Oracle® Cloud

Migrating Infrastructure Classic Workloads to Oracle Cloud Infrastructure

F13813-13
May 2020

Oracle Cloud Migrating Infrastructure Classic Workloads to Oracle Cloud Infrastructure,

F13813-13

Contents

5 Identify and Translate Resources in Your Source Environment

6 Migrate Users and Groups

11 Select a Method to Migrate Database Instances

12 Migrate Databases Using the Migration Tools

13 Learn About Migrating a Database Cloud Service Deployment to a Virtual Machine Database System

19 Migrate Object Storage

Preface

Topics:

This guide summarizes the steps required to migrate the resources and data in your Oracle Cloud Infrastructure Compute Classic, Oracle Cloud Infrastructure Object Storage Classic, Oracle Database Classic Cloud Service, Oracle Cloud Infrastructure Load Balancing Classic, and Oracle Cloud Infrastructure FastConnect Classic accounts, to your second generation Oracle Cloud Infrastructuretenancy.

Audience

This document is intended for users who are considering migrating their compute, storage, networking, and database resources to Oracle Cloud Infrastructure. To complete the migration procedures described in this document, you must have access to an Oracle Cloud Infrastructure tenancy with policies that allow you to create the required resources.

This document assumes that you are familiar with Oracle Cloud Infrastructure Compute Classic and Object Storage Classic. If you're considering migrating your Load Balancer Classic, FastConnect Classic, or Oracle Database Cloud Service resources, then it is assumed that you're familiar with those services as well.

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 Resources

For more information, see these Oracle resources:

- Oracle Cloud Infrastructure documentation

  https://docs.cloud.oracle.com/iaas/Content/home.htm
- Oracle Cloud Infrastructure Getting Started

  https://docs.cloud.oracle.com/iaas/Content/GSG/Concepts/baremetalintro.htm
- Oracle Cloud Infrastructure Object Storage Classic documentation

  https://docs.oracle.com/en/cloud/iaas/storage-cloud/index.html
- Oracle Cloud Infrastructure Compute Classic documentation

  https://docs.oracle.com/en/cloud/iaas/compute-iaas-cloud/index.html
- Oracle Cloud Infrastructure Load Balancing Classic documentation

  https://docs.oracle.com/en/cloud/iaas/load-balancer-cloud/index.html
- Oracle Database Cloud Service documentation

  https://docs.oracle.com/en/cloud/paas/database-dbaas-cloud/index.html

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.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction to Migrating to Oracle Cloud Infrastructure

Oracle provides tools, guidelines, and instructions that can help you migrate your existing Oracle Cloud resources from an environment based on Oracle Cloud Infrastructure Compute Classic to Oracle Cloud Infrastructure.

Topics:

- Understand the Benefits of Moving to Oracle Cloud Infrastructure
- Consider a General Migration Strategy

Understand the Benefits of Moving to Oracle Cloud Infrastructure

Oracle strongly encourages customers to migrate their existing cloud resources from Oracle Cloud Infrastructure Compute Classic regions. There are several advantages to doing so.

In Oracle Cloud, you provision resources in specific data regions, which are localized geographic locations. In general, a region supports one of two infrastructure platforms: Oracle Cloud Infrastructure or Oracle Cloud Infrastructure Compute Classic. Oracle Cloud Infrastructure is Oracle's more modern infrastructure platform that's based on the latest cloud technologies and standards, and typically provides better performance than Oracle Cloud Infrastructure Compute Classic. Oracle Cloud Infrastructure also provides more predictable pricing and lower costs (OCPU per hour). Most importantly, Oracle continues to invest in Oracle Cloud Infrastructure, including the addition of new regions, services and features.

You can take advantage of these additional administrative features in Oracle Cloud Infrastructure when you migrate from Oracle Cloud Infrastructure Compute Classic:

- Organize cloud resources in logical compartments, and assign fine-grained access policies to each compartment.
- Distribute cloud resources across availability domains and fault domains for high availability and disaster recovery.

Consider a General Migration Strategy

When migrating to Oracle Cloud Infrastructure there are several distinct strategies that you can consider. These high-level strategies can help guide you through the planning of your migration and help you prioritize your mirgration requirements.

Strategy	Description	Example
Rehost	This strategy for migration is also referred to as "Lift and Shift". It involves taking a snapshot of the application server VM's on the source environment including the OS, boot record, converting into an Oracle Cloud Infrastructure-compatible format (qcow2, vmdk) and importing it on Oracle Cloud Infrastructure as a custom Image, and then re-instantiating the image.	You have non-Oracle applications, running on a VM on Oracle Cloud Infrastructure Compute Classic. These applications cannot be easily rebuilt. In this case, you create a snapshot of the entire image (OS, application, configuration information, and data), import it into Oracle Cloud Infrastructure and rerun it there–essentially, you are rehosting the application server from Oracle Cloud Infrastructure Compute Classic to Oracle Cloud Infrastructure. The application can also be an on-premise database deployment, which was installed and configured on the source virtual machine.
Replatform	This strategy involves rebuilding or redeploying the application on an upgraded operating system.	You create a new Oracle Cloud Infrastructure virtual machine, based on a new version of the Oracle Linux operating system with the latest security updates. You can then redeploy your application on the new virtual machine and the new operating system. For example, consider the task of migrating PeopleSoft using Cloud Manager. This operation reinstalls PeopleSoft on a new Oracle Cloud Infrastructure VM, and moves just the configurations and data over.
Refactor	This strategy involves redesigning and refactoring your application with cloud design points to make it more cloud native.	You redesign and rewrite your application to make extensive use of Oracle Cloud Infrastructure constructs and Oracle Cloud Infrastructure PaaS services, so you get the maximum benefit out of moving to the new infrastructure. This may be the only solution to migrating complex network configurations.

Strategy	Description	Example
Retire/Replace	These are not covered in detail here. With this strategy, you basically discard your application or buy another application. In these situations, there are no migration scenarios.	N/A

2

Review the Oracle Cloud Infrastructure Compute Classic Architecture

Before you begin a migration, it is critical that you review the current Oracle Cloud Infrastructure Compute Classic architecture and fully understand the artifacts of the environment and how they relate to each other. It is also important to your specific application requirements and the context of the migration.

For example, compile a list of all applications that you want to migrate to the Oracle Cloud Infrastructure environment. Consider where each application currently resides (on-prem, Oracle Cloud Infrastructure Compute Classic, other cloud, etc.). Plan for you migration based on your specific applications and their network, database, VM, block storage, HA, and DR considerations.

General Considerations

The following table describes the general considerations migrating to Oracle Cloud Infrastructure.

Category	Considerations	Notes
General	• Customer technical expertise • Timing and downtime expectations • Business constraints	Setting expectations for a migration project is critical. Depending on the characteristics of the source environment, the downtime can be considerable, so it's important to make plans for the preparation, the actual migration, and for validation of the new environment before cutover.
Environment Information	• Development • Test • Production	Knowing the purpose of the environment can help in any required architecture redesign, and can help determine downtime requirements.
Financial Account Information	Current subscription type: • Non-metered • Metered (traditional) • Government/Public Sector • Universal Credits	Make sure you have migrated your Oracle Cloud Infrastructure Compute Classic account to the new Universal Credits subscription model. This migration must be performed before making the physical workload move. Contact your Sales Representative for more information.

Category	Considerations	Notes
Data Region Location	• Current data region/data center • Availability of Oracle Cloud Infrastructure data center	When migrating from Oracle Cloud Infrastructure Compute Classic to Oracle Cloud Infrastructure, the data center and data region are critical in determining how easy or difficult the migration process is. The data region can determine the connection types available between the Oracle Cloud Infrastructure Compute Classic shared network or IP network and the Oracle Cloud Infrastructure VCN.
Services used	• IaaS Only • IaaS and PaaS • Lift and shift applications – Apps Unlimited – Fusion Middleware	Identifying the main applications and services running within an environment helps to determine the most appropriate migration strategy for each workload.

Network Considerations

The following table describes the network considerations for migrating to Oracle Cloud Infrastructure.

Category	Considerations
General network requirements	• Shared network usage • Number of IP networks • Number of external IPs • Bandwidth requirements • Load Balancer information • Outbound proxy for external Internet access • Communication between data centers • DNS usage

Category	Considerations
Network security	• Security policies that exist in this environment • Security rules, security lists (Shared network) • ACLs, Virtual NIC Sets, and IP Prefix Sets (IP Network) • Additional security features needed beyond layer-3/4 filtering • Additional filtering needed (for example, layer-7) The configuration of security rules is especially important and can introduce a layer of complexity to a migration project. It's important to understand that there is not necessarily a 1-to-1 mapping of these features from the Oracle Cloud Infrastructure Compute Classic network to the VCN on Oracle Cloud Infrastructure.
Oracle Cloud Infrastructure Compute Classic to Oracle Cloud Infrastructure Network Connection	• Ashburn or London options • Other options to consider When migrating from Oracle Cloud Infrastructure Compute Classic to Oracle Cloud Infrastructure, the data center and data region are critical in determining how easy or difficult the migration will be. The data region can determine the connection types available between the Oracle Cloud Infrastructure Compute Classic shared network or IP network and the Oracle Cloud Infrastructure VCN. This connection is used primarily for moving the workloads, data, and other artifacts from your Oracle Cloud Infrastructure Compute Classic environment to your Oracle Cloud Infrastructure environment.
On-Premise to Oracle Cloud connection	• FastConnect Classic • VPN (Corente) • VPN as a Service (VPNaaS)

Database Considerations

The following table describes the database considerations for migrating to Oracle Cloud Infrastructure.

Category	Considerations
General	• Number of databases to migrate • Purpose of each database • Dependencies (what applications depend on each database) • Average size of each database
Oracle Databases	• Type of Database deployment: Oracle Database Classic Cloud Service or on-premises software installed on a VM? • Version and Edition of each database

Category	Considerations
Third-party Databases	- Brand, version and edition of each third-party database
Migration Method	- Are there any restrictions that would prevent the use of DataGuard as the primary tool for migrating the data? - What is the backup method and schedule for each database

Virtual Machine Considerations

The following table describes the virtual machine considerations for migrating to Oracle Cloud Infrastructure.

Question	Notes
How is access to the instance secured?	For example: SSH for Linux instances, WinRM or Remote Desktop for Windows instances.
Is there a bastion host?	For accessing this instance, a best practice is to configure a bastion (jump) host.
How is the system patched?	Are systems patched after initial provisioning?
Is there a way to audit the fleet of VM's for patches?	How to tell which VMs need additional patches? Especially CVE patches.
Is malware / anti-virus installed?	Which anti-virus vendor?
How are system level logs captured?	Syslogs for Unix. Event logs for Windows. Ideally connect to a log analytics system (Splunk, ELK, Graylog, ...)
Is the image hardened?	Review CIS (https://www.cisecurity.org/cis-benchmarks/) benchmarks for hardening systems.
What monitoring of the system is in place?	At a minimum CPU / memory / disk should be monitored. A better solution would be to alert based on these metrics.The best solution would be to provide a mechanism for auto-scaling.
Is there a firewall running on this instance?	Local firewall setting may affect remote access independent of any network security rule.
Does the system sync time using NTP?	Verify the NTP servers are accessible from Oracle Cloud Infrastructure, or consider using the Oracle Cloud Infrastructure NTP service.
How are the attached disks backed up?	Verify there is a plan for backup / restore.
Are fault domains being leveraged?	Verify that fault domains are being considered as compute instances are provisioned.

Block Storage Considerations

The following table describes the block storage considerations for Oracle Cloud Infrastructure Compute Classic to Oracle Cloud Infrastructure migration.

Question	Notes
Verify performance (IOPS, latency, throughput) is reasonable for your workload.	Use fio or Cloud Harmony benchmark to gather benchmark numbers. For details, see https://docs.cloud.oracle.com/iaas/Content/Block/Concepts/blockvolumeperformance.htm.
Verify block volume backup plan.	Ideally this should be automated or use policy based backups.
When using iSCSI, enable CHAP authentication.	For security, always enable CHAP authentication for iSCSI devices.

Custom Image Considerations

The following table describes the custom image considerations for Oracle Cloud Infrastructure Compute Classic to Oracle Cloud Infrastructure migration.

Question	Notes
Be aware of limitations (size, reserved IP addresses, Windows export...) custom images	Documented at: https://docs.cloud.oracle.com/iaas/Content/Compute/Tasks/managingcustomimages.htm
Since images can be shared across regions, upload images only as needed for startup time.	Trade off management of images versus startup time for a new instance.

Application-Level Disaster Recovery Considerations

The following list describes the application-level disaster recovery considerations for Oracle Cloud Infrastructure Compute Classic to Oracle Cloud Infrastructure migration.

- Is the application accessed via a DNS FQDN or by IP address directly?
- Will failover between prod and DR be accomplished by making DNS changes?
- Are there any other IP requirements between DR, prod and any other environments or are these largely undefined/nonexistent (such as using the same IP addressing for both prod and DR, etc.)?

3 Review the List of Available Migration Tools

The following tools are available to help migrate specific Oracle Cloud Infrastructure Compute Classic resources to Oracle Cloud Infrastructure.

Tools for Migrating Infrastructure Resources

You can use the following tools to identify resources in your Oracle Cloud Infrastructure Compute Classic environment and to migrate workloads to your Oracle Cloud Infrastructure tenancy.

These tools can help you to set up your network in Oracle Cloud Infrastructure and migrate your VMs and block storage volumes over to the target environment.

- Oracle Cloud Infrastructure Classic Discovery and Translation Tool
  This tool automates the discovery of all resources in your Oracle Cloud Infrastructure Compute Classic, Oracle Cloud Infrastructure Object Storage Classic, and Oracle Cloud Infrastructure Load Balancing Classic account. Among other capabilities, this tool can generate:

  - Reports that itemize the resources it finds in the specified source environment. You can use these reports to analyze your existing resources and identify or enumerate the resources to be migrated.

  - A list of the existing virtual machines in the source environment. You can use this list as input to Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool.

  - Terraform configuration files containing information about the network in the source environment. You can use this Terraform configuration to set up your network in the target environment.

  For information about installing and running the standalone version of this tool, see Identify and Translate Resources in Your Source Environment.

  For information about running this in an instance created using the Oracle Cloud Infrastructure Classic Migration Tools image, see Set Up the Migration Tools.

- Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool
  This tool automates the migration of virtual machines and block storage volumes from the source environment to the target environment. This tool is provided as an image in your Oracle Cloud Infrastructure Compute Classic account. You can create an instance using this image, to run this tool.

  For information about setting up an instance using the Oracle Cloud Infrastructure Classic Migration Tools image, see Set Up the Migration Tools.

  For information about using this tool, see Migrate Virtual Machines and Block Storage to Oracle Cloud Infrastructure.

- Oracle Cloud Infrastructure Classic Block Volume Backup and Restore Tool

This tool automates the migration of remote snapshots of storage volumes as well as scheduled backups. This tool is provided as an image in your Oracle Cloud Infrastructure Compute Classic account. You can create an instance using this image, to run this tool.

For information about setting up an instance using the Oracle Cloud Infrastructure Classic Migration Tools image, see Set Up the Migration Tools.

For information about using this tool, see Migrate Remote Snapshots and Scheduled Backups.

Tools for Migrating Databases

There are a variety of tools and processes available to migrate your databases to Oracle Cloud Infrastructure.

This guide describes how to migrate database instances using Oracle Cloud Infrastructure Classic Database Backup Migration Tool, which is available on instances created using the Oracle Cloud Infrastructure Classic Migration Tools image. This guide also describes how to migrate database instances using Oracle Data Guard.

Oracle Cloud Infrastructure Classic Database Backup Migration Tool migrates databases by using Oracle Recovery Manager (RMAN) to create a backup that can be restored to a new DB System in Oracle Cloud Infrastructure. For more information, see Migrate Databases Using the Migration Tools.

Alternatively, you can use Oracle Data Guard to migrate your database instances from Oracle Database Classic Cloud Service to a DB System in Oracle Cloud Infrastructure. When you use Oracle Data Guard to perform the migration, the source database is the primary database and the target database is the standby database.

- For information about migrating a single database instance, see Learn About Migrating a Database Cloud Service Deployment to a Virtual Machine Database System.
- For information about migrating a RAC database, see Learn About Migrating a Multi-Node Database Cloud Service Deployment to Virtual Machine Database System.

Tools for Migrating Object Storage

You can use the following tools to migrate your object storage resources from Oracle Cloud Infrastructure Object Storage Classic to Oracle Cloud Infrastructure Object Storage.

- If you use the Oracle Cloud Infrastructure Storage Software Appliance (also known as the Oracle Storage Cloud Software Appliance) to store data in your Oracle Cloud Infrastructure Object Storage Classic account, then you can migrate your data to your Oracle Cloud Infrastructure Object Storage account by using the Oracle Cloud Infrastructure Storage Gateway.
  For more information, see Migrate Storage Appliance Data to Oracle Cloud Infrastructure Storage Gateway.
- If you don't use the Oracle Cloud Infrastructure Storage Software Appliance, then you can use the `rclone` command to migrate your object storage data.
  For more information, see Migrate Object Storage Using Rclone.

Tools for Migrating an Oracle Java Cloud Service Instance

Based on your requirement, select one of the following tools to migrate your Oracle Java Cloud Service instances to Oracle Cloud Infrastructure.

Oracle recommends migrating your existing domains in Oracle Java Cloud Service to Oracle WebLogic Server for Oracle Cloud Infrastructure.

You cannot migrate your Oracle Java Cloud Service – Enterprise - Government and Oracle Java Cloud Service - High Performance - Government instances to Oracle WebLogic Server for Oracle Cloud Infrastructure. You can migrate these instances to Oracle Java Cloud Service on Oracle Cloud Infrastructure.

- Use Oracle Cloud Infrastructure Classic Java Migration Tool to migrate an Oracle Java Cloud Service instance from your Oracle Cloud Infrastructure Classic account to Oracle WebLogic Server for Oracle Cloud Infrastructure. See Migrate an Instance to Oracle WebLogic Server for Oracle Cloud Infrastructure in *Migrating Oracle Java Cloud Service Instances to Oracle Cloud Infrastructure Using Migration Tools*.
- Only if Oracle WebLogic Server is not available in your Oracle Cloud Infrastructure region, you can migrate Oracle Java Cloud Service applications to Oracle Java Cloud Service on Oracle Cloud Infrastructure. You can use Application Migration or Oracle Cloud Infrastructure Classic Java Migration Tool to perform this migration. Oracle recommends that you use Application Migration to migrate Oracle Java Cloud Service applications to Oracle Java Cloud Service on Oracle Cloud Infrastructure. See Migrate an Instance to Oracle Java Cloud Service Using Application Migration Service in *Migrating Oracle Java Cloud Service Instances to Oracle Cloud Infrastructure Using Migration Tools*.

  Application Migration does not support the migration of WebLogic Server domains that include these types of resources:

  - Custom Identity or Trust Keystore
  - Foreign JNDI Provider
  - Foreign JMS Server
  - JMS Bridge Destination
  - Storage-and-Forward (SAF) Context
  - JavaMail Session
  - WebLogic Diagnostic Framework (WLDF) REST Notification Endpoint

  If your source Oracle Java Cloud Service instance uses these resource types, then Oracle recommends using the Oracle Cloud Infrastructure Classic Java Migration Tool instead of Application Migration. See Migrate an Instance to Oracle Java Cloud Service Using Classic Tools in *Migrating Oracle Java Cloud Service Instances to Oracle Cloud Infrastructure Using Migration Tools*.

For more information about migrating an Oracle Java Cloud Service instance, see Learn About Migrating to Oracle Cloud Infrastructure in *Migrating Oracle Java Cloud Service Instances to Oracle Cloud Infrastructure Using Migration Tools*.

4 Set Up the Migration Tools

If you are planning to migrate your infrastructure resources from Oracle Cloud Infrastructure Compute Classic to Oracle Cloud Infrastructure, you can use Oracle-provided migration tools to make the process quicker and easier.

To access the Oracle-provided migration tools, create a migration controller instance in your Oracle Cloud Infrastructure Compute Classic account using the Oracle Cloud Infrastructure Classic Migration Tools image. Look for the most recent migration tools image in the list of Oracle-provided images. When your migration controller instance is ready, you can configure the instance and then start using the preinstalled tools to do the following:

- Get a list of resources in your source environment.
- Create networking objects in your target environment.
- Migrate your instances and block volumes.
- Migrate storage volume backups.
- Migrate a single instance database using RMAN.

About Required Services and Roles

To use the migration tools to migrate your resources, you'll need the following services and roles:

- Oracle Cloud Infrastructure Compute Classic: You'll need the `Compute_Operations` role to create the migration controller instance and to create snapshots of the boot and block volumes.
- Oracle Cloud Infrastructure: Ensure that you have policies in place that allow you to read the required OCIDs from the Web Console. You'll also need to create an API user, who must belong to a group that has policies in place to create the required resources.

Complete the Prerequisites

Before you create the migration controller instance, Control-S, complete the following prerequisites.

- Verify that you have sufficient quota to launch the instance. For the migration controller instance to be created in your Oracle Cloud Infrastructure Compute Classic account, you'll need the capacity to create one instance with a sufficient number of OCPUs for your migration.
- Generate an SSH key to connect to Control-S.
- Generate an API signing PEM key for Oracle Cloud Infrastructure.
- Ensure that you have an API user set up in your Oracle Cloud Infrastructure environment:
  1. Create an API user in Oracle Cloud Infrastructure.

2. Upload the API signing key for this user.
3. Ensure that this user is added to the required groups and has the required policies in place to create VMs and storage volumes in the required compartment.
4. Keep a copy of the API access PEM key. You'll need to make this key available on Control-S later.

- Verify that your API access to Oracle Cloud Infrastructure is set up correctly. Use the CLI to run a few commands to ensure that you can connect to the Oracle Cloud Infrastructure account and have the required permissions.

Launch the Migration Controller Instance (Control-S) in the Source Environment

In your Oracle Cloud Infrastructure Compute Classic account, create the source controller (Control-S) instance with the following configuration.

1. The Control-S instance must be created in the same identity domain and site as the resources that you want to migrate. You can use the web console or any other interface to create an instance with the following specifications:
   - **Image:** OL_7.5_UEKR4_x86_64_MIGRATION. This image is available under **Oracle Images** in the console.
   - **Shape:** General Purpose oc5 (4 OCPUs, 30 GB RAM) or any other shape with a sufficient number of OCPUs.
   - **SSH Key:** Associate an SSH public key with the Control-S instance. You'll use the corresponding private key to connect to the Control-S instance. Note that this key *isn't* the same as the SSH key pair used to access Linux source instances from Control-S when you migrate instances and block storage.
   - **Network:** Ensure that you select the *shared* network with a persistent public IP address.
     - Select the `default` security list that allows SSH inbound.
     - Ensure that security rules are in place to allow SSH outbound, SMB inbound, and HTTPS outbound traffic.
     - If you want to migrate instances that have interfaces on one or more IP networks only, then configure interfaces of the Control-S instance on the relevant IP networks as well, so that the Control-S instance can access the source instances that you want to migrate.
   - **Storage:** Use the default bootable storage volume.
2. The Control-S instance and associated storage volumes created for migration are by default billed at the applicable rates for your account. However, you can rename these resources so that the multipart name includes `/oraclemigration` as a container. Resources created in this `/oraclemigration` container aren't billed to your account.

   If you create the Control-S instance using the API, CLI, or a Terraform configuration, you can specify multipart resource names as `/Compute-example/user@example.com/`**`oraclemigration`**`/resource-name` when you create the resources.

If you create the Control-S instance and storage volumes using the web console, then after the instance is created, modify the orchestration to move the instance and storage volumes to the `/oraclemigration` container.

To move the Control-S instance and storage volumes into the `/oraclemigration` container:

a. Log in to the Oracle Cloud Infrastructure Compute Classic web console and go to the Instances page.

b. When the Control-S instance is created with status **Running,** click the **Orchestrations** tab.

c. To move the Control-S instance to the `/oraclemigration` container, you can suspend the orchestration. Go to the relevant orchestration and from the ☰ menu, select **Suspend**.

d. After the orchestration status changes to **Suspended,** from the ☰ menu, select **Update**.

e. On the update page, in the Instance section, click the ☰ menu and select **Edit JSON.**

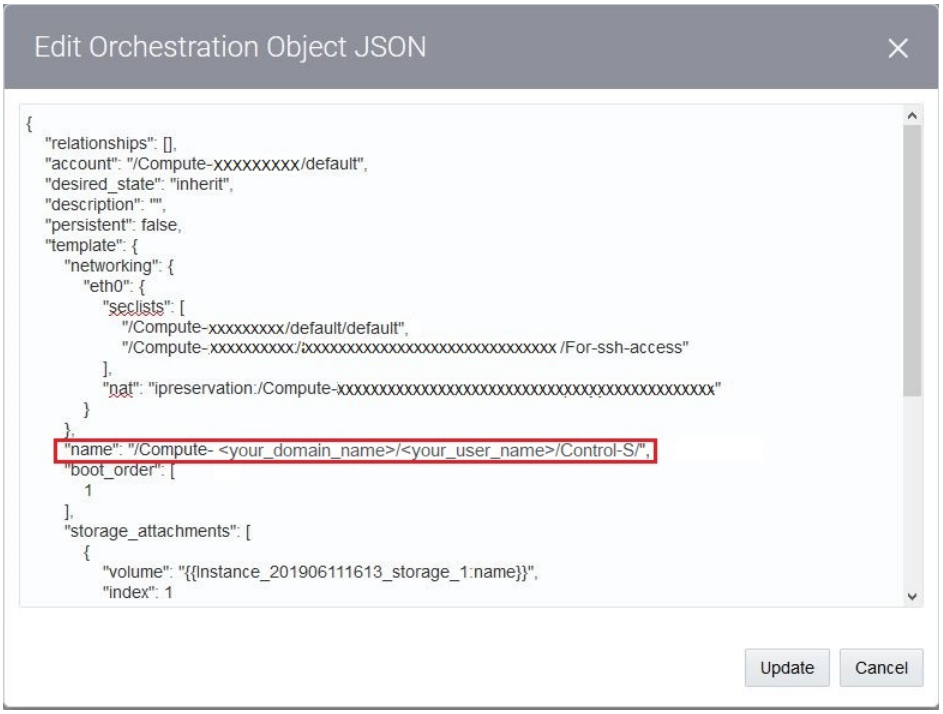
f. In the Edit Orchestration Object JSON window, look for the instance name. This is usually displayed within the `template` section, after `networking`.

```
 "name": "/Compute-example/user@example.com/instance-name",
```

Modify the instance name to include the `/oraclemigration` container:

```
 "name": "/Compute-example/user@example.com/oraclemigration/
instance-name",
```

Click **Update.**



```
{
   "relationships": [],
   "account": "/Compute-xxxxxxxxx/default",
   "desired_state": "inherit",
   "description": "",
   "persistent": false,
   "template": {
      "networking": {
         "eth0": {
            "seclists": [
               "/Compute-xxxxxxxxx/default/default",
               "/Compute-xxxxxxxxxx/xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx /For-ssh-access"
            ],
            "nat": "ipreservation:/Compute-xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
         }
      },
      "name": "/Compute- <your_domain_name>/<your_user_name>/Control-S/",
      "boot_order": [
         1
      ],
      "storage_attachments": [
         {
            "volume": "{{Instance_201906111613_storage_1:name}}",
            "index": 1
```

g. To move storage volumes to the `/oraclemigration` container, you must terminate the orchestration. This step destroys all persistent and nonpersistent objects created by the orchestration. Do this only if you haven't made any changes to your instance or storage volumes that you want to preserve. On the Orchestrations page, go to the relevant orchestration and from the ☰ menu, select **Terminate.**

h. After the orchestration status changes to **Stopped,** from the ☰ menu, select **Update.**

i. On the update page, in the Storage Volume section, go to the relevant storage volume, click the ☰ menu and select **Edit JSON.**

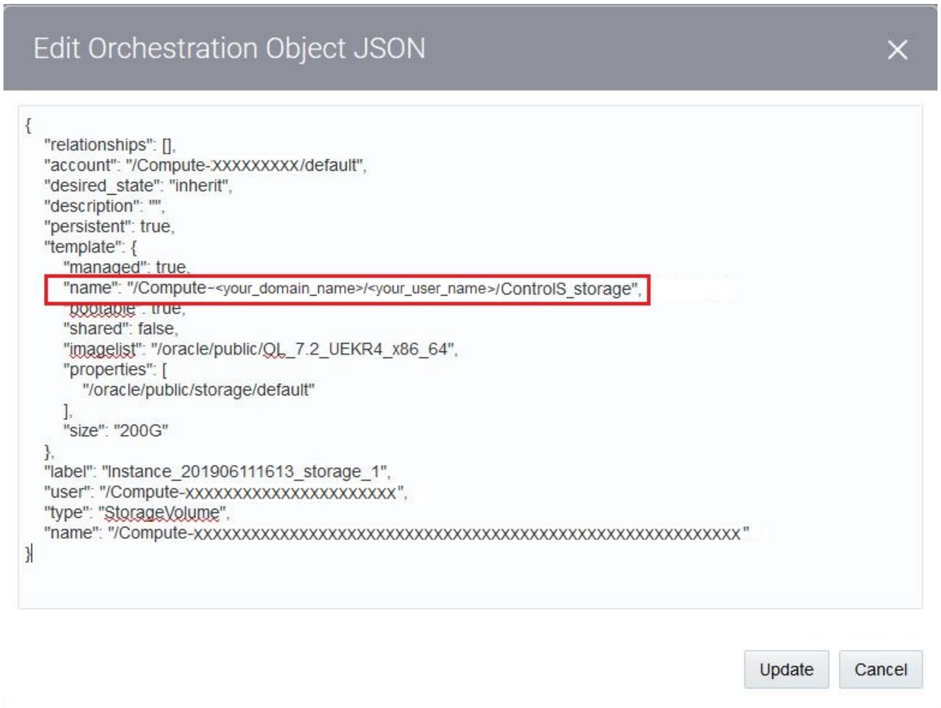
j. In the Edit Orchestration Object JSON window, look for the storage volume name in the `template` section:

```
 "name": "/Compute-example/user@example.com/storage-volume-name",
```

Modify the instance name to include the `/oraclemigration` container:

```
 "name": "/Compute-example/user@example.com/oraclemigration/storage-
volume-name",
```

Click **Update.**



```
{
   "relationships": [],
   "account": "/Compute-XXXXXXXXX/default",
   "desired_state": "inherit",
   "description": "",
   "persistent": true,
   "template": {
      "managed": true,
      "name": "/Compute-<your_domain_name>/<your_user_name>/ControlS_storage",
      "bootable": true,
      "shared": false,
      "imagelist": "/oracle/public/OL_7.2_UEKR4_x86_64",
      "properties": [
         "/oracle/public/storage/default"
      ],
      "size": "200G"
   },
   "label": "Instance_201906111613_storage_1",
   "user": "/Compute-xxxxxxxxxxxxxxxxxxxxxx",
   "type": "StorageVolume",
   "name": "/Compute-xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
}
```

k. Repeat these steps for any other storage volume in this orchestration that you want to move to the `/oraclemigration` container.

l. When you've updated all the relevant resources, start the orchestration. On the Orchestrations page, go to the relevant orchestration and from the ☰ menu, select **Start**.

> **Note:**
>
> When your instance and other resources are created in the `/oraclemigration` container, they are listed in the web console with this container name prefixed to the user-specified name. So if you had named your Control-S instance **Control-S** it will now appear with the name **oraclemigration/Control-S**.

When the instance status is displayed as Running, you can log in to the instance as the `opc` user from your local system, using your SSH private key.

Install Terraform

You can use Oracle Cloud Infrastructure Classic Discovery and Translation Tool to generate Terraform configurations for various resources. You can then apply the Terraform configuration to create the specified resources in your Oracle Cloud Infrastructure tenancy.

To use Terraform, download the appropriate package from the Terraform web site and install it on your Control-S instance.

> **Note:**
>
> Recent versions of the Oracle Cloud Infrastructure Classic Migration Tools image have Terraform preinstalled. However, if you created your Control-S instance prior to May 2019, Terraform might not be preinstalled on the instance.

1. Use SSH to log in to your Control-S instance as the opc user.
2. Use yum to install Terraform along with the required providers.

   ```
   sudo yum-config-manager --enable ol7_developer

   sudo yum install -y terraform.x86_64 terraform-provider-oci.x86_64
   terraform-provider-baremetal.x86_64
   ```

3. To verify your installation and check the version, run the following command:

   ```
   terraform -v
   ```

   The Terraform version is displayed in the output, as shown in this example:

   ```
   Terraform v0.11.10
   ```

Get Started with the Migration Tools

You can now start using the migration tools available on the Control-S instance. Each of the migration tools requires you to provide information about the source and target environment including user credentials, service details, access points, and so on. Provide the required information in the appropriate format.

For Oracle Cloud Infrastructure Compute Classic, this information is generally provided in the /home/opc/.opc/profiles/default file and for Oracle Cloud Infrastructure, this information is generally provided in the /home/opc/.oci/config file. See the relevant sections of this document for more information on setting up these files.

The following tools are preinstalled on this instance:

- Oracle Cloud Infrastructure Classic Discovery and Translation Tool: You can access this tool using the opcmigrate commands. This tool helps you to discover resources in your source environment and generate Terraform configuration files that you can use to create the corresponding resources in your target environment. See Identify and Translate Resources in Your Source Environment.
- Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool: You can access this tool by using the opcmigrate migrate instance commands. This tool helps you to migrate instances and block volumes. See Migrate Virtual Machines and Block Storage to Oracle Cloud Infrastructure.
- Oracle Cloud Infrastructure Classic Block Volume Backup and Restore Tool: You can access this tool using the opcmigrate migrate rsm commands. This tool

helps you to migrate storage backups created using remote snapshots or scheduled backups. See Migrate Remote Snapshots and Scheduled Backups.

- Oracle Cloud Infrastructure Classic Database Backup Migration Tool: You can access this tool using the `opcmigrate migrate database` commands. This tool helps you to migrate single instance deployments of Oracle Database Classic Cloud Service using RMAN. See Migrate Databases Using the Migration Tools.
- Oracle Cloud Infrastructure Classic Java Migration Tool: You can access this tool using the opcmigrate migrate jcs commands. This tool helps you to migrate an Oracle Java Cloud Service instance. See Learn About Migrating to Oracle Cloud Infrastructure in *Migrating Oracle Java Cloud Service Instances to Oracle Cloud Infrastructure Using Migration Tools.*

5
Identify and Translate Resources in Your Source Environment

If you want to migrate resources from your Oracle Cloud Infrastructure Compute Classic account to your Oracle Cloud Infrastructure tenancy, then you'll need a complete list of your existing resources. Oracle Cloud Infrastructure Classic Discovery and Translation Tool allows you to generate reports of your object storage account as well as the networking objects, virtual machine instances, and load balancers in your Oracle Cloud Infrastructure Compute Classic and Oracle Cloud Infrastructure Load Balancing Classic account.
You can use the output generated by this tool to analyze the networking objects that you'll need to set up in your Oracle Cloud Infrastructure tenancy and to identify the virtual machine instances and block storage volumes that you want to migrate. You can select an output format that works best for your requirements and you can also filter the output using various commands and options provided by this tool.

You can generate reports in the following formats:

- **JSON:** The default format. If you want to use Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool to migrate your resources, you can generate a list of instances that can be used as input for that tool.
- **Graph:** You can use the graphical output to get a visual representation of your resources. You can also filter the graphical output to exclude or include specified object types or to focus on specific objects.
- **Spreadsheet:** Use this format to view a list of different resource types in separate worksheets. The spreadsheet also includes a summary of your environment and a count of each resource type.
- **Terraform:** Use this format if you want to use Terraform to set up your resources on Oracle Cloud Infrastructure. Note that you must review the generated Terraform module carefully before using it to create resources on Oracle Cloud Infrastructure. You'll also need to provide input in a variables file to enable access to the Oracle Cloud Infrastructure tenancy.

> **Note:**
>
> The reports generated by the tool can contain sensitive data about your cloud environment, such as personally identifiable information, system identifiers, security configurations, system initialization scripts and default passwords. By default, the reports generated by this tool are stored in the local directory where the tool is run. Make sure that access to these files is restricted and that you delete the files as soon as possible after your migration is complete.

After you've generated reports for each of your resources, evaluate the options to create the network in the target environment before you migrate your virtual machine instances and block storage volumes.

After you've set up the network in your target environment, you can use the output provided by Oracle Cloud Infrastructure Classic Discovery and Translation Tool as input to Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool to specify the resources that you want to migrate. This simplifies the process of moving your compute and block storage resources to your Oracle Cloud Infrastructure tenancy.

You can also use the remote storage migration commands and the database migration commands to migrate those resources to Oracle Cloud Infrastructure.

Considerations for Using Oracle Cloud Infrastructure Classic Discovery and Translation Tool

Before you run Oracle Cloud Infrastructure Classic Discovery and Translation Tool, consider the following suggestions.

- The standalone distribution of this tool doesn't modify any resources in the source environment. It is recommended that you run this tool as a user with the minimum required read-only access. The recommended user privileges are:
  - Compute Classic: `Compute.Compute_Monitor`
  - Load Balancer Classic: `LBAAS_READONLYGROUP`
  - Storage Classic: `Storage_ReadOnlyGroup`
- If you run this tool in an instance created with the Oracle Cloud Infrastructure Classic Migration Tools image, then you can use the `opcmigrate migrate` set of commands to migrate VMs, block storage, storage snapshots, and database instances. Some of those commands require read-write access to your Oracle Cloud Infrastructure Compute Classic account as well as your Oracle Cloud Infrastructure tenancy. Ensure that sufficient privileges are in place in both environments.
- This tool generates Terraform configuration files that can be used to create resources in a single availability domain on Oracle Cloud Infrastructure.

Prepare to Use Oracle Cloud Infrastructure Classic Discovery and Translation Tool

If you want to use Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool along with Oracle Cloud Infrastructure Classic Discovery and Translation Tool, create a migration controller instance in your Oracle Cloud Infrastructure Compute Classic account using the Oracle Cloud Infrastructure Classic Migration Tools image.

For information about creating a migration controller instance, see Complete the Prerequisites and Launch the Migration Controller Instance (Control-S) in the Source Environment. If you've already created this instance, you can use the same instance for this procedure. You don't need to create it again.

Oracle Cloud Infrastructure Classic Discovery and Translation Tool is preinstalled on this instance, so you can start using this tool right away on all instances created with this image.

If you want to run Oracle Cloud Infrastructure Classic Discovery and Translation Tool on any other system, follow the procedure to install the tool on any system running Oracle Linux 7.x, Windows, or MacOS. Note, however, that not all features of this tool are available when you download and install the tool on other systems.

Before you run the tool, ensure that you've set up your profile and provided the credentials required to allow the tool to connect to your Oracle Cloud Infrastructure Compute Classic account. If you're using Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool on an instance created using the Oracle Cloud Infrastructure Classic Migration Tools image, skip the installation procedures and set up your profile now. See Set Up Your Profile.

