (optional)
  "network_route": [ <networkblock_route>,... ] (optional)
  "default_gateway": <boolean>, (optional)
  "provider": <boolean>, (optional)
  "primary": <boolean>, (optional)
}
```

The following table describes the attributes for `networkblock_options`.

Fields	Description
hostname	Specifies the host name associated with the instance network IP. When multiple host names are specified on multiple networks, the primary host name must be indicated with boolean <code>primary</code> attribute. Otherwise, the host name on the first network (sorted by key) is used as the instance host name.
network_route	A list of routing rules.
default_gateway	A Boolean flag that indicates that this network's gateway must be used for default routing. The default is false if not set.
provider	A Boolean flag that can be used to configure the instance as a provider (full membership) on an underlying service network.
primary	A Boolean flag that indicates that the host name specified for this network must be used as the instance host name. The default is false if not specified. If not specified as true for any network, then the first network (sorted by key) will be used as the instance host name.

networkblock_dns Attributes

```
{
  "search_domains": <string>,
  "servers": <string>,
}
```

The following table describes the attributes for `networkblock_dns`.

Fields	Description
search_domains	A single string in comma-separated value format that specifies the set of DNS search domains to configure for the instance.
servers	A single string in comma-separated value format that specifies the set of DNS servers to configure for the instance.

networkblock_route Attributes

```
{  
  "type": "rule" | "static" | "policy"  
  "name": "SameNICReply" # only if "rule"  
  "ip_network": <string>, # only if "static" or "policy"  
  "via": <string>, # only if "static" or "policy"  
}
```

The following table describes the attributes for `networkblock_route`.

Fields	Description
type	A string that must be <code>rule</code> , <code>static</code> , or <code>policy</code> . All other properties must be specified as documented according to this value.
name	Only valid if type is <code>rule</code> . A string that specifies the name of the rule to use. Currently only <code>SameNICReply</code> is accepted. This rule results in a source-based routing rule created on the instance, designed to route reply packets back through the same network interface on which the packets to the vNet IP were received.
ip_network	Only valid if the type is <code>static</code> or <code>policy</code> . A string that specifies the destination IP address range in the CIDR format.
via	Only valid if type is <code>static</code> or <code>policy</code> . A string that specifies the gateway IP address.

B

Sample Orchestration

Sample orchestrations are provided in this section for your reference. Substitute the variables with your tenant names, network, and storage information.

Simple Single Instance

This orchestration creates a simple instance.

```
{
  "name": "/mytenant/public/single_instance",
  "description": "Orchestration containing a single instance",
  "oplans": [
    {
      "obj_type": "launchplan",
      "label": "single_instance_plan",
      "objects": [
        {
          "instances": [
            {
              "name": "/mytenant/public/single_instance",
              "label": "single_instance",
              "shape": "oc3",
              "imagelist": "/oracle/public/linux6_16.3.1_64",
              "networking": {
                "net0": {
                  "vnet": "/mytenant/public/ipoib1-vnet"
                },
                "net1": {
                  "vnetreservation": "/mytenant/public/vnetreservation_1"
                },
              },
            }
          ]
        }
      ]
    }
  ]
}
```

Simple Single Volume

This orchestration creates only a storage volume. It does not create an instance.

```
{
  "description": "Orchestration containing a single storage volume",
  "name": "/mytenant/public/voll",
```

```

"oplans": [
  {
    "ha_policy": "monitor",
    "label": "tools",
    "obj_type": "storage/volume",
    "objects": [
      {
        "name": "/mytenant/public/voll",
        "properties": [
          "/oracle/public/storage/default"
        ],
        "size": "1G"
      }
    ]
  }
]
}

```

Single Instance with Storage Attachment

This orchestration creates an instance with a storage attachment and two network connections. One network connection is based on vNet and the other one is based on vNet reservation.

```

{
  "description": "Instance with storage attachment and vnetreservation",
  "name": "/mytenant/public/instance_with_attachment",
  "oplans": [
    {
      "label": "instance-with-attachment",
      "obj_type": "launchplan",
      "objects": [
        {
          "instances": [
            {
              "name": "/mytenant/public/instance_ref_sv",
              "label": "instance-with-attachment",
              "shape": "oc3",
              "imagelist": "/oracle/public/linux6_16.3.1_64",
              "networking": {
                "net0": {
                  "vnet": "/mytenant/public/ipoib1-vnet"
                },
                "net1": {
                  "vnetreservation": "/mytenant/public/vnetreservation_1"
                }
              },
              "storage_attachments": [
                {
                  "index": 1,
                  "volume": "/mytenant/public/voll"
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}

```

```

        }
    ]
}
]
}
```

Instance with Storage Volume

This orchestration creates an instance and storage volume. The storage volume is attached to the instance.

```
{
  "name": "/mytenant/public/mytenant_sv_instance",
  "description": "Orchestration containing a storage volume and an
instance using that volume",
  "oplans": [
    {
      "label": "instance-with-attachment",
      "obj_type": "launchplan",
      "objects": [
        {
          "instances": [
            {
              "imagelist": "/oracle/public/linux6_16.3.1_64",
              "label": "instance-with-attachment",
              "name": "/mytenant/public/instance/with/attachment",
              "networking": {
                "net0": {
                  "vnet": "/mytenant/public/ipoib1"
                }
              },
              "shape": "oc3",
              "storage_attachments": [
                {
                  "index": 1,
                  "volume": "/mytenant/public/vol10"
                }
              ]
            }
          ]
        }
      ],
      "label": "volume-vol10",
      "obj_type": "storage/volume",
      "objects": [
        {
          "name": "/mytenant/public/vol10",
          "properties": [
            "/oracle/public/storage/default"
          ],
          "size": "10G"
        }
      ]
    }
  ]
}
```

```

        }
    ],
    "relationships": [
        {
            "oplan": "instance-with-attachment",
            "to_oplan": "volume-vol10",
            "type": "depends"
        }
    ]
}

```

Multiple Instances with Instance Relationships

This orchestration creates three instances in which two instances have same node relationship and two instances have different node relationships.

```

{
    "description": "3 instances with same node/different node relationships",
    "name": "/mytenant/public/relationships",
    "oplans": [
        {
            "ha_policy": "",
            "label": "related-instances",
            "obj_type": "launchplan",
            "objects": [
                {
                    "instances": [
                        {
                            "imagelist": "/oracle/public/linux6_16.3.1_64",
                            "label": "instance-a",
                            "name": "/mytenant/public/a",
                            "networking": {
                                "net0": {
                                    "vnet": "/mytenant/public/ipoib1-vnet"
                                }
                            },
                            "shape": "oc3"
                        },
                        {
                            "imagelist": "/oracle/public/linux6_16.3.1_64",
                            "label": "instance-b",
                            "name": "/mytenant/public/b",
                            "networking": {
                                "net0": {
                                    "vnet": "/mytenant/public/ipoib1-vnet"
                                }
                            },
                            "shape": "oc3"
                        },
                        {
                            "imagelist": "/oracle/public/linux6_16.3.1_64",
                            "label": "instance-c",
                            "name": "/mytenant/public/c",
                            "networking": {
                                "net0": {

```

```
        "vnet": "/mytenant/public/ipoib1-vnet"
    }
},
"shape": "oc3"
},
"relationships": [
{
    "instances": [
        "instance-a",
        "instance-b"
    ],
    "type": "same_node"
},
{
    "instances": [
        "instance-b",
        "instance-c"
    ],
    "type": "different_node"
}
]
}
]
```