Some features of the tool might require credentials to access yourOracle Cloud Infrastructure tenancy as well. Follow the steps in specific sections of this document to provide the required information for each feature.

Install Oracle Cloud Infrastructure Classic Discovery and Translation Tool

To install Oracle Cloud Infrastructure Classic Discovery and Translation Tool you'll need Python 3.6.6 or higher. To view graphs generated by the tool and to generate PDFs of graphs, you'll need Graphviz 2.30.1 or higher.

The steps to install the tool vary slightly depending on the OS of the system that you want to install it on. Currently, you can install this tool on the following operating systems:

- Oracle Linux 7.x
- MacOS
- Microsoft Windows

Install Oracle Cloud Infrastructure Classic Discovery and Translation Tool on Oracle Linux

On an Oracle Linux system, use pip to install Oracle Cloud Infrastructure Classic Discovery and Translation Tool. You can use yum to install pip along with the required Python and Graphviz packages.

1. To install pip, Python, and Graphviz, run the following commands:

   ```
   sudo yum install yum-utils

   sudo yum-config-manager --enable ol7_developer_EPEL

   sudo yum install python36 python36-setuptools graphviz

   sudo easy_install-3.6 pip
   ```

2. Download the latest version of Oracle Cloud Infrastructure Classic Discovery and Translation Tool from Oracle Technology Network.

3. Navigate to the directory where the tool is saved. Use pip to install the tool.

   ```
   pip install ./opcmigrate-<version>-py3-none-any.whl
   ```

   If you're prompted to provide the user for this command, specify the opc user, or use sudo to install the packages as root.

   > **Note:**
   >
   > If you also have Python 2.x installed on this system, you might need to use pip3 instead of pip to install the tool.

4. To verify that the tool was successfully installed, run the command with the --help option. The tool returns a list of commands and general options.

   ```
   opcmigrate --help
   ```

5. To verify the version of the tool, run the command with the --version option.

   ```
   opcmigrate --version
   ```

Install Oracle Cloud Infrastructure Classic Discovery and Translation Tool on MacOS

On MacOS, use pip to install Oracle Cloud Infrastructure Classic Discovery and Translation Tool. You can use brew to install pip along with the required Python and Graphviz packages.

1. To install pip, Python, and Graphviz, run the following command:

   ```
   brew install python3 graphviz pango gts librsvg imagemagick
   ```

2. Download the latest version of Oracle Cloud Infrastructure Classic Discovery and Translation Tool from Oracle Technology Network.
3. Navigate to the directory where the tool is saved. Use pip to install the tool.

   ```
   pip install ./opcmigrate-<version>-py3-none-any.whl
   ```

   If you're prompted to provide the user for this command, specify the opc user.

   > **Note:**
   >
   > If you also have Python 2.x installed on this system, you might need to use pip3 instead of pip to install the tool.

4. To verify that the tool was successfully installed, run the command with the `--help` option. The tool returns a list of commands and general options.

   ```
   opcmigrate --help
   ```

5. To verify the version of the tool, run the command with the `--version` option.

   ```
   opcmigrate --version
   ```

Install Oracle Cloud Infrastructure Classic Discovery and Translation Tool on Windows

On Windows, download and install Graphviz and Python. Then use `pip` to install Oracle Cloud Infrastructure Classic Discovery and Translation Tool.

1. Download and install Graphviz from https://graphviz.gitlab.io/download/.
2. Update the PATH environment variable to include the `/bin` folder in the directory where you installed Graphviz. By default, on a 64-bit Windows system, Graphviz is installed in: `C:\Program Files (x86)\Graphviz<version>`
3. Download Python 3.6.6 from https://www.python.org/downloads/. If you are running a 64-bit version of Windows, then ensure that you download the AMD64 version of Python.
4. Install Python. During the installation, ensure that you select the option to add Python to the PATH environment variable.
5. In a Windows command prompt, enter the following command to upgrade the pip package:

   ```
   python -m pip install --upgrade pip
   ```

6. Download the latest version of Oracle Cloud Infrastructure Classic Discovery and Translation Tool from Oracle Technology Network.
7. Navigate to the directory where the tool is saved. Use `pip` to install the tool.

   ```
   pip install ./opcmigrate-<version>-py3-none-any.whl
   ```

   > **Note:**
   >
   > If you also have Python 2.x installed on this system, you might need to use `pip3` instead of `pip` to install the tool.

8. To verify that the tool was successfully installed, run the command with the `--help` option. The tool returns a list of commands and general options.

   ```
   opcmigrate --help
   ```

9. To verify the version of the tool, run the command with the `--version` option.

   ```
   opcmigrate --version
   ```

Set Up Your Profile

Oracle Cloud Infrastructure Classic Discovery and Translation Tool connects to your source environment using connection information that you provide in a profile file.

The information you provide in the profile file includes the user name or identity for each service in the source environment, as well as the service end point and region. If you want to run the tool in multiple regions or tenancies, you can create separate profile files for each region and tenancy.

1. You'll need the user name and API end point for each service. Look up service-specific details in your Oracle Cloud Dashboard.
2. Create the directory for the profile file, if the directory doesn't already exist. By default, profile files are created in the directory `~/.opc/profiles`. If you create profiles in a location other than `~/.opc/profiles`, when you run the tool, provide the full path to the profile location by using the `--profile-directory` option.
3. Use the following template to create your profile file. Save this profile with the file name `default` in the path `~/.opc/profiles`. Replace the sample values with values specific to each service.

   Note:

   If you use Oracle Cloud Infrastructure Classic Discovery and Translation Tool on an instance created with the Oracle Cloud Infrastructure Classic Migration Tools image:

   - The `profiles` directory and a default profile might already exist. However, the default profile might contain only the `"compute"` section.
   - If you run the Control-S setup command for Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool, it generates the default profile using information provided by you in the `secret.yml` file. The Control-S setup command overwrites any existing default profile.

   Modify the default profile or create a new profile, as required.

```
{
  "global": {
    "format": "text",
    "debug-request": false
  },
  "compute": {
    "user": "/Compute-example/user@example.com",
    "endpoint": "compute.uscom-central-1.oraclecloud.com"
  },
  "lbaas": {
    "user": "user@example.com",
    "endpoint":
"lbaas-00000000000000000000000000000000.balancer.oraclecloud.com",
    "region": "uscom-central-1"
  },
  "paas": {
    "user": "user@example.com",
    "identity_id": "idcs-00000000000000000000000000000000",
    "endpoint": "psm.us.oraclecloud.com",
    "region": "uscom-central-1"
  },
  "object_storage": {
    "auth-endpoint": "uscom-central-1.storage.oraclecloud.com/auth/
v1.0",
    "user": "Storage-example:user@example.com",
    "endpoint": "uscom-central-1.storage.oraclecloud.com/v1/Storage-
example"
  }
}
```

Passwords aren't specified in the profile file for security reasons. You'll be prompted to provide the password for each service when you run the tool.

4. If you create multiple profiles in the `~/.opc/profiles` directory, use the `--profile` option to specify the profile you want to use when you run the tool. If no profile is specified, the `default` profile is used.

Upgrade Oracle Cloud Infrastructure Classic Discovery and Translation Tool

If you've already installed an earlier version of Oracle Cloud Infrastructure Classic Discovery and Translation Tool on your system, then you can use `pip` to uninstall the old version and install the latest version.

Note:

You can upgrade only a standalone installation of Oracle Cloud Infrastructure Classic Discovery and Translation Tool. If you are using Oracle Cloud Infrastructure Classic Discovery and Translation Tool on an instance created with the Oracle Cloud Infrastructure Classic Migration Tools image, then to upgrade to the latest version of the installed tools, use the latest image to create an instance.

On a local system with a standalone installation of Oracle Cloud Infrastructure Classic Discovery and Translation Tool, to upgrade, do the following:

1. Uninstall the currently installed version of the tool. Navigate to the directory where the tool is saved and enter the following command:

   ```
   pip uninstall ./opcmigrate-<old-version>-py3-none-any.whl
   ```

2. Download the latest version of Oracle Cloud Infrastructure Classic Discovery and Translation Tool.

3. Navigate to the directory where the latest version of the tool is saved. Use `pip` to install the tool as before.

   ```
   pip install ./opcmigrate-<latest-version>-py3-none-any.whl
   ```

   If you're prompted to provide the user for this command, specify the `opc` user.

4. To verify the version of the tool, run the command with the `--version` option.

   ```
   opcmigrate --version
   ```

Run Oracle Cloud Infrastructure Classic Discovery and Translation Tool to Generate Reports

You can use various commands and options to specify the output format of the report and to filter the output according to your requirements.

Learn About Commonly Used Commands and Options

Here are some of the commonly used commands and options that allow you to customize your reports as required. To view a complete list of commands and for detailed information about the options and permitted values for each command, run the tool with the `--help` option.

To view help on all commands and options, use:

```
opcmigrate --full-help
```

- To generate JSON formatted resource files and reports, use:
  - `opcmigrate discover:` Generates a report of all available network, instance, load balancer, PaaS, and object storage resources in the site, in JSON format. The data in this output processed by other `opcmigrate` commands.
  - `opcmigrate network:` Generates a report of the networking objects. The report contains the security lists in the shared network and a list of all the IP networks along with the associated access control lists, security rules, and instances in each IP network.
  - `opcmigrate instances-export:` Generates a file in JSON or YAML format that lists all the instances along with information about the operating system of each instance. This file can be passed as input to Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool.
  - `opcmigrate anonymize:` Removes sensitive data from a specified file.
  - `opcmigrate summary:` Displays on the terminal a summary of the resources in the site.
- To generate a spreadsheet, use `opcmigrate report`. This command generates a spreadsheet with separate worksheets for each resource type, listing all the resources in the site.
- To generate a graph of relationships between resources, use `opcmigrate graph`. This command generates a Graphviz graph and a PDF of the relationships

between the discovered resources. You can specify a number of options to customize the graphs generated by this command.

- To focus on specified resources or resource types, include and exclude resources, or filter resources, use `opcmigrate plan create`. This command generates a plan for the specified resources. In addition to allowing you to filter resources in the output, this command allows you to set certain resource-level attributes for individual objects. For example, for any given object, you can specify if you want that object to be migrated or not. You can then provide the output of this command as input to the `opcmigrate generate` command to generate a Terraform configuration for the specified resources.
- To generate a Terraform configuration file, use `opcmigrate generate`. This command generates a Terraform configuration file that can be used to create resources on Oracle Cloud Infrastructure.
- To migrate boot and block volumes, instances, remote snapshots, scheduled backups, and single instance deployments of Oracle Database Classic Cloud Service using RMAN, use the `opcmigrate migrate` set of commands. These commands aren't explained in detail in this section. For information on using this tool to migrate these resources, see:
  - Migrate Virtual Machines and Block Storage to Oracle Cloud Infrastructure
  - Migrate Databases Using the Migration Tools
  - Migrate Remote Snapshots and Scheduled Backups

Generate a Summary and JSON Output

Use the following commands and options to generate a comprehensive JSON formatted resource cache of your resources or to view a brief summary of your resources.

- To generate a JSON formatted file that contains information about all the networking objects, instances, storage volumes, and other resources in the site:

  ```
  opcmigrate discover
  ```

  This command writes the output to a file with the name `resources-*.json` where * indicates the profile name. This file is stored in the same directory where the command is run. You can use this file as input to other commands, to filter and sort the data.
- To specify a profile other than `default`:

  ```
  opcmigrate --profile <profile> discover
  ```

  The profile name is included in the file name of reports generated by the `opcmigrate discover` command.
- By default, a summary view of object storage containers is listed in the report. To fetch the full file names:

  ```
  opcmigrate discover --with-storage-objects
  ```

- To view a list summarizing the resource types and the number of resources in your source environment:

  ```
  opcmigrate summary
  ```

  This command takes as input a report generated by the `opcmigrate discover` command. You must run that command first, before running `opcmigrate summary`. With this command, output is displayed on the terminal and no output file is generated. Use standard commands to write the output to a file, if required.
- To generate a file with the network details:

  ```
  opcmigrate network
  ```

  This command takes as input a report generated by the `opcmigrate discover` command. You must run that command first, before running `opcmigrate network`. This report shows the security rules and instances in each security list in the shared network. It also lists the access control lists, security rules, and instances in each IP network.

  You can filter the output by resource type, to view only the shared network or only IP networks. You can also sort the output to display data by secrule in the shared network, or by vNICset in IP networks, and so on. For more information about command options, run the command with the `--help` option.
- The reports generated by the `opcmigrate discover` command contain sensitive data about your source environment. If you want to remove sensitive data from these files, including user ids, email addresses, instance initialization scripts, and unique service ids, use the `anonymize` command.

  ```
  opcmigrate anonymize --file resources-default.json --output resources-
  anonymized.json
  ```

Generate a Spreadsheet

To generate an Excel spreadsheet with separate worksheets for each resource type, use the following command.

- `opcmigrate report`

Generate Graphical Reports

To generate a graph of the relationships between resources, use the following command:

- `opcmigrate graph`

  This command generates a Graphviz graph as well as a PDF file.
  By default, the graph excludes certain object types such as orchestrations, machine images, and image lists, as well as common or default resources and PaaS resources. Use the `--with-orchestrations`, `--no-filter`, and `--with-paas` options to include these resources in the graph.

  This command also provides options to do the following:

- Focus on a specified resource using `--focus`.
- Exclude specified resources or resource types using `--exclude`.
- Show only specified resources or resource types using `--include`.
- Use a specified graph layout engine using `--engine`.

For more information about command options, run the command with the `--help` option.

Create a Migration Plan

You can use the resources file generated by `opcmigrate discover` to create a migration plan file. A migration plan file allows you to apply various filters to include or exclude objects. You can also set certain object-level migration attributes in this file.

A migration plan file is created by using a resources file as input. You must have already run `opcmigrate discover` to generate a resources file, before you create a migration plan.

To create a migration plan, run `opcmigrate plan create`. You can specify a number of options while creating a migration plan, to include or exclude specific resources or resource types. Use the `--help` option for information about the available options.

You can specify a resources file by using the `--file` option. If no resources file is specified, the command looks for the `resources-default.json` file in the current directory.

By default, the plan file is named `plan-default.json.` If you run the `opcmigrate plan create` command multiple times, the output file is overwritten each time by default. Note, however, that in earlier versions of the tool, the output of this command wasn't written to a file but was displayed on the screen by default. To save the migration plan to a file with a different name, use the `--output` option.

```
opcmigrate plan create --file resources-default.json --output migration-
plan.json
```

If you want to print the output to standard output (stdout), use the option `-o -`.

```
opcmigrate plan create --file resources-default.json -o -
```

For each object listed in the migration plan file, you can specify if you want to migrate that object by modifying the value of the `opc_migrate_include` attribute. For customer-created resources, this attribute is set to true by default. For Oracle-defined resources, this attribute is generally set to false by default.

For some objects, such as databases and storage volumes, you can also specify other object-level migration attributes.

If you want to use this plan as input to generate a list of instances, you can scan through the list of instances and make the required modifications, if any, to include or exclude instances from being migrated.

You can use the `--include, --exclude,` and `--no-filter` options to alter the scope of the generated plan. For example, to generate a plan with specified resource types, use:

```
opcmigrate plan create --include instance ip_network vnic interface
security_rule vnic_set acl security_protocol ip_address_prefix_set --
output plan.json
```

Generate a List of Instances to Migrate

You can use Oracle Cloud Infrastructure Classic Discovery and Translation Tool to generate a list of instances and storage volumes that you want to migrate. You can then use this list as input to Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool.

Use the `opcmigrate instances-export` command to generate a list of instances in your Oracle Cloud Infrastructure Compute Classic account. This command requires as input a plan generated by the `opcmigrate plan create` command. You can also specify a resources file generated by the `opcmigrate discover` command. If no resources file is specified, the command looks for the `resources-default.json` file in the current directory.

To generate a list of instances, do the following:

1. Run `opcmigrate discover` to generate a resource file. By default the resource file is named `resources-default.json` and it is saved in the current directory.

   ```
   opcmigrate discover
   ```

2. Run `opcmigrate plan create` to create a migration plan. You can specify a number of options while creating a migration plan, to include or exclude specific resources or resource types. Use the `--help` option for information about the available options.

   ```
   opcmigrate plan create --file resources-default.json --output migration-
   plan.json
   ```

3. Modify the migration plan file, if required. For each object listed in the migration plan file, you can specify whether you want to migrate that object or not. Scan through the list of instances and make the required modifications, if any, to include or exclude instances from being migrated.
4. Run `opcmigrate instances-export` to generate the list of instances to be migrated. By default, this command generates the output in JSON format, which can be used to create job files for migration. Use the `--format` option if you want to generate the output in YAML.
   By default, the output of this command is displayed on the terminal. Use standard commands to write the output to a file, if required.

   ```
   opcmigrate instances-export --file resources-default.json --plan
   migration-plan.json > instances.yaml
   ```

Generate Terraform Configuration Files

If you want to use Terraform to set up resources in your Oracle Cloud Infrastructure tenancy, you can use `opcmigrate generate` to generate Terraform configuration files.

The `opcmigrate generate` command requires a migration plan file and a resources file as input.
You can specify a plan file by using the `--plan` option. If no plan file is specified, the command uses the `plan-default.json` file in the current directory. Note that in earlier versions of the tool, `--plan` was a required option and the `opcmigrate generate` command didn't use the `plan-default.json` file by default.

You can optionally also specify a resources file by using the `--file` option. If no resources file is specified, the command looks for the `resources-default.json` file in the current directory.

By default, the Terraform configuration file is named `generate-default.tf.` If you run the `opcmigrate generate` command multiple times, the output file is overwritten each time by default. Note, however, that in earlier versions of the tool, the output of this command wasn't written to a file but was displayed on the screen by default. To save the configuration to a file with a different name, use the `--output` option.

To generate a Terraform configuration file, do the following:

1. Run `opcmigrate discover` to generate a resources file. By default the resources file is named `resources-default.json` and it is saved in the current directory.

   ```
   opcmigrate discover
   ```

2. Run `opcmigrate plan create` to create a migration plan. You can specify a number of options while creating a migration plan, to include or exclude specific resources or resource types. Use the `--help` option for information about the available options.

   ```
   opcmigrate plan create --file resources-default.json --output migration-
   plan.json
   ```

   If you want to print the output to standard output (stdout), use the option `-o -`.

   ```
   opcmigrate plan create --file resources-default.json -o -
   ```

3. Modify the migration plan file, if required. For each object listed in the migration plan file, you can specify whether you want to migrate that object or not. For some objects, such as databases and storage volumes, you can also specify other object-level migration attributes.
4. Run `opcmigrate generate` to generate the Terraform configuration file.

   ```
   opcmigrate generate --plan migration-plan.json --output main.tf
   ```

   The generated Terraform configuration file includes:

   - The `oci` provider definition.

- Variable declarations for the required input variables.
- A data source for the for the availability domains. AD1 is specified by default.
- One or more `oci_core_vcn` resources and `oci_core_subnet` resources, based on the Oracle Cloud Infrastructure Compute Classic shared network and IP networks.
- Security lists and security rules or network security groups (NSGs) and NSG security rules, based on the IP networks being migrated.
- A `oci_core_route_table` route table for each subnet.
- Instance definitions for each of the instances in the resources file that is identified for migration in the migration plan. Instances are associated with the appropriate resources, including storage volumes, subnets, and IP addresses.

5. By default, security rules created in Oracle Cloud Infrastructure Compute Classic are mapped to NSG security rules in Oracle Cloud Infrastructure.

```
opcmigrate generate --plan migration-plan.json --output main.tf
```

If the private IP addresses of IP networks connected to an IP network exchange can't be represented in a single /16 CIDR prefix, then multiple VCNs must be created. These VCNs are connected using VCN peering. However, NSG security rules can only make reference to NSGs within the same VCN. NSG security rules can't reference NSGs across peered VCNs. In this case, no NSGs and NSG security rules are generated by default in the Terraform configuration.

In such cases, to generate a set of security lists and security rules to be used in the Oracle Cloud Infrastructure network, use the `--with-security-rule-union` option.

```
opcmigrate generate --with-security-rule-union --plan migration-
plan.json --output main.tf
```

With the `--with-security-rule-union` option, the generated Terraform configuration creates VCNs and subnets in your Oracle Cloud Infrastructure environment along with the required security lists and security rules. The security lists created by this Terraform configuration contains a union of all the security rules related to the IP networks being migrated.

> **Caution:**
>
> When you use the `--with-security-rule-union` option, review the generated Terraform configuration carefully before applying it. Due to differences in the way security rules are implemented in Oracle Cloud Infrastructure Compute Classic and Oracle Cloud Infrastructure, the security rules implemented by the Terraform configuration in the target environment might expose instances to more traffic than intended.

Troubleshooting

Here are a few tips for dealing with errors that might occur while installing and using Oracle Cloud Infrastructure Classic Discovery and Translation Tool.

- **Error authenticating with your Oracle Cloud Infrastructure Compute Classic, Oracle Cloud Infrastructure Object Storage Classic, Oracle Cloud Infrastructure Load Balancing Classic, or PaaS account.**
  When you run the `opcmigrate discover` command, the tool attempts to connect to your accounts using the user names supplied in the profile file. If authentication fails, ensure that the correct user name is supplied for each account. Also check that the correct REST API endpoints are provided. The endpoint should be just the domain name without the `https://` prefix. The endpoint URLs can be found on the Service Details page for each service.
- **Warnings and errors while running the `opcmigrate graph` command.**
  When you use the opcmigrate graph command, you might see the following errors:

  ```
  - Warning: Overlap value "prism" unsupported - ignored-
  Error: remove_overlap: Graphviz not built with triangulation library
  ```

  These errors indicate that the default Graphviz distribution is missing some required dependencies. You might need to reinstall Graphviz from an alternative distribution, or build from source with the `gts` and `pango` options enabled. For example, on an Oracle Linux system, to build GTS dependency from source run the following commands:

  ```
  cd/usr/local/src
  wget http://gts.sourceforge.net/tarballs/gts-snapshot-121130.tar.gz
  tar zxvf gts-snapshot-121130.tar.gz
  cd gts-snapshot-121130
  sudo yum install gcc glib-devel
  ./configure
  make
  sudo make install
  ```

  To build Graphviz from source, run the following commands:

  ```
  cd /usr/local/src
  wget https://graphviz.gitlab.io/pub/graphviz/stable/SOURCES/
  graphviz.tar.gz
  tar xzvf graphviz.tar.gz
  cd graphviz-2.40.1
  sudo yum install gcc-c++ libstdc++-devel gd-devel pango-devel
  ./configure
  make
  sudo make install
  ```

- **The `opcmigrate graph` command takes a long time to run or doesn't complete**Large graphs may take a long time to compete using the default graph layout engine. To reduce the time taken to create graphs, you can:

  - Use the `--engine` option to specify an alternative graph layout engine. The `sfdp` engine is recommended for large graphs.
  - Reduce the size of the graph by limiting the types of nodes displayed. Use the `--focus, --exclude`, or `--include` options.

- **The `opcmigrate instances-export` command doesn't return the `os` and `osSku` values.**
  When you run the `opcmigrate instances-export` command, you might see the following errors:

  ```
  PandaExportGenerator: WARNING: machine image details not found for
  instance://Compute-a000000/user@example.com
  ```

  This error indicates that the machine image used to launch an instance is no longer present in the environment. The machine image and image list have been deleted after the instance was launched. You can ignore this warning. However, if you want to provide the output of the `opcmigrate instances-export` command as input to Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool, it's recommended that you specify the `os` and `osSku` value for all instances, if possible.

- **The Terraform output generated by the `opcmigrate generate` command doesn't include any security rules.**
  By default, the `opcmigrate generate` command doesn't include output that maps the Oracle Cloud Infrastructure Compute Classic security rules to Oracle Cloud Infrastructure security rules. To generate security rules, use the `--with-security-rule-union` option.

  > **Caution:**
  >
  > When you use the `--with-security-rule-union` option, review the generated Terraform configuration carefully before applying it. Due to differences in the way security rules are implemented in Oracle Cloud Infrastructure Compute Classic and Oracle Cloud Infrastructure, the security rules implemented by the Terraform configuration in the target environment might expose instances to more traffic than intended.

6 Migrate Users and Groups

When migrating customers from Oracle Cloud Infrastructure Compute Classic to Oracle Cloud Infrastructure, one of the key tasks is to ensure that the users have equivalent access privileges to resources in Oracle Cloud Infrastructure, as they had in Oracle Cloud Infrastructure Compute Classic.

Background

This section provides some basic concepts you should understand before you start the migration of users and groups to Oracle Cloud Infrastructure.

Traditional Cloud Account versus Cloud Account with Identity Cloud Service in Oracle Cloud Infrastructure Compute Classic

Oracle Cloud Infrastructure Compute Classic users get access to one of the following two identity management systems or identity providers (IDPs):

- Cloud Account with Identity Cloud Service: This is the new identity management system to manage users and roles for Universal Credits subscription users.
- Traditional Cloud Account: This uses Shared Idenity Management (SIM) to manage the users and roles in the account.

For the purposes of this document, it is assumed that the Oracle Cloud Infrastructure Compute Classic user is already using a Cloud Account with Identity Cloud Service. Procedures and processes are available to migrate existing users to the newer subscription model and identity management system .

Oracle Cloud Infrastructure and User Account Federation

Oracle Cloud Infrastructure has its own native Identity and Access Management (IAM) system to manage users, groups, and policies. However, it also provides a feature that allows you to federate users with an external identity provider (IDP). By default, Oracle Identity Cloud Service is set up as a federated IDP for all Oracle Cloud Infrastructure tenancies.

This means that you can continue to sign in and manage the Oracle Cloud Infrastructure resources with the users and roles created in Oracle Cloud Infrastructure Compute Classic that uses Oracle Identity Cloud Service. You simply assign the Oracle Cloud Infrastructure Compute Classic users to specific groups with specific policies in the Oracle Cloud Infrastructure IAM. Alternatively, if you want to remove the dependency on Oracle Identity Cloud Service. then you can recreate the users in the Oracle Cloud Infrastructure native IAM system.

In this document, we'll continue to use the pre-existing Oracle Cloud Infrastructure Compute Classic users and the Oracle Identity Cloud Service authentication. Then, we'll assign those users to the required groups and policies in Oracle Cloud Infrastructure.

For more information about Oracle Cloud Infrastructure IAM, see Oracle Cloud Infrastructure Security.

Compare Oracle Cloud Infrastructure and Oracle Cloud Infrastructure Compute Classic Features and Concepts

Here are some of the differences between Oracle Cloud Infrastructure and Oracle Cloud Infrastructure Compute Classic users.

Table 6-1 Comparison of Oracle Cloud Infrastructure Compute Classic and Oracle Infrastructure Cloud Users

Oracle Cloud Infrastructure Compute Classic Users	Oracle Infrastructure Cloud Users
Individual users can be granted specific roles (such as `Compute.Compute_Operations` for managing the Compute Classic Cloud service).	Privileges are granted through policy statements, and these policy statements can be applied only to a group (not an individual user). For example, you must create a group *Compute_Users* and then assign the appropriate policies to the group. The members of this group can then manage Oracle Cloud Infrastructure service operations.
Users can inherit the roles by being a member of a particular user group in Oracle Identity Cloud Service.	An Oracle Cloud Infrastructure group cannot contain Oracle Identity Cloud Service users directly; instead, it can map only to a group in Oracle Identity Cloud Service.

Before you can assign Oracle Cloud Infrastructure Compute Classic users to specific Oracle Cloud Infrastructure policies, you must first make sure the Oracle Cloud Infrastructure Compute Classic users are assigned to specific groups in Oracle Identity Cloud Service. These groups can then be mapped to a specific group with specific privileges in Oracle Cloud Infrastructure.

Assign Oracle Cloud Infrastructure Policies to Federated Oracle Identity Cloud Service Users

This section provides a typical procedure for configuring your existing Oracle Cloud Infrastructure Compute Classic users to manage Oracle Cloud Infrastructure resources, as part of an overall migration from Oracle Cloud Infrastructure Compute Classic to Oracle Cloud Infrastructure.

Verify that Your Oracle Cloud Infrastructure Account is Federated with Oracle Identity Cloud Service

Oracle Cloud Infrastructure tenancies created on December 18, 2017 or later are automatically federated with Oracle Identity Cloud Service.

If your tenancy was created before December 18, 2017, and you want to set up a federation with Oracle Identity Cloud Service, see Federating with Oracle Identity Cloud Service.

To verify your Oracle Cloud Infrastructure account is federated with Oracle Identity Cloud Service:

1. Go to the Oracle Cloud Infrastructure Console and sign in with your Oracle Cloud Infrastructure login and password.
2. Open the navigation menu. Under **Governance and Administration**, go to **Identity** and click **Federation**.

You should see **OracleIdentityCloudService** listed as an identify provider at the top of the page. If you don't see it listed, then see Federating with Oracle Identity Cloud Service.

Run Reports to List the Oracle Cloud Infrastructure Compute Classic Users, Groups, and Assigned Privileges in Identity Cloud Service

Before you configure the Oracle Cloud Infrastructure environment for Oracle Cloud Infrastructure Compute Classic users, it is helpful to know the complete list of users and groups, as well as the roles you have assigned to specific users in the Oracle Cloud Infrastructure Compute Classic environment. Note that Oracle Cloud Infrastructure Compute Classic users are created using the Oracle Identity Cloud Service.

To export the list of groups in spreadsheet format:

1. Sign in to your account and click **Users** and then click the link to go to the Oracle Identity Cloud Service Console.
2. Click **Groups**.
3. Click **Export**, and then select **Export All** to export all groups.
4. In the Export Groups window, click **Export Groups**.
5. After Oracle Identity Cloud Service creates the export file, a Job ID link appears. Click the link.
6. In the Jobs page, review the job details such as how many groups you exported, how many groups Oracle Identity Cloud Service exported successfully, and how many groups can't be exported because of a system error.
7. Click **Download**.

To export the list of users in spreadsheet format:

1. In the Oracle Identity Cloud Service Console, open the Navigation menu on the top left, and then click **Users**.
2. To export all user accounts, click **Export**, and then select **Export All**.

3. In the **Export Users?** dialog box, click **Export Users**.
4. After Oracle Identity Cloud Service creates the export file, you need to review the results.
   - If the job can be processed immediately, then a dialog box appears with the Job ID link for your import job. Click the link and review the details that appear on the Jobs page.
   - If the job cannot be processed immediately, then a message appears with a Schedule ID in it. Copy that Schedule ID, and use it to search for the job on the Jobs page. The job will appear when processing completes.
5. On the Jobs page, locate the job that you want to view, and then click **View Details**.
6. Click **Download**.

To view or download a report that shows the roles assigned to users in the Oracle Identity Cloud Service user database:

1. In the Identity Console, open the Navigation menu on the top left, and then click **Reports**.
2. In the Reports page, expand the **Applications** node.
3. Click the **Application Role Privileges** report. Detailed report information appears.
4. Filter the data that appears in the Application Role Privileges report by performing one of the following options:
   - To view application role grants and revokes for applications that are configured in Oracle Identity Cloud Service over a period of days, click 30 Days or 60 Days or 90 Days.
   - To specify a custom date range, click Custom Dates. To activate a date picker tool to select this date range, click the Calendar icon in the Start Date and End Date fields.

   Tip: You can sort the report data each column in the table in ascending or descending order by clicking the arrow next to the column title.
5. To download a PDF version of the report, click **Download Report**.

Create Groups in Identity Cloud Service for Each Required Role

In this step, you create a new group in the Identity Cloud Service Console for each of the user roles that you want to map to an Oracle Cloud Infrastructure policy. Policies are permissions you assign Oracle Cloud Infrastructure users to perform specific tasks.

For example, you typically want all Oracle Cloud Infrastructure Compute Classic users who are assigned privileges to manage Oracle Cloud Infrastructure Compute Classic virtual machines, to have similar privileges in Oracle Cloud Infrastructure so that they can manage the virtual machines. To do this:

1. Sort the Application Role Privileges report you generated to identify all the users assigned the `Compute.ComputeOperations` role.
2. In the Identity Console, open the Navigation menu on the top left, and then click **Groups**.
3. Click **Add**.

4. Create a new group called, `ComputeAdmins_IDCS` and click **Next**.
5. Select all the users that are currently assigned the `Compute.ComputeOperations` role to the new group.
6. Click **Finish**.

Optionally, you can assign applications to the group from the **Access** tab. For more information on user roles, see Add Users and Assign Roles.

Create a New Oracle Cloud Infrastructure Group for Your Compute Administrators

1. Go to the Oracle Cloud Infrastructure console.
2. Open the navigation menu. Under **Governance and Administration**, go to **Identity** and click **Groups**.

   A list of the groups in your tenancy is displayed.
3. Click **Create Group**.
4. Enter the following:
   - **Name:** Enter a name to identify the IDCS-based Compute administrators, such as "ComputeAdmins_IDCS". Note that you cannot change this name later.
   - **Description:** A friendly description. You can change this later if you want to.
   - **Tags:** Optionally, you can apply tags. If you have permissions to create a resource, you also have permissions to apply free-form tags to that resource. To apply a defined tag, you must have permissions to use the tag namespace. For more information about tagging, see Resource Tags. If you are not sure if you should apply tags, skip this option (you can apply tags later) or ask your administrator.
5. Click **Create Group**.

Map the Oracle Identity Cloud Service Group to the Oracle Cloud Infrastructure Group

The groups you create in Oracle Identity Cloud Service get access through groups you define in Oracle Cloud Infrastructure. Before your Oracle Identity Cloud Service groups can get access, you must create groups in Oracle Cloud Infrastructure with the desired permissions and then map your Oracle Identity Cloud Service groups to these. You can add permissions to the Oracle Cloud Infrastructure groups before or after you complete the mapping.

1. Open the Oracle Cloud Console.
2. Open the navigation menu. Under **Governance and Administration**, go to **Identity** and click **Federation**.
3. On the list of identity providers, click **OracleIdentityCloudService**.
4. Click **Edit Mapping**.
5. Provide the client ID and secret when prompted.

For information on how to get the client ID and secret, see Get required Information from Oracle Identity Cloud Service.

6. Click **+ Add Mapping**.
7. Select the Oracle Identity Cloud Service group from the list under **Identity Provider Group**.
8. Select the IAM group you want to map from the list under **Oracle Cloud Infrastructure Group**.
9. Repeat the **+ Add Mapping** steps for each mapping you want to create, and then click **Submit**.

If the mapping is successful, then Oracle Cloud Infrastructure Compute Classic users will be automatically mapped to the Oracle Cloud Infrastructure group. However, you can't see the members of the mapped group in the Groups page. To view the federated users of the mapped group, navigate to the Users page.

Create a Policy to Grant the Group Permissions on Oracle Cloud Infrastructure Resources

The group you created in Identity Console gets permissions to access resources in Oracle Cloud Infrastructure through the policy you assign to the Oracle Cloud Infrastructure group. Before you complete this step, you need to decide what permissions you want to give your new group.

Some of the common policies (permissions) you could assign your groups are:

- Allow network admins to manage load balancers
- Allow Compute admins to manage instances or launch instances
- Allow admins to access specific data region
- Allow network admins to manage all components of cloud network

When you assign these policies to a group, the users in the group will be able to carry out the specifc tasks in the policy. For more information, see Getting Started with Policies and Common Policies.

Prerequisite: The group and compartment that you're writing the policy for must already exist.

1. Go to the Oracle Cloud Infrastructure console.
2. Open the navigation menu. Under **Governance and Administration**, go to **Identity** and click **Policies**.

   A list of the policies in the compartment you're viewing is displayed.
3. If you want to attach the policy to a compartment other than the one you're viewing, select the desired compartment from the list on the left. Where the policy is attached controls who can later modify or delete it (see Policy Attachment).
4. Click **Create Policy**.
5. Enter the following:
   a. **Name**: A unique name for the policy. The name must be unique across all policies in your tenancy. You cannot change this later.
   b. **Description**: A friendly description. You can change this later if you want to.

- c. **Policy Versioning:** Select **Keep Policy Current** if you'd like the policy to stay current with any future changes to the service's definitions of verbs and resources. Or if you'd prefer to limit access according to the definitions that were current on a specific date, select **Use Version Date** and enter that date in format YYYY-MM-DD format. For more information, see Policy Language Version.
- d. **Statement**: A policy statement. For the correct format to use, see Policy Basics and also Policy Syntax. If you want to add more than one statement, click **+**.

For example:

To allow your group to manage all resources within a specified compartment enter a statement like the following:

```
Allow group <Oracle Cloud Infrastructure_group_name> to manage all-resources in
compartment <compartment_name>
```

6. Click **Create**.

You have a group called `ComputeAdmins_IDCS` in Oracle Cloud Infrastructure and it is mapped to a group called `OCI_Adminsitrators` in the Identity Cloud Service Console.

In the Oracle Cloud Infrastructure console, create and assign the following policies to the `ComputeAdmins_IDCS` group:

- Allow Compute admins to manage instances or launch instances
- Allow Compute admins to manage all components of cloud network

When you assign these policies to the group, users of the OCI_Adminstrators group in the Identity Cloud Service Console can manage networks and Compute instances in Oracle Cloud Infrastructure.

Verify Your Migration

After completing the migration steps, verify that your migration was successful, in the Oracle Cloud Infrastructure.

In Oracle Cloud Infrastructure, navigate to the Users page. A list of migrated users are listed in the page, which indicates that your Oracle Cloud Infrastructure Compute Classic users were successfully migrated.

For more information on how to verify the migration, see the section After the Federation Set Up in Oracle Cloud Infrastructure documentation.

7 Understand the Oracle Cloud Infrastructure Network Resources

There are certain key differences between the network models in Oracle Cloud Infrastructure Compute Classic and Oracle Cloud Infrastructure. This chapter helps you understand the network resources in Oracle Cloud Infrastructure and how they map to the network resources in Oracle Cloud Infrastructure Compute Classic.

While in some cases the differences in the network models in these two environments could have an impact on your network design and implementation, using the Terraform configuration generated by Oracle Cloud Infrastructure Classic Discovery and Translation Tool replicates the network environment from your Oracle Cloud Infrastructure Compute Classic account in your Oracle Cloud Infrastructure tenancy. When you use this Terraform configuration to migrate your network, you don't need to map any of your network resources manually. The information provided here helps you to understand how the tool maps the network resources in your source environment to the slightly different network resources in the target environment.

Understand the Oracle Cloud Infrastructure Compute Classic Network Models

Here's a brief description of the shared network and IP networks used in your Oracle Cloud Infrastructure Compute Classic account.

Term or Concept	Description	Oracle Cloud Infrastructure Compute Classic Documentation
IP Networks	Oracle Cloud Infrastructure Compute Classic IP networks allow you to define multiple, independent IP networks that can optionally be connected through an IP network exchange. Access control lists (ACLs) contain security rules that are applied to a group of network interfaces (vNICsets) across multiple networks.	Configuring IP Networks
Shared Network	The Oracle Cloud Infrastructure Compute Classic shared network is a single, flat network. Instances are grouped by security lists. Security rules define what traffic is allowed to a group of instances in a security list.	Configuring the Shared Network

Understand the Oracle Cloud Infrastructure Network Model

Here's a brief description of virtual cloud networks, subnets, and availability domains used in your Oracle Cloud Infrastructure tenancy.

Term or Concept	Description	Oracle Cloud Infrastructure Documentation
Virtual Cloud Networks (VCNs)	A virtual, private network that you set up in Oracle data centers. It closely resembles a traditional network, with firewall rules and specific types of communication gateways that you can choose to use. A VCN covers a single, contiguous IPv4 CIDR block of your choice.	• Overview of Networking • Default Components that Come With Your VCN
Subnets	Subdivisions you define in a VCN (for example, 10.0.0.0/24 and 10.0.1.0/24). Subnets contain virtual network interface cards (VNICs), which attach to instances. Subnets act as a unit of configuration within the VCN: All VNICs in a given subnet use the same route table, security lists, and DHCP options. You can designate a subnet as either public or private when you create it.	Overview of Networking
Availability Domain	Each subnet in a VCN exists in a single **availability domain** and consists of a contiguous range of IP addresses that do not overlap with other subnets in the VCN.	Regions and Availability Domains

Understand How Your Oracle Cloud Infrastructure Compute Classic Network Resources Map to Oracle Cloud Infrastructure Network Resources

The following table provides some basic information about how the elements of your Oracle Cloud Infrastructure Compute Classic network map to the corresponding Oracle Cloud Infrastructure network elements.

Understand How Oracle Cloud Infrastructure Compute Classic Network Concepts Map to Oracle Cloud Infrastructure Network Concepts

Oracle Cloud Infrastructure Compute Classic Network Resource	Oracle Cloud Infrastructure Network Resource
Shared Network	A single subnet in a VCN.
IP Network	Subnets within a single VCN OR Multiple VCNs with local peering configured – if the subnets span different parent CIDR block ranges and need to be interconnected
Corente VPN or VPNaaS	IPSec VPN
Oracle Cloud Infrastructure FastConnect Classic	Oracle Cloud Infrastructure FastConnect

Understand How Oracle Cloud Infrastructure Compute Classic Shared Network Concepts Map to Oracle Cloud Infrastructure Network Concepts

Oracle Cloud Infrastructure Compute Classic Shared Network Resource	Oracle Cloud Infrastructure Network Resource
Security lists	A security list applied to a subnet in a VCN or a set of network security group (NSG) security rules.
Security rules	An Ingress and Egress security rule within a security list or a security rule in an NSG.
Security applications	The TCP, UDP or ICMP options within a security rule.
Security IP lists	No direct equivalent. Security rules must be defined for a single source or destination IP prefix. In NSGs, security rules can use the same NSG or another NSG in the same VCN as a source or destination.

Understand How Oracle Cloud Infrastructure Compute Classic IP Network Concepts Map to Oracle Cloud Infrastructure Network Concepts

Oracle Cloud Infrastructure Compute Classic IP Network Resource	Oracle Cloud Infrastructure Network Resource
IP network exchange	Partially maps to a VCN. IP network exchanges provide connectivity between IP networks. In Oracle Cloud Infrastructure subnets under a VCN are connected by default. If an IP network translates to multiple subnets across multiple VCNs, then a local peering gateway (LPG)is required to connect the subnets.
Virtual NIC sets	No direct equivalent. However, creating a network security group (NSG) allows you to specify a set of vNICs in a VCN and apply a set of security rules to this set of vNICs.
Access Control Lists (ACLs) ACLs are applied to a set of vNICs. The vNICs don't have to be within a single IP network or in IP networks connected to an IP network exchange.	A security list applied to a subnet in a VCN or a set of security rules in an NSG. Security lists in Oracle Cloud Infrastructure are applied at the subnet level and can't be applied to specific vNICs. NSG security rules are applied to the set of vNICs specified in the NSG. These vNICs must be in a single VCN.
Routes	Routes
Security rules	An ingress and egress security rule within a security list or an NSG security rule.
IP Address Prefix Sets	No direct equivalent. Security rules must be defined for a single source or destination IP prefix. In NSGs, security rules can use the same NSG or another NSG in the same VCN as a source or destination.

Understand How Oracle Cloud Infrastructure Compute Classic Security Rules Map to Oracle Cloud Infrastructure NSG Security Rules

If you use IP networks in Oracle Cloud Infrastructure Compute Classic, then Access Control Lists (ACLs) are used to apply a set of security rules to a set of instance interfaces.

In Oracle Cloud Infrastructure, network security groups (NSGs) are used to apply a set of security rules to a set of instance interfaces in a VCN.

However, because of differences in the way in which NSG security rules are defined compared to security rules in ACLs, you must keep the following considerations in mind when you start planning your network migration.

- Oracle Cloud Infrastructure Compute Classic allows you to set up a large number of security lists and security rules. Oracle Cloud Infrastructure permits a smaller number of NSGs and NSG security rules. If you use a large number of security lists and security rules in your source environment, you might not be able to directly migrate your network architecture to Oracle Cloud Infrastructure. Check the number of NSGs and NSG security rules you'll need and find out your tenancy limits before migrating the network to your target environment.

- Each security rule in a given ACL might translate to one or more NSG security rules in the target environment. For example, in Oracle Cloud Infrastructure Compute Classic, a security rule can specify a list of ports or a list of IP addresses in the source or destination, while in Oracle Cloud Infrastructure each of these fields can take only a single value.

Considerations for Setting Up Your Oracle Cloud Infrastructure Network

When you migrate your network elements to Oracle Cloud Infrastructure, consider the following points for determining DNS names and the CIDR block size and prefix for your VCNs and subnets.

VCN and Subnet CIDR Prefixes

When you create a VCN or a subnet in Oracle Cloud Infrastructure, you must specify the IP address range for the VCN or subnet in the form of a CIDR prefix. You must select this CIDR prefix carefully, because you can't change it after the VCN or subnet has been created.

If you use Oracle Cloud Infrastructure Classic Discovery and Translation Tool to migrate your network, the tool generates Terraform configuration to replicate the network in your source environment. The tool takes the following points into consideration when designing the network in the target environment. If you want to design your network manually or if you want to modify the Terraform configuration generated by the tool, then consider the following points:

- It is recommended that the private IP addresses associated with each instance be retained in the migration process. Design the network architecture in the target environment carefully to ensure that private IP addresses can be migrated wherever possible.
- For a VCN, select a CIDR block that can accommodate all IP networks that need to be interconnected, if possible.
  For example, consider IP networks with the following CIDR prefixes, connected to an IP network exchange:
  - `192.168.1.0/24`
  - `192.168.2.0/24`
  - `192.168.3.0/20`

  These IP networks can be migrated as one or more subnets in a single VCN. In this case, you can create the VCN with the CIDR range `192.168.0.0/16`.

  However, consider IP networks with the following CIDR prefixes, connected to an IP network exchange.
  - `192.168.1.0/24`
  - `172.16.1.0/24`

Although you want to enable connectivity across the subnets that these IP networks are migrated to, you can't create the required subnets in a single VCN with a /16 CIDR prefix. You must either migrate one of the IP networks to a different CIDR block, or if you want to retain the same IP addresses, then you must migrate these two IP networks to separate VCNs and enable VCN peering across those networks.

- If you need to connect two VCNs using VCN peering, remember that the peered VCNs must have non-overlapping CIDR prefixes.
- Select a VCN and a subnet CIDR block that is large enough so that you can add more VMs in the same VCN and subnet later on, if required.
- For a subnet, select a CIDR block large enough to accommodate the private IP addresses all of the instances that you want to migrate to this subnet, if possible.
- As far as possible, select a CIDR block for each subnet that can include all the IP addresses of instances in the source environment that you want to migrate to this subnet. If a single subnet maps to a single IP network, the subnet's CIDR should map to the IP network's address range, whenever possible.

DNS Names

DNS names used in Oracle Cloud Infrastructure Compute Classic will change when your instances are migrated to Oracle Cloud Infrastructure. When specifying DNS names, consider the following.

- In Oracle Cloud Infrastructure Compute Classic, the way DNS names are derived depends on whether instances are in the shared network or in an IP network.
  - In the shared network, DNS names for the private IP are derived from the host name by appending the domain `<accountName>.oraclecloud.internal`. This domain name is assigned by the system and can't be changed.
  - In IP networks, you can specify the hostname and the DNS name separately, and multiple DNS names are allowed, with each DNS name resolving to the private IP. You can specify any FQDNs, as required.
- In Oracle Cloud Infrastructure, DNS names for the private IP are derived from the host name by appending the domain `<subnetLabel>.<VcnLabel>.oraclevcn.com`. Here, the subnet label and VCN label can be user-specified, but `oraclevcn.com` is assigned by the system and can't be changed.

Since there is no one-to-one mapping of DNS names between Oracle Cloud Infrastructure Compute Classic and Oracle Cloud Infrastructure, consider the following recommendations when assigning DNS names in Oracle Cloud Infrastructure:

- If the instance host name is specified in Oracle Cloud Infrastructure Compute Classic, use that for both the instance name and the host name. The DNS name will be derived from this host name by appending the domain name as described above.
- If no host name is specified in Oracle Cloud Infrastructure Compute Classic, but the `dns` attribute is specified in the networking section of the instance orchestration, pick the first name in the `dns` list. Use the host name part of the dns name as the instance host name.
- If no host name is specified and the `dns` attributed is also not specified in the instance orchestration, generate a host name from the instance name.

Note that the DNS name always changes during migration, as the domain for Oracle Cloud Infrastructure Compute Classic is different from the domain for Oracle Cloud Infrastructure.

Applications that Use DNS

If the instances that you are migrating host applications that use DNS, then consider the differences between DNS features in Oracle Cloud Infrastructure Compute Classic and Oracle Cloud Infrastructure and select a strategy to migrate your instances so that your applications continue to work without requiring configuration changes.

In Oracle Cloud Infrastructure Compute Classic, the top-level domain is `oraclecloud.internal`. A fully-qualified domain name (FQDN) is assigned to each instance by default. You can specify an FQDN to override the default value.

In Oracle Cloud Infrastructure Compute Classic, external DNS resolution isn't supported. Only instances in a tenancy can resolve the IP addresses of other instances in the same tenancy.

In Oracle Cloud Infrastructure, the top-level domain is `oraclevcn.com`. You can specify a DNS label for each VCN and subnet that you create, as well as a host name for each instance. The FQDN of an instance has the form: `<hostname>.<subnet DNS label>.<VCN DNS label>.oraclevcn.com`

Before you migrate your instances, consider the following strategies:

- Use custom DNS servers to preserve FQDNs
- Specify search domains and host names

Preserve FQDNs with Custom DNS Servers

With this strategy, you preserve the fully qualified domain name of each instance, so that the applications don't need any configuration changes when you migrate them to Oracle Cloud Infrastructure.

- A pair of instances in Oracle Cloud Infrastructure run a DNS server (for example, Bind 9).
  - The IP address to name mappings are extracted from the resources JSON file and configured in the DNS servers.
  - Security rules allow port 53 TCP/UDP ingress from the migrated instances.
- Migrated instances are created as follows:
  - With the same static IP addresses as they had in Oracle Cloud Infrastructure Compute Classic.
  - Their DHCP option `custom_dns_servers` is configured with the IP addresses of the DNS server instances.
    See: https://www.terraform.io/docs/providers/oci/r/core_dhcp_options.html
  - Security rules allow port 53 TCP/UDP egress to the DNS server instances.

Use Search Domains and Host Names

This strategy works if the applications to be migrated can be configured to have URLs or server names make reference to short host DNS names. For example, the applications can be configured to reference `http://foo/some/path/` instead of `http://foo.compute-608547156.oraclecloud.internal./some/path/`. Note that this configuration might already be the default, because in Oracle Cloud Infrastructure Compute Classic, the default search domain is already set for this to work out of the box.

The advantage of this solution is that it doesn't require a set of dedicated DNS servers and IP addresses can be automatically allocated to the migrated instances in Oracle Cloud Infrastructure.

If required, you can change the applications to connect with the short host name instead of the FQDN.

Instances are started with proper search domain specified as part of the `search_domain_names` DHCP options. See: https://www.terraform.io/docs/providers/oci/r/core_dhcp_options.html

Typically, the search domain is set to: `<subnet DNS label>.<VCN DNS label>.oraclevcn.com.`, since the FQDN is typically: `<host name>.<subnet DNS label>.<VCN DNS label>.oraclevcn.com.`.

8

Create a Virtual Cloud Network in Oracle Cloud Infrastructure

You must set up your virtual cloud network (VCN) in your Oracle Cloud Infrastructure tenancy, before you create compute, load balancer, database, or other resources.

Complete the Prerequisites

Before you create the Oracle Cloud Infrastructure VCN, you should do the following.

- Review the Oracle Cloud Infrastructure Compute Classic Architecture
- Understand the Oracle Cloud Infrastructure Network Resources
- You should be familiar with the fundamentals of networking in Oracle Cloud Infrastructure. For information about setting up a cloud network in Oracle Cloud Infrastructure, see:
  - Overview of Networking
  - VCNs and Subnets
- Oracle Cloud Infrastructure Classic Discovery and Translation Tool can help you to identify and filter information about the networking resources in your Oracle Cloud Infrastructure Compute Classic environment. If you use this tool to identify resources in your Oracle Cloud Infrastructure Compute Classic account, then you can use the reports generated by this tool to help design and set up the networking objects in your Oracle Cloud Infrastructure tenancy. You should be familiar with this tool and its commands and options. For information about using this tool, see Identify and Translate Resources in Your Source Environment.
- The Terraform configurations generated by Oracle Cloud Infrastructure Classic Discovery and Translation Tool can be used to automate the set up of resources in your Oracle Cloud Infrastructure tenancy. To use Terraform to apply those configurations in the target environment, you should be familiar with Terraform.
- Before you apply the Terraform configuration, you'll need to create a compartment in your Oracle Cloud Infrastructure tenancy. It's strongly recommended that you create a new compartment dedicated to the migration process. Don't use the root compartment for migration.

Considerations for Migrating Your Network

You must carefully consider several aspects of your network design when you migrate your network from Oracle Cloud Infrastructure Compute Classic to Oracle Cloud Infrastructure.

Public IP Addresses

When you migrate your network, some network configuration might change. For example, the public IP addresses associated with your instances on Oracle Cloud

Infrastructure Compute Classic can't be reused on Oracle Cloud Infrastructure. This is because the range of public IP addresses available in both services is different. Evaluate the impact of this change on your applications and architecture before you start your migration.

Security Rules

You'll see some changes is in the way security rules are designed and applied in Oracle Cloud Infrastructure compared to Oracle Cloud Infrastructure Compute Classic.

- In Oracle Cloud Infrastructure Compute Classic, if you use IP networks, then security rules are applied to groups of vNICs called *vNICsets*. If you use the shared network, security rules are applied to a set of instance interfaces that are members of a *security list*.
- In Oracle Cloud Infrastructure, using *network security groups (NSGs)* allows you to specify a set of vNICs in a VCN and apply a set of security rules to these vNICs.
- NSGs, however, are only scoped to affect vNICs within the same VCN. NSGs don't affect vNICs in peered VCNs and NSG security rules can't reference NSGs from peered VCNs. If your IP networks are migrated as separate but connected VCNs, the Terraform configuration generated by Oracle Cloud Infrastructure Classic Discovery and Translation Tool sets up VCN peering along with the required security lists and security rules to provide that connectivity. For more information about NSGs and NSG security rules, see Network Security Groups in the Oracle Cloud Infrastructure documentation.
- When you use the Terraform configuration generated by Oracle Cloud Infrastructure Classic Discovery and Translation Tool to migrate your network, your network is migrated as follows:
  - If you have multiple IP networks connected with an IP network exchange:
    - If the address ranges of all the IP networks can be expressed as a single CIDR address range no larger than /16, then your network is migrated as a single VCN in Oracle Cloud Infrastructure, with the appropriate NSGs and NSG security rules to permit and restrict traffic between sets of vNICs in the VCN.
    - If the address ranges of all IP networks can't be expressed as a single /16 CIDR address range, then:
      1. The IP networks are migrated to separate VCNs.
      2. In this case, NSG security rules can't be used to permit and restrict traffic between sets of vNICs in different VCNs. Security lists and security rules are created to permit or restrict traffic between instances in a VCN.
      3. Connectivity across the VCNs is implemented using LPGs.
  - If you have multiple IP networks that aren't connected with an IP network exchange, then each IP network is migrated as a separate VCN, with the appropriate NSGs and NSG security rules to permit and restrict traffic between sets of vNICs within each VCN.

Multiple vNICs

If instances in your Oracle Cloud Infrastructure Compute Classic account have multiple vNICs:

1. You might need to use a different instance shape in Oracle Cloud Infrastructure to support multiple vNICs. Smaller shapes support a smaller number of vNIC attachments. Depending on the number vNICs required, the instances shape (OCPU count) may need to be increased. See Map Oracle Cloud Infrastructure Compute Classic Instance Shapes to Oracle Cloud Infrastructure Shapes.
2. Launch the instance with the primary vNIC on the appropriate primary subnet. If you use the Terraform configuration generated by Oracle Cloud Infrastructure Classic Discovery and Translation Tool to launch your instances, this is done automatically.
3. Additional vNICs must be attached as secondary vNICs after the instance is launched. See https://docs.cloud.oracle.com/iaas/Content/Network/Tasks/managingVNICs.htm. If you use the Terraform configuration generated by Oracle Cloud Infrastructure Classic Discovery and Translation Tool to launch your instances, the additional vNICs are attached automatically.

Migrate the Shared Network

If your source environment uses the shared network, then your network is migrated to Oracle Cloud Infrastructure with a single VCN that has a single subnet.

In Oracle Cloud Infrastructure Compute Classic, the shared network doesn't allow you to select or specify private IP addresses and private IP addresses aren't persistent. However, when you create the VCN and subnet in Oracle Cloud Infrastructure, you can use the `--shared-network-prefix` option to specify the IP address range for private IP addresses. Primary private IP addresses are persistent in Oracle Cloud Infrastructure.
To migrate your shared network to the target environment use Oracle Cloud Infrastructure Classic Discovery and Translation Tool. This tool simplifies the process of setting up your network in Oracle Cloud Infrastructure. You can use this tool to discover all the security rules applied to each security list in your source environment. You can then generate a Terraform module to create a VCN, subnet, network security groups (NSGs) and NSG security rules for your shared network.

You must have already installed Oracle Cloud Infrastructure Classic Discovery and Translation Tool and run the `opcmigrate discover` command before you run the following commands.

1. To view the security rules associated with a security list in the shared network in Oracle Cloud Infrastructure Compute Classic, run the following command:

   ```
   opcmigrate network --shared-grouping seclist
   ```

   The output of this command provides a list of security rules associated with each security list.
2. To generate Terraform, use the following commands:

   ```
   opcmigrate plan create --output migration-plan.json
   ```

   ```
   opcmigrate generate --plan migration-plan.json --output main.tf
   ```
3. Review the generated Terraform and make any required modifications before creating the network and applying the security rules to subnets in Oracle Cloud Infrastructure.

4. If you want to enable direct access to the public Internet for VMs launched in a VCN, use the Oracle Cloud Infrastructure Console to create an Internet Gateway for the VCN. To create an Internet Gateway using the Console:

   a. Open the navigation menu. Under **Core Infrastructure**, go to **Networking** and click **Virtual Cloud Networks**.

   b. Click the VCN you're interested in.

   c. Click **Internet Gateways**.

   d. Click **Create Internet Gateway**.

   e. Enter the following:

      - **Create in Compartment:** Leave as is.

      - **Name:** A friendly name for the Internet Gateway. It doesn't have to be unique, and it cannot be changed later in the Console (but you can change it with the API). Avoid entering confidential information.

      - **Tags:** Optionally, you can apply tags. If you are not sure if you should apply tags, skip this option. You can apply tags later.

   f. Click **Create**.
      Your internet gateway is created and displayed on the **Internet Gateways** page. Ensure that you have a route rule that allows traffic to flow to the gateway.

5. If required, you can enable access to the public Internet for VMs that have only private IP addresses, by using a NAT gateway. To create a NAT gateway using the Console:

   a. Open the navigation menu. Under **Core Infrastructure**, go to **Networking** and click **Virtual Cloud Networks**.

   b. Click the VCN you're interested in.

   c. On the left side of the page, click **NAT Gateways**.

   d. Click **Create NAT Gateway**.

   e. Enter the following values:

      - **Create in compartment**: The compartment where you want to create the NAT gateway, if different from the compartment you're currently working in.

      - **Name:** A friendly name for the NAT gateway. It doesn't have to be unique. Avoid entering confidential information.

      - **Tags:** Optionally, you can apply tags. If you are not sure if you should apply tags, skip this option. You can apply tags later.

   f. Click **Create NAT Gateway**.
      The NAT gateway is then created and displayed on the **NAT Gateways** page. Ensure that you have a route rule that directs the desired traffic from the subnet to the NAT gateway. Do this for each subnet that needs to access the gateway.

6. Migrate your instances and block volumes. You can use Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool to automate this process. See Migrate Virtual Machines and Block Storage to Oracle Cloud Infrastructure.

7. After your instances are migrated, launch your instances in the appropriate subnet and add the vNICs to the appropriate NSGs. The NSGs that vNICs are added to should correspond to the security lists that the instance was added to in the source environment.

   The process of launching instances in appropriate subnets can be performed manually or by applying the Terraform configuration generated by Oracle Cloud Infrastructure Classic Discovery and Translation Tool. If you use the Terraform configuration generated by Oracle Cloud Infrastructure Classic Discovery and Translation Tool to launch your VMs, the VMs are automatically launched in the appropriate subnet and the vNICs are automatically added to the appropriate NSGs. For more details about launching your VMs after migration, see Launch VMs in the Target Environment.

Migrate IP Networks

If your source environment uses IP networks, then, to recreate your network in the target environment using network security groups (NSGs), you can use Oracle Cloud Infrastructure Classic Discovery and Translation Tool.

This tool simplifies the process of setting up your network in Oracle Cloud Infrastructure. You can use this tool to discover all the security rules applied to each IP network in your source environment. You can then generate a Terraform module to create the corresponding VCNs and subnets, and – wherever possible – the required NSGs and NSG security rules for each of your IP networks. If VCNs corresponding to separate IP networks need to be connected using VCN peering, the Terraform configuration sets up the components required for VCN peering.

You must have already installed Oracle Cloud Infrastructure Classic Discovery and Translation Tool and run the `opcmigrate discover` command before you run the following commands.

1. To view the security rules applied to vNICs in an IP network, use the following command:

   ```
   opcmigrate network --ipnetwork-grouping ipnetwork
   ```

   The output of this command provides a list of ACLs along with the associated security rules applied to the vNICs in an IP network. These security rules are translated into a set of security rules in network security groups (NSGs) in Oracle Cloud Infrastructure.

2. To generate Terraform, use the following commands:

   ```
   opcmigrate plan create --output migration-plan.json

   opcmigrate generate --plan migration-plan.json --output main.tf
   ```

> **Note:**
>
> If you are migrating the shared network as well, then you need to run these commands just once. These commands generate the Terraform configuration for the shared network as well as your IP networks, along with instance configurations.

By default, the Terraform configuration creates the required VCNs and subnets along with the required NSGs and the NSG security rules. The generated configuration exactly replicates the security context created by the vNICsets, security rules, and ACLs in your IP networks.

If the private IP addresses of IP networks connected to an IP network exchange can't be represented in a single /16 CIDR prefix, then multiple VCNs must be created. These VCNs are connected using VCN peering. However, NSG security rules can only make reference to NSGs within the same VCN. NSG security rules can't reference NSGs across peered VCNs. In this case, no NSGs or NSG security rules are generated by default. Use the `--with-security-rule-union` option to generate security lists and security rules. This option generates a security list with a union of all the security rules related to the specified IP networks.

> **Caution:**
>
> When you use the `--with-security-rule-union` option, review the generated Terraform configuration carefully before applying it. The security rules generated with this option might expose instances to more traffic than intended.

3. Review the generated Terraform and make any required modifications before creating the network and applying the security rules in Oracle Cloud Infrastructure.
4. If you want to enable direct access to the public Internet for VMs launched in a VCN, use the Oracle Cloud Infrastructure Console to create an Internet Gateway for the VCN. To create an Internet Gateway using the Console:
   a. Open the navigation menu. Under **Core Infrastructure**, go to **Networking** and click **Virtual Cloud Networks**.
   b. Click the VCN you're interested in.
   c. Click **Internet Gateways**.
   d. Click **Create Internet Gateway**.
   e. Enter the following:
      - **Create in Compartment:** Leave as is.
      - **Name:** A friendly name for the Internet Gateway. It doesn't have to be unique, and it cannot be changed later in the Console (but you can change it with the API). Avoid entering confidential information.
      - **Tags:** Optionally, you can apply tags. If you are not sure if you should apply tags, skip this option. You can apply tags later.
   f. Click **Create**.

Your internet gateway is created and displayed on the **Internet Gateways** page. Ensure that you have a route rule that allows traffic to flow to the gateway.

5. If required, you can enable access to the public Internet for VMs that have only private IP addresses, by using a NAT gateway. To create a NAT gateway using the Console:

   a. Open the navigation menu. Under **Core Infrastructure**, go to **Networking** and click **Virtual Cloud Networks**.

   b. Click the VCN you're interested in.

   c. On the left side of the page, click **NAT Gateways**.

   d. Click **Create NAT Gateway**.

   e. Enter the following values:

      - **Create in compartment**: The compartment where you want to create the NAT gateway, if different from the compartment you're currently working in.

      - **Name:** A friendly name for the NAT gateway. It doesn't have to be unique. Avoid entering confidential information.

      - **Tags:** Optionally, you can apply tags. If you are not sure if you should apply tags, skip this option. You can apply tags later.

   f. Click **Create NAT Gateway**.
   The NAT gateway is then created and displayed on the **NAT Gateways** page. Ensure that you have a route rule that directs the desired traffic from the subnet to the NAT gateway. Do this for each subnet that needs to access the gateway.

6. If you migrate the shared network as well as one or more IP networks, the shared network and the IP networks are migrated as separate VCNs. If you need to enable traffic across those VCNs, you will need to set up VCN peering. See Connect VCNs Using Local Peering Gateways.

7. Migrate your instances and block volumes. You can use Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool to automate this process. See Migrate Virtual Machines and Block Storage to Oracle Cloud Infrastructure.

8. After your instances are migrated, launch your instances in the appropriate subnet for each instance and add the vNICs to the appropriate NSGs. The subnet an instance is launched in should correspond to the IP network that the instance had an interface on in the source environment. The NSGs that vNICs are added to should correspond to the vNICsets the vNICS belonged to in the source environment.

   The process of launching instances in appropriate subnets can be performed manually or by applying the Terraform configuration generated by Oracle Cloud Infrastructure Classic Discovery and Translation Tool. If you use the Terraform configuration generated by Oracle Cloud Infrastructure Classic Discovery and Translation Tool to launch your VMs, the VMs are automatically launched in the appropriate subnet and the vNICs are automatically added to the appropriate NSGs. For more details about launching your VMs after migration, see Launch VMs in the Target Environment.

Connect VCNs Using Local Peering Gateways

VCN peering can be required when you migrate multiple IP networks connected to an IP network exchange. If you use the Terraform configuration generated by Oracle Cloud Infrastructure Classic Discovery and Translation Tool to migrate your network, VCN peering is set up automatically when required.

However, if you migrate the shared network as well as one or more IP networks, the shared network and the IP networks are migrated as separate VCNs. In this case, VCN peering across these VCNs isn't set up by default. If you need to enable traffic across these VCNs, you will need to set up VCN peering.
VCN peering involves the following steps:

1. Create local peering gateways (LPGs) in each VCN.
2. Establish the connection.
3. Update route tables to enable traffic between the peered VCNs as desired.
4. Update security lists to enable traffic between the peered VCNs as desired.

This procedure assumes that you have policies in place.
For example, if you belong to the `NetworkAdmin` group or the `Administrators` group, you might have either of the following policies already in place:

```
Allow group NetworkAdmin to manage virtual-network-family in tenancy
```

Or:

```
Allow group Administrators to manage all-resources in tenancy
```

If you need to set up IAM policies required for this task, or for more information about VCN peering concepts and procedures, see Local VCN Peering in the Oracle Cloud Infrastructure documentation.

To set up VCN peering using the Console:

1. Create the LPG in each of the VCNs that you want to connect.
   a. Open the navigation menu. Under **Core Infrastructure**, go to **Networking** and click **Virtual Cloud Networks**.
   b. Click the VCN you're interested in.
   c. On the Virtual Cloud Network Details page, click **Local Peering Gateways** on the left of the page.
   d. Click **Create Local Peering Gateway**.
   e. Enter the following:
      - **Name:** A friendly name for the LPG. It doesn't have to be unique, and it cannot be changed later in the Console (but you can change it with the API). Avoid entering confidential information.
      - **Create in compartment**: The compartment where you want to create the LPG, if different from the compartment you're currently working in.

- **Associate with Route Table (optional)**: Leave this field blank. It isn't required for this migration scenario.
- **Tags:** Optionally, you can apply tags. If you are not sure if you should apply tags, skip this option. You can apply tags later.

f. Click **Create.** The LPG is created and displayed on the **Local Peering Gateways** page.

g. Repeat these steps to create the LPG in the other VCN.

2. In the context of VCN peering, one network is considered the requestor and the other network is considered the acceptor. The VCN peering connection is initiated by the requestor and accepted by the acceptor. After both LPGs have been set up, to establish the VCN peering connection:

   - Open the navigation menu. Under **Core Infrastructure**, go to **Networking** and click **Virtual Cloud Networks**.

   a. Click the requestor LPG's VCN.

   b. On the Virtual Cloud Network Details page, click **Local Peering Gateways** on the left of the page.

   c. For the requestor LPG that you want to use, click **View Peering Gateway.**

   d. On the Local Peering Gateway Details page, click **Establish Connection**.

   e. Select the acceptor LPG that you want to peer with.

   f. Click **Establish Peering Connection**.

   The connection is established and the LPG's state changes to PEERED. The details of each LPG update to show the **Peer VCN CIDR Block** for the other VCN.

3. Next, create or edit the route table entries in the route table for each VCN, to ensure that traffic intended for the peered subnet is routed to the appropriate LPG.

   a. Determine which subnets in the requestor VCN need to communicate with the acceptor VCN.

   b. Update the route table for each of those subnets to add a rule that directs traffic destined for the acceptor VCN to your LPG:

   i. Open the navigation menu. Under **Core Infrastructure**, go to **Networking** and click **Virtual Cloud Networks**.

   ii. Click the requestor VCN.

   iii. Click **Route Tables**.

   iv. Select the default route table.

   v. Click **Edit Route Rules**.

   vi. Click **+ Another Route Rule** and enter the following:

   - **Target Type:** Local Peering Gateway.
   - **Destination CIDR Block:** The acceptor VCN's CIDR block. If you want, you can specify a subnet or particular subset of the peered VCN's CIDR.
   - **Target Compartment:** The compartment where the LPG is located, if not the current compartment.
   - **Target:** The LPG.

        vii. Click **Save**. Any subnet traffic with a destination that matches the rule is now routed to your LPG.

    c. Repeat these steps to update the route table for the acceptor VCN as well.

4. Verify that the appropriate security rules apply to each of the subnets participating in the peered connection. Add the following security rules, as required:

    - Ingress rules for the types of traffic you want to allow from the other VCN's CIDR block or specific subnets.

    - Egress rules to allow outgoing traffic from your VCN to the other VCN. If the subnet already has a broad egress rule for all types of protocols to all destinations (0.0.0.0/0), then you don't need to add specific egress rules.

    To add or edit security rules in security lists applied to your subnets:

    a. In the Console, while viewing the VCN you're interested in, click **Security Lists**.

    b. Click the security list you're interested in.

    c. Click **Edit All Rules** and create one or more rules, each for the specific type of traffic you want to allow.

    d. Click **Save Security List Rules** at the bottom of the dialog box.

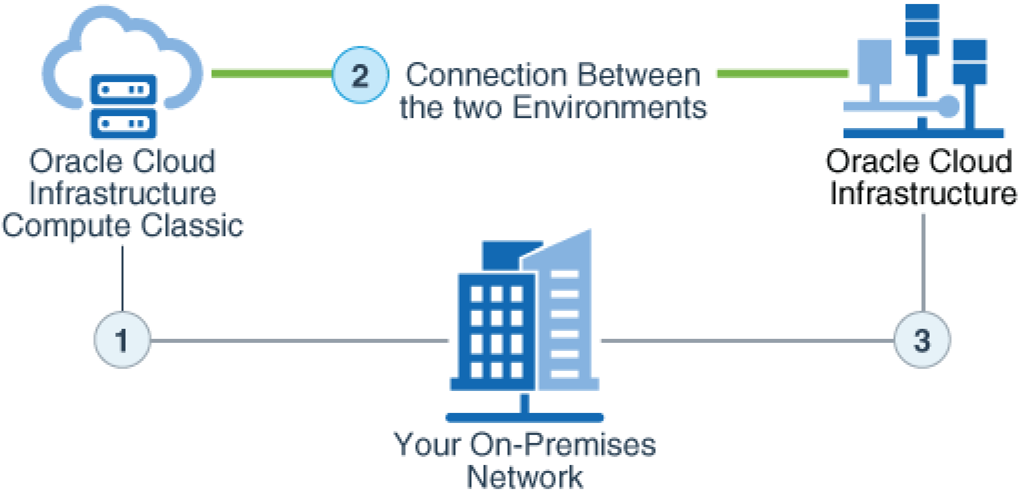
    e. Repeat these steps for all subnets in both VCNs, as required.

9 Connect the Source and Target Networks

After you create the Virtual Cloud Network (VCN) in your Oracle Cloud Infrastructureenvironment, you should consider your requirements for connecting to the new network.

Understanding Connections to Your New VCN

The following is a high-level diagram of the connections between your Oracle Cloud Infrastructure Compute Classic network (the source of your migration), the Oracle Cloud Infrastructure network (the target of your migration), and your on-premises data center network.


Oracle Cloud Infrastructure Compute Classic
2
Connection Between the two Environments
Oracle Cloud Infrastructure
1
3
Your On-Premises Network

For information about the connection between your on-premises data center and your Oracle Cloud networks, see Connect Your On-Premises Network to Your Oracle Cloud Infrastructure Network. These connections are identified by items 1 and 3 in the diagram.

About Connecting the Source and Target Networks

When migrating from Oracle Cloud Infrastructure Compute Classic to Oracle Cloud Infrastructure, you can configure a connection between an IP network in Oracle Cloud Infrastructure Compute Classic and a subnet in the Oracle Cloud Infrastructure VCN.

This connection is not necessarily a requirement. Depending on the types of workloads and application data you are migrating and the tools you are using, such a connection is typically optional.

In some cases, such as PaaS instances, you may need to establish a connection between the Oracle Cloud Infrastructure Compute Classic and the Oracle Cloud Infrastructure environments using a FastConnect or IPSec tunnel. If you are using tools provided in the Oracle Cloud Infrastructure Classic Migration Tools image to migrate your workloads, you don't have to establish a connection.

For example, this connection is not required if you are migrating workloads using migration tools such as Oracle Cloud Infrastructure Classic Discovery and Translation

Tool or Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool. However, when you migrate specific platform cloud service workloads to Oracle Cloud Infrastructure, you might need to use this connection to transfer your applications, data, and configuration settings to the new Oracle Cloud Infrastructureenvironment. Refer to the specific migration documentation or contact your Oracle representative for more information.

Instructions for connecting the two networks are available in the Access to Oracle Cloud Infrastructure Classic topic in the Oracle Cloud Infrastructure documentation.

For information about connecting the shared network in Oracle Cloud Infrastructure Compute Classic environment to Oracle Cloud Infrastructure, see Set Up VPN Connection between Shared Network and Oracle Cloud Infrastructure in *Using Oracle Cloud Infrastructure Compute Classic*.

Note:

If the same IP address ranges have been retained in the Oracle Cloud Infrastructure network configuration that are used in the Oracle Cloud Infrastructure Compute Classic network configuration, then you should not connect both networks at the same time over IPSec or FastConnect as this will introduce routing conflicts, unless a NATing solution is place.

10 Connect Your On-Premises Network to Your Oracle Cloud Infrastructure Network

After you migrate to the new Oracle Cloud Infrastructure network, you can connect your on-premises data center to your Oracle Cloud Infrastructure network. This connection is important if you run workloads across your on-premises and cloud-based infrastructure or if for any other reason you need to extend your network to include your on-premises as well as your cloud-based resources.

You can use any of the following options to establish a connection from your on-premises data center to your existing Oracle Cloud Infrastructure Compute Classic resources:

Connection Option	Description	Oracle Cloud Infrastructure Compute Classic Documentation Topics
VPN access using Corente Services Gateway	Use this solution when you use Oracle Cloud Infrastructure Compute Classic shared networks. Corente is an Oracle-provided IPSec solution. Corente Services Gateway acts as a proxy to facilitate secure access and data transfer to your instances. All VPN connections to your multitenant Oracle Cloud Infrastructure Compute Classic site use a Corente Services Gateway instance in the cloud.	Connecting to Instances in a Multitenant Site Using VPN
VPN as a Service (VPNaaS)	Use this VPN approach when you use Oracle Cloud Infrastructure Compute Classic IP networks.	Setting Up a VPN Connection Using VPNaaS
FastConnect Classic	FastConnect Classic allows you to access Oracle Cloud services using a direct connection from your on-premises data centers.	Connecting to Instances Using FastConnect

Oracle Cloud Infrastructure provides both VPN and FastConnect options for connecting your on-premises data center to the Oracle Cloud Infrastructure network.

- IPSec VPN Overview
- FastConnect Overview

After you migrate your resources to your Oracle Cloud Infrastructure environment, you'll want to either reconfigure your existing FastConnect or IPSec VPN connection to point to the new Oracle Cloud Infrastructure network, or you'll want to create a new FastConnect or IPSec VPN connection to Oracle Cloud Infrastructure, so you can run both connections in parallel.

About Migrating from FastConnect Classic to Oracle Cloud Infrastructure FastConnect

This document provides instructions for migrating FastConnect Classic users to Oracle Cloud Infrastructure FastConnect.

In most cases, it's recommended that you keep your existing FastConnect Classic links established during the migration of your data between Oracle Cloud Infrastructure Compute Classic and Oracle Cloud Infrastructure data centers. You can leverage your FastConnect Classic public peering session to access public endpoints from both data centers during this period. Depending on your workload and your migration strategy, you can benefit from FastConnect Classic public peering global prefixes advertisements.

If you need to migrate your FastConnect Classic link to a different geographical location or facility, then you must carefully plan for the migration in consultation with your FastConnect Classic provider, network service provider, and data center providers, to anticipate any potential delays or service interruptions.

> **Note:**
>
> Migrating to a different location can have cost and performance impacts. Evaluate these changes with your network service provider prior to choosing a target location.

Before You Begin

Before you begin migrating Oracle Cloud Infrastructure FastConnect Classic, identify the following information:

- The location where Oracle Cloud Infrastructure FastConnect Classic has been provisioned.
- The connectivity model.
- Your FastConnect provider.
- The subscribed bandwidth.
- The peering types enabled.
- The target region that you want to migrate to.

For details about the connectivity models, providers, peering types, or other information, see https://cloud.oracle.com/en_US/fastconnect. For information about Oracle Cloud Infrastructure regions, see About Regions and Availability Domains in Oracle Cloud Infrastructure documentation.
To find the required information, check the initial order that you had placed for Oracle Cloud Infrastructure FastConnect Classic or refer to the excel file that was created in the activation phase of the service. You may have received this information in an email

from `saas_provisioning@custhelp.com`. If you can't find this information, contact your Oracle Cloud representative.

Options for Migrating FastConnect Classic

The migration process varies depending on your location, connectivity models, and partners. This document describes the migration process for the following scenarios:

- Migration process for standard/colocation edition:
  - If your FastConnect Classic location is Slough and your target Oracle Cloud Infrastructure region is London
  - For any other source and target regions
- Migration process for partner/provider edition:
  - If your FastConnect Classic location is Slough or Amsterdam
  - If your FastConnect Classic location is Ashburn and your target Oracle Cloud Infrastructure region is also Ashburn
  - For any other source and target regions, where your current provider operates in the target region
  - For any other source and target regions, where your current provider doesn't operate in the target region

Migrate FastConnect Classic Standard or Colocation Edition

With FastConnect Classic standard or colocation edition, you manage your own equipment hosted in a FastConnect Classic location. A network service provider operates the link between your on-premises location and the FastConnect Classic location.

If your FastConnect Classic location is Slough and your target Oracle Cloud Infrastructure region is London, then when you are ready to start the migration, perform the following steps:

1. In your Oracle Cloud Infrastructure tenancy, use the Console to create a new cross-connect and get the LOA.
2. Submit a change request to your data center provider (Equinix) to update the existing cross-connect with the information provided in the new LOA.
3. When the cross-connect is updated, use the Oracle Cloud Infrastructure Console to create a new virtual circuit.
4. Finalize the BGP peering session configuration on your equipment.

For all other source or target locations, perform the following steps:

1. Identify a data center provider for your target Oracle Cloud Infrastructure region. For a list of providers, see https://cloud.oracle.com/en_US/fastconnect/emea-providers
2. Provide the data center provider's address to your network service provider. Plan with your network service provider for establishing the new link.
3. Rent space and ship your equipment to the target location provided by the data center provider.

4. In your Oracle Cloud Infrastructure tenancy, use the Console to create a new cross-connect and get the LOA.
5. Submit a cross-connect request to your data center provider and provide them the new LOA.
6. When the cross-connect is updated, use the Oracle Cloud Infrastructure Console to create a new virtual circuit.
7. Finalize the BGP peering session configuration on your equipment.

Migrate FastConnect Classic Partner or Provider Edition

With FastConnect Classic provider or partner edition, you rely on FastConnect Classic service providers to establish and maintain end-to-end connectivity between your on-premises location and your FastConnect Classic location.

Complete these steps in the following scenarios:

- If your FastConnect Classic location is Slough or Amsterdam
- If your FastConnect Classic location is Ashburn and your target Oracle Cloud Infrastructure region is also Ashburn
- For any source and target regions, if your current FastConnect provider *operates* in the target region

In these scenarios, to perform the migration, complete the following steps:

1. In your Oracle Cloud Infrastructure tenancy, use the Console to create a new virtual circuit and get the OCID of the new circuit.
2. Submit a change request to your FastConnect provider to update the existing virtual circuit with the new circuit OCID and location.
3. When the circuit is updated, finalize the BGP peering session configuration with your FastConnect provider.

For a list of FastConnect provider locations, see https://cloud.oracle.com/en_US/fastconnect/providers.

If your current FastConnect provider *doesn't operate* in the target region, then you must subscribe to another FastConnect provider. Perform the following steps:

1. Follow the standard procedure to provision a new FastConnect circuit with a new FastConnect Provider in the target region. For information about setting up FastConnect on Oracle Cloud Infrastructure, see FastConnect: With an Oracle Provider in Oracle Cloud Infrastructure documentation.
2. When the new circuit is provisioned, terminate your contract with your existing FastConnect provider.

About Migrating Your IPSec VPN Connection

If you use an IPSec VPN connection to connect to instances in your Oracle Cloud Infrastructure Compute Classic account, then you can set up an IPSec VPN connection to the VCN in your Oracle Cloud Infrastructure tenancy as well.

Before You Begin

Before you begin migrating the IPSec VPN connection, ensure that you have completed the following tasks.

- Identify whether the on-premises router is a route-based or policy-based device.
- Identify if the on-premise device is placed behind a NAT device.
- Ensure that you have set up an IPSec VPN connection in your Oracle Cloud InfrastructureVCN. The IPSec connection contains multiple IPSec tunnels for redundancy. For each IPSec tunnel, collect the following information:
  - The IP address of the Oracle IPSec tunnel endpoint (the VPN headend)
  - The pre-shared key (PSK)

Configure the Customer-Premises Equipment

After setting up the IPSec VPN connection in Oracle Cloud Infrastructure, configure the customer-premises equipment, or CPE in your on-premises environment.

The required configuration depends on the type of the on-premises router. See Configuring Your CPE in Oracle Cloud Infrastructure documentation.

Take care of the following additional configuration requirements while peering the Oracle Cloud Infrastructure DRG with a policy-based device in your on-premise environment. There are additional requirements that you must meet if the on-premise device is place behind a NAT device.

1. If you are using a policy-based device in your on-premises environment, ensure the following additional configuration for your on-premise device:
   a. Create a single SPI with a single destination IP address. Oracle recommends using a single SPI with the following values:
      - Source IP address: Any (0.0.0.0/0)
      - Destination IP address: VCN CIDR (example: 10.120.0.0/20)
      - Protocol: IPv4

   b. Make sure the single SPI matches any traffic that needs to go from your on-premises network across the IPSec tunnel to the VCN. The VCN CIDR must not overlap with your on-premises network.

   c. Ensure that the on-premise device is always the initiator of the tunnel. You don't have to meet this requirement only if you create the single SPI with both Source IP address and Destination IP address as Any (0.0.0.0/0).

   d. Ensure that the encryption domain in your on-premise environment has only one local subnet and one remote subnet. If multiple subnets are present or if you use multiple subnets with Oracle Cloud Infrastructure Compute Classic, you'll need to perform route summarization.
2. If the on-premise device is placed behind a NAT device:
   a. Oracle recommends that you disable NAT-T at your on-premise device when establishing IPSec tunnels with Oracle Cloud Infrastructure. Unless you have multiple CPEs sharing the same NAT IP, NAT-T is not required.

b. While creating the CPE in Oracle Cloud Infrastructure, you specify the public IP address of the on-premise device. Your on-premise IPSec local identity must match your CPE public IP. If your CPE is behind a NAT device but does not support setting the IPSec local identity, file a ticket at My Oracle Support for help in configuring your CPE and bringing up the tunnels.

After configuring the CPE, ensure that the IPSec tunnel enters the UP state and that you can send and receive traffic over the IPSec tunnel.

11
Select a Method to Migrate Database Instances

Many cloud infrastructure environments rely on databases to support applications deployed on virtual machines. There are several options for migrating your Oracle Database Classic Cloud Service instances to Oracle Cloud Infrastructure, depending on the type of database, the amount of downtime you can plan for, and the tools you have available.

Database Migration Options Covered In This Guide

This migration guide describes three different database migration options. The following table provides a summary of each one.

Database Migration Option	Description	Required Tools and Technologies	Typical Use Case
Migrate Databases Using the Migration Tools	The Oracle Cloud Infrastructure Classic Database Backup Migration Tool creates a Recovery Manager (RMAN) backup of your Oracle Database Classic Cloud Service instances. RMAN backups are stored in the associated Oracle Cloud Infrastructure Object Storage Classic container. The migration tool automatically transfers the backup to Oracle Cloud Infrastructure as a standalone backup. You can restore this standalone backup to a new Oracle Cloud Infrastructure Database system.	• Oracle Cloud Infrastructure Migration Tools • Oracle Database Recovery Manager (RMAN)	This option is best suited for migrating database instances in your development and test environments. In this scenario, you can use the same migration tool that you used to migrate your infrastructure resources. The downtime depends upon the size of the database and the speed of the network.

Database Migration Option	Description	Required Tools and Technologies	Typical Use Case
Learn About Migrating a Database Cloud Service Deployment to a Virtual Machine Database System	With this migration option, you use Oracle Data Guard to migrate a single-instance database, created with Oracle Database Classic Cloud Service on a Compute Classic server, to an Oracle Cloud Infrastructure Virtual Machine Database System.	Oracle Data Guard	Use this option if you are using applications that depend on Oracle Data Guard. Downtime is kept to a minimum and is equal to the time it takes for Oracle Data Guard to perform a switchover from the source database to the target (standby) database. This can vary, depending on the network speed and the load on the database at the time.
Learn About Migrating a Multi-Node Database Cloud Service Deployment to Virtual Machine Database System	This migration option also uses Oracle Data Guard, but it is designed to migrate a multi-node Oracle Real Application Clusters (RAC) database to a Virtual Machine Database System on Oracle Cloud Infrastructure.	Oracle Data Guard	Use this option if you are using applications that depend on Oracle Data Guard. Downtime is kept to a minimum and is equal to the time it takes for Oracle Data Guard to perform a switchover from the source database to the target (standby) database. This can vary, depending on the network speed and the load on the database at the time.

Database Migration Option	Description	Required Tools and Technologies	Typical Use Case
Migrate New Data in Real Time from Oracle Database Classic to Oracle Cloud Infrastructure	This migration option is a real-time migration. The source is an Oracle database Classic instance and the target is an Oracle database system on a virtual machine on OCI. This migration method watches the source database and updates the target with every change that happens on the source. Every new insert, delete, or update that happens on the source will be reflected in the target database in real-time. For this method, there is no need to shutdown any of the databases. This method is applicable for Oracle databases 11g to 11g, 11g to 12c, and 12c to 12c.	Oracle Data Integration Platform Cloud	Use this migration option for the following scenarios: • When you don't want to shut down your source production database. • For real time migration of Oracle database 11g to 12c, which is more complicated with other migration methods. • When your new database only requires a copy of your source database from a certain point of time that you choose. This time has to be after the time that you start the migration. The target database can be used for data analytics, to take the query load off your source database. It can also be used as a more current database, which doesn't include historical data from the source database. For a complete migration of the source database, including the

Database Migration Option	Description	Required Tools and Technologies	Typical Use Case
			historical data, use Oracle Data Pump or Oracle Recovery Manager (RMAN) to copy the initial load from source to the target first and perform this real-time migration after that.

Additional Database Migration Options

You can find additional options to migrate the database instances on the Oracle Help Center's Solution page.

12 Migrate Databases Using the Migration Tools

Use the Oracle Cloud Infrastructure Classic Discovery and Translation Tool to migrate Oracle Database Classic Cloud Service instances to Oracle Cloud Infrastructure Database systems. You can migrate one or more database instances at a time. This tool creates a Recovery Manager (RMAN) backup of your Oracle Database Classic Cloud Service instance. RMAN backups are stored in the associated Oracle Cloud Infrastructure Object Storage Classic container. The migration tool automatically transfers the backup to Oracle Cloud Infrastructure as a standalone backup. You can restore this standalone backup to a new Oracle Cloud Infrastructure Database system. Oracle Cloud Infrastructure offers multiple options for restoring a standalone backup.

This solution is best suited for migrating database instances in your development and test environments. The database is restored only till the point when the backup was taken. Any subsequent transactions which are not part of backup are not available when the database is restored.

Workflow

Here's an overview of the high-level steps required to migrate your database instances from Oracle Database Classic Cloud Service to Oracle Cloud Infrastructure. You can use Oracle Cloud Infrastructure Classic Discovery and Translation Tool to automate this process.

1. Create an instance in Oracle Cloud Infrastructure Compute Classic using the Oracle Cloud Infrastructure Classic Migration Tools image.
2. Identify the database instances that you want to migrate.
3. Run the `opcmigrate migrate database migrate` command to initiate the migration process.
   This command creates a Recovery Manager (RMAN) backup of your Oracle Database Classic Cloud Service instance and transfers the backup to Oracle Cloud Infrastructure as a standalone backup.
4. Run the `opcmigrate migrate database list` command to view the status of the migration process.
5. Restore the standalone backup of the source database. Oracle Cloud Infrastructure offers multiple options for restoring the backup.

Complete the Prerequisites

Before you begin, complete the following prerequisites.

- To use the Oracle Cloud Infrastructure Classic Database Backup Migration Tool to migrate your database instances, you must create an instance in your Oracle Cloud Infrastructure Compute Classic account using the Oracle Cloud Infrastructure Classic Migration Tools image. Ensure that you have sufficient quota for this instance to be created. For information about creating your Control-S instance, see Complete the Prerequisites and Launch the Migration Controller Instance (Control-S) in the Source Environment. If you've already created this instance earlier in your migration process, you can use the same instance for this procedure.

- The `opcmigrate migrate database migrate` command of Oracle Cloud Infrastructure Classic Database Backup Migration Tool creates an RMAN backup of the database instance in Oracle Cloud Infrastructure. RMAN backup is temporarily stored in the associated Oracle Cloud Infrastructure Object Storage Classic container before the tool uploads it to Oracle Cloud Infrastructure. Ensure that you have sufficient quota and required storage space to save the backup in your source environment and in Oracle Cloud Infrastructure.
- Ensure that you have a virtual cloud network (VCN) with subnets set up in a compartment that you have access to.
- Ensure that the API access PEM key does not have a pass phrase. The Oracle Cloud Infrastructure Classic Database Backup Migration Tool requires that you have set up an API user in your Oracle Cloud Infrastructure environment.
- Ensure that the `oracle` user of the source database system has the `SYSDBA` role and can perform password-less sudo.

Plan for the Migration

To use Oracle Cloud Infrastructure Classic Database Backup Migration Tool to migrate your database instances, you must create and configure an instance in your Oracle Cloud Infrastructure Compute Classic account using the Oracle Cloud Infrastructure Classic Migration Tools image.

1. Create or update the `.opc/profiles/default` file on Control-S. The Oracle Cloud Infrastructure Classic Database Backup Migration Tool used to migrate database instances requires access to the Oracle Database Classic Cloud Service account.

   > **Note:**
   >
   > If you run the Control-S setup command for Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool, it generates the default profile using information provided by you in the `secret.yml` file. The Control-S setup command overwrites any existing default profile. Verify and update the existing profile, or create a new profile, if required.

   a. You'll need the SID, IP address, edition, and name of the Oracle Database Classic Cloud Service instance that you want to migrate. Look this up in your Oracle Cloud Dashboard. The edition value can be **SE** for Standard Edition, **EE** for Enterprise Edition, **EE_HP** for Enterprise Edition High Performance, and **EE_EP** for Enterprise Edition Extreme Performance.

   b. You'll also need the private SSH key file corresponding to the public SSH key that is associated with your database instance. Ensure the SSH key pair was generated password-less, and is sized -b 4096 or smaller. Use the following command to base64-encode the private SSH key.

```
base64 -i --wrap=0 /home/opc/.ssh/name_of_private_key_file
```

c. Create or locate your profile file. This file contains the credentials and other information required to access your source environment. The default location for this file is `~/.opc/profiles/default`. If required, you can create multiple profiles and use the `--profile` option to specify the profile you want to use when you run the tool. If you create profiles in a location other than `~/.opc/profiles`, provide the full path to the profile location by using the `--profile-directory` option, along with the `--profile` option to specify the profile file name. If no profile is specified, the `~/.opc/profiles/default` profile is used.

d. Use the following template to add the `"database"` section to your profile file. You can specify details of one or more database instances that you want to migrate. Replace the sample values with values specific to your account.

```
"database": {
  "orcl-db1": {
    "credentials": "LS0tLS1...WS0tLS0tCg==",
    "source_ip": "129...",
    "edition": "SE",
    "sid": "prodenv"
  },
  "orcl-db2": {
    "credentials": "LS0tLS1...WS0tLS0tCg==",
    "source_ip": "129...",
    "edition": "EE",
    "sid": "ORCL"
  }
}
```

Where,

`orcl-db1` and `orcl-db2` are the names of the Oracle Database Classic Cloud Service instances that you want to migrate from your source environment.

`credentials` is the base64-encoded format of the private SSH key file that corresponds to the public SSH key that is associated with the Oracle Database Classic Cloud Service instance that you want to migrate.

`source_ip` is the IP address of the Oracle Database Classic Cloud Service instance. This IP address is used to SSH to the instance.

`edition` is the Oracle Database Classic Cloud Service instance software edition that you want to migrate from your source environment.

`sid` is the SID associated with the source Oracle Database Classic Cloud Service instance.

2. Get details of your target environment. Log in to the Oracle Cloud Infrastructure Console and collect the required information. You'll need the following:

   - The user OCID. From the menu, choose Identity and then Users.
   - The API PEM key fingerprint. Click the user to view user details. The API Keys section displays the PEM key fingerprint.

- The compartment OCID. From the menu, choose Identity and then Compartments. The standalone backup is created in this compartment.
- The tenancy OCID and the tenancy region. From the menu, choose Administration and then choose Tenancy Details.
- The Availability Domain. From the menu, choose **Networking** and then **Virtual Cloud Networks**. Click the VCN that you've created for this migration. The **Subnets** section displays the availability domain. The standalone backup is created in this availability domain.

3. Create or edit the `/home/opc/.oci/config` file on Control-S. This file contains the credentials and other information required to access your target environment. If you've used Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool earlier in the migration process, this would already have been created. However, the file might not contain all of the information required by the Oracle Cloud Infrastructure Classic Database Backup Migration Tool. For example, you might have to add information about the availability domain.
   If you run the Control-S setup command for Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool, it generates the `.oci/config` file, using information provided by you in the `secret.yml` file. The Control-S setup command overwrites any existing `.oci/config file`.

   Use the following template to create or update the `.oci/config` file. Replace the sample values with values specific to your tenancy.

   ```
   [DEFAULT]
   user=ocid1.user.oc1..aaaaa...
   fingerprint=81:45:aa:2...
   key_file=<path to api pem key>
   compartment=ocid1.compartment.oc1...
   tenancy=ocid1.tenancy.oc1..aaaaaaa...
   region=us-ashburn-1
   ad=ilMx:US-ASHBURN-AD-1
   ```

Migrate Database Instances

Use the Oracle Cloud Infrastructure Classic Database Backup Migration Tool to migrate database instance to Oracle Cloud Infrastructure. You can run this tool on an instance in Oracle Cloud Infrastructure Compute Classic that was created using the Oracle Cloud Infrastructure Classic Migration Tools image.

> **Note:**
>
> If you download and install Oracle Cloud Infrastructure Classic Discovery and Translation Tool on your local system, you won't be able to use the `opcmigrate migrate database` commands, as these commands aren't included in that distribution of the tool. To use the `opcmigrate migrate database` commands, you must create an instance using the Oracle Cloud Infrastructure Classic Migration Tools image.

1. To create an RMAN back up of the source database instance and to save it as a standalone backup in Oracle Cloud Infrastructure, run the following command:

   ```
   opcmigrate migrate database migrate
   ```

   Sample output:

   ```
   opcmigrate-database: INFO: Launching migration for database: orcl-db
   ```

   Where, `orcl-db` is the name of the source database system.

2. To view the status of a migration job, run the following command:

   ```
   opcmigrate migrate database list
   ```

   Sample output:

   ```
   opcmigrate migrate database list
   opcmigrate-database: INFO: Get list of migrations in progress
   opcmigrate-database: INFO: SID     Source IP
   Status                          Backup Name      Backup OCID
   opcmigrate-database: INFO: ORCL1   129.150.8...
   SUCCESS                         Backup_ORCL_1
   ocid1.dbsystem.oc1.iad.re7...
   opcmigrate-database: INFO: ORCL2   129.150.1...
   CONNECT_FAILURE_FOR_SOURCE_HOST
   ```

   Note down the backup OCID as you'll have to provide this information later.

   In the response, look at the value returned for the `Status` field. Its value changes to **Success** when the back up to Oracle Cloud Infrastructure is complete.

The database backup is created and then it is copied to Oracle Cloud Infrastructure. This standalone backup is created in the compartment and the availability domain that you have specified in the `.oci/config` file. Note down the name of the backup the backup OCID.
After the database backup is available in Oracle Cloud Infrastructure, create a new Oracle Cloud Infrastructure Database instance from the standalone backup.

Restore a Database Instance

Create a new Oracle Cloud Infrastructure Database instance from the standalone backup in Oracle Cloud Infrastructure. Select a method to restore the database and create a new database instance on Oracle Cloud Infrastructure.

Restore a Database Using the Console or the API

- Using the Oracle Cloud Infrastructure web console. See Recovering a Database from Object Storage in *Oracle Cloud Infrastructure documentation*.
- Using the Oracle Cloud Infrastructure API. See Using the API in *Oracle Cloud Infrastructure documentation*.

Restore a Database Using Terraform

Before you begin, you must install and configure Terraform. Instructions to set up, configure, and run Terraform are outside the scope of this document.

To restore Oracle Cloud Infrastructure Database instances from the standalone backup using the terraform file generated by the Oracle Cloud Infrastructure Classic Discovery and Translation Tool:

1. Update the `~/.opc/profiles/default` file on Control-S. You'll need to provide your user name, the identity domain and region in which the Oracle Database Classic Cloud Service instances that you want to migrate has been provisioned, and the REST endpoint to access the PaaS account. Look this up in your Oracle Cloud Dashboard as follows:

   a. Sign in to the Oracle Cloud My Services application. The My Services dashboard is displayed.

   b. Look for **Compute Classic** on the Dashboard, and then select **View Details** from the **Actions** menu.

   c. In the **Overview** tab, under **Additional Information**, note down the **Identity Service Id**. For example, `idcs-9cd522ebea844aa1ab64c6773da59a54`.

   d. Click **Open Service Console**. The Compute Classic console is displayed.

   e. Click **Site** near the top of the page, and then note down the value selected in the **Site** drop-down list. This is the region in which your database instance has been provisioned.

   f. Identify the endpoint for your PaaS account based on the region or value that you have noted down from the **Site** drop-down list.

      - If your Oracle cloud account is provisioned in the `uscom` region, use `psm.us.oraclecloud.com`.
      - If your Oracle cloud account is provisioned in the `aucom` region, use `psm.aucom.oraclecloud.com`.
      - Otherwise, use `psm.europe.oraclecloud.com`.

   g. Locate your profile file, which contains the information required to access your source environment. The default location for this file is `~/.opc/profiles/default`.

   h. Use the following template to add the `"paas"` section to your profile file. Replace the sample values with values specific to your account. You'll need to specify the user name that you provided when you logged in to use the service, the **Identity Service Id**, the region or value that you noted down from the **Site** drop-down list, and the endpoint.

```
"paas": {
  "user": "acme@example.com",
  "identity_id": "idcs-9bd5....",
  "endpoint": "psm.us.oraclecloud.com",
  "region": "uscom-central-1"
},
```

2. Generate a list of database resources in your source environment. On the Control-S instance, run the following command. While running this command, you can specify the profile file that you have updated in the previous step.

```
opcmigrate discover
```

The output of this command is a file named `resources-default.json`, unless you specified a name for the output file while running the command. This file contains details of the database resources in your source environment. It is created in the directory where you run the command.

3. Log in to the Oracle Cloud Infrastructure Console to gather the OCID of the subnet in which you want to restore the database instances. From the menu, choose **Networking** and then **Virtual Cloud Networks**. Click the VCN that you've created for this migration. The **Subnets** section displays the OCID. Copy the OCID of the subnet as you'll have to provide this information.

4. Run the following command to generate a plan that includes information about the Oracle Cloud Infrastructure Database instances that you want to create.

Syntax

```
opcmigrate plan create --focus database_cloud --db-subnet-id
DB_SUBNET_ID --db-sshkeys DB_SSHKEYS -o plan_file.json
```

Where,

- `DB_SUBNET_ID` is the OCID of the subnet where you want to restore the database instance. If the plan file contains information about multiple databases, all these databases will be restored in the specified subnet.
- `DB-SSHKEYS` is the public SSH key that you want to associate with the restored Oracle Cloud Infrastructure Database instances.
- `plan_file.json` is the path to the output plan file. If you don't provide the `-o` option, the generated plan is stored in a file named `plan-default.json` in the current directory. Note that, if a file with the same name already exists in the current directory, it will be overwritten.

Example

```
opcmigrate plan create --focus database_cloud --db-subnet-id
ocid1.subnet... --db-sshkeys "ssh-rsa AAA..." -o dbmigrate-plan.json
```

The plan file is created at the location you specify for the `-o` option. You'll use this plan file to generate a terraform file.

5. Run the following command to generate a terraform file.

Syntax

```
opcmigrate generate --plan plan_file.json -o terraform_file.tf
```

Where, `terraform_file.tf` is path to output terraform file. If you don't provide this option, the terraform is available on the console but the terraform file is not created.

Example

```
opcmigrate generate --plan dbmigrate-plan.json -o dbmigrate.tf
```

6. Open the generated terraform file in any text editor of your choice. Add the following line just before the closing braces (}) for each database system to define the storage size for the database.

   ```
   data_storage_size_in_gb = "storage_size_of_database_in_GB"
   ```

   You must specify only one of the following values as the storage size in GB for the Oracle Cloud Infrastructure Database: 256, 512, 1024, 2048, 4096, 6144, 8192, 10240, 12288, 14336, 16384, 18432, 20480, 22528, 24576, 26624, 28672, 30720, 32768, 34816, 36864, 38912, or 40960. The storage size that you define must be larger than the size of the standalone backup from which you want to restore.

   A snippet of a sample terraform file is provided below for your reference to help you identify the place where you have to define the storage size of the database.

   ```
   # database_name
   resource "oci_database_db_system" "database_name_xxxxxx" {
   ....
   # Add the storage size for the database system here.
   }
   ```

7. The terraform file defines three variables for every standalone backup. Note down the names of these variables for every standalone backup that you want to restore. The value for these variables is not assigned in this terraform file. You'll assign values for these variables in the terraform variables file. A snippet of a sample terraform file is provided below for your reference.

   ```
   ...
   variable "ocic_dbaas_to_oci_database_edition" {
   ...
   }
   variable "database_name_xxxxxx_admin_password" {}
   variable "database_name_xxxxxx_backup_id" {}
   variable "database_name_xxxxxx_backup_tde_password" {}

   provider "oci" {
   ...
   }
   ```

8. Create or update an existing terraform variables file. Enter the names of the variables that you have noted down from the generated terraform file, and then define values for these variables.

   - `database_name_xxxxxx_admin_password`: Specify a single password for the SYSTEM and SYS users, as well as the backup TDE password on the Oracle Cloud Infrastructure Database instance. Ensure that you conform to the standards set for passwords in Oracle Cloud Infrastructure.
   - `database_name_xxxxxx_backup_id`: Specify the OCID of the standalone backup that was created when you ran the `opcmigrate migrate database list` command.

- *database_name*_xxxxxx_backup_tde_password: Specify the password for the SYS admin on the source Oracle Database Classic Cloud Service database that you want to migrate. When the database is created, the password for SYS user and the backup TDE password are the same. If you have not changed these values, then you can provide the SYS admin password.

9. Apply the terraform configuration to create Oracle Cloud Infrastructure Database instances.
10. Confirm that your Oracle Cloud Infrastructure Database instances are being created. Open the navigation menu in the Oracle Cloud Infrastructure console. Under **Database**, click **Bare Metal, VM, and Exadata**, and then click **DB Systems**. You'll see that the database is in the provisioning state.

Troubleshooting

Here are a few tips for dealing with errors that might occur while migrating database instances using Oracle Cloud Infrastructure Classic Discovery and Translation Tool.

To view the status of a migration job, run the following command:

```
opcmigrate migrate database list
```

In the response, look at the value returned for the Status field.

Code	Status	Description	Troubleshoot
000	Success	The database backup to Oracle Cloud Infrastructure has been completed successfully. This backup is available as a standalone backup in Oracle Cloud Infrastructure tenancy.	Not applicable
001	Verifying	Checks are in progress to ensure that enough space is available in source database system to temporarily store the backup.	Not applicable
002	Backing Up	The database backup is in progress.	Not applicable
003	Uploading	The database backup is in progress.	Not applicable
004	Restoring	The database backup is in progress.	Not applicable
100	Decoding failed for credential	The private key that you have specified in the profile file is not base64 encoded.	You must specify the base-64 encoded private SSH key in the "database" section of the profile file located at ~/.opc/profiles/default.
101	Connect failure for source host	Connection could not be established to the source database host.	Ensure that you can SSH to the source database host using the credentials that you have provided in the profile file.
102	Authentication failure for source host root user	SSH connectivity to the source database host could not be established using the specified credentials.	Ensure that you have provided the correct credentials in the "database" section of the profile file located at ~/.opc/profiles/default.

Code	Status	Description	Troubleshoot
103	User sudo failed	Password-less access to the source database host as a root user was not possible with the currently specified user.	Ensure that the `oracle` user has password-less access to the source database host as a root user.
104	Oracle user invalid	The provided `oracle` user or the specified user does not exist.	Ensure that you provide the correct `oracle` user and that the user has the required permissions. Also ensure that the credentials are in the required format.
105	User sudo to oracle user failed	The `oracle` user does not have root access	Ensure that the `oracle` user has root permission
106	Oracle user insufficient privilege	Provided `oracle` user does not have sufficient privileges.	Ensure that the `oracle` user you specify has the required privileges.
107	Oracle user not SYSDBA	The `oracle` user or the specified user does not have the SYSDBA role.	Ensure that the `oracle` user you specify has the required roles assigned.
108	Oracle script installation failed	The required CLI script could not be installed from Oracle github.	Ensure that an outbound connection can be established from the source database host. If required, set up the security rules to permit outbound connection.
109	Oracle Cloud Infrastructure configuration file cannot be read	The Oracle Cloud Infrastructure configuration file either does not exist or it can't be read.	Ensure that the Oracle Cloud Infrastructure configuration file exists and it can be read. The expected permission on the file is 400.
110	Oracle Cloud Infrastructure configuration file parsing error	Oracle Cloud Infrastructure configuration file could not be parsed.	Ensure that the configuration file is a valid JSON file and it has the correct structure.
111	Oracle Cloud Infrastructure configuration file missing key_file key	The Oracle Cloud Infrastructure configuration file does not have the `key_file` value or the content of `key_file` is empty.	Ensure that you provide `key_file` value in Oracle Cloud Infrastructure configuration file and the file has valid content.
112	Oracle Cloud Infrastructure script failed to create backup	Failed to create backup in Oracle Cloud Infrastructure. The script could not create the database back up.	View the log file located at `/var/log/dbmigration.log` to identify the errors that occurred, and then identify the corrective measures.
113	Oracle Cloud Infrastructure configuration file missing availability domain	Oracle Cloud Infrastructure configuration file missing availability domain.	Ensure that you specify the availability domain in the Oracle Cloud Infrastructure configuration file.
114	Oracle Cloud Infrastructure configuration key file does not exist or cannot be read	Oracle Cloud Infrastructure key file does not exist or cannot be parsed.	Ensure that you provide a valid key file with the proper access permissions.
115	Invalid source SID	Source SID is invalid.	Ensure that the SID of the source host is valid.
150	Size verification failure	There is not enough space in the source database system to temporarily store the database backup file.	Ensure that the required space is available in the source database system.
151	Backup failure	Unknown error	View the log file located at `/var/log/dbmigration.log` to identify the errors that occurred, and then identify the corrective measures.

Code	Status	Description	Troubleshoot
500	Internal server error	Internal error	View the log file located at `/var/log/dbmigration.log` to identify the errors that occurred, and then identify the corrective measures.

13

Learn About Migrating a Database Cloud Service Deployment to a Virtual Machine Database System

If you want to migrate a single-instance database, created with Oracle Database Classic Cloud Service on an Oracle Cloud Infrastructure Compute Classic server, to an Oracle Cloud Infrastructure Virtual Machine Database System, then you can perform the database migration by using Oracle Data Guard.

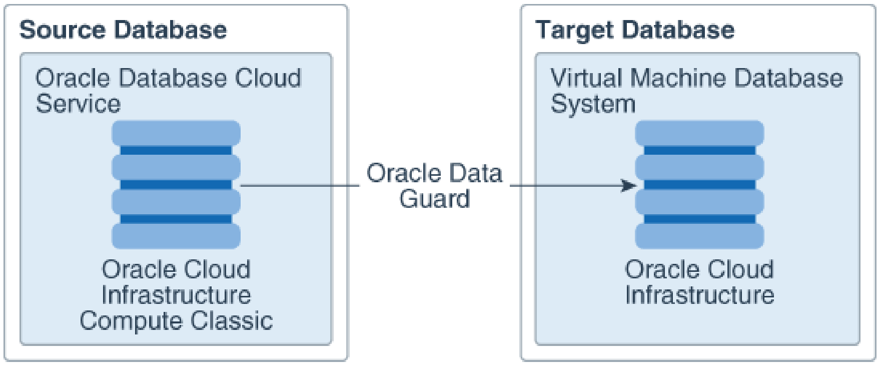
This procedure can be used as part of an overall migration of your Oracle Cloud environment to Oracle Cloud Infrastructure.

Architecture

You can migrate Oracle Database releases 12.1.0.2 and 12.2.0.1. Before you migrate your database, you must have an Oracle Database Classic Cloud Service instance that you want to migrate, and an Oracle Cloud Infrastructure Server with Oracle Database installed.

When you use Oracle Data Guard to perform the migration, the source database is the primary database, and the target database is the standby database.

The following diagram shows the migration process:


Source Database
Oracle Database Cloud Service
Oracle Cloud Infrastructure Compute Classic
Oracle Data Guard
Target Database
Virtual Machine Database System
Oracle Cloud Infrastructure

To perform the migration, you must follow these general steps:

1. Plan the migration.
   When you plan the database migration, you begin by inventorying the source environment (the primary database) and then you decide on the best migration strategy. To inventory the source environment, you must perform tasks such as determining the sizes of database files and checking which disaster recovery plans are in place. To decide on the best strategy, you should plan, for example, the best time of day to perform the migration.
2. Prepare for the migration.
   To prepare to migrate the source database (the primary database) to the target environment (the standby database), you must perform tasks such as ensuring

that the database that you want to migrate is running, installing the latest patches for both databases so that they are patched at the same level, and ensuring that the 1521 port is open between the primary database and the standby database.

> **Note:**
>
> Oracle recommends using the same database name for both databases so that applications can automatically fall over to the new database.

3. Perform the migration.
   You can perform the database migration by configuring the primary database (the source database) and the standby database (the target database) for Oracle Data Guard, copying the TDE wallets from the primary database to the standby database, and then completing the standby database configuration.

Required Tools to Perform the Migration

Before you begin the database migration, in addition to knowledge of Oracle Cloud Infrastructure, you must have knowledge in several areas of Oracle Database tools.

The tools with which you must be familiar are as follows:

- SQL*Plus
- Oracle Data Guard
- Oracle Flashback Technology and `spfiles`
- Familiarity with using the `srvctl` and `dgmgrl` utilities
- Familiarity with editing the `tnsnames.ora`, `listener.ora`, `sqlnet.ora`, and `oraenv` files
- Familiarity with performing Oracle Data Guard switchover operations
- (Optionally) Familiarity with generating Oracle Automatic Workload Repository and Oracle Automatic Database Diagnostic Monitor reports
- (Optionally) Familiarity with Oracle Automatic Storage Management Cluster File System (Oracle ACFS)

About Required Services and Roles

This solution requires Oracle Cloud Infrastructure.

The roles that are described in this topic will be in the target database only if they were in the source database. Oracle Database creates these roles (as well as other roles) during the installation process to better enforce separation of duty.

The following table lists the roles that you will need to complete this Oracle Cloud Infrastructure solution.

Service Name: Role	Required to...
Administrator (`SYSDBA` and `SYSOPER` privileges)	Perform SYS-related administration tasks.

Service Name: Role	Required to...
Administrator (`SYSDBG` privilege)	Perform Oracle Data Guard tasks, if you are using Oracle Database 12c release 2 (12.1.0.2) or later.
Administrator (`SYSKM` privilege)	Perform Transparent Data Encryption tasks, if you are using Oracle Database 12c release 2 (12.1.0.2) or later.

Plan the Database Migration

To plan the single instance of DBCS migration, you must inventory the source environment and decide on the best migration strategy.

Inventory the Source Environment

Inventorying the environment ensures that you have the supported Oracle Database versions and configurations that are required for migration.

- Ensure that you have the supported versions and configurations. Here is the list of combinations that are supported by Oracle Data Guard for migration.

Database versions and configurations	Source (Primary)	Target (Standby)
Database Service	Oracle Database Classic Cloud Service (Standalone)	Oracle Cloud Infrastructure Virtual Machine 1 Node Database System
Database Version	– 12.1.0.2 – 12.2.0.1	– 12.1.0.2 – 12.2.0.1
Database Storage	Filesystem for : – 12.1.0.2 – 12.2.0.1	ASM for version 12.2.0.1 and version 12.1.0.2 databases

- Determine the size of the source database .

  You can view the source database size from the Oracle Database Classic Cloud Service service console. Identify the size of the OCPUs, Memory and Storage for the Oracle Database Classic Cloud Service instance. This information will enable you to identify the appropriate Oracle Cloud Infrastructure Virtual Machine Database shape which maps to the source database size. Virtual Machine Database system is available in fixed shapes. Ensure that the shape chosen for creating database should be able to accommodate the source database plus any future sizing requirements. A thumb rule is to use a shape similar or higher in size than your target database.

- Determine the workload level.
  You can generate an Oracle Automatic Workload Repository report to find a sample of the workload for the source database. Alternatively, if you have an Oracle Diagnostics Pack and Oracle Tuning Pack license, you can generate an Automatic Database Diagnostic Monitor report to find the source database performance over a period of time between specified snapshots. The time model statistics, operating system statistics, and wait events provide a relatively clear measure of the workload, in terms of the operating system capacity.

- Determine the environment variables that have been set in the source database. You may want to use these same settings in the target database.
- Check the database character set.
  You can find the database character set by issuing the following query:

  ```
  SELECT * FROM NLS_DATABASE_PARAMETERS;
  ```

  You will need to ensure that the target database will also have this character set.
- Determine the disaster recovery plan that is currently in place.

Decide on the Best Migration Strategy

After you inventory your environment, you should decide on the best migration strategy.

Consider the following before you begin the migration process:

- Take a backup of your primary database before starting the migration
- The best time of day to perform the migration
- Downtime requirements
- Database size
- Security considerations
- A strategy for large workloads

Prepare For Migration

To prepare for the migration of a single instance Oracle Database Classic Cloud Service to an Oracle Cloud Infrastructure server, you must perform multiple preparatory tasks before migration can start.

Ensure That the Database to be Migrated Is Running

Before you begin the migration process, you must check that the source database (the primary database) to be migrated is running.

1. Use SSH to sign in to the server where the source database (the primary database) to be migrated is located.
2. Sign in as the database software owner `oracle`.

   ```
   sudo su
   su - oracle
   ```

3. Go to the `$ORACLE_HOME` location.

   ```
   cd $ORACLE_HOME
   ```

   If the `$ORACLE_HOME` location has not been set, then use the `oraenv` script (located in the `/usr/local/bin` directory) to set the environment, including `$PATH`, so the `lsnrctl` and `sqlplus` commands can resolve without using full path names.

4. Check the listener status.

   ```
   lsnrctl status
   ```

5. If the listener is not running (for example, the output has error `TNS-12541: TNS:no listener`), then start the listener.

   ```
   lsnrctl start
   ```

6. Check that the database is running.

   ```
   sqlplus / as sysdba
   ```

   This command should connect you to the database instance and the `SQL>` prompt should appear.

7. Check if the database is running in Read Write Mode

   ```
   SELECT NAME, OPEN_MODE FROM V$DATABASE;
   ```

   Output similar to the following appears:

   ```
   NAME              OPEN_MODE
   --------------    ---------
   source_db_name    READ WRITE
   ```

8. Exit the SQL*Plus

   ```
   EXIT
   ```

Ensure That All Database Components on the Source Database Are Installed on the Target Database

You can find the components that are installed on the source database (the primary database) by querying the `DBA_REGISTRY` data dictionary view.

1. Use SSH to sign in to the source database server.
2. Sign in to SQL*Plus as an administrator user.

   For example:

   ```
   sqlplus sys / as sysdba
   Password: password
   ```

3. Make a note of the version and edition of the software that is displayed in the opening banner
4. Exit SQL*Plus.

   ```
   EXIT
   ```

5. Use the `opatch` inventory command to find the latest patch set that has been applied.

   For example:

   ```
   $ORACLE_HOME/OPatch/opatch lsinventory
   ```

6. Repeat these steps on the target database (the standby database).

   The target database should have the same database version, and **same or later** bundle patch/PSU version installed than primary database.

Create a Standby Database for the Oracle Cloud Infrastructure System

You must create a standby database (the target database) on the Oracle Cloud Infrastructure Virtual Machine Database system, in addition to the database that is currently on this system. An Oracle database on Virtual Machine Database system can accommodate multiple databases on a single host. The process for creating this database creates a starter database during provisioning. Create the database system with the host name, shape, and CPU count that your site requires. The steps below provide the detailed process for creating the Oracle Cloud Infrastructure Virtual Machine Database system.

If you want to learn more about the Virtual Machine Database systems, you can refer to Oracle Cloud Infrastructure documentation.

Generate SSH Key Pair

To gain local access to the tools, utilities and other resources on Oracle Cloud Infrastructure Virtual Machine Database system, you use Secure Shell (SSH) client software to establish a secure connection and log in as the user `oracle` or the user `opc`. To access the standby Virtual Machine Database system, using SSH, you have to use SSH key pair instead of a password to authenticate a remote user. A key pair consists of a private key and public key. You keep the private key on your computer and provide the public key every time you launch an instance. To create key pairs, you can use a third-party tool such as OpenSSH on UNIX-style systems (including Linux, Solaris, BSD, and OS X) or PuTTY Key Generator on Windows.

Create Virtual Cloud Network

When you work with Oracle Cloud Infrastructure, one of the first steps is to set up a virtual cloud network (VCN) for your cloud resources. Ensure that you have set up a VCN before creating a standby database. You can refer to Oracle Cloud Infrastructure documentation for more information on how to create a VCN.

Verify the Virtual Machine Database Shapes Supported by your Tenancy

When you sign up for Oracle Cloud Infrastructure, a set of service limits are configured for your tenancy. The service limit is the quota or allowance set on a resource. For example, your tenancy is allowed a maximum number of compute instances per availability domain. These limits are generally established with your Oracle account representative when you purchase Oracle Cloud Infrastructure.

When you create a standby Virtual Machine Database system, you have to ensure that the Virtual Machine database shape that you select, should closely map to the primary(source) instance. You also MUST ensure that the selected shape is supported by your tenancy.

Verify your tenancy limits and usage (by region):

> **Note:**
>
> If a given resource type has limits per availability domain, the limit and usage for each availability domain is displayed.

1. Open the Oracle Cloud Services Dashboard. Open the **User** menu and click Tenancy: <your_tenancy_name>.
2. Click **Service Limits** on the left side of the page.
   Your resource limits and usage for the specific region are displayed, broken out by service.
3. Click on **Database**, and verify the Virtual Machine database shapes supported by your tenancy.

Your selection of the standby database shape should be a combination of shape that closely maps to primary(source) instance shape along with supported database shapes in your tenancy. Virtual Machine Database system is available in fixed data sizing shapes. Ensure that the shape chosen for creating database should be able to accommodate the source database plus any future sizing requirements. A thumb rule is to use a shape similar or higher in size than source database.

Create Standby Virtual Machine Database System

1. Login to your Oracle Cloud Services Dashboard
2. Open the navigation menu. Under Services, click **Database** (NOT Database Classic).
3. Choose your **Compartment**.
4. Click **Launch DB System**.
5. In the Launch DB System dialog, enter the following:
   a. **DB System Information**
      - **Compartment**: By default, the DB system launches in your current compartment and you can use the network resources in that compartment. Click the **click here** link in the dialog box if you want to enable compartment selection for the DB system, network, and subnet resources.
      - **Display Name**: A friendly, display name for the DB system. The name doesn't need to be unique. An Oracle Cloud Identifier (OCID) will uniquely identify the DB system.
      - **Availability Domain**: The **availability domain** in which the DB system resides.
      - **Shape Type**: Select **Virtual Machine**

- **Shape**: The shape to use to launch the DB system. The shape determines the type of DB system and the resources allocated to the system. Choose the VM DB Shape that you identified from the previous section.
- **Total Node Count**: The number of nodes in the DB system. The number depends on the shape you select. You can specify 1 or 2 nodes for virtual machine DB systems, except for VM.Standard2.1 and VM.Standard1.1, which are single-node DB systems.
- **Oracle Database Software Edition**: The database edition supported by the DB system. You can mix supported database versions on the DB system, but not editions. (The database edition cannot be changed and applies to all the databases in this DB system). Choose a database edition which is same or higher than the primary database.
- **Available Storage Size (GB)**: Enter a size with at least the same size as your primary (source) server.
- **License Type**: The type of license you want to use for the DB system. Your choice affects metering for billing.
  - **License included** means the cost of the cloud service includes a license for the Database service.
  - **Bring Your Own License (BYOL)** means you are an Oracle Database customer with an Unlimited License Agreement or Non-Unlimited License Agreement and want to use your license with Oracle Cloud Infrastructure. This removes the need for separate on-premises licenses and cloud licenses.
- **SSH Public Key**: The public key portion of the key pair you want to use for SSH access to the DB system.Use the public key that you generated in the previous section.

b. **Network Information**

- **Virtual Cloud Network Compartment**: The compartment containing the network in which to launch the DB system.
- **Virtual Cloud Network:** The VCN in which to launch the DB system. Select the VCN that you created in the previous section.
- **Subnet Compartment**: The compartment containing a subnet within the cloud network to attach the DB system to.
- **Client Subnet:** The subnet to which the DB system should attach.
- **Hostname Prefix**: Your choice of host name for the DB system. The host name must begin with an alphabetic character, and can contain only alphanumeric characters and hyphens (-).
  - The maximum number of characters allowed is 30. The host name must be unique within the subnet. If it is not unique, the DB system will fail to provision.
- **Host Domain Name**: The domain name for the DB system. If the selected subnet uses the Oracle-provided Internet and VCN Resolver for DNS name resolution, this field displays the domain name for the subnet and it can't be changed. Otherwise, you can provide your choice of a domain name. Hyphens (-) are not permitted.

- **Host and Domain URL**: Combines the host and domain names to display the fully qualified domain name (FQDN) for the database. The maximum length is 64 characters.

c. **Database Information**

- **Database Name**: The name for the database. The database name must begin with an alphabetic character and can contain a maximum of eight alphanumeric characters. Special characters are not permitted.
- **Database Version**: The version of the initial database created on the DB system when it is launched. After the DB system is active, you can create additional databases on it. You can mix database versions on the DB system, but not editions.
- **PDB Name**: The name of the pluggable database. The PDB name must begin with an alphabetic character, and can contain a maximum of 8 alphanumeric characters. The only special character permitted is the underscore (_).
- **Database Admin Password**: A strong password for SYS, SYSTEM, TDE wallet, and PDB Admin. The password must be 9 to 30 characters and contain at least 2 uppercase, 2 lowercase, 2 numeric, and 2 special characters. The special characters must be _, #, or -. The password must not contain the username (SYS, SYSTEM, and so on) or the word "oracle" either in forward or reversed order and regardless of casing.
- **Confirm Database Admin Password**: Re-enter the Database Admin Password you specified.
- **Automatic Backup**: Check the check box to enable automatic incremental backups for this database.
- **Database Workload**: Select the workload type that best suits your application.
  - Online Transactional Processing (OLTP) configures the database for a transactional workload, with a bias towards high volumes of random data access.
  - Decision Support System (DSS) configures the database for a decision support or data warehouse workload, with a bias towards large data scanning operations.
- **Character Set**: The character set for the database. The default is AL32UTF8.
- **National Character Set**: The national character set for the database. The default is AL16UTF16.
- **Tags**: Optionally, you can apply tags. If you have permissions to create a resource, you also have permissions to apply free-form tags to that resource. To apply a defined tag, you must have permissions to use the tag namespace. For more information about tagging, see Resource Tags. If you are not sure if you should apply tags, skip this option (you can apply tags later) or ask your administrator.

6. Click **Launch DB System**. The DB system appears in the list with a status of Provisioning. The DB system's icon changes from yellow to green (or red to indicate errors).
7. Wait for the DB system's icon to turn green, with a status of Available, and then click the highlighted DB system name. Details about the DB system are displayed.

8. Note the IP addresses; you'll need the private or public IP address, depending on network configuration, to connect to the DB system.
9. Check the SYS password on the primary (source) database. If it does not meet the Oracle Cloud Infrastructure password requirements, then change it to match the password that you created for the standby database (the target database).

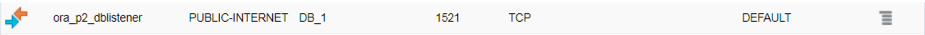
Ensure That Port 1521 Is Open Between the Primary Database and the Standby Database

You must ensure that the primary database (the source database) and the standby database (the target database) can connect to each other through port 1521.

Enable Communication from Standby Database to Primary Database

On primary database (the source database), open port 22 and 1521 for ingress traffic from standby database (the target database).

1. On Oracle Database Service Console, enable **ora_p2_dblistener** rule to open ingress traffic on port 1521 from public internet.
   a. Login to your **Oracle Cloud Services Dashboard**
   b. Open the navigation menu. Under Services, click **Database Classic.**
   c. Click ☰ next to your DBCS deployment, select **Access Rules** from the menu displayed.
   d. Enable rule **ora_p2_dblistener** to open ingress traffic on port 1521



ora_p2_dblistener	PUBLIC-INTERNET	DB_1	1521	TCP	DEFAULT	☰

2. From the standby database SSH to primary database (the source database) on port 22. Port 22 on primary database is open by default from public internet.

```
[opc@<standby_db_name> ~]$ ssh -i <private key> opc@<Primary DB IP>
[opc@<primary_db_name> ~]$
```

   Exit from the primary database base.

```
[opc@<primary_db_name> ~]$ exit
[opc@<standby_db_name> ~]$
```

3. On the standby database change the user to *root*:

```
sudo su
```

4. Set the TCP Socket size.

```
sysctl -w net.core.rmem_max=10485760
sysctl -w net.core.wmem_max=10485760
```

5. On the standby database, switch back to user *oracle*. Add the TNS entry for primary database to `$ORACLE_HOME/network/admin/tnsnames.ora`.

   ```
   <Primary DB Name>=
     (DESCRIPTION =
   (ADDRESS = (PROTOCOL = TCP)(HOST = <Primary DB Public IP>)(PORT =
   1521))
   (CONNECT_DATA =
   (SERVER = DEDICATED)
   (SERVICE_NAME = <Primary DB Service Name>)
    )
   )
   ```

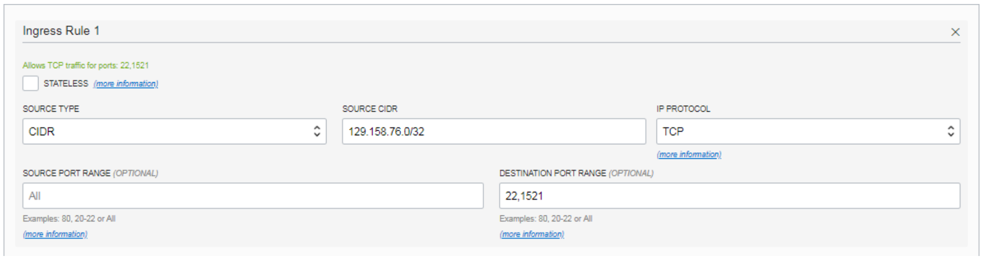
6. Test `sqlplus` connection from standby to primary database.

   ```
   sqlplus sys/<password>@<Primary DB Name> as sysdba

   SQL*Plus: Release 12.1.0.2.0 Production on Tue Feb 20 02:18:35 2018
   Copyright (c) 1982, 2014, Oracle.  All rights reserved.

   Connected to:
   Oracle Database 12c EE Extreme Perf Release 12.1.0.2.0 - 64bit
   Production
   With the Partitioning, Oracle Label Security, OLAP, Advanced Analytics
   and Real Application Testing options
   SQL>
   ```

Enable Communication from Primary Database to Standby Database System

On the standby database, open port 22 and 1521 for ingress traffic from primary database.

1. Note down the public IP of primary database instance from the console.
2. Login to Oracle Cloud Infrastructure Service console.
3. Open the Oracle Cloud Infrastructure Navigation menu. Under **Core Infrastructure**, go to **Networking** and click **Virtual Cloud Networks**.
4. Click the cloud network that you have used to create yourOracle Cloud Infrastructure Virtual Machine Database System.
5. Click **Security Lists**.
6. Click the security list you're interested in. It is generally the **Default Security List**.
7. Click **Edit All Rules**.
8. Click **+ Another Ingress Rule**. Add primary databse IP to **Source CIDR** . Update the **Destination Port Range** to 22,1521.



Ingress Rule 1

Allows TCP traffic for ports: 22,1521

□ STATELESS *(more information)*

SOURCE TYPE: CIDR

SOURCE CIDR: 129.158.76.0/32

IP PROTOCOL: TCP *(more information)*

SOURCE PORT RANGE *(OPTIONAL)*: All

Examples: 80, 20-22 or All *(more information)*

DESTINATION PORT RANGE *(OPTIONAL)*: 22,1521

Examples: 80, 20-22 or All *(more information)*

9. Click **Save Security List Rules**
10. From the primary database, test SSH on port 22 to standby database System.

    ```
    ssh -i <private key> opc@<standby DB IP>
    [opc@<standby DB IP> ~]$
    ```

    Exit from the primary database base.

    ```
    [opc@<primary_db_name> ~]$ exit
    [opc@<standby_db_name> ~]$
    ```

11. On the standby database change the user to *root*:

    ```
    sudo su
    ```

12. Set TCP Socket size

    ```
    sysctl -w net.core.rmem_max=10485760
    sysctl -w net.core.wmem_max=10485760
    ```

13. Add TNS entry for standby database to `$ORACLE_HOME/network/admin/tnsnames.ora`. Replace standby database server name with the public IP in TNS entry.

    ```
    <Standby DB Name>=
    (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = <Standby DB Public IP>)(PORT =
    1521))
    (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = <Standby DB Service Name>)
    )
    )
    ```

14. Use the utility tnsping to check the primary-standby DB connection.

    ```
    [oracle@<primary_db_name>]$ tnsping <Standby tns entry>
    TNS Ping Utility forLinux: Version 12.1.0.2.0- Production on 20-
    FEB-201802:23:01
    Copyright (c) 1997, 2014, Oracle.  All rights reserved.
    Used parameter files:
    /u01/app/oracle/product/12.1.0/dbhome_1/network/admin/sqlnet.ora

    Used TNSNAMES adapter to resolve the alias
    ```

```
Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST =
<Standby IP>)(PORT
= 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = <Primary
DB name>)))
OK (20msec)
```

15. Test `sqlplus` connection from primary to standby database on port 1521.

    ```
    sqlplus sys/<password>@<Standby DB Name> as sysdba
    SQL*Plus: Release 12.1.0.2.0Production on Tue Feb 2002:23:112018
    Copyright (c) 1982, 2014, Oracle.  All rights reserved.

    Connected to:
    Oracle Database 12c EE Extreme Perf Release 12.1.0.2.0- 64bit Production
    With the Partitioning, Real Application Clusters, Automatic Storage
    Management, Oracle Label Security,
    OLAP, Advanced Analytics and Real Application Testing options
    SQL>
    ```

Ensure That Bundle Patches Have Been Applied and Are in Sync

You must ensure that the patch level on the primary database is earlier or the same as the patch level on the standby database.

The patches on the Oracle Database Classic Cloud Service instance (on Oracle Cloud Infrastructure Compute Classic) must be the same as the Oracle Cloud Infrastructure Virtual Machine Database System, and must be manually applied. Look out for one-off patches that have been applied to the primary database, and if there is a need to apply them to the standby database as well.

1. Use SSH to sign in to the primary database server.

   ```
   sudo su
   su - oracle
   ```

2. List the patch level on the primary database as follows:
   a. To find a brief listing of patches:

      ```
      $ORACLE_HOME/OPatch/opatch lspatches
      ```

   b. To find a detailed listing of patches:

      ```
      $ORACLE_HOME/OPatch/opatch lsinventory
      ```

3. Make a note of the patch level.
4. Use SSH to sign in to the server where the standby database is located.
5. Check the patch level on the standby database by running the opatch lsinventory command.
6. Compare the patch level on the primary database with the patch level on the standby database. Ensure that the standby system has a bundle patch that is either equal to or later than the bundle patch that is on the primary database. If you find that you must install a patch on the Oracle Database Classic Cloud

Service primary database (the source database) instance, then ensure that the patch is not later than the patch on the Oracle Cloud Infrastructure standby database (the target database) instance.

7. Ensure that the bundle patches that have been applied include the fix for Bug **18633374** (My Oracle Support Note - Doc ID 1918906.1). This patch must be applied to both primary and standby database.

   a. Download the patch from My Oracle Support Note - Doc ID 1918906.1 on your local system

   b. Transfer the patch to the database server on which the patch needs to be applied.

      ```
      ls -rlt p18633374_12102171017_Linux-x86-64.zip
      unzip p18633374_12102171017_Linux-x86-64.zip
      ```

   c. Check for conflicts

      ```
      cd 18633374/
      $ORACLE_HOME/OPatch/opatch prereq CheckConflictAgainstOHWithDetail -
      ph
          ./
      ```

      Given below is the **sample output** of the above command, which shows that patches 28210208 and 27351628

      ```
      Conflict with 28210208
      Conflict details:
       /u01/app/oracle/product/12.1.0/dbhome_1/lib/libserver12.a:krb.o
       /u01/app/oracle/product/12.1.0/dbhome_1/lib/libserver12.a:krbi.o

      Conflict with 27351628
      Conflict details:
        /u01/app/oracle/product/12.1.0/dbhome_1/lib/libserver12.a:krbb.o
      ```

   d. Shutdown the database

      ```
      sqlplus / as sysdba
      SQL> shut immediate
       Database closed.
       Database dismounted.
       ORACLE instance shut down.
       SQL> exit
       Disconnected from Oracle Database 12c EE Extreme Perf Release
      12.1.0.2.0 - 64bit Production
       With the Partitioning, Oracle Label Security, OLAP, Advanced
      Analytics
       and Real Application Testing options
      ```

   e. Stop the Listener

      ```
      $ lsnrctl stop
      LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 19-FEB-2018
      13:02:25
      ```

```
Copyright (c) 1991, 2014, Oracle. All rights reserved.
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<Database
host name>)(PORT=1521)))
 The command completed successfully
```

f. Rollback the conflicting patches

Execute this command for all the conflicting patches:

```
$ORACLE_HOME/OPatch/opatch rollback -id <patch_id>
```

Sample Output

```
Oracle Interim Patch Installer version 12.2.0.1.9
 Copyright (c) 2018, Oracle Corporation. All rights reserved.

 Oracle Home : /u01/app/oracle/product/12.1.0/dbhome_1
 Central Inventory : /u01/app/oraInventory
 from : /u01/app/oracle/product/12.1.0/dbhome_1/oraInst.loc
 OPatch version : 12.2.0.1.9
 OUI version : 12.1.0.2.0
 Log file location : /u01/app/oracle/product/12.1.0/dbhome_1/
cfgtoollogs/opatch/opatch2018-02-19_13-02-42PM_1.log

 Patches will be rolled back in the following order:
 24401351
 The following patch(es) will be rolled back: 24401351
Please shutdown Oracle instances running out of this ORACLE_HOME on
the local system.
 (Oracle Home = '/u01/app/oracle/product/12.1.0/dbhome_1')

 Is the local system ready for patching? [y|n]
 y
 User Responded with: Y
Rolling back patch 24401351...
RollbackSession rolling back interim patch '24401351' from OH
'/u01/app/oracle/product/12.1.0/dbhome_1'
Patching component oracle.rdbms, 12.1.0.2.0...
 RollbackSession removing interim patch '24401351' from inventory
 Log file location: /u01/app/oracle/product/12.1.0/dbhome_1/
cfgtoollogs/opatch/opatch2018-02-19_13-02-42PM_1.log
OPatch succeeded.
```

g. Apply patch

```
pwd
/home/oracle/18633374
[oracle@<database host> 18633374]$ $ORACLE_HOME/OPatch/opatch apply
```

h. Restart the database

```
sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Mon Feb 19 13:04:48 2018
Copyright (c) 1982, 2014, Oracle. All rights reserved.
```

```
startupConnected to an idle instance.
SQL> startup
```

i. Start the listener

```
$ lsnrctl start
```

> **Note:**
>
> Ensure that once the migration is complete, you have to rollback patch 18633374, and re-apply the conflicting patches.

8. If you must install a later patch on the standby database (the target database), then access My Oracle Support.
9. Download the correct version of the patch to the standby database (the target database).
10. Extract the bundle patch.
11. List the available patches.

    ```
    $ORACLE_HOME/OPatch/opatch lspatches
    ```

12. Apply the patch.

    ```
    $ORACLE_HOME/OPatch/opatch apply patch_number
    ```

Compare Timezone Levels

The timezone of both the primary and standby database systems should be same. In case there is a difference, corresponding timezone patch to be applied. You can check for the timezone on both the systems, using the following command:

```
sudo su
su - oracle
sqlplus / as sysdba

SQL> select * from v$timezone_file;
FILENAME                VERSION     CON_ID
-------------------- ---------- ----------
timezlrg_28.dat              28          0
```

The output of the above command must be same on both primary and standby databases.

Migrate the Database

To perform the migration of a single instance Oracle Database Classic Cloud Service server to an Oracle Cloud Infrastructure Virtual Machine database system, you can use Oracle Data Guard. You must configure the database on Oracle Cloud Infrastructure Compute Classic as the primary database (the source database), which you migrate to a standby database (the target database) on Oracle Cloud Infrastructure on Virtual Machine Database systems.

Configure the Primary (Source) Database

To configure the primary database (the source database), you configure Oracle Data Guard and modify the listener.ora and tnsnames.ora files for the standby database (the target database).

Configure the Primary Database for the Standby Database

In this configuration, you configure the primary (source) database to use Oracle Data Guard.

1. Use SSH to sign in to the primary (source) database server.

   ```
   sudo su
   su - oracle
   ```

2. Sign in to the database instance as a user who has administrator privileges. For example:

   ```
   sqlplus / as sysoper
   ```

3. Ensure that the database is in ARCHIVELOG mode

   ```
   SQL> archive log list
   Database log mode Archive Mode
   Automatic archival Enabled
   Archive destination USE_DB_RECOVERY_FILE_DEST
   Oldest online log sequence 9
   Next log sequence to archive 11
   Current log sequence 11
   SQL>
   ```

   In the output above, notice that the value of Database log mode is set to **Archive Mode**, and the value of Automatic archival is **Enabled**.

4. If the output for Database log mode is **No Archive Mode** and the output for Automatic archival is **Disabled**, then do the following:

   a. Shut down the database.

      ```
      SHUTDOWN IMMEDIATE
      ```

   b. Restart the database in mount mode

      ```
      STARTUP MOUNT
      ```

   c. Enable archive log mode.

      ```
      ALTER DATABASE ARCHIVELOG;
      ```

   d. Ensure that the database is now in archive log mode.

   ```
   ARCHIVE LOG LIST
   ```

   e. The output for the Database log mode should be **Archive Mode** and the output for Automatic archival is **Enabled**.

   f. Open the database.

   ```
   ALTER DATABASE OPEN;
   ```

5. Connect with the SYSDBA administrator privilege.

   ```
   CONNECT / AS SYSDBA
   ```

6. For a multitenant environment, do the following:

   a. Check the status of the PDBS.

   ```
   SHOW PDBS
   ```

   b. If the PDBS are not open, then open them.

   ```
   ALTER PLUGGABLE DATABASE ALL OPEN;
   ```

7. Ensure that the database is in force logging mode. For example:

   ```
   SELECT NAME, OPEN_MODE, FORCE_LOGGING FROM V$DATABASE;
   ```

8. If necessary, enable force logging.

   ```
   ALTER DATABASE FORCE LOGGING;
   ```

9. Check the configuration.

   ```
   SELECT NAME, CDB, OPEN_MODE, FORCE_LOGGING FROM V$DATABASE;
   ```

   The FORCE_LOGGING column should be YES.

10. Use the SHOW PARAMETER command to check the following database parameters:

    a. DB_NAME : It is recommended to use the same name as the target database.

    b. DB_UNIQUE_NAME: Ensure that this name is different from the name used on the target database.

    c. REMOTE_LOGIN_PASSWORD_FILE: This parameter must be set to EXCLUSIVE.

11. Ensure that the Flashback in ON. If its not ON, use sql command ALTER DATABASE FLASHBACK ON;

    ```
    select flashback_on from v$database;
    FLASHBACK_ON
    ------------------
    YES
    ```

```
show parameter flashback_retention_target
NAME                                 TYPE        VALUE
------------------------------------ -----------
------------------------------
db_flashback_retention_target        integer     1440
SQL>
```

Add Static Services to the Primary Database `listener.ora` File

You must add a new static listener to the primary (source) database `listener.ora` file and restart the listener.

1. Use SSH to sign in to the primary (source) database server.
2. At the command line, connect as `root`.

   ```
   sudo su -
   ```

3. Sign in as the database software owner `oracle`.

   ```
   su - oracle
   ```

4. Modify the `$ORACLE_HOME/network/admin/listener.ora` file to include the static listener. The following example shows the format to use for one static listener:

   ```
   SID_LIST_LISTENER=
     (SID_LIST=
       (SID_DESC=
       (SDU=65535)
       (GLOBAL_DBNAME = <primary_db_unique_name>.<primary_db_domain>)
       (SID_NAME = <source_db_name>)
       (ORACLE_HOME=<oracle_home_directory>)
       (ENVS="TNS_ADMIN=<oracle_home_directory>/network/admin")
       )
       (SID_DESC=
       (SDU=65535)
       (GLOBAL_DBNAME =
   <primary_db_unique_name>_DGMGRL.<primary_db_domain>)
       (SID_NAME = <source_db_name>)
       (ORACLE_HOME=<oracle_home_directory>)
       (ENVS="TNS_ADMIN=<oracle_home_directory>/network/admin")
       )
     )
   ```

5. Stop the listener.

   ```
   lsnrctl stop listener
   ```

6. Restart the listener.

   ```
   lsnrctl start listener
   ```

7. Check the listener status.

   ```
   lsnrctl status
   ```

Configure the Primary Database Parameters

After you configure the primary (source) database and add static services to the primary database `listener.ora` file, you can configure the Oracle Data Guard parameters on the primary database.

> **Note:**
>
> Ensure that the source database is in ARCHIVELOG MODE with FLASHBACK enabled by default. It is recommended to have DB_BLOCK_CHECKSUM=FULL. If they are any performance issues then switch to DB_BLOCK_CHECKING=MEDIUM

1. Use SSH to sign in to the primary database (the source database) server.
2. Sign in to the database instance as a user who has the `SYSDBA` administrator privilege.

   ```
   sqlplus / as sysdba
   Enter password: password
   ```

3. Enable automatic standby file management.

   ```
   ALTER SYSTEM SET STANDBY_FILE_MANAGEMENT=AUTO SID='*' SCOPE=SPFILE;
   ```

4. Set the archive lag target.

   ```
   ALTER SYSTEM SET ARCHIVE_LAG_TARGET=1800 SID='*' SCOPE=SPFILE;
   ```

5. Identify the Oracle Broker configuration file names and locations. The following statements depend on the type of database storage.

   ```
   ALTER SYSTEM SET DG_BROKER_CONFIG_FILE1='/u02/app/oracle/oradata/
   <source_db_name>/dr1<source_db_name>.dat' SCOPE=BOTH;
   ALTER SYSTEM SET DG_BROKER_CONFIG_FILE2='/u03/app/oracle/
   fast_recovery_area/<source_db_name>/dr2<source_db_name>.dat' SCOPE=BOTH;
   ```

6. Enable the Oracle Broker DMON process for the database.

   ```
   ALTER SYSTEM SET DG_BROKER_START=TRUE SCOPE=BOTH;
   ```

7. Set the DB_BLOCK_CHECKING and DB_BLOCK_CHECKSUM parameters.

   ```
   ALTER SYSTEM SET DB_BLOCK_CHECKING=FULL SCOPE=BOTH;
   ALTER SYSTEM SET DB_BLOCK_CHECKSUM=FULL SCOPE=BOTH;
   ```

8. Set the log buffer to 256 megabytes.

   ```
   ALTER SYSTEM SET LOG_BUFFER=268435456 SCOPE=SPFILE;
   ```

9. Set the DB_LOST_WRITE_PROTECT parameter to TYPICAL.

   ```
   ALTER SYSTEM SET DB_LOST_WRITE_PROTECT=TYPICAL SCOPE=BOTH;
   ```

10. Enable the database flashback feature. The minimum recommended value for DB_FLASHBACK_RETENTION_TARGET is 120 minutes.

    ```
    ALTER DATABASE FLASHBACK ON;
    ALTER SYSTEM SET DB_FLASHBACK_RETENTION_TARGET=120;
    ALTER SYSTEM ARCHIVE LOG CURRENT;
    ```

11. Add the standby redo logs, based on the online redo log. You can use the query below to determine the number and size (in bytes) of the ORLs. The size of the standby redo logs must be the same as the online redo logs, but you must add one or more additional standby redo logs than there are online redo logs. In the following example, four online redo logs exist, so you must add at least five standby redo logs. In other words, you must specify the current redo logs plus at least one, and then use the same size for it as the original redo logs.

    a. Execute the following query to determine the number, and size in bytes, of the Oracle redo logs.

       ```
       SELECT GROUP#, BYTES FROM V$LOG;
       ```

       The output should be similar to the following.

       ```
       GROUP# BYTES
       ------ ----------
       1      1073741824
       2      1073741824
       3      1073741824
       ```

    b. Specify the current redo logs plus one more, and use the same size as the current redo logs. For example:

       ```
       alter database add standby logfile thread 1
       group 4('/u04/app/oracle/redo/stby_redo01.log') size 1073741824,
       group 5('/u04/app/oracle/redo/stby_redo02.log') size 1073741824,
       group 6('/u04/app/oracle/redo/stby_redo03.log') size 1073741824,
       group 7('/u04/app/oracle/redo/stby_redo04.log') size 1073741824;
       ```

    c. Verify that you created the correct number of standby redo logs.

       ```
       SELECT GROUP#, BYTES FROM V$STANDBY_LOG;
       ```

       Output similar to the following should appear:

       ```
       GROUP#      BYTES
       ---------- ----------
       ```

```
4    1073741824
5    1073741824
6    1073741824
7    1073741824
```

Configure the Standby (Target) Database

To configure the standby (target) database, you must modify the `oratab`, `listener.ora`, and `tnsnames.ora` files.

Drop Standby database

This step cleans up the initial database for creating a physical standby on Virtual Machine Database system.

> **Note:**
>
> Capture *db_unique_name* on standby database. It is mandatory to use same *db_unique_name* for standby database creation. The *db_unique_name* is case sensitive.

1. Use SSH to sign in to the standby database (the target database) server.
2. At the command line, connect as `root`.

   ```
   sudo su -
   ```

3. Sign in as the database software owner `oracle`.

   ```
   sudo su - oracle
   ```

4. Stop the database

   ```
   srvctl stop database -d <standby_db_unique_name>
   ```

5. Start the database in mount mode

   ```
   srvctl start database -d <standby_db_unique_name> -o mount
   ```

6. Login to the database as user `sysdba`

   ```
   sqlplus / as sysdba
   SQL*Plus: Release 12.1.0.2.0 Production on Sat Feb 17 18:21:20 2018
   Copyright (c) 1982, 2014, Oracle.  All rights reserved.
   Connected to:
   Oracle Database 12c EE Extreme Perf Release 12.1.0.2.0 - 64bit
   Production
   With the Partitioning, Real Application Clusters, Automatic Storage
   ```

```
Management, OLAP,
Advanced Analytics and Real Application Testing options
```

7. Drop the database

   ```
   alter system enable restricted session;
   System altered.
   drop database;
   ```

Add Static Services to the Standby Database listener.ora File

After you add static services to the standby database (the target database) `listener.ora` file, you must restart the listener. .

1. Use SSH to sign in to the standby (target) database server.
2. At the command line, connect as `root`.

   ```
   sudo su
   ```

3. Sign in as the database software owner `oracle`.

   ```
   su - oracle
   ```

4. Execute the `oraenv` script, which sets the `$ORACLE_HOME` environment variable.

   ```
   . oraenv
   ```

   Output similar to the following should appear:

   ```
   ORACLE_SID = [oracle] ? db_name
   The Oracle base has been set to /u01/app/oracle
   ```

5. SSH to the standby database system, log in as the `opc` or `root` user, and sudo to the *grid* user.

   ```
   sudo su - grid
   ```

6. Modify the `/u01/app/12.2.0.1/grid/network/admin/listener.ora` file to include the static listener. The first static listener shown below is required for Oracle Recovery Manager (Oracle RMAN) duplicate.

   ```
   SID_LIST_LISTENER=
     (SID_LIST=
       (SID_DESC=
       (SDU=65535)
       (GLOBAL_DBNAME = <standby db_unique_name>.<standby db_domain>)
       (SID_NAME = <standby oracle_sid>)
       (ORACLE_HOME=<oracle home directory>)
       (ENVS="TNS_ADMIN=<oracle home directory>/network/admin")
       )

     )
   ```

7. Use the `srvctl` utility to stop the listener.

   ```
   srvctl stop listener -l LISTENER
   ```

8. Restart the listener.

   ```
   srvctl start listener -l LISTENER
   ```

9. Check the listener status.

   ```
   lsnrctl status
   ```

   Sample Output

   ```
   LSNRCTL for Linux: Version 12.2.0.1.0 - Production on 20-FEB-2018
   02:45:31
   Copyright (c) 1991, 2016, Oracle.  All rights reserved.
   Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER)))
   STATUS of the LISTENER
   ------------------------
   Alias                     LISTENER
   Version                   TNSLSNR for Linux: Version 12.2.0.1.0 -
   Production
   Start Date                19-FEB-2018 12:14:06
   Uptime                    0 days 14 hr. 31 min. 24 sec
   Trace Level               off
   Security                  ON: Local OS Authentication
   SNMP                      OFF
   Listener Parameter File   /u01/app/12.2.0.1/grid/network/admin/
   listener.ora
   Listener Log File         /u01/app/grid/diag/tnslsnr/migtest/listener/
   alert/log.xml
   Listening Endpoints Summary...
     (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER)))
     (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=10.0.1.2)(PORT=1521)))
     (DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=<host name>)(PORT=5500))
   (Security=(my_wallet_directory=/u01/app/oracle/product/12.1.0.2/
   dbhome_1/admin/<Standby_db_name>/xdb_wallet))(Presentation=HTTP)
   (Session=RAW))
   Services Summary...
   Service "+APX" has 1 instance(s).
     Instance "+APX1", status READY, has 1 handler(s) for this service...
   Service "+ASM" has 1 instance(s).
     Instance "+ASM1", status READY, has 1 handler(s) for this service...
   Service "+ASM_DATA" has 1 instance(s).
     Instance "+ASM1", status READY, has 1 handler(s) for this service...
   Service "+ASM_RECO" has 1 instance(s).
     Instance "+ASM1", status READY, has 1 handler(s) for this service...
   Service "<StandbyDB>" has 1 instance(s).
     Instance "<standby_db_name>.<standby_db_domain>", status READY, has 1
   handler(s) for this service...
   Service "<standby_db_name>" has 1 instance(s).
     Instance "<standby_db_name>.<standby_db_domain>", status READY, has 1
   handler(s) for this service...
   ```

```
Service "<standby_db_name>.<standby_db_domain>" has 2 instance(s).
  Instance "MIGTEST", status UNKNOWN, has 1 handler(s) for this
service...
  Instance "<StandbyDB>", status READY, has 1 handler(s) for this
service...
Service "<StandbyDB>" has 1 instance(s).
  Instance "<StandbyDB>", status READY, has 1 handler(s) for this
service...
Service "<StandbyDB>" has 1 instance(s).
    Instance "<StandbyDB>", status READY, has 1 handler(s) for this
service...
The command completed successfully
```

> **Note:**
>
> In the above output, you may see the new listener with status UNKNOWN. This is an expected output.

Copy TDE Wallets from the Primary Database to the Standby Database

You can manually copy the TDE wallet files from the primary database (the source database) system to the standby database (the target database) system by using Secure Copy Protocol (SCP).

Compress the TDE Wallet

You must perform this operation in the primary database (the source database).

1. Use SSH to sign in to the primary database (the source database) server.
2. At the command line, connect as root.

   ```
   sudo su -
   ```

3. Sign in as the database software owner `oracle`.

   ```
   su - oracle
   ```

4. To find the wallet location, sign in to the primary database (the source database) instance with the `SYSDBA` administrator privilege.

   ```
   sqlplus / as sysdba
   ```

5. Query the `WRL_PARAMETER` column of the `V$ENCRYPTION_WALLET` dynamic view to find the directory where the wallet is located.

   ```
   SELECT * FROM V$ENCRYPTION_WALLET;
   ```

6. Exit SQL*Plus.

   ```
   exit
   ```

7. Go to the directory where the wallet files are located.For example:

   ```
   cd /u01/app/oracle/admin/source_db_unique_name
   ```

8. Use the tar command to compress the TDE wallet.For example:

   ```
   tar cvf tde_wallet.tar ./tde_wallet
   ```

   Output similar to the following appears:

   ```
   ./tde_wallet/
   ./tde_wallet/ewallet.p12
   ./tde_wallet/cwallet.sso
   ./tde_wallet/ewallet_2018021607225910.p12
   ```

Copy the TDE Wallet and Set Permissions on the Wallet Directory

After you back up the TDE wallet file, you must create a directory for the wallet and set permissions on this directory.

1. Copy the wallet tar file to a temp directory. For example:

   ```
   cp tde_wallet.tar /tmp/
   ```

2. Exit twice to become the OCP user.

   ```
   $ exit
   # exit
   ```

3. Use SCP to copy the wallet files from the primary database (the source database) to the standby database (the target database), in the `/opt/oracle/dcs/commonstore/wallets/tde/$ORACLE_UNQNAME` directory.For example:

   ```
   scp -i /home/opc/.ssh/privateKey /tmp/tde_wallet.tar opc@<Standby DB
   IP>:/tmp/
   ```

   Output similar to the following appears:

   ```
   tde_wallet.tar
   100% 20KB 20.0KB/s 00:00
   ```

4. Use SSH to sign in to the target database server and then sign in as database software owner `oracle`.
5. Go to the target wallet directory.For example:

   ```
   cd /opt/oracle/dcs/commonstore/wallets/tde
   ```

6. Check that the correct wallet is in this directory.

   ```
   ls <Standby DB Unique Name>
   ```

7. Back up the wallet file.For example:

   ```
   mv standby_db_unique_name standby_db_unique_name.old
   ```

8. Create a directory in which to store the wallet.For example:

   ```
   mkdir standby_db_unique_name
   ```

9. Check the permissions on the wallet directory.

   ```
   ls -ld standby_db_unique_name
   ```

10. If necessary, give the database software owner oracle read, write, and execute permissions.

    ```
    chmod 700 standby_db_unique_name
    ```

11. Check the permissions again.

    ```
    ls -ld standby_db_unique_name
    ```

12. Copy the wallet tar file to the current directory.

    ```
    cp /tmp/tde_wallet.tar .
    ```

13. Check the permissions.

    ```
    ls -rlt
    ```

    Output similar to the following appears:

    ```
    total 124
    drwx------ 2 oracle oinstall 20480 Feb 16 09:25
    standby_db_unique_name.old
    drwx------ 2 oracle oinstall 20480 Feb 16 10:16 standby_db_unique_name
    -rw-r--r-- 1 oracle oinstall 20480 Feb 16 10:17 tde_wallet.tar
    ```

Complete the TDE Wallet Process

You must extract the TDE wallet file tar and then move its contents to the wallet directory on the standby database (the target database).

1. On the standby database (the target database), ensure that you are in the correct wallet directory. For example:

   ```
   pwd

   # Output similar to the following should appear:
   /opt/oracle/dcs/commonstore/wallets/tde
   ```

2. Extract the tar file.

   ```
   tar xvf tde_wallet.tar
   ```

   Output similar to the following should appear:

   ```
   ./tde_wallet/ewallet.p12
   ./tde_wallet/ewallet_2018050819024979.p12
   ./tde_wallet/cwallet.sso
   ```

3. Move the `tde_wallet` contents to the wallet directory on the standby database (the target database).

   ```
   mv ./tde_wallet/* ./target_db_unique_name
   ```

4. Remove the `tde_wallet` contents from the standby database (the target database).

   ```
   rm -rf ./tde_wallet
   ```

Configure the Standby Initialization Parameter File and Start the Instance in NOMOUNT Mode

After you configure the standby initialization file, then you can restart the database in `NOMOUNT` mode.

1. Use SSH to sign in to the standby database (the target database) server.
2. Sign in as the database software owner `oracle`.

   ```
   su - oracle
   ```

3. Execute the `oraenv` script, which sets the `$ORACLE_HOME` environment variable.

   ```
   . oraenv
   ```

4. Go to the `dbs` directory.

   ```
   cd $ORACLE_HOME/dbs
   ```

5. Create a temporary initialization parameter file, named `init<Standby>.ora`. For example:

   ```
   echo "*.db_name='standby_db_name'" > $ORACLE_HOME/dbs/
   initstandby_db_name.ora
   echo "*.db_unique_name='standby_db_unique_name'" >> $ORACLE_HOME/dbs/
   initStandby_db_name.ora
   echo "*.db_domain='standby_db_domain'" >> $ORACLE_HOME/dbs/
   initStandby_db_name.ora
   ```

6. Back up the existing password file, if one exists. For example:

   ```
   mv $ORACLE_HOME/dbs/orapwtarget $ORACLE_HOME/dbs/orapwtarget.old
   ```

7. Create a new password file. For example:

   ```
   orapwd file=$ORACLE_HOME/dbs/orapwtarget
   password=admin_password_for_primary entries=5
   ```

8. Connect to the standby database (the target database) instance as a user who has the SYSOPER administrator privilege. For example:

   ```
   sqlplus / as sysoper
   ```

9. Shut down the database. For example:

   ```
   shutdown immediate [ If the Database is already stopped, this may throw
   error]
   ```

10. Restart the database in `NOMOUNT` mode using the `init_target.ora` initialization parameter file.

    ```
    startup force nomount PFILE=?/dbs/initStandby_db_name.ora
    ```

Duplicate the Target Database for the Standby from the Active Database

You can execute a script to duplicate the standby database (the target database). If the primary database (the source database) is large, then you can allocate additional channels to improve its performance. For a newly installed database, one channel typically runs the database duplication in a couple of minutes. Ensure that no errors occur after you run the Oracle Recovery Manager (Oracle RMAN) duplication operation. If errors occur, then restart the database by using the initialization parameter file (not spfile), in case it is generated under the $ORACLE_HOME/dbs directory as part of the Oracle RMAN duplication process.

1. Connect to the standby database (the target database) as the database software owner `oracle`.

   ```
   su - oracle
   ```

2. Create a script file `dup.rcv`, where you will copy the commands and fill in the environment specific variables specified in the later step.

   ```
   vi dup.rcv
   run {
   allocate channel prmy1 type disk;
   allocate channel prmy2 type disk;
   allocate channel prmy3 type disk;
   allocate channel prmy4 type disk;
   allocate auxiliary channel stby1 type disk;
   allocate auxiliary channel stby2 type disk;
   allocate auxiliary channel stby type disk;
   duplicate target database for standby from active database dorecover
   ```

```
spfile
parameter_value_convert '/u02/app/oracle/oradata/
<source_db_name>','+DATA'
set db_unique_name='<target_db_unique_name>'
set db_create_file_dest='+DATA'
set db_create_online_log_dest_1='+RECO'
set db_recovery_file_dest='+RECO'
set audit_file_dest='/u01/app/oracle/admin/db_name/adump'
set control_files='+DATA','+RECO'
set dg_broker_config_file1='+DATA/<target_db_unique_name>/
dr1<target_db_unique_name>.dat'
set dg_broker_config_file2='+RECO/<target_db_unique_name>/
dr2<target_db_unique_name>.dat'
set dispatchers='(PROTOCOL=TCP) (SERVICE=<target_db_name>XDB)'
set instance_name='<target_db_name>'
set db_domain='<target_db_domain>'
set db_recovery_file_dest='+RECO'
;
}
```

3. With the standby database (the target database) in `NOMOUNT` mode, connect to `Oracle RMAN`.

```
rman
RMAN> connect target sys@<primary_db_tnsnames_name>
target database Password: password
RMAN> connect auxiliary sys@<standby_db_tnsnames_name>
auxiliary database Password: password
```

4. Execute the following script to duplicate the target database for a standby database from an active database. The following example shows the dup.rcv script, which **must** be user-created, and is based on the My Oracle Support note `2369137`. In this example, the `dup.rcv` script has been customized to push duplication (image copies) from the file system to Oracle Automatic Storage Management (Oracle ASM). Other options such as from the file system to Oracle Automatic Storage Management Cluster File System, or Oracle ASM to Oracle ASM, would require changes to the file destination parameters and the file name conversion parameters.

   Modify the dup.rcv file with environment specific parameters.

```
@dup.rcv
RMAN> run {
2> allocate channel prmy1 type disk;
3> allocate channel prmy2 type disk;
4> allocate channel prmy3 type disk;
5> allocate channel prmy4 type disk;
6> allocate auxiliary channel stby1 type disk;
7> allocate auxiliary channel stby2 type disk;
8> allocate auxiliary channel stby type disk;
9> duplicate target database for standby from active database dorecover
10> spfile
11> parameter_value_convert '/u02/app/oracle/oradata/
<source_db_name>','+DATA'
12> set db_unique_name='<target_db_unique_name>'
13> set db_create_file_dest='+DATA'
14> set db_create_online_log_dest_1='+RECO'
15> set db_recovery_file_dest='+RECO'
16> set audit_file_dest='/u01/app/oracle/admin/db_name/adump'
17> set control_files='+DATA','+RECO'
18> set dg_broker_config_file1='+DATA/<target_db_unique_name>/
dr1<target_db_unique_name>.dat'
19> set dg_broker_config_file2='+RECO/<target_db_unique_name>/
dr2<target_db_unique_name>.dat'
20> set dispatchers='(PROTOCOL=TCP) (SERVICE=<target_db_name>XDB)'
21> set instance_name='<target_db_name>'
22> set db_domain='<target_db_domain>'
23> set db_recovery_file_dest='+RECO'
24> ;
25> }
```

Post Oracle Recovery Manager Duplication Steps

After you complete the Oracle Recovery Manager (Oracle RMAN) duplication operation, you should perform clean-up tasks.

Enable Oracle Flashback

You should enable Oracle Flashback.

1. Use SSH to sign in to the standby database (the target database) server.
2. Sign in as the database software owner `oracle`.

   ```
   sudo su - oracle
   ```

3. Sign in to the database instance as a user who has the `SYSOPER` administrator privilege. For example:

   ```
   sqlplus / as sysoper
   Enter password: password
   ```

4. Enable Oracle Flashback.

   ```
   ALTER DATABASE FLASHBACK ON;
   ```

5. Connect as a user with the `SYSDBA` administrator privilege.

   ```
   CONNECT / AS SYSDBA
   ```

6. Set the flashback retention target.

   ```
   ALTER SYSTEM SET DB_FLASHBACK_RETENTION_TARGET=120;
   ```

Move the `spfile` File to Oracle Automatic Storage Management

You should move the `spfile` file to Oracle Automatic Storage Management.

1. Use SSH to connect to the standby database (the target database) server.

2. Sign in to the database instance as a user who has the SYSOPER administrator privilege.
3. Create and move the `spfile` file to Oracle Automatic Storage Management.
   a. Create the pfile.

      ```
      create pfile='/tmp/init<target_db_name>.ora' from spfile;
      ```

   b. Shut down the database

      ```
      shutdown immediate
      ```

   c. Restart the database in `MOUNT` mode by using the `initdb_name.ora` file that you just created.

      ```
      startup mount pfile='/tmp/init<target_db_name>.ora';
      ```

   d. Create the `spfile` file.

      ```
      create spfile='+DATA' from pfile='/tmp/init<target_db_name>.ora';
      ```

4. Exit SQL*Plus.
5. As the `grid` user, find the `spfile` file on Oracle Automatic Storage Management by using the `asmcmd` command.

   ```
   asmcmd
   ASMCMD> cd +DATA/<target_db_unique_name>/PARAMETERFILE/
   ASMCMD> ls -lt
   ```

   Output similar to the following appears:

   ```
   Type           Redund  Striped  Time             Sys  Name
   PARAMETERFILE  UNPROT  COARSE   APR 09 16:00:00  Y    spfile.
   262.973010033
   ```

   Make a note of the ASM name (spfile.262.973010033), which you will need in the next task.

Change the init<target_db_name>.ora File to Reference the spfile File

You can modify the `init<target_db_name>.ora` file to reference the spfile file.

1. Use SSH to connect to the standby database (the target database) server.
2. Sign in as the database software owner `oracle`.

   ```
   su - oracle
   ```

3. Execute the oraenv script to set the `$ORACLE_HOME` environment variable.

   ```
   . oraenv
   ```

4. Go to the $ORACLE_HOME/dbs directory.

```
cd $ORACLE_HOME/dbs
```

5. Ensure that the init<target_db_name>.ora file is in this directory.

```
ls *.ora
```

6. Change the init<target_db_name>.ora file to refer to the spfile file.For example:

```
mv spfile<target_db_name>.ora spfile<target_db_name>.ora.stby
mv init<target_db_name>.ora init<target_db_name>.ora.stby
echo "SPFILE='+DATA/target_db_unique_name/PARAMETERFILE/spfile.
262.973010033'" > init<target_db_name>.ora
cat init<target_db_name>.ora --To check the file
```

   In this output, spfile.262.973010033 is the name of the file that you generated when you moved the spfile file to Oracle Automatic Storage Management in the previous task.

   Output similar to the following appears:

```
SPFILE='+DATA/<target_db_unique_name>/PARAMETERFILE/spfile.
262.973010033'
```

Modify and Start the Standby Database in MOUNT Mode

You can use the srvctl to modify and start the standby database (the target database).

1. Use SSH to sign in to the standby database (the target database) server.
2. Sign in as the database software owner oracle.

```
su - oracle
```

3. Use srvctl to modify and start the standby database (the target database) in MOUNT mode. For example:

```
srvctl modify database -db <target_db_unique_name> -role
PHYSICAL_STANDBY -s "READ ONLY"  -spfile +DATA/target_db_unique_name/
PARAMETERFILE/spfile.262.973010033

srvctl config database -db <target_db_unique_name>
```

4. Sign in to the database instance as a user who has the SYSOPER administrator privilege.

```
sqlplus / as sysoper
```

5. Shut down the database, and then exit SQL*Plus.

   ```
   SHUTDOWN IMMEDIATE
   EXIT
   ```

6. Start the database in MOUNT mode by using `srvctl`.

   ```
   srvctl start database -db <target_db_unique_name> -o mount
   ```

7. Sign in to the database instance as a user who has the `SYSDBA` administrator privilege.

   ```
   sqlplus / as sysdba
   ```

8. Query the `V$DATABASE` dynamic view to ensure that the database is in `MOUNT` mode.

   ```
   SELECT NAME, OPEN_MODE FROM V$DATABASE;

   #Output similar to the following appears:

   NAME              OPEN_MODE
   --------------    ---------
   source_db_name    MOUNTED
   ```

Set the Database and Log File Name Conversion Parameters on the Primary Database

You must set the conversion parameters for the database and the log file name on the primary database (the source database).

1. Use SSH to sign in to the primary database (the source database) server.
2. Sign in as the database software owner `oracle`.

   ```
   sudo su - oracle
   ```

3. Sign in to the database instance as a user who has the SYSDBA administrator privilege. For example:

   ```
   sqlplus / as sysdba
   Enter password: password
   ```

4. Check the `CONVERT` parameter.

   ```
   SHOW PARAMETER CONVERT#Output similar to the following
   appears:NAME                                 TYPE        VALUE
   ------------------------------------ ----------- ------
   db_file_name_convert                 string
   log_file_name_convert                string
   pdb_file_name_convert                string
   ```

   The `VALUE` column should be empty (null). If there is a value, then make a note of this value for after the migration is complete. After the migration is complete, these values are set to null.

5. Check the names of the data files.

   ```
   SELECT NAME FROM V$DATAFILE;
   ```

   Output similar to the following appears:

   ```
   NAME
   -----------------------------------------------------------------------
   /u02/app/oracle/oradata/SOURCE_DB_NAME/system01.dbf
   /u02/app/oracle/oradata/SOURCE_DB_NAME/sysaux01.dbf
   /u02/app/oracle/oradata/SOURCE_DB_NAME/undotbs01.dbf
   /u02/app/oracle/oradata/SOURCE_DB_NAME/pdbseed/system01.dbf
   /u02/app/oracle/oradata/SOURCE_DB_NAME/users01.dbf
   /u02/app/oracle/oradata/SOURCE_DB_NAME/pdbseed/sysaux01.dbf
   /u02/app/oracle/oradata/SOURCE_DB_NAME/pdb1/system01.dbf
   /u02/app/oracle/oradata/SOURCE_DB_NAME/pdb1/sysaux01.dbf
   /u02/app/oracle/oradata/SOURCE_DB_NAME/pdb1/SAMPLE_SCHEMA_users01.dbf
   /u02/app/oracle/oradata/SOURCE_DB_NAME/pdb1/example01.dbf
   ```

   Note down the path in the output above.

6. Check the `V$LOGFILE` dynamic view.

   ```
   SELECT MEMBER FROM V$LOGFILE;Output similar to the following
   appears:MEMBER
   -------------------------------------------------------------------
   /u04/app/oracle/oradata/redo/redo03.log
   /u04/app/oracle/oradata/redo/redo02.log
   /u04/app/oracle/oradata/redo/redo01.log
   /u03/app/oracle/fast_recovery_area/SOURCE_DB_NAME/onlinelog/
   o1_mf_4_fddlmffq_.log
   /u03/app/oracle/fast_recovery_area/SOURCE_DB_NAME/onlinelog/
   o1_mf_5_fddlvjo1_.log
   /u03/app/oracle/fast_recovery_area/SOURCE_DB_NAME/onlinelog/
   o1_mf_6_fddlvjs4_.log
   /u03/app/oracle/fast_recovery_area/SOURCE_DB_NAME/onlinelog/
   o1_mf_7_fddlvjys_.log
   /u03/app/oracle/fast_recovery_area/SOURCE_DB_NAME/onlinelog/
   o1_mf_8_fddlvk7x_.log
   /u03/app/oracle/fast_recovery_area/SOURCE_DB_NAME/onlinelog/
   o1_mf_9_fddlvkfj_.log
   ```

   Note down the path in the output above.

7. Use the information from this output to set the DB_FILE_NAME_CONVERT parameter.

> **Note:**
>
> Note that the name of the source database in this output is case sensitive.

```
ALTER SYSTEM SET DB_FILE_NAME_CONVERT='+DATA',
'/u02/app/oracle/oradata/SOURCE_DB_NAME/' SID='*' SCOPE=SPFILE;
```

8. Set the `LOG_FILE_NAME_CONVERT` parameter. For example:

   ```
   ALTER SYSTEM SET LOG_FILE_NAME_CONVERT='+RECO','/u04/app/oracle/
   redo/','+RECO',
   '/u03/app/oracle/fast_recovery_area/SOURCE_DB_NAME/onlinelog/' SID='*'
   SCOPE=SPFILE;
   ```

9. Restart the database.

   ```
   SHUTDOWN IMMEDIATE,
   STARTUP
   ```

Set the Database and Log File Name Conversion Parameters on the Standby Database

You must set the conversion parameters for the database and the log file name on the standby database (the target database).

1. Use SSH to sign in to the standby database (the target database) server.
2. Sign in as the database software owner `oracle`.

   ```
   sudo su - oracle
   ```

3. Sign in to the database instance as a user who has the `SYSDBA` administrator privilege. For example:

   ```
   sqlplus / as sysdba
   ```

4. Check the `CONVERT` parameter.

   ```
   SHOW PARAMETER CONVERT
   Output similar to the following appears:
   NAME                                 TYPE        VALUE
   ------------------------------------ ----------- ------
   db_file_name_convert                 string
   log_file_name_convert                string
   pdb_file_name_convert                string
   ```

   The `VALUE` column should be empty (null). If there is a value, then make a note of this value for after the migration is complete. After the migration is complete, these values are set to null.

5. Check the names of the data files.

   ```
   SELECT NAME FROM V$DATAFILE;
   Output similar to the following appears:
   NAME
   ------------------------------------------------
   +DATA/target_db_unique_name/DATAFILE/system.273.972998889
   +DATA/target_db_unique_name/DATAFILE/sysaux.272.972998889
   +DATA/target__unique_name/DATAFILE/undotbs1.270.972998945
   +DATA/target_db_unique_name/690A484F7D3F1B6EE05332C6120A3C84/DATAFILE/
   system.266.972998961
   +DATA/target_db_unique_name/DATAFILE/users.263.972998969
   +DATA/target_db_unique_name/690A484F7D3F1B6EE05332C6120A3C84/DATAFILE/
   sysaux.269.972998945
   +DATA/target_db_unique_name/690A76A8ED011E29E05332C6120AD40F/DATAFILE/
   system.265.972998945
   +DATA/target_db_unique_name/690A76A8ED011E29E05332C6120AD40F/DATAFILE/
   sysaux.264.972998889
   +DATA/target_db_unique_name/690A76A8ED011E29E05332C6120AD40F/DATAFILE/
   users.261.972998971
   +DATA/target_db_unique_name/690A76A8ED011E29E05332C6120AD40F/DATAFILE/
   example.267.972998887
   ```

6. Check the V$LOGFILE dynamic view.

   ```
   SELECT MEMBER FROM V$LOGFILE;
   Output similar to the following appears:
   MEMBER
   -------------------------------------------------------------------
   +RECO/target_db_unique_name/ONLINELOG/group_3.264.972998987
   +RECO/target_db_unique_name/ONLINELOG/group_2.263.972998987
   +RECO/target_db_unique_name/ONLINELOG/group_1.257.972998985
   +RECO/target_db_unique_name/ONLINELOG/group_4.265.972998987
   +RECO/target_db_unique_name/ONLINELOG/group_5.266.972998987
   +RECO/target_db_unique_name/ONLINELOG/group_6.267.972998989
   +RECO/target_db_unique_name/ONLINELOG/group_7.268.972998989
   +RECO/target_db_unique_name/ONLINELOG/group_8.269.972998989
   +RECO/target_db_unique_name/ONLINELOG/group_9.270.972998989
   ```

7. Use the information from this output to set the DB_FILE_NAME_CONVERT parameter.For example:

   ```
   ALTER SYSTEM SET DB_FILE_NAME_CONVERT='/u02/app/oracle/oradata/
   SOURCE_DB_NAME/','+DATA' SID='*' SCOPE=SPFILE;
   ```

8. Set the LOG_FILE_NAME_CONVERT parameter. For example:

   ```
   ALTER SYSTEM SET LOG_FILE_NAME_CONVERT='/u04/app/oracle/oradata/
   redo/','+RECO','/u03/app/oracle/fast_recovery_area/SOURCE_DB_NAME/
   onlinelog/','+RECO' SID='*' SCOPE=SPFILE;
   ```

9. Restart the database.

   ```
   srvctl stop database -db target_db_unique_name
   srvctl startup database -db target_db_unique_name -o mount
   ```

Configure the Database with Oracle Data Guard Broker

You can use the `dgmgrl` utility to configure either the primary database (the source database) or the standby database (the target database) with Oracle Data Guard Broker.

1. Use SSH to sign in to the primary database (the source database) or the standby database (the target database) server.
2. Start the `dgmgrl` command line utility as user `SYS` from either the primary or the standby database system. For example, to log in to a primary database whose TNS name is `OCIC-ORCL`:

   ```
   dgmgrl sys@ocic-orcl
   Enter password: password
   ```

3. Using the `dgmgrl` utility, create the Oracle Data Guard configuration and identity for the primary and standby databases. For example:

   ```
   create configuration <configuration_name> as primary database is
   <source_db_unique_name> connect identifier is <Source_tns_name>; --
   Uses the source TNS name

   add database <target_db_unique_name> as connect identifier is
   <target_tns_name>; --Uses the target TNS name
   ```

4. Enable the configuration.

   ```
   enable configuration
   ```

5. Show the Oracle Data Guard configuration on the standby database.

   ```
   show configuration
   #Output similar to the following appears:
   Configuration - configuration_name
   Protection Mode: MaxPerformance
   Members:
    source_db_unique_name         - Primary database
    target_db_unique_name         - Physical standby database

   Fast-Start Failover: DISABLED

   Configuration Status:
   SUCCESS   (status updated 12 seconds ago)
   ```

Validate Oracle Data Guard Broker on the Primary Database and the Standby Database

You can use SQL*Plus to validate Oracle Data Guard Broker on the primary database (the source database) and the standby database (the target database).

Validate Oracle Data Guard Broker on the Primary Database

You can use SQL*Plus to validate Oracle Data Guard Broker on the primary database (the source database).

1. Use SSH to sign in to the primary database (the source database) server.
2. Connect as a user who has the SYSDBA administrator privilege. For example, for a primary database whose TNS name is OCIC-ORCL:

   ```
   connect sys@ocic-orcl as sysdba
   Enter password: password
   ```

3. Query the V$DATABASE dynamic view.

   ```
   SELECT FORCE_LOGGING, FLASHBACK_ON, OPEN_MODE, DATABASE_ROLE,
   SWITCHOVER_STATUS, DATAGUARD_BROKER, PROTECTION_MODE FROM V$DATABASE;
   ```

4. Output similar to the following appears:

   ```
   FORCE_LOGGING                           FLASHBACK_ON       OPEN_MODE
   --------------------------------------- ------------------ -----------
   DATABASE_ROLE    SWITCHOVER_STATUS    DATAGUAR PROTECTION_MODE
   ---------------- -------------------- -------- --------------------
   YES                                     YES                READ WRITE
   PRIMARY          TO STANDBY           ENABLED  MAXIMUM PERFORMANCE
   ```

   In the output, the DATABASE_ROLE should be PRIMARY and OPEN_MODE should be READ WRITE.

Validate Oracle Data Guard Broker on the Standby Database

You can use SQL*Plus to validate Oracle Data Guard Broker on the standby database (the target database).

1. Use SSH to sign in to the standby database (the target database) server.
2. Connect as a user who has the SYSDBA administrator privilege.
3. Query the V$DATABASE dynamic view.

   ```
   SELECT FORCE_LOGGING, FLASHBACK_ON,
   OPEN_MODE, DATABASE_ROLE, SWITCHOVER_STATUS,
   DATAGUARD_BROKER, PROTECTION_MODE
   FROM V$DATABASE;
   ```

Output similar to the following appears:

```
FORCE_LOGGING                           FLASHBACK_ON       OPEN_MODE
--------------------------------------- ------------------ -----------
DATABASE_ROLE    SWITCHOVER_STATUS    DATAGUAR PROTECTION_MODE
---------------- -------------------- -------- --------------------
YES                                     YES                MOUNTED
PHYSICAL STANDBY NOT ALLOWED          ENABLED  MAXIMUM PERFORMANCE
```

The output should show `DATABASE_ROLE` as `PHYSICAL STANDBY`and `OPEN_MODE` as `MOUNTED`.

4. Verify that the Oracle Data Guard processes are initiated in the standby database.

   ```
   SELECT PROCESS,PID,DELAY_MINS FROM V$MANAGED_STANDBY;
   ```

   Output similar to the following appears:

   ```
   PROCESS   PID                      DELAY_MINS
   --------- ------------------------ ----------
   ARCH      9207                              0
   ARCH      9212                              0
   ARCH      9216                              0
   ARCH      9220                              0
   RFS       1065                              0
   RFS       1148                              0
   RFS       1092                              0
   MRP0      972                               0
   RFS       1208                              0
   ```

   The output should indicate that the processes are running with little or no delay. If the `DELAY_MINS` for `MRP0`, the databases are synchronized.

5. Check the`LOG_ARCHIVE_DEST` parameter.

   ```
   SHOW PARAMETER LOG_ARCHIVE_DEST_
   ```

   Output similar to the following appears:

   ```
   NAME                         TYPE      VALUE
   ---------------------------- --------- -----------------------------
   log_archive_dest_1           string
                                          location=USE_DB_RECOVERY_FILE_
                                          DEST, valid_for=(ALL_LOGFILES,
                                          ALL_ROLES)
   log_archive_dest_10          string
   log_archive_dest_11          string
   log_archive_dest_12          string
   log_archive_dest_13          string
   log_archive_dest_14          string
   log_archive_dest_15          string
   ...
   log_archive_dest_2           string    service="oci-orcl", ASYNC
   ```

```
                                       NOAF FIRM delay=0 optional
                                       compression=disable
                                       max_failure=0 max_connections
                                       =1 reopen=300 db_unique_name=
                                       "source_db_unique_name"
                                       net_timeout=30, valid_for=
                                       (online_logfile,all_roles)
...
```

The output should be similar to the output for `log_archive_dest_2`, with the service pointing to the standby database (the target database), which in this example is `oci-orcl`.

6. Check the `LOG_ARCHIVE_CONFIG` parameter.

   ```
   SHOW PARAMETER LOG_ARCHIVE_CONFIG#
   ```

   Output similar to the following appears:

   ```
   NAME                         TYPE      VALUE
   ---------------------------- ---------
   -----------------------------------------------------------
   log_archive_config           string
   dg_config=(source_db_unique_name,target_db_unique_name)
   ```

7. Check the `FAL_SERVER` parameter.

   ```
   SHOW PARAMETER FAL_SERVER
   ```

   Output similar to the following appears:

   ```
   NAME                        TYPE      VALUE
   --------------------------- --------- ----------
   fal_server                  string    <tns_entry_of_primary>
   ```

8. Check the `LOG_ARCHIVE_FORMAT` parameter.

   ```
   SHOW PARAMETER LOG_ARCHIVE_FORMAT
   ```

   Output similar to the following appears:

   ```
   NAME                        TYPE      VALUE
   --------------------------- --------- --------------
   log_archive_format          string    %t_%s_%r.dbf
   ```

Complete the Validation on the Primary Database

You can use `dgmrgl` to complete the Oracle Data Guard Broker validation on the primary database (the source database).

1. Use SSH to sign in to the primary database (the source database) server.
2. Repeat steps 5 through 8 in the topic **Validate Oracle Data Guard Broker on the Standby Database** on the primary database (the source database).
3. Start the `dgmgrl` command line utility:

   ```
   dgmgrl
   ```

4. Connect as user `SYS` from either the primary or the standby database system. For example, to log in to a primary database whose TNS name is `OCIC-ORCL`:

   ```
   connect sys@primary_db_tnsnames_name
   Enter password: password
   ```

5. Check the Oracle Data Guard configuration.

   ```
   show configuration verbose
   ```

   Output similar to the following appears:

   ```
   Configuration - configuration_name

     Protection Mode: MaxPerformance
     Members:
     source_db_unique_name            - Primary database
       target_db_unique_name          - Physical standby database

     Properties:
       FastStartFailoverThreshold      = '30'
       OperationTimeout                = '30'
       TraceLevel                      = 'USER'
       FastStartFailoverLagLimit       = '30'
       CommunicationTimeout            = '180'
       ObserverReconnect               = '0'
       FastStartFailoverAutoReinstate  = 'TRUE'
       FastStartFailoverPmyShutdown    = 'TRUE'
       BystandersFollowRoleChange      = 'ALL'
       ObserverOverride                = 'FALSE'
       ExternalDestination1            = ''
       ExternalDestination2            = ''
       PrimaryLostWriteAction          = 'CONTINUE'

   Fast-Start Failover: DISABLED
   ```

6. Check the status on the standby database (the target database). For example:

   ```
   show database verbose target_db_unique_name
   ```

   After you complete these steps, you must test that the Oracle Data Guard configuration is functioning as expected by performing switchover operations in both directions.

Perform the Migration

To complete the migration, you must perform a switchover operation from the primary database (the source database) to the standby database (the target database).

1. Use SSH to sign in to the primary database (the source database) server.
2. Start the `dgmgrl` utility. For example, for a source database whose TNS name is `OCIC-ORCL`:

   ```
   dgmgrl sys@ocic-orcl
   Enter password: password
   ```

3. Check the configuration.

   ```
   show configuration verbose
   ```

4. In the configuration verbose output, check the `StaticConnectIdentifier` setting.

   This setting should point to the standby database (the target database) connection ID. The `Database Status` setting should say `SUCCESS`.
5. If necessary, use `dgmgrl` to change the `StaticConnectIdentifier` setting to point to the correct TNS net services name. For example:

   ```
   edit database source_db_unique_name set property
   staticConnectidentifier='source_TNS_name';
   edit database target_db_unique_name set property
   staticConnectidentifier='target_TNS_name';
   ```

6. Check the configuration for the primary database (the source database).

   ```
   show database verbose source_db_unique_name
   ```

   The database verbose output should show that the role is primary and the setting for `StaticConnectIdentifier` is the same as `DGConnectIdentifier`.
7. Perform a switchover operation to the standby database (the target database).

   ```
   switchover to target_db_unique_name
   ```

   The output should indicate that the switchover operation is occurring between the two databases.

Post-Migration Steps

After you complete the migration of an Oracle database from an Oracle Cloud Infrastructure Compute Classic server to an Oracle Cloud Infrastructure server that uses a Virtual Machine Database system, you should validate the migration, and then remove the configuration from the primary database (the source database).

Test the Oracle Data Guard Configuration on the Standby Database

At this stage, the target database is now the primary database. The source database is now the standby database.

You can test the Oracle Data Guard connection on the target database, by performing a switchover operation with the source database. This switchover operation will make the target database take the standby role again. The purpose of this test is to prove that you can return to the original configuration in case the target database is not functional.

1. Use SSH to sign in to the standby database (the target database) server.
2. Start the `dgmgrl` utility. For example:

   ```
   dgmgrl sys@target_db
   Enter password: password
   ```

3. Perform a switchover operation to the source database, which will make the target database take the standby role.

   ```
   switchover to source_db_unique_name;
   ```

   The output should indicate that the switchover operation is occurring between the two databases.
4. (Optional) To prevent changes to the new standby database until the new primary database is determined to be fully functional, temporarily disable the `Redo Apply` feature.

   ```
   edit database source_db_unique_name set state = 'APPLY-OFF';
   ```

   If you perform another switchover operation so that the target database is now the standby database, you can perform an `APPLY-OFF` operation to prevent the source database from being updated. This enables the target database to be put in service, and keeps the source database as a point-in-time backup in case of a logical failure in the new configuration.
5. (Optional) To restart the apply feature:

   ```
   edit database source_db_unique_name set state = 'APPLY-ON';
   ```

6. Exit `dgmgrl`.

   ```
   exit
   ```

7. Perform a switchover operation to the target database, which will make the source database the standby role.

   ```
   switchover to target_db_unique_name;
   ```

   The output should indicate that the switchover operation is occurring between the two databases.

8. Test the connection to the new primary database. For example, after exporting the target unique name, connect as user SYS and select from an encrypted table space. In this example, the HR.EMPLOYEES table is encrypted.

   ```
   exit
   ```

9. Test the connection to the new primary database. For example, after exporting the target unique name, connect as user SYS and select from an encrypted table space. In this example, the HR.EMPLOYEES table is encrypted.

   ```
   export ORACLE_UNQNAME=target_db_unique_name

   sqlplus sys@target_TNS_name
   Password: password

   SQL> ALTER SESSION SET CONTAINER = PDB1;
   SQL> SELECT * FROM HR.EMPLOYEES;

   SQL> EXIT
   ```

Clean Up the Standby Database

After you complete and test the migration, you can remove the Oracle Data Guard configuration from the standby database (the target database). You do not need to remove the original source database. At this stage, the standby database is the new source database.

1. Use SSH to sign in to the standby database (the target database) server and sign in to the Oracle Data Guard dgmgrl utility.
2. Check the configuration.

   ```
   show configuration
   ```

3. If the configuration does not show Protection Mode: MaxPerformance, then set Oracle Data Guard to use the MaxPerformance protection mode.

   ```
   edit configuration set protection mode as maxperformance
   ```

4. Disable and then remove the configuration.

   ```
   edit database source_db_unique_name set state = 'APPLY-OFF';

   disable configuration;

   remove configuration;

   exit
   ```

5. Connect to the database instance as a user who has the SYSDBA administrator privilege.For example:

   ```
   sqlplus / as sysdba
   ```

6. Check the DG_BROKER_CONFIG_FILE parameters.

   ```
   SHOW PARAMETER DB_BROKER_CONFIG_FILE
   ```

   The output should list the associated data and recovery files for this configuration, typically named dg_broker_config_file1 and dg_broker_config_file2.

7. Start another terminal window, and sign in to asmcmd as the grid user.

8. Remove the Oracle Data Guard configuration files that were listed when you checked the DG_BROKER_CONFIG_FILE parameters.

9. Return to the window that is running SQL*Plus.

10. Execute the following ALTER SYSTEM statements:

    ```
    ALTER SYSTEM SET DG_BROKER_START=FALSE SID='*' SCOPE=BOTH;
    ALTER SYSTEM SET DG_BROKER_CONFIG_FILE1='' SID='*' SCOPE=SPFILE;
    ALTER SYSTEM SET DG_BROKER_CONFIG_FILE2='' SID='*' SCOPE=SPFILE;
    ALTER SYSTEM RESET LOG_ARCHIVE_CONFIG SID='*' SCOPE=SPFILE;
    ```

11. Check the following parameters:

    ```
    SHOW PARAMETER DB_FILE_NAME_CONVERT
    SHOW PARAMETER LOG_FILE_NAME_CONVERT
    SHOW PARAMETER LOG_ARCHIVE_DEST
    SHOW PARAMETER LOG_ARCHIVE_DEST_STATE
    SHOW PARAMETER STANDBY_ARCHIVE_DEST
    SHOW PARAMETER FAL
    ```

12. If any of the preceding parameters is set, then reset the parameters to use blank values. For example, for STANDBY_ARCHIVE_DEST:

    ```
    ALTER SYSTEM SET STANDBY_ARCHIVE_DEST='' SID='*' SCOPE=SPFILE;
    ```

13. Restart the database.

    ```
    SHUTDOWN IMMEDIATE
    STARTUP
    ```

14. Drop the standby logs from the primary database (the source database).

    a. Find the group numbers for the standby database redo logs that are on the new primary database (which was formerly the target database).

       ```
       SELECT GROUP# FROM V$STANDBY_LOG;Output similar to the following
       appears:    GROUP#
       ----------
                5
                6
                7
                8
                9
       ```

b. Remove the standby logs. For example:

```
ALTER DATABASE DROP STANDBY LOGFILE GROUP 5;
ALTER DATABASE DROP STANDBY LOGFILE GROUP 6;
ALTER DATABASE DROP STANDBY LOGFILE GROUP 7;
ALTER DATABASE DROP STANDBY LOGFILE GROUP 8;
ALTER DATABASE DROP STANDBY LOGFILE GROUP 9;
```

15. (Optional) Change the DB_BLOCK_CHECKSUM and DB_BLOCK_CHECKING parameters.

    The default values are DB_BLOCK_CHECKSUM=TYPICAL and DB_BLOCK_CHECKING=FALSE.

16. Exit SQL*Plus.

```
EXIT
```

Reapply Rolled Back Patches (if any) on Primary

After you complete the migration, you should reapply the patches(if any) that you had rolled back as part of the **Prepare** section. This step should be performed on primary database.

> **Note:**
>
> This step is applicable **ONLY** if you had applied patch for bug 18633374 in the **Prepare** step, and have rolled back any patches as part of that procedure.

1. Login to the primary database as *oracle*
2. Roll back patch 18633374.

```
$ORACLE_HOME/OPatch/opatch rollback -id 18633374
```

   Wait for the roll back to complete.

3. Go to My Oracle Support (MOS) page. Use the **Search** box to find the MOS note for the patch number that was rolled back as part of **Prepare** step .
4. Follow the instructions in the **ReadMe** file to reapply the patch.
5. Repeat the steps 2 and 3 for all the patches that were rolled back.

14 Migrate New Data in Real Time from Oracle Database Classic to Oracle Cloud Infrastructure

You can migrate a single-instance database, created on Oracle Cloud Infrastructure Classic, to an Oracle Cloud Infrastructure virtual machine database system, by using Data Integration Platform Cloud. You can use this procedure as part of an overall migration of your Oracle Cloud environment to Oracle Cloud Infrastructure.

About Using Data Integration Platform Cloud to Migrate Oracle Database Cloud Deployment to Oracle Cloud Infrastructure

Using Data Integration Platform Cloud (DIPC), you can perform real-time data migration. The DIPC migration process watches the source database and updates the target with every change that happens on the source.

- When to Use DIPC for Migrating an Oracle Database Cloud Deployment to OCI
- Architecture
- What You Need to Perform the Migration
- How to Migrate an Oracle Database Cloud Deployment to OCI Using DIPC

When to Use Data Integration Platform for Migrating an Oracle Database Cloud Deployment to Oracle Cloud Infrastructure

Use Data Integration Platform Cloud to migrate new data in real time from Oracle Database Classic to Oracle Cloud Infrastructure.

To use Data Integration Platform Cloud for migration, your source database should be an Oracle Database Classic instance and the target should be an Oracle database system on a virtual machine on OCI. This migration is application for Oracle database 11g to 11g, 11g to 12c, and 12c to 12c.

This migration method watches the source database and updates the target with every change that happens in the source. Every new insert, delete, or update that happens on the source will be reflected in the target database in real-time. Before you start this real-time migration, you must create the target database with the same schemas and tables that you want to replicate.

Use DIPC for migration in the following scenarios:

- When you don’t want to shut down your source production database.

- When you want to perform real-time migration of Oracle databases 11g to 12c, which is more complicated with other migration methods.
- When your new database only requires a copy of your source database from a certain point of time that you choose. This point of time has to be after you start the migration. The target database can be used for data analytics, to take the query load off your source database. It can also be used as a more current database, which doesn't include historical data from the source database.

Limitations:

- Replicate Data does not perform an initial load. For a complete migration of the source database, including the historical data, use Oracle Data Pump or Oracle Recovery Manager (RMAN) to copy the initial load from source to the target first and perform this real-time migration after that.
- The change in the source database isn't available for processing until after the Data Manipulation Language (DML) commands such as insert, update and delete have been committed. Therefore, only transactions are replicated in Replicate Data.
- Data Definition Language (DDL) operations such as drop or create are not part of the replication process.
- Replicate Data is one to one and uni-directional, which means that it accepts one source and one target for each task. If you want to include several sources to deliver data to one target, you can create a Replicate data task for each source and have them all deliver to the same target.

Architecture

You can migrate your data from an Oracle Database Classic instance hosted on Oracle Cloud Infrastructure Classic to a virtual machine database system on Oracle Cloud Infrastructure (OCI). Oracle Database versions 11.2.0.4, 12.1.02 and 12.2.0.1 are all supported.

Here is how data flows when Data Integration Platform Cloud replicates your data:

Your source database is on OCI Classic. You create a target database on OCI. An application on OCI called Oracle Data Integration Platform Cloud (DIPC) orchestrates data replication from the source database to the target database. To perform the replication, you set up two DIPC remote agents on two compute instances: one for source and one for target. The agents have some binaries called Oracle GoldenGate (OGG) that can send data information from source to target.

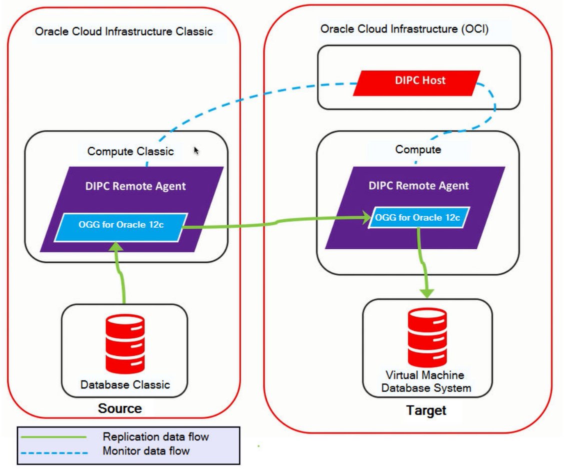
The data flows from source database directly to the source agent and from there to the target agent and finally to the target database. It doesn't go to the DIPC host. Only monitoring information is exchanged between the agents and the DIPC host.

The data that flows from source agent to the target agent has to pass through a firewall. You open up some ports on the target compute instance, so that the data can come in. You don't need to open any ports on the source side, because it's a one-way replication and data only goes out.

The connections between each database and its agent is through a private IP address. The compute instance on OCI Classic should be on the same network as the Database Classic instance. Similarly, the Compute instance on the OCI system should be on the same VCN as the virtual machine database system. With this setup, you don't need VPN to have the agents access the databases.

The connection between the agents and their DIPC host is through an outgoing https connection. Information only goes out to the DIPC host for monitoring the data flow.

The following diagram shows the migration process:



What You Need to Perform the Migration

Before you begin the database migration, in addition to knowledge of Oracle Cloud Infrastructure, you must have knowledge in several areas of Oracle Database tools.

You will need the following components or tools:

- An Oracle Cloud Infrastructure account with a role to create compartments, subnets, target database, and Data Integration Platform instances.
- An account on Oracle Support to create a service request.
- Tools to create SSH keys, such as PuTTY or SSH
- Tool to connect to VMs, such as PuTTY
- Tool to connect to databases, such as SQL developer or you can SSH to the database directly

You should be familiar with the following tasks:

- Setting up compartments, subnets, and databases in Oracle Cloud Infrastructure (OCI)
- Creating SSH keys using either PuTTY or SSH
- Using SQL*Plus
- Performing Oracle database management tasks, such as creating schemas, users, and tables.

> **Note:**
>
> You must create a service request through Oracle Support. Ask them to reserve a Data Integration Platform Cloud instance for you. Otherwise, you will get an error message that you don't have enough resources.

How to Migrate an Oracle Database Cloud Deployment to Oracle Cloud Infrastructure by Using Data Integration Platform

You perform the following tasks to migrate an Oracle Database Classic cloud deployment to Oracle Cloud Infrastructure by using Data Integration Platform Cloud.

- Prepare for Migration
  - Create a Data Integration Platform Cloud instance
  - Set up the agents
  - Prepare source database
  - Prepare target database
- Migrate the Database
  - Create Connections
  - Create a Replicate Data Task
  - Run the Replicate Data Task

Prepare for Migration

To prepare for migration with Data Integration Platform Cloud perform the following tasks:

1. Create a Data Integration Platform Cloud Instance
2. Set Up the Agents
3. Prepare Source Database
4. Prepare Target Database

Create a Data Integration Platform Cloud Instance

Data Integration Platform Cloud (DIPC) is an all-in-one platform with graphical menu options for data preparation, integration, replication and quality management.

Here are the steps to create a Data Integration Platform Cloud instance:

1. Log in to **Oracle Cloud Infrastructure** home page.
2. Click the **menu** option for **Oracle Cloud**.
3. Click **My Services Dashboard**.
4. If you land on the Guided Journey page, then click **Dashboard**.
5. Click **Customize Dashboard**.
6. Under **Integration**, click **Show** for **Data Integration Platform**.
7. **Close** the Customize Dashboard window.
8. Confirm that **Data Integration Platform** appears as a tile on your dashboard.
9. In the Data Integration Platform tile, click **Open Service Console** from the **Action Menu**.

10. If you encounter a welcome page, click **Go to Console**.
11. Don't click the prerequisite link on the page, because it is an old link and gives an error message. Currently the prerequisites are created for your when you create the DIPC instance. So there are no prerequisites, if you can already create an instance.
12. Click **Create Instance**.
13. Complete the Instance details. For example:
    - **Instance Name: `ABCMigration`**
    - Description: **`Data Migration from the ABC database to OCI`**
    - **Email:** an email address to receive a notification when your instance is ready
    - **Region: No Preference**
    - **Tags:** Click the Click to Create a Tag button and create a tag called **Migration** and then click OK. (optional)
    - **License Type:** Subscribe to a **new Data Integration Platform Cloud software license** and the Data Integration Platform Cloud.
14. Click **Next**.
15. Complete the Service Details. For example:
    - **Service Edition: `Enterprise Edition`**

      > **Note:**
      >
      > You must either choose Enterprise or Governance Edition. Enterprise Edition offers choices for selective copy or real time replication of your data sources. This edition also offers extract, load and transform (ELT) options for your data entities. Governance Edition includes Enterprise features plus additional data quality management options. Standard Edition is for ELT only and does not offer you a choice to perform a migration.

    - **Data Volume - GB:** Select an option to be billed based upon. You're charged a monthly flat rate based on the volume of data processed per hour for data movement from/to Data Integration Platform Cloud. For more information, see Pricing.
    - **Instance Reserved:** Enable this checkbox if an instance has already been reserved for you. If you have not reserved an item, you can't create an instance. Submit a request to reserve a DIPC instance for you through Oracle Support.
16. Click **Next**.
17. If you need to make changes, use the Previous button to navigate back to the Instance or Details pages, and then come back to this page and confirm and create the instance.
18. Confirm that the name of your instance is in the list of **Instances** and the field **Created On** displays the creation date.

Set Up the Agents

Agents are applications that you allow to securely connect to data sources and orchestrate data integration activities. An agent's components depends on the activities that it orchestrates. For example, data replication and adding data to data lake have different components.

For data migration between two Oracle databases, you set up two agents:

- A source agent to capture data activities from the source and send it to the target agent
- A target agent to receive information from the source agent and apply the data changes to the target database

The best location for the source agent is on OCI-Classic where the Oracle Database Classic is. Similarly, the best place for the target agent is on OCI, in the same region as the target database system. A Compute instance is a physical location that you can create on OCI or OCI classic and then copy your agent binaries there.

Here are the steps to set up your agents:

1. Download Agent Binaries
2. Create a Location for Agents
3. Register the Agents
4. Run the Agents

Download Agent Binaries

To set up an agent, you must first download the proper agent binaries from the Data Integration Platform console to your machine.

If both source and target databases have the same version, then you can download the binaries once and use it twice.

Here is how you can get your agent binaries:

1. On My Services Dashboard, click the **Action Menu** for the **Data Integration Platform** tile and then click **Open Service Console**.
2. In the **Instances** page, locate your Data Integration Platform Cloud instance. For example, `ABCMigration`
3. From the instance's **Manage this instance** menu, select **Data Integration Platform Console**.
4. Click **Download** in the **Agents** tile.
5. Select the operating system and components:
   - **Operating System:** This option is not your database's operating system. It is the operating system of the location that the agent will be installed. Select Linux 64-bit, because both the OCI and OCI Classic Compute instances have Linux operating systems.
   - **Connectors/Components:** For this option, either choose Oracle 11g (OGG) or Oracle 12c (OGG). This option provides the replication binaries that match your Oracle database version.

6. Click **Download**.
7. **Save** the binaries so that later you can transfer it to a Compute instance.

Create a Location for Agents

Each Data Integration Platform Cloud agent must have access to a data source to orchestrate its data integration activities. Compute instances are ideal locations for agents that manage cloud database activities.

For an Oracle Database Classic, the best place to put the agent binaries is on a Compute Classic instance. Similarly, for a database on OCI, the best place for the agent is on a Compute instance on OCI. This way, you don't need to open any ports between each database and its agent and all that communication is done through private IP addresses. The only port you open, is on the target Compute instance to allow the source agent to send information to the target agent.

Here are the steps to create a location for your source and target agents.

1. Create a Compute Instance for Source Agent
2. Add Java to Your Compute Instance
3. Create a Compute Instance for the Target Agent
4. Open Ports on Target Compute Instance

Create a Compute Instance for Source Agent

If you have an Oracle Database Classic, then, place its agent binaries on a Compute Classic instance located in the same region as the database.

Here are the steps to create a Compute instance for your source agent:

1. Go to **My Services Dashboard**.
2. Click **Customize Dashboard**.
3. Click Show for **Compute Classic**.
4. Close the Customize Dashboard window.
5. In the **Compute Classic** tile, click **Open Service Console** from the **Action Menu**.
6. Click **Create Instance**.
7. Click **Customize**.
8. For **Image**, select the latest one. For example, OL_7.2_UEKR4_x86_64 which is an Oracle Linux 7.2 UEKR4 and proceed to the Shape page.
9. For **Shape**, select **oc5** which gives you **4 OCPUs and 30GB** of memory.
10. Proceed to the **Instance** page.
11. For **Placement**, select **the same domain or location** as the **source database**.
12. Add your public Key to **SSH Keys**.
13. Proceed to the **Network** page.
14. For Network options, select **IP Network**.
15. Click the **Action Menu** for the listed default network and then select **Update**.
16. Set the **Public IP Address** to **Auto Generated** and then click **Update**.

17. Name your **DNS Hostname Prefix** and then proceed to the next page.
18. Click the **Action Menu** for the listed default storage and then select **Update**.
19. Update the storage **Size** to **120 GB**.
20. Confirm that the new storage size is 120 GB and then proceed to the **Review** page.
21. Review all your options and either go back to modify the fields or if you agree with your options, then click **Create**.

Related Topics

- Creating an Oracle Linux Instance Using the Oracle Cloud Infrastructure Compute Classic Web Console
- Oracle Cloud Infrastructure Compute Classic Help Center

Add Java to Your Compute Instance

Data Integration Platform agents refer to $JAVA_HOME to perform their activities. You must set up $JAVA_HOME to point to JDK with a Java version of 1.8 or higher, before you set up your agents.

Compute instances don't come with JDK, so you must install JDK on your Compute instances and then have $JAVA_HOME point to JDK.

1. Download JDK with Java version 1.8 or higher from Java SE Downloads
2. Use an application that has Secure Copy Protocol (SCP) such as WinSCP or SSH to transfer the JDK execution file to your Compute instance. Provide the IP address and the private key that matches the public key that you gave to your Compute instance to connect to it. The username is `opc`.
3. Use PuTTY or SSH to connect to the Compute instance. For instructions on how to connect to instances with PuTTY, see Connect to an Oracle Cloud Instance From Windows Using PuTTY.
4. Install JDK on the Compute instance.
5. Ensure that your environment is pointing to the correct Java.
6. Ensure that `JAVA_HOME` points to your JDK. If not, then set it to point to the right location.

Example:

```
[xxx]$ java - version
java version "1.8.0_91"

[xxx]$ which java
~/Public/JDK/bin/java

[xxx]$ echo $JAVA_HOME
/home/oracle/Public/JDK
```

Create a Compute Instance for the Target Agent

The Compute instance that you create for the target agent is similar to the source Compute, except that it's on Oracle Cloud Infrastructure and it must be much larger than the source Compute. Ensure that the storage size that you allocate for the target is 2TB or larger.

The target agent receives and stores all the information about the changes in the source database in the target Compute storage. Because these files are constantly growing, you must have a large storage to ensure a great performance. Here are some guidelines to create a Compute instance on OCI.

1. Log in to Oracle Cloud Infrastructure.
2. Select a **region** on the top navigation bar that matches your target database region.
3. From the Action Menu, go to **Compute** and then **Instances**.
4. Click **Create Instance**.
5. Select **Virtual Machine** for instance type.
6. For **Shape** select **VM.Standard2.1**
7. Select **Custom boot volume size**.
8. Enter `2000`, because the size you enter is in GBs and 2TB is equivalent to 2000 GB.
9. Enter an SSH public key with the .pub extension. Ensure that you have the matching private key on your machine.
10. Refer to Launching Your First Linux Instance for additional guidance.
11. Click **Create**.
12. Install JDK for Java 1.8 or higher on your Compute instance.
13. Have $JAVA_HOME point to your JDK.

Related Topics

- Overview of the Compute Service
- Add Java to Your Compute Instance
  Data Integration Platform agents refer to $JAVA_HOME to perform their activities. You must set up $JAVA_HOME to point to JDK with a Java version of 1.8 or higher, before you set up your agents.

Open Ports on Target Compute Instance

Data Integration Platform Cloud source agents send data replication information to target agents. The only way for the target agents to receive this information is to open ports on the Oracle Cloud Infrastructure (OCI) Compute instance where they are located.

You must open port 7809 for the agent application that manages the replication and 7819-7829 for the data collector application. Here are steps on opening ports for a Compute instance on OCI:

1. Log in to OCI.
2. Go to the region that your target database is located. For example, `us-phoenix-1`.
3. From the Actions Menu, go to **Compute** and then **Instances**.
4. Click the Compute instance hat you have allocated for your agent.
5. Click the link in the **Subnet** field. For example, `Public_Subnet ilMx:PHX-AD-3`

6. Click **Security Lists** from the list of **Resources**.
7. Click the default security list for your VCN. For example, `Default Security List for DIPC-VCN`.
8. In the **Resources** section, go to **Ingress Rules**.
9. Click **Add Ingress Rules**.
10. Enter the following information for your ingress rule:
    - **SOURCE TYPE:** CIDR
    - **SOURCE CIDR:** `0.0.0.0/0`
    - **IP PROTOCOL:** TCP
    - **SOURCE PORT RANGE:** All
    - **DESTINATION PORT RANGE:** `7809`
11. Click **+ Additional Ingress Rule**.
12. Enter the following information for your ingress rule:
    - **SOURCE TYPE:** CIDR
    - **SOURCE CIDR:** `0.0.0.0/0`
    - **IP PROTOCOL:** TCP
    - **SOURCE PORT RANGE:** All
    - **DESTINATION PORT RANGE:** `7819-7829`
13. Click **Add Ingress Rules**.

Register the Agents

After you download your Data Integration Platform agent components, you perform certain steps and send special parameters to the Data Integration Platform host, so that it can discover the agent. This procedure is called registering the agent.

Here are steps to register an agent. **Perform these steps twice**. Once for the source agent and once for the target agent.

1. Use an application that has Secure Copy Protocol (SCP) such as WinSCP or SSH to transfer the agent zip file to the Compute instance you have allocated for the agent.

   Provide the IP address and the private key that matches the public key that you gave to your Compute instance to connect to it. The username is `opc`.
2. Use PuTTY or SSH to connect to the Compute instance.

   For PuTTY instructions, see Connect to an Oracle Cloud Instance From Windows Using PuTTY.
3. Unzip the agent package that's called something like `DIPCAgent-18.4.3_Linux-x64.zip`, into `<agent_unzip_loc>`.
4. Create a file under `<agent_unzip_loc>/dicloud` and call it `agent_register.sh`.

5. Copy the following code into the agent_register.sh file and then save it.

   ```
   ./dicloudConfigureAgent.sh <agent_name> -recreate -authType=OAUTH2 -
   idcsServerUrl=https://idcs-xxx.identity.oraclecloud.com -
   agentIdcsScope=https://xxx.adipc.opc.oraclecloud.com:
   443urn:opc:resource:consumer::all -agentClientId=xxx -
   agentClientSecret=xxx -dipchost=xxx -dipcport=443 -
   user=<your_Oracle_Cloud_user_name> -
   password=<your_Oracle_Cloud_password>
   ```

   The dicloudConfigureAgent.sh command sends several parameters to the Data Integration Platform host. With these parameters, the Data Integration Platform host can recognize the details of the agent location and display it in the **Agents** page.

   The -recreate parameter tells the dicloudConfigureAgent command to create a directory called <agent_name> and place all the agent configuration files, such as agent.properties in this directory. If an agent with the name <agent_name> already exists in the dicloud directory, then the existing agent directory is renamed and kept, before an <agent_name> directory is recreated.

6. Find the values for the following required parameters and update the agent_register.sh file one by one, with the correct values.

   - idcsServerUrl:
     a. On My Services Dashboard, click **Customize Dashboard**.
     b. Click Show for **Identity Cloud** and then close the window.
     c. On the Identity Cloud tile, click the **Action Menu**.
     d. Click **View Details**.
     e. From the **identity** section, copy the value displayed for **Service Instance URL**.
     f. **Remove** the /ui/v1/adminconsole section from the <idcsServerUrl>/ui/v1/adminconsole and copy the <idcsServerUrl> section for the value of the idcsServerUrl parameter.
   - agentIdcsScope, agentClientId, agentClientSecret: Create a confidential application and get the values from its detail page. See Create a Confidential Application for Agents.
   - dipchost:
     a. In the Data Integration Platform Console, go to the **Agents** page.
     b. Find the URL of the Agents page in the browser which has a format of https://<dipchost>/dicloud/app/agents
     c. For the value of the dipchost parameter, copy the <dipchost> section of the URL.
   - dipcport: 443
   - username: username for your Oracle Cloud user account
   - password: password for your Oracle Cloud user account

Before you execute `agent_register.sh`, ensure that `$JAVA_HOME` is set correctly and you are using JDK for Java 1.8 or higher. See Add Java to Your Compute Instance

7. To register the agent, execute the `agent_register.sh` file.

Create a Confidential Application for Agents

A confidential application is a server-side application that uses OAUTH 2.0 to keep the confidentiality of the client secret. You can create a confidential application for your Data Integration Platform Cloud host. Then you can provide the client secret for the application to your agent so it can securely communicate with the DIPC host.

Here are the steps to make create a confidential application for your agent:

1. On My Services Dashboard, click **Customize Dashboard**.
2. Click Show for **Identity Cloud** and then close the window.
3. On the Identity Cloud tile, click the **Action Menu**.
4. Click **View Details**.
5. Click **Open Service Console**.
6. Click **Applications**.
7. Click **Add**.
8. Click **Confidential Application**.
9. In the **App Details**, enter a name for your application such as `DIPC_Agent` and then click **Next**.
10. In the **Client** page, click **Configure this application as a client now**.
11. For **Allowed Grant Types**, select **Resource Owner**, **Client Credentials**, **JWT Assertion** and **Refresh Token**.
12. Click **Add** for **Scope**.
13. Find the item with the format of `DIPCINST_<Your_DIPC_Instance_Name>` .See Create a Data Integration Platform Cloud Instance if you don't remember your Data Integration Platform instance name.
14. Click the arrow that indicates **Select scopes for this resource**.
15. Select the **checkbox** on the page for the resource and then click **Back**.
16. Select the checkbox for `DIPCINST_<Your_DIPC_Instance_Name>`.
17. Click **Add**.
18. Copy the link displayed for **Allowed Scope** to replace the value of the `agentIdcsScope` parameter in the `agent_register.sh` file. See Register the Agents
19. Click **Next**.
20. In the **Resources** page, click **Skip for Later** and then click **Next**.
21. In the **Web Tier Policy** page, click **Skip for Later** and then click **Next**.
22. Click **Finish**.
23. In the **Application Added** window, copy **Client ID** to replace the value of the `agentClientId` parameter in the `agent_register.sh` file. See Register the Agents.

24. In the same **Application Added** window, copy **Client Secret** to replace the value of the `agentClientSecret` parameter in the `agent_register.sh` file. See Register the Agents.
25. Click **Close**.
26. Click **Activate**.
27. Click **Activate Application**.

> **Note:**
>
> You must activate the application. Otherwise, your agent can't communicate with the Data Integration Platform host.

Run the Agents

After you register your agents with their Data Integration Platform host, you must start them, so you can use them for your data integration tasks.

Perform the following steps to start your agent:

1. Access the Compute instance that has a registered source agent, by using PuTTYor SSH.
2. Navigate to the `<agent_unzip_loc>/dicloud/agent/<your_agent_name>/bin` directory.

   If you didn't name your agent, then the default for `<your_agent_name>` is something like `dipcagent001`.
3. Run the following command:

   ```
   ./startAgentInstance.sh
   ```
4. Go to the **Agents** page in the DIPC console.
5. Find your instance name in the list of agents.
6. Ensure that its status is **Running**.
7. Repeat the same procedure for the **target agent**.

Prepare Source Database

To prepare a source database for migration with Data Integration Platform Cloud, you must either create a user on the source that is specific to the migration process or give an existing user specific data integration rights.

Perform the following steps to create a user at the CDB level:

1. Connect to the VM of the source database through SSH, PuTTY or SQL Developer.

   For guidance, see Find the IP Address of an Oracle Cloud Service VM, Change Private Key Format to Use with PuTTY and Connect to a Cloud VM on Windows with PuTTY. To connect to the VM with PuTTY use the `opc` username.
2. sudo to the oracle user with the `sudo -su oracle` command.
3. Ensure that the source database is running.

For guidance, see Ensure That the Database to be Migrated Is Running

4. Connect to the database on the VM, at the CDB level, as the admin user.
5. Run the following commands to create a user called C##GGSRC:

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
ALTER SYSTEM SET ENABLE_GOLDENGATE_REPLICATE=TRUE SCOPE=BOTH;
CREATE USER C##GGSRC IDENTIFIED BY <PASSWORD> DEFAULT TABLESPACE USERS
TEMPORARY TABLESPACE TEMP;
GRANT RESOURCE, CONNECT, DBA to C##GGSRC CONTAINER=ALL;
EXEC DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE =>'C##GGSRC, CONTAINER
=> 'ALL');
```

6. Ensure that you can log in to the source database as C##GGSRC.

Prepare Target Database

Before you migrate your data, you must create a database system with a virtual machine shape on Oracle Cloud Infrastructure. Then you must set up the target database with the same schema and structure as the source database.

> **Note:**
>
> Data Integration Platform Cloud's Replicate Data task doesn't perform an initial copy of the source, so you'll get all the changes from the point of time that you start your job. If you want the target database to be exactly like the source database, then you must perform an initial load first. For initial load, you can use the opcmigrate tool to create a copy of the source database. This method creates the database for you. See Migrate Databases Using the Migration Tools.

If you don't want an initial load, then create a **database system** with a **virtual machine** shape type. Then create the the schema that you want to replicate from the source database. Then create the tables and the structure for this schema. Replicate Data task doesn't perform any DDL commands.

After you define the schema and structure for the target database, then perform the following steps:

1. Connect to the VM of the target database.

   You can find the IP address of the database in the database detail page. For guidance with PuTTY, see Change Private Key Format to Use with PuTTY and Connect to a Cloud VM on Windows with PuTTY. To connect to the VM with PuTTY use the opc username.
2. sudo to the oracle user with the sudo -su oracle command.
3. Ensure that the target database is running.
4. Connect to the database at the PDB level.

   For example for a PDB with name PDB1:

```
CONNECT SYS/<PASSWORD>#@PDB1 AS SYSDBA
```

5. Run the following commands to create a user called DIPC_TRG:

```
CREATE USER DIPC_TGT IDENTIFIED BY PASSWORD;
GRANT CONNECT, RESOURCE, CREATE SESSION, SELECT ANY TABLE, CREATE ANY
TABLE, CREATE ANY VIEW, CREATE ANY PROCEDURE, CREATE DATABASE LINK,
SELECT ANY DICTIONARY, EXP_FULL_DATABASE, IMP_FULL_DATABASE TO DIPC_TGT;
GRANT READ, WRITE ON DIRECTORY DATA_PUMP_DIR TO DIPC_TGT;
GRANT UNLIMITED TABLESPACE TO DIPC_TGT;
EXEC DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE ('DIPC_TGT');
```

6. Ensure that you can log in to the target database as DIPC_TRG.

Migrate the Database

To perform a real-time migration you can use the Replicate Data Task of Oracle Data Integration Platform Cloud. The Replicate Data Task captures changes in your data source and updates the target in real time with that change.

Here is a summary of steps to use the Replicate Data Task:

1. Create Connections
2. Create a Replicate Data Task
3. Run the Replicate Data Task

Create Connections

A **Connection** in Data Integration Platform Cloud is a set up to establish a connection to a data source. The first step for a migration or replication is to define the source and target Connections.

After you create your **Connections**, they are ready to be used in your data integration tasks including the Replicate Data Task.

If a source database is a PDB database, the Replicate Data Task captures the data from the source at the Container Database (CDB) level. Even though you select a schema within a PDB to be replicated, the task happens at the CDB level. Therefore, if your source is an Oracle 12c with PDB, you must create an additional CDB Connection for the Replicate Data Task. Replicate Data Task doesn't use a CDB connection for data delivery, so you don't need to set it up for the target connection.

Here are the type of Connections that you must create for your real-time migration:

Table 14-1 Source Connection Types

Source Database	Connection Type
Oracle 12c	Oracle CDB and then Oracle
Oracle 11g	Oracle

Table 14-2 Target Connection Types

Target Database	Connection Type
Oracle 12c	Oracle
Oracle 11g	Oracle

1. Find Source Database Information
2. Find Target Database Information
3. Create a CDB Connection
4. Create a Source Connection
5. Create a Target Connection

Find Source Database Information

When you create an Oracle Connection in Data Integration Platform Cloud, you must provide host name, service name and the database port number.

Here are steps to find Oracle Database Classic connection information:

1. Go to **My Services** Dashboard.
2. Open Service Console for **Database Classic**.
3. Click the name of the source Database Classic instance.
4. Copy the value from the **Public IP** field.
5. Use SSH or PuTTY to connect to the VM that hosts the source database by using its IP address.
    - **PuTTY example:** Connect to a Cloud VM on Windows with PuTTY
    - **SSH example:** `xxxx$ ssh opc@<source database IP address>`
6. Use `sudo` to become the `oracle` super user:

    ```
    [opc@<database> ~] sudo su - oracle
    [oracle@<database> ~]
    ```

7. Enter the following command:

    ```
    [oracle@<database> ~ lsnrctl status
    ```

8. Find the host name, port and service name from the output.

Table 14-3 Source Database Connection Information

Field Name for Connection	Relevant String	Section to Copy
Hostname	`Connecting to (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST = <hostname.compute-xxx.oraxxx)(PORT = <port number>))`	`hostname`

Port	`Connecting to (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST = <hostname.compute-xxx.oraxxx)(PORT = <port number>))`	`<port number>`
		`default value: 1521`

Service Name for CDB Connection	`Instance "ORCL", status ready, has 1 handler(s) for this service....`
	`Service "ORCL.xxx.oraclecloud.xxx" has 1 instance(s).`
		`ORCL.xxx.oraclecloud.xxx`

Service Name for PDB Connection	`Instance "ORCL", status ready, has 1 handler(s) for this service....`
	`Service "<PDB name>.xxx.oraclecloud.xxx" has 1 instance(s).`
		`<PDB name>.xxx.oraclecloud.xxx`
		`default value: pdb1.xxx.oraclecloud.xxx`

> **Note:**
>
> You can use the same procedure to find service name for Oracle 11g databases that don't have a PDB or CDB.

Find Target Database Information

When you create an Oracle Connection in Data Integration Platform Cloud, you must provide host name, service name and the database port number.

Here are steps to find connection information for an Oracle database system on Oracle Cloud Infrastructure:

1. Go to Oracle Cloud Infrastructure **Home** page.
2. Click the menu option for **Oracle Cloud**.
3. In the Database category, click **Bare Metal, VM, and Exadata**.
4. Copy the **Public IP** address for the target database.

5. Use SSH or PuTTY to connect to the VM that hosts the target database by using its IP address.
   - **PuTTY example:** Connect to a Cloud VM on Windows with PuTTY
   - **SSH example:** `xxxx$ ssh opc@<source database IP address>`
6. Use `sudo` to become the `oracle` super user:

   ```
   [opc@<database> ~] sudo su - oracle
   [oracle@<database> ~]
   ```

7. Enter the following command:

   ```
   [oracle@<database> ~ lsnrctl status
   ```

8. Find the host name, port and service name from the output.

Table 14-4 Target Database Connection Information

Field Name for Connection	Relevant String	Section to Copy
Hostname	`Connecting to (DESCRIPTION = (ADDRESS = (PROTOCOL = tcp)(HOST = <private_ip_address>)(PORT = <port number>))`	`<private_ip_address> default value: 10.0.0.2 or 10.0.0.3`

Port	`Connecting to (DESCRIPTION = (ADDRESS = (PROTOCOL = tcp)(HOST = <private_ip_address>)(PORT = <port number>))`	`<port number>`
		`default value: 1521`

Service Name for PDB Connection	`Instance "ORCL", status ready, has 1 handler(s) for this service....`
	`Service "<PDB name>.xxx.oraclevcn.com" has 1 instance(s).`
		`<PDB name>.xxx.oraclecloud.xxx`
		`default value: pdb1.xxx.oraclecloud.xxx`

> **Note:**
>
> You can use the same procedure to find service name for Oracle 11g databases that don't have a PDB or CDB.

Create a CDB Connection

For PDB database migration, you must first create an Oracle CDB Connection. Then you create a PDB Connection for your source database and associate this CDB Connection within your PDB connection.

> **Note:**
>
> A CDB Connection for a Replicate Data Task only works if the source agent is set up with **Oracle 12c (OGG)** binaries. You can download these binaries from the Agents page. See Download Agent Binaries

To create a CDB Connection:

1. Go to the **Agents** page of the Data Integration Platform Console.
2. Ensure that the source agent is listed and running.
3. Go to the **Home** page of the Data Integration Platform Console.
4. Click **Create** in the **Connection** tile.
5. Complete the fields in the **General Information** section.

   The Identifier is to identify this connection through a string with no spaces. If you want to edit this auto-generated field, then you must only include capital letters, numbers, and underscores (_) with no space.

   - **Agent:** `<source agent>`
   - **Type:** Oracle Database
   - **SubType:** Oracle CDB
6. In the **Connection Settings** enter the Hostname, Port and Service Name. See Find Source Database Information .
7. Enter the rest of the fields for the connection:
   - **Username:** Use the username starting with C## that you created for capturing data such as `C##GGSRC`. See Prepare Source Database.
   - **Password:** Enter the password associated with the username.
8. Click **Test Connection**.
9. If the test succeeds, then click **Save**.

Create a Source Connection

Before you can perform a migration or replication in Data Integration Platform Cloud, you must create a Connection to the source database.

> **Note:**
>
> An Oracle 12c Connection for a Replicate Data task only works if its agent is set up with the **Oracle 12c (OGG)** binaries. Similarly, an Oracle 11g Connection only works for a Replicate Data Task if its agent is set up with the **Oracle 11g (OGG)** binaries. You can download these binaries from the Agents page. See Download Agent Binaries

To create a source Connection:

1. Go to the **Agents** page of the Data Integration Platform Console.
2. Ensure that the source agent is listed and running.
3. Go to the **Home** page of the Data Integration Platform Console.
4. Click **Create** in the **Connection** tile.
5. Complete the fields in the **General Information** section.

   The Identifier is to identify this connection through a string with no spaces. If you want to edit this auto-generated field, then you must only include capital letters, numbers, and underscores (_) with no space.

   - **Agent:** `<source agent>`
   - **Type:** Oracle Database
   - **SubType:** Oracle
6. In the **Connection Settings** enter the Hostname, Port and Service Name. See Find Source Database Information .
7. Enter the rest of the fields for the connection:
   - **Username:** Use the username starting with C## that you created for capturing data such as `C##GGSRC`. See Prepare Source Database.
   - **Password:** Enter the password associated with the username.
   - **Schema Name:** You can use the default or any schema that you wish to replicate. You can only use one schema for the Replicate Data Task.
   - **CDB Connection:** The CDB connection you created for the source connection, if your source is a PDB.
8. Click **Test Connection**.
9. If the test succeeds, then click **Save**.

Create a Target Connection

Before you can perform a migration or replication in Data Integration Platform Cloud, you must create a Connection to the target database.

> **Note:**
>
> An Oracle 12c Connection for a Replicate Data task only works if its agent is set up with the **Oracle 12c (OGG)** binaries. Similarly, an Oracle 11g Connection only works for a Replicate Data Task if its agent is set up with the **Oracle 11g (OGG)** binaries. You can download these binaries from the Agents page. See Download Agent Binaries

To create a target Connection:

1. Go to the **Agents** page of the Data Integration Platform Console.
2. Ensure that the target agent is listed and running.
3. Go to the **Home** page of the Data Integration Platform Console.
4. Click **Create** in the **Connection** tile.
5. Complete the fields in the **General Information** section.

   The Identifier is to identify this connection through a string with no spaces. If you want to edit this auto-generated field, then you must only include capital letters, numbers, and underscores (_) with no space.

   - **Agent:** `<target agent>`
   - **Type:** Oracle Database
   - **SubType:** Oracle
6. In the **Connection Settings** enter the Hostname, Port and Service Name. See Find Target Database Information.
7. Enter the rest of the fields for the connection:
   - **Username:** Use the username you created for Data Integration Platform Cloud such as `DIPC_TGT`. See Prepare Target Database.
   - **Password:** Enter the password associated with the username.
   - **Schema Name:** Use the same schema as the source.
   - **CDB Connection:** Leave blank.
8. Click **Test Connection**.
9. If the test succeeds, then click **Save**.

Create a Replicate Data Task

Here’s how you can set up a Replicate Data Task in Data Integration Platform Cloud.

1. Go to the **Home** page of the **Data Integration Platform Console**.
2. Click **Create** in the **Replicate Data** tile.

3. Name and describe your task in the in the **General Information** section. The Identifier is to identify this task through a string with no spaces. If you want to edit this auto-generated field, then you must only include capital letters, numbers, and underscores (_) with no space.
4. Select an encryption level for the captured data that's sent across the network. The higher the number, the harder it is for hackers to decrypt the data. Choosing **None** will send your original data, without encryption to the target.
5. If you choose an encryption level, then generate a wallet according to Replicate Data - Generate Wallet and Master Key instructions, before you run the Replicate Data task.
6. Click **Design** to select your connections and mapping pattern.
7. Click **Source** to display properties for the source database.
8. In the **Properties** for Source, in the **Details** tab, select the **PDB Connection** that you have created for the source.
9. In the **Properties** for Source, in the **Schemas** tab, select a schema from the list and then click the plus icon to add it as a mapping rule.

   For example, if you select schema `A` and then click the plus icon, the rule `Include A.*` appears in the **Rule Applied** section.

   Only **one rule** is allowed, so only one schema can be included in a mapping rule. You can further narrow down a mapping rule for a schema by including a pattern. For example, instead of `A.*`, you can edit the field to `A.EMP*` to only include tables in schema `A` that start with `EMP`.

   To reset a mapping rule, you can click **Reset**. The **default** rule uses the schema assigned to the **Source Connection**. You can edit the default schema for a Connection in its detail page or override the rule here by replacing it with another schema by using the plus icon.
10. Click **Target** to display properties for the target database.
11. In the Properties for **Target**, in the **Details** tab, select the target database Connection.
12. In the Properties for **Target**, in the **Schemas/Topics** tab, select a schema in the target for an **auto-match** mapping pattern. See Replicate Data - What's Auto-match?:
13. For Mapping Pattern, select **auto-match**.
14. Click **Save** to save your task and run it later.

Run the Replicate Data Task

Here are the steps to run a Replicate Data Task:

1. Go to the **Catalog** page of the **Data Integration Platform Console**.
2. Change the filter option from **All** to **Tasks**.
3. Click the name of your **Replicate Data Task**.
4. If you want to change the schema, or view the details of this task, click **Edit**.
5. If you are happy with the setup, then click **Run**.
6. Wait until the **Start Delivery** option in the Monitor page shows a **Running** status.

7. Connect to the target database with SQL Developer. See Connect to a Database Cloud Service with Oracle SQL Developer. .

8. Go to the schema you chose for the target and perform some queries to monitor the replication.

Related Topics

- Monitor a Replicate Data Task

15

Learn About Migrating a Multi-Node Database Cloud Service Deployment to Virtual Machine Database System

If you want to migrate a two-node Oracle Real Application Cluster (Oracle RAC) database created using Oracle Database Classic Cloud Service to an Oracle Cloud Infrastructure Virtual Machine Database System, then you can perform the database migration by using Oracle Data Guard.

For more information about the source and target databases, see the following table:

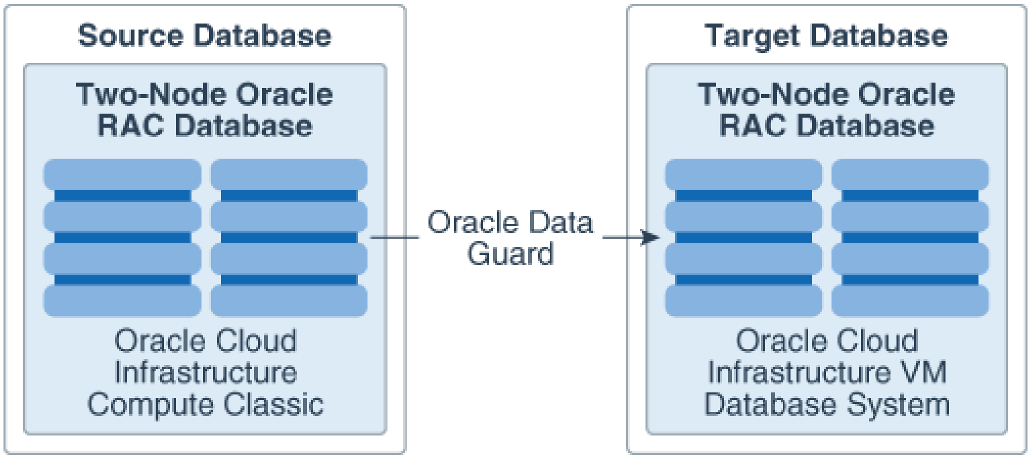
Information	Source Database	Target Database
Platform	Oracle Cloud Infrastructure Compute Classic	Oracle Cloud Infrastructure Compute
Database Type	Oracle Real Application Cluster (RAC) Database on Database Classic Cloud Service Classic	Oracle RAC Database on Virtual Machine Database System
Creation Mechanism	Database Classic Cloud Service UI, CLI, API	Oracle Cloud Infrastructure UI, CLI, API
Size	2 Node	2 Node

Architecture

You can migrate Oracle Database releases 12.1.0.2 and 12.2.0.1. Before you migrate your database, you must have Oracle RAC database on Oracle Database Classic Cloud Service and a two-node Oracle RAC Database running on Oracle Cloud Infrastructure.

When you use Oracle Data Guard to perform the migration, the source database is the primary database, and the target database is the standby database.

The following diagram shows the migration process:


Source Database
Two-Node Oracle RAC Database
Oracle Cloud Infrastructure Compute Classic
Oracle Data Guard
Target Database
Two-Node Oracle RAC Database
Oracle Cloud Infrastructure VM Database System

To perform the migration, you must follow these general steps:

1. Plan the migration.
   When you plan the database migration, you begin by inventorying the source environment (the primary database) and then you decide on the best migration strategy. To inventory the source environment, you must perform tasks such as determining the sizes of database files and checking which disaster recovery plans are in place. To decide on the best strategy, you should plan, for example, the best time of day to perform the migration.
2. Prepare for the migration.
   To prepare to migrate the source database (the primary database) to the target environment (the standby database), you must perform tasks such as ensuring that the database that you want to migrate is running, installing the latest patches for both databases so that they are patched at the same level, and ensuring that the 1521 port is open between the primary database and the standby database. In this solution, the net service name for the source (primary) database is `OCIC-ORCL` and the net service name for the target (standby) database is `OCI-ORCL` .

   > **Note:**
   >
   > Oracle recommends using the same database name for both databases so that applications can automatically fail over to the new database.

3. Perform the migration.
   You can perform the database migration by configuring the primary database (the source database) and the standby database (the target database) for Oracle Data Guard, copying the TDE wallets from the primary database to the standby database, and then completing the standby database configuration.

Required Tools to Perform the Migration

Before you begin the database migration, in addition to knowledge of Oracle Cloud Infrastructure, you must have knowledge in several areas of Oracle Database tools.

The tools with which you must be familiar are as follows:

- SQL*Plus

- Oracle Data Guard
- Oracle Flashback Technology and `spfiles`
- Familiarity with using the `srvctl` and `dgmgrl` utilities
- Familiarity with editing the `tnsnames.ora`, `listener.ora`, `sqlnet.ora`, and `oraenv` files
- Familiarity with performing Oracle Data Guard switchover operations
- (Optionally) Familiarity with generating Oracle Automatic Workload Repository and Oracle Automatic Database Diagnostic Monitor reports
- (Optionally) Familiarity with Oracle Automatic Storage Management Cluster File System (Oracle ACFS)

About Required Services and Roles

This solution requires Oracle Cloud Infrastructure.

Role	Required to...
Administrator (`SYSDBA` and `SYSOPER` privileges)	Perform SYS-related administration tasks.
Administrator (`SYSDBG` privilege)	Perform Oracle Data Guard tasks, if you are using Oracle Database 12c release 2 (12.1.0.2) or later.
Administrator (`SYSKM` privilege)	Perform Transparent Data Encryption tasks, if you are using Oracle Database 12c release 2 (12.1.0.2) or later.

Plan

To plan the Oracle RAC database migration, you must inventory the source environment and decide on the best migration strategy.

Inventory the Source Environment

Inventorying the environment includes tasks such as ensuring that you have the supported Oracle Database versions and configurations.

- Ensure that you have the supported versions and configurations.

  At a minimum, you should have at least Oracle Database 12c release 1 (12.1.0.2) (standalone).
- Determine the size of the database files of the source database.

  You can find the total size of the database files of the database that you plan to migrate, including redo log files sizes, by executing the `du -s` command at the command line. For example:

  ```
  du -s /u01/app/oracle/*
  ```

  This value provides information about how much space to allocate for the target database system. Check the name and sizes of data files by querying the

V$DATAFILE and V$TEMPFILE dynamic views. If you are using Oracle Automatic Storage Management, then check the data files used by ASM as well.

- Determine the workload level.
  You can generate an Oracle Automatic Workload Repository report to find a sample of the workload for the source database. Alternatively, if you have an Oracle Diagnostics Pack and Oracle Tuning Pack license, you can generate an Automatic Database Diagnostic Monitor report to find the source database performance over a period of time between specified snapshots. The time model statistics, operating system statistics, and wait events provide a relatively clear measure of the workload, in terms of the operating system capacity.
- Determine the environment variables that have been set in the source database.

  You may want to use these same settings in the target database.
- Check the database character set.

  You can find the database character set by issuing the following query:

  ```
  SELECT * FROM NLS_DATABASE_PARAMETERS;
  ```

  You will need to ensure that the target database will also have this character set.
- Determine the disaster recovery plan that is currently in place.
  For example, if Oracle Data Guard is already deployed, then you can create a standby database for the migration procedure. (This migration solution will use Oracle Data Guard for the migration.) If off-site backups are used, then you should plan on making a new backup to ship to Oracle Cloud, using Oracle Recovery Manager (Oracle RMAN).

Decide on the Best Migration Strategy

After you inventory your environment, you should decide on the best migration strategy.

Consider the following before you begin the migration process:

- Take a backup of your current RAC database before starting the migration
- The best time of day to perform the migration
- Downtime requirements
- Database size
- The source database and the target database platform (endian)
- Security considerations
- A strategy for large workloads

Prepare Oracle RAC

To prepare for the migration of a Oracle Real Application Cluster (Oracle RAC) database to an Oracle Cloud Infrastructure server, you must perform multiple preparatory tasks before migration can start.

Add Entries for the Database Instances

Update the `/etc/oratab` file on the Oracle RAC nodes and add an entry for your database instance as follows:

1. Use SSH to sign in to the first node of the source database (the primary database) to be migrated.
2. Update `/etc/oratab` to add the database instance ID to the database entry by doing the following:
   1. Edit `/etc/oratab`:

      ```
      sudo vi /etc/oratab
      ```

   2. Add an entry for your database in the following format:

      ```
      $ORACLE_SID:$ORACLE_HOME:N
      ```

      Example for node 1:

      ```
      orcl1:/u01/app/oracle/product/12.1.0.2/dbhome_1:N
      ```

      Example for node 2:

      ```
      orcl2:/u01/app/oracle/product/12.1.0.2/dbhome_1:N
      ```

3. Run the `oraenv` script to set environment variables of the database such as `$ORACLE_HOME`:

   ```
   . oraenv
   ```

4. Repeat the previous steps for the second Oracle RAC node.

Ensure That the Database to be Migrated Is Running

Before you begin the migration process, you must check that the source database (the primary database) to be migrated is running.

1. Use SSH to sign in to the server where the source database (the primary database) to be migrated is located.
2. Sign in as the database software owner `oracle`.

   ```
   sudo su - oracle
   ```

3. Execute the oraenv script, which sets the $ORACLE_HOME environment variable.

   ```
   . oraenv
   ```

4. Go to the $ORACLE_HOME location.

   ```
   cd $ORACLE_HOME
   ```

   If the $ORACLE_HOME location has not been set, then use the oraenv script (located in the /usr/local/bin directory) to set the environment, including $PATH, so the lsnrctl and sqlplus commands can resolve without using full path names.

5. Check the listener status.

   ```
   srvctl status listener
   ```

6. If the listener is not running (for example, the output has error TNS-12541: TNS:no listener), then switch to OS user grid and start the listener.

   ```
   sudo su - grid
   srvctl start listener
   ```

7. Check that the database is running.

   ```
   sqlplus / as sysdba
   ```

   This command should connect you to the database instance and the SQL> prompt should appear.

8. Check if the database is running in Read Write Mode

   ```
   SELECT NAME, OPEN_MODE FROM V$DATABASE;
   ```

   Output similar to the following appears:

   ```
   NAME              OPEN_MODE
   --------------    ---------
   source_db_name    READ WRITE
   ```

9. Exit the SQL*Plus

   ```
   EXIT
   ```

Ensure That All Database Components on the Source Database Are Installed on the Target Database

You can find the components that are installed on the source database (the primary database) by querying the DBA_REGISTRY data dictionary view.

1. Use SSH to sign in to the source database server.
2. Sign in to SQL*Plus as an administrator user.

For example:

```
sqlplus sys / as sysdba
Password: password
```

3. Make a note of the version of the software that is displayed in the opening banner.
4. Exit SQL*Plus.

   ```
   EXIT
   ```

5. Use the `opatch` inventory command to find the latest patch set that has been applied.

   For example:

   ```
   $ORACLE_HOME/OPatch/opatch lsinventory
   ```

6. Repeat these steps on the second node of the source database and both nodes of the target database (the standby database).
7. The second node of the source database and both nodes of the target database should have the **same or later** versions installed.

Create a Standby Database for the Oracle Cloud Infrastructure System

You must create a standby database (the target database) on the Oracle Cloud Infrastructure, in addition to the database that is currently on this system. The creation process for creating this database creates a starter database during provisioning. Create the database system with the host name, shape, and CPU count that your site requires.

Generate SSH Key Pair

To gain local access to the tools, utilities and other resources on Oracle Cloud Infrastructure Virtual Machine Database system, you use Secure Shell (SSH) client software to establish a secure connection and log in as the user `oracle` or the user `opc`. To access the standby Virtual Machine Database system, using SSH, you have to use SSH key pair instead of a password to authenticate a remote user. A key pair consists of a private key and public key. You keep the private key on your computer and provide the public key every time you launch an instance. To create key pairs, you can use a third-party tool such as OpenSSH on UNIX-style systems (including Linux, Solaris, BSD, and OS X) or PuTTY Key Generator on Windows.

Create Virtual Cloud Network

When you work with Oracle Cloud Infrastructure, one of the first steps is to set up a virtual cloud network (VCN) for your cloud resources. Ensure that you have set up a VCN before creating a standby database. You can refer to Oracle Cloud Infrastructure documentation for more information on how to create a VCN.

Verify the Virtual Machine Database Shapes Supported by your Tenancy

When you sign up for Oracle Cloud Infrastructure, a set of service limits are configured for your tenancy. The service limit is the quota or allowance set on a resource. For example, your tenancy is allowed a maximum number of compute instances per availability domain. These limits are generally established with your Oracle account representative when you purchase Oracle Cloud Infrastructure.

When you create a standby Virtual Machine Database system, you have to ensure that the Virtual Machine database shape that you select, should closely map to the primary(source) instance. You also MUST ensure that the selected shape is supported by your tenancy.

Verify your tenancy limits and usage (by region):

> **Note:**
>
> If a given resource type has limits per availability domain, the limit and usage for each availability domain is displayed.

1. Open the Oracle Cloud Services Dashboard. Open the **User** menu and click Tenancy: <your_tenancy_name>.
2. Click **Service Limits** on the left side of the page.
   Your resource limits and usage for the specific region are displayed, broken out by service.
3. Click on **Database**, and verify the Virtual Machine database shapes supported by your tenancy.

Your selection of the standby database shape should be a combination of shape that closely maps to primary(source) instance shape along with supported database shapes in your tenancy. Virtual Machine Database system is available in fixed data sizing shapes. Ensure that the shape chosen for creating database should be able to accommodate the source database plus any future sizing requirements. A thumb rule is to use a shape similar or higher in size than source database.

Create Standby Virtual Machine Database System

> **Note:**
>
> Ensure the database is created with the same parameters, such as character set, as the primary database (the source database).

1. Login to your Oracle Cloud Services Dashboard
2. Open the navigation menu. Under Services, click **Database** (NOT Database Classic).
3. Under **Database**, click **Bare Metal, VM, and Exadata**.
4. Select the **compartment** in which you want to work.

5. Click **Launch DB System**.
6. In the **Launch DB** System wizard, enter the following:
    1. **DB System Information**
        - **Compartment**: By default, the DB system launches in your current compartment and you can use the network resources in that compartment. Click the **click here** link in the dialog box if you want to enable compartment selection for the DB system, network, and subnet resources.
        - **Display Name**: A friendly, display name for the DB system. The name doesn't need to be unique. An Oracle Cloud Identifier (OCID) will uniquely identify the DB system.
        - **Availability Domain**: The **availability domain** in which the DB system resides.
        - **Shape Type**: Select **Virtual Machine**
        - **Shape**: The shape to use to launch the DB system. The shape determines the type of DB system and the resources allocated to the system. Choose the Virtual Machine Database Shape that you identified from the previous section.
        - **Total Node Count**: The number of nodes in the DB system. The number depends on the shape you select. You must specify 2.
        - **Oracle Database Software Edition**: The database edition supported by the DB system. Choose a database edition which is same or higher than the primary database.
        - **Available Storage Size (GB)**: Enter a size with at least the same size as your primary (source) server.
        - **License Type**: The type of license you want to use for the DB system. Your choice affects metering for billing.
            - **License included** means the cost of the cloud service includes a license for the Database service.
            - **Bring Your Own License (BYOL)** means you are an Oracle Database customer with an Unlimited License Agreement or Non-Unlimited License Agreement and want to use your license with Oracle Cloud Infrastructure. This removes the need for separate on-premises licenses and cloud licenses.
        - **SSH Public Key**: The public key portion of the key pair you want to use for SSH access to the DB system. Use the public key that you generated in the previous section.
    2. **Network Information**
        - **Virtual Cloud Network:** The VCN in which to launch the DB system. Select the VCN that you created in the previous section.
        - **Subnet Compartment**: The compartment containing a subnet within the cloud network to attach the DB system to.
        - **Client Subnet:** The subnet to which the DB system should attach.
        - **Hostname Prefix**: Your choice of host name for the DB system. The host name must begin with an alphabetic character, and can contain only alphanumeric characters and hyphens (-).

    - The maximum number of characters allowed is 30. The host name must be unique within the subnet. If it is not unique, the DB system will fail to provision.

  - **Host Domain Name**: The domain name for the DB system. If the selected subnet uses the Oracle-provided Internet and VCN Resolver for DNS name resolution, this field displays the domain name for the subnet and it can't be changed. Otherwise, you can provide your choice of a domain name. Hyphens (-) are not permitted.
  - **Host and Domain URL**: Combines the host and domain names to display the fully qualified domain name (FQDN) for the database. The maximum length is 64 characters.

c. **Database Information**

  - **Database Name**: The name for the database. The database name must begin with an alphabetic character and can contain a maximum of eight alphanumeric characters. Special characters are not permitted. Specify a name that is different from the primary database (the source database) name.
  - **Database Version**: The version of the initial database created on the DB system when it is launched. After the DB system is active, you can create additional databases on it. You can mix database versions on the DB system, but not editions.
  - **PDB Name**: Omit this setting because the pluggable database (PDB) will be created later on, when you perform the Oracle RMAN duplicate step.
  - **Database Admin Password**: Enter the same SYS password that is used for the primary database (the source database). It should be a strong password for SYS, SYSTEM, TDE wallet, and PDB Admin. The password must be 9 to 30 characters and contain at least 2 uppercase, 2 lowercase, 2 numeric, and 2 special characters. The special characters must be _, #, or -. The password must not contain the username (SYS, SYSTEM, and so on) or the word "oracle" either in forward or reversed order and regardless of casing. (If the primary database SYS password does not fit this requirement, then you can change it after you complete these settings.)
  - **Confirm Database Admin Password**: Re-enter the Database Admin Password you specified.
  - **Automatic Backup**: Check the check box to enable automatic incremental backups for this database.
  - **Database Workload**: Select the workload type that best suits your application.
    - Online Transactional Processing (OLTP) configures the database for a transactional workload, with a bias towards high volumes of random data access.
    - Decision Support System (DSS) configures the database for a decision support or data warehouse workload, with a bias towards large data scanning operations.
  - **Character Set**: The character set for the database. The default is AL32UTF8.

- **National Character Set**: The national character set for the database. The default is AL16UTF16.
- **Tags**: Optionally, you can apply tags. If you have permissions to create a resource, you also have permissions to apply free-form tags to that resource. To apply a defined tag, you must have permissions to use the tag namespace. For more information about tagging, see Resource Tags. If you are not sure if you should apply tags, skip this option (you can apply tags later) or ask your administrator.

7. Click **Launch DB System**. The DB system appears in the list with a status of Provisioning. The DB system's icon changes from yellow to green (or red to indicate errors).
8. Wait for the DB system's icon to turn green, with a status of Available, and then click the highlighted DB system name. Details about the DB system are displayed.
9. Note the IP addresses; you'll need the private or public IP address, depending on network configuration, to connect to the DB system.
10. Check the `SYS` password on the primary (source) database. If it does not meet the Oracle Cloud Infrastructure password requirements, then change it to match the password that you created for the standby database (the target database).

Ensure That Port 1522 and 1521 Is Open Between the Primary Database and the Standby Database

You must ensure that port 1522 is open on the primary database (the source database) and port 1521 on the standby database (the target database) to allow the databases to connect.

Enable Communication from the Oracle Cloud Infrastructure System to the Oracle Cloud Infrastructure Classic System

On the Oracle Cloud Infrastructure Classic system, you must open port 22 and 1522 for ingress traffic from the Oracle Cloud Infrastructure virtual machine system.

Configure the Standby Database to Primary Database Communication Settings in the Oracle Cloud My Services Console

You can use the Oracle Cloud My Services console to configure the first part of the communication settings.

1. Sign in to My Services console.
2. From the Dashboard, click **Database Classic.**
3. On Service: Oracle Database Classic Cloud Service, click **Open Service Console,** in the right-hand side of the page.
4. From the ☰ menu for your database deployment, select **Access Rules**.

   The Access Rules page is displayed.
5. Locate the ora_p2_dblistener rule to enable ingress traffic on port 1522 from the public internet.
6. From the ☰ menu for the located rule, select **Enable**.

The Enable Access Rule window is displayed.

7. Click **Enable**.

   The Enable Access Rule window closes and the rule is displayed as enabled in the list of rules. The given port on the compute node is opened to the public internet.

8. Create the source-db-ssh rule to open ingress traffic on port 22 from the public internet.

   if port 22 is not already open add this access rule.

   In the "**Create Access Rule**" wizard enter the following information:

   a. **Rule Name**: source-db-ssh
   b. **Source**: PUBLIC-INTERNET (select from drop down menu)
   c. **Destination**: The database name from the drop down menu
   d. **Destination Port(s)**: 22
   e. **Protocol**: TCP (select from drop down menu)

9. Click **Create**.

10. Set **Status** to Enabled.

11. Use SSH to sign in to the server where the standby database (the target database) is located.

12. Test SSH on port 22 from the Oracle Cloud Infrastructure system to the Oracle Database Classic Cloud Service system.

    Port 22 on the Oracle Database Classic Cloud Service system is open by default from the public internet.

    a. Transfer the SSH private key to the `/home/opc/.ssh` directory by using a secure file transfer utility such as SCP.
    b. As the Oracle Cloud Infrastructure *opc* user, test SSH by running the following command. In this example, the private key is named privateKey:

    ```
    ssh -i /home/opc/.ssh/privateKey opc@source_ip
    ```

Complete the Oracle Cloud Infrastructure to Oracle Cloud Infrastructure Classic Communication Settings on the Command Line

You must set a TCP socket size, edit the etc/host file, and update the `tnsnames.ora` file on the primary database (the source database).

1. SSH to the first node of the primary database (the source database)

2. Connect as root.

   ```
   sudo su -
   ```

3. (Optional) Set the TCP socket size. For example:

   ```
   sysctl -w net.core.rmem_max=10485760
   sysctl -w net.core.wmem_max=10485760
   ```

> **Note:**
>
> The TCP socket size is set to increase performance during the migration. This setting may not be ideal for production databases.

4. Connect as the oracle user.

   ```
   su - oracle
   ```

5. Execute the oraenv script, which sets the $ORACLE_HOME environment variable.

   ```
   . oraenv
   ```

6. On both Oracle RAC nodes, in the `$ORACLE_HOME/network/admin/tnsnames.ora` file on the primary database (the source database) on Oracle Cloud Infrastructure Classic, add a TNS entry similar to the following:

   > **Note:**
   >
   > Replace source_node1_ip, source_node2_ip, source_server_name with the parameters of the primary database (the source database). Replace target_node1_ip,target_node2_ip, target_server_name with the parameters of the standby database (the target database).

   ```
   MIGRAC_OCIC_s1 =
     (DESCRIPTION =
       (ADDRESS_LIST=
         (ADDRESS = (PROTOCOL = TCP)(HOST = source_node1_ip)(PORT = 1522))
         (ADDRESS = (PROTOCOL = TCP)(HOST = source_node2_ip)(PORT = 1522))
       )
       (CONNECT_DATA =
         (UR=A)
         (SERVER = DEDICATED)
         (SERVICE_NAME = source_service_name)
       )
     )

   MIGRAC_OCI_s2 =
     (DESCRIPTION =
       (ADDRESS_LIST=
         (ADDRESS = (PROTOCOL = TCP)(HOST = target_node1_ip)(PORT = 1521))
         (ADDRESS = (PROTOCOL = TCP)(HOST = target_node2_ip)(PORT = 1521))
       )
       (CONNECT_DATA =
         (UR=A)
         (SERVER = DEDICATED)
         (SERVICE_NAME = target_service_name)
       )
     )
   ```

> **Note:**
>
> (UR=A) is optional for RAC.

7. Test the SQL*Plus connect from the standby database (the target database) system to the primary database (target database) on port 1521.

   ```
   sqlplus sys@migrac_oci_s2 as sysdba
   Enter password: password

   SQL> SELECT NAME FROM V$DATABASE;
   ```

8. Exit SQL*Plus.

   ```
   exit
   ```

9. Repeat these steps on the second node of the primary database (source database).

Enable Communication from the Oracle Cloud Infrastructure Classic System to the Oracle Cloud Infrastructure System

On the Oracle Cloud Infrastructure Database system, you must open ports 22 and 1521 for ingress traffic from the Oracle Cloud Infrastructure Classic system.

Configure the Oracle Cloud Infrastructure Classic to Oracle Cloud Infrastructure Communication Settings in the Oracle Cloud Infrastructure Console

You can use the Oracle Cloud Infrastructure console to configure the first part of the communication settings.

1. Make a note of the public IP address of the Oracle Cloud Infrastructure Classic server.
2. Sign in to the Oracle Cloud Infrastructure console.
3. On the left side of the page, select your compartment, and then select the **Networking** tab at the top of the page.
4. On the Virtual Cloud Networks in the Compartment page, select the name of your network.
5. On the page that is labeled with the network name, select **Security Lists** from the left navigation pane.
6. On the Security Lists page, select the list that you want to view.
7. On the Security_list_name Security List for network_name page, click **Edit all rules**.
8. Scroll to find the rule that you want to change, or click **Add Rule** for either the Ingress or Engress rule.
9. Update the database system security list to the Oracle Cloud Infrastructure Classic server public IP that you obtained in the first step.

By default, port 22 for SSH is enabled on Oracle Cloud Infrastructure systems for traffic from the public internet.

a. Set **SOURCE CIDR** to the IP address on the Oracle Cloud Infrastructure Compute Classic server.
b. Set **IP PROTOCOL** to TCP.
c. Set **SOURCE PORT RANGE** to ALL.
d. Set **DESTINATION PORT RANGE** to 1521.

Complete the Oracle Cloud Infrastructure Classic to Oracle Cloud Infrastructure Communication Settings on the Command Line

You must edit the `$ORACLE_HOME/network/admin/tnsnames.ora` file, set the TCP socket size on the standby database (the target database).

1. SSH to the first node of the standby database (the target database).
2. On the standby database (the target database), connect as the database software owner oracle.

   ```
   sudo su - oracle
   ```

3. Execute the oraenv script, which sets the $ORACLE_HOME environment variable.

   ```
   . oraenv
   ```

4. On both nodes, in the `$ORACLE_HOME/network/admin/tnsnames.ora` file on the Oracle Cloud Infrastructurevirtual machine system, add a TNS entry for each of the two databases:

   In this example, the primary database TNS name is `OCIC-ORCL` and the standby database TNS name is `OCI-ORCL`.

   > **Note:**
   >
   > Replace `source_node1_ip, source_node2_ip, source_server_name` with the parameters of the primary database (the source database).
   >
   > Replace `target_node1_ip, target_node2_ip, target_server_name` with the parameters of the standby database (the target database)

   ```
   MIGRAC_OCIC_s1 =
     (DESCRIPTION =
       (ADDRESS_LIST=
         (ADDRESS = (PROTOCOL = TCP)(HOST = source_node1_ip)(PORT = 1522))
         (ADDRESS = (PROTOCOL = TCP)(HOST = source_node2_ip)(PORT = 1522))
       )
       (CONNECT_DATA =
         (UR=A)
         (SERVER = DEDICATED)
         (SERVICE_NAME = source_service_name)
       )
   ```

```
  )

MIGRAC_OCI_s2 =
  (DESCRIPTION =
    (ADDRESS_LIST=
      (ADDRESS = (PROTOCOL = TCP)(HOST = target_node1_ip)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = target_node2_ip)(PORT = 1521))
    )
    (CONNECT_DATA =
      (UR=A)
      (SERVER = DEDICATED)
      (SERVICE_NAME = target_service_name)
    )
  )
```

5. Use SQL*Plus to test connection from the Oracle Cloud Infrastructure system to the Oracle Cloud Infrastructure Classic database on port 1521.

   ```
   sqlplus sys@migrac_orcl_s2 as sysdba
   Enter password: password
   ```

   (At this stage, any SQL statements you execute will fail because the standby database has not been fully configured yet.)

6. At the command line, connect as `root`.

   ```
   sudo su -
   ```

7. (Optional) On both nodes, set the TCP socket size.For example:

   ```
   sysctl -w net.core.rmem_max=10485760
   sysctl -w net.core.wmem_max=10485760
   ```

   > **Note:**
   >
   > The TCP socket size is set to increase performance during the migration. This setting may not be ideal for production databases.

8. Repeat these steps on the second node of the standby database (target database).

Ensure That Bundle Patches Have Been Applied and Are in Sync

Ensure that the patch level on the primary database (the source database) Oracle Cloud Infrastructure Classic system is earlier or the same as the patch level on the standby database (the target database) Oracle Cloud Infrastructure system.

1. Use SSH to sign in to the first node of the primary database (the source database).
2. Check the patch level on the primary database as follows:

a. To find a brief listing of patches:

   ```
   $ORACLE_HOME/OPatch/opatch lspatches
   ```

b. To find a detailed listing of patches:

   ```
   $ORACLE_HOME/OPatch/opatch lsinventory
   ```

3. Make a note of the patch level.
4. Repeat steps 1 to 3 for the second node of the primary database (the source database).
5. Use SSH to sign in to the nodes of the standby database (the target database).
6. Check the patch level on the nodes of the standby database by running the opatch `lsinventory` command.
7. Compare the patch levels of the nodes of the databases. Ensure that the standby system has a bundle patch that is either equal to or later than the bundle patch that is on the primary database.
8. If you must install a later patch on the standby database (the target database), then access My Oracle Support: https://support.oracle.com/
9. Download the correct version of the patch to the standby database (the target database).
10. Extract the bundle patch.
11. List the available patches.

    ```
    $ORACLE_HOME/OPatch/opatch lspatches
    ```

12. Apply the patch.

    ```
    $ORACLE_HOME/OPatch/opatch apply patch_number
    ```

Migrate Oracle RAC

To perform the migration of a Oracle RAC database from an Oracle Cloud Infrastructure Classic server to an Oracle Cloud Infrastructure Database system, you can use Oracle Data Guard. You must configure the database on Oracle Cloud Infrastructure Classic as the primary database (the source database), which you migrate to a standby database (the target database) on Oracle Cloud Infrastructure on virtual machine systems.

Configure the Primary (Source) Database

To configure the primary database (the source database), you configure Oracle Data Guard and modify the `listener.ora` and `tnsnames.ora` files for the standby database (the target database).

Configure the Primary Database for the Standby Database

In this configuration, you configure the primary (source) database to use Oracle Data Guard.

1. Use SSH to sign in to the primary database (the source database) server.
2. On the standby database (the target database), connect as the database software owner oracle.

   ```
   sudo su - oracle
   ```

3. Execute the oraenv script, which sets the $ORACLE_HOME environment variable.

   ```
   . oraenv
   ```

4. Sign in to the database instance as a user who has administrator privileges. For example:

   ```
   sqlplus / as sysoper
   ```

5. Ensure that the database is in `ARCHIVELOG` mode.

   ```
   ARCHIVE LOG LIST
   ```

6. If the output for `Database log mode` is `No Archive Mode` and the output for `Automatic archival` is `Disabled`, then do the following:
   a. Exit SQL*Plus

      ```
      SQL> exit
      ```

   b. Shut down the database.

      ```
      srvctl stop database -db db_unique_name
      ```

   c. Restart the database in mount mode

      ```
      srvctl start database -db db_unique_name -o mount
      ```

   d. Sign in to the database instance as a user who has administrator privileges. For example:

      ```
      sqlplus / as sysoper
      ```

   e. Enable archive log mode.

      ```
      ALTER DATABASE ARCHIVELOG;
      ```

   f. Ensure that the database is now in archive log mode.

      ```
      ARCHIVE LOG LIST
      ```

The output for the output for Database log mode should be `Archive Mode` and the output for `Automatic archival` is `Enabled`.

g. Open the database.

```
ALTER DATABASE OPEN;
```

7. Connect with the `SYSDBA` administrator privilege.

```
CONNECT / AS SYSDBA
```

8. For a multitenant environment, do the following:

   a. Check the status of the PDBS.

   ```
   SHOW PDBS
   ```

   b. If the PDBS are not open, then open them.

   ```
   ALTER PLUGGABLE DATABASE ALL OPEN;
   ```

9. Ensure that the database is in force logging mode. For example:

```
SELECT NAME, OPEN_MODE, FORCE_LOGGING FROM V$DATABASE;
```

10. If necessary, enable `force logging`.

```
ALTER DATABASE FORCE LOGGING;
```

11. Check the configuration.

```
SELECT NAME, CDB, OPEN_MODE, FORCE_LOGGING FROM V$DATABASE;
```

The `FORCE_LOGGING` column should be `YES`.

12. Use the `SHOW PARAMETER` command to check the following database parameters:

   a. `DB_NAME` and `DB_UNIQUE_NAME`: Ensure that these names are different from the names that are used on the target database.

   b. `REMOTE_LOGIN_PASSWORDFILE`: This parameter must be set to `EXCLUSIVE`.

Add Static Services to the Primary Database listener.ora File

In this section, you must add a new static listener to `listener.ora` and restart the listener.

1. Use SSH to sign in to the first node of the primary (source) database.
2. At the command line, connect as `grid user`.

```
sudo su - grid
```

3. Modify the `/u01/app/12.2.0.1/grid/network/admin/listener.ora` file to include the static listener. The following example shows the format to use for one static listener:

   ```
   SID_LIST_LISTENER =
     (SID_LIST =
       (SID_DESC =
         (SDU=65535)
         (GLOBAL_DBNAME = source_db_unique_name.source_db_domain)
         (ORACLE_HOME = source_oracle_home)
         (ENVS="TNS_ADMIN= source_oracle_home/network/admin")
         (SID_NAME = source_db_name)
       )
     )
   ```

4. Stop the listener.

   > **Note:**
   >
   > Stopping and starting the static listener can affect new connections to the database for a few seconds.

   ```
   srvctl stop listener -l LISTENER
   ```

5. Restart the listener.

   ```
   srvctl start listener -l LISTENER
   ```

6. Check the listener status.

   ```
   lsnrctl status
   ```

7. Validate that there are entries in the output with the status `UNKNOWN`.
8. Repeat the previous steps for the second node of the primary database (the source database).

Configure the Primary Database Parameters

After you configure the primary (source) database and add static services to the primary database `listener.ora` file, you can configure the Oracle Data Guard parameters on the primary database.

1. Use SSH to sign in to the first RAC node of the primary database (the source database).
2. Sign in to the database instance as a user who has the `SYSDBA` administrator privilege.

   ```
   sqlplus / as sysdba
   Enter password: password
   ```

3. Enable automatic standby file management.

   ```
   ALTER SYSTEM SET STANDBY_FILE_MANAGEMENT=AUTO SID='*' SCOPE=BOTH;
   ```

4. Set the archive lag target.

   ```
   ALTER SYSTEM SET ARCHIVE_LAG_TARGET=1800 SID='*' SCOPE=BOTH;
   ```

5. Identify the Oracle Broker configuration file names and locations. The following statements depend on the type of database storage.

   ```
   ALTER SYSTEM SET DG_BROKER_CONFIG_FILE1='/u02/app/oracle/oradata/
   <db_unique_name>/dr1<db_unique_name>.dat' SID='*' SCOPE=BOTH;
   SCOPE=BOTH;
   ALTER SYSTEM SET DG_BROKER_CONFIG_FILE2='/u03/app/oracle/
   fast_recovery_area/<db_unique_name>/dr2<db_unique_name>.dat' SID='*'
   SCOPE=BOTH;
   ```

6. Enable the Oracle Broker DMON process for the database.

   ```
   ALTER SYSTEM SET DG_BROKER_START=TRUE SCOPE=BOTH;
   ```

7. (Optional) Set the `DB_BLOCK_CHECKING` and `DB_BLOCK_CHECKSUM` parameters.

   ```
   ALTER SYSTEM SET DB_BLOCK_CHECKING=FULL SID='*' SCOPE=BOTH;
   ALTER SYSTEM SET DB_BLOCK_CHECKSUM=FULL SID='*' SCOPE=BOTH;
   ```

8. (Optional) Set the log buffer to 256 megabytes.

   ```
   ALTER SYSTEM SET LOG_BUFFER=268435456 SID='*' SCOPE=BOTH;
   ```

9. Set the `DB_LOST_WRITE_PROTECT` parameter to `TYPICAL`.

   ```
   ALTER SYSTEM SET DB_LOST_WRITE_PROTECT=TYPICAL SID='*' SCOPE=BOTH;
   ```

10. Enable the database flashback feature. The minimum recommended value for `DB_FLASHBACK_RETENTION_TARGET` is 120 minutes.

    ```
    ALTER DATABASE FLASHBACK ON;
    ALTER SYSTEM SET DB_FLASHBACK_RETENTION_TARGET=120;
    ALTER SYSTEM ARCHIVE LOG CURRENT;
    ```

11. Add the standby redo logs, based on the online redo log. You can use the query below to determine the number and size (in bytes) of the ORLs. The size of the standby redo logs must be the same as the online redo logs, but you must add one or more additional standby redo logs than there are online redo logs. In the following example, four online redo logs exist, so you must add at least five standby redo logs. In other words, for each thread, you must specify the current redo logs plus at least one, and then use the same size for it as the original redo logs.

a. Execute the following query to determine the number, and size in bytes, of the Oracle redo logs.

```
SELECT GROUP#, BYTES FROM V$LOG;
```

The output should be similar to the following.

```
GROUP# BYTES
------ ----------
1      1073741824
2      1073741824
3      1073741824
4      1073741824
```

b. For the first thread, specify the current redo logs plus one more, and use the same size as the current redo logs. For example:

```
ALTER DATABASE ADD STANDBY LOGFILE THREAD 1
GROUP 5 SIZE 1073741824,
GROUP 6 SIZE 1073741824,
GROUP 7 SIZE 1073741824,
GROUP 8 SIZE 1073741824,
GROUP 9 SIZE 1073741824;
```

c. For the second thread, specify the current redo logs plus one more, and use the same size as the current redo logs. For example:

```
ALTER DATABASE ADD STANDBY LOGFILE THREAD 2
GROUP 10 SIZE 1073741824,
GROUP 11 SIZE 1073741824,
GROUP 12 SIZE 1073741824,
GROUP 13 SIZE 1073741824,
GROUP 14 SIZE 1073741824;
```

d. Verify that you created the correct number of standby redo logs.

```
SELECT GROUP#, BYTES FROM V$STANDBY_LOG;
```

Output similar to the following should appear:

```
    GROUP#      BYTES
---------- ----------
         5 1073741824
         6 1073741824
         7 1073741824
         8 1073741824
         9 1073741824
        10 1073741824
        11 1073741824
        12 1073741824
        13 1073741824
        14 1073741824
10 rows selected.
```

Configure the Standby (Target) Database

To configure the standby (target) database, you must drop the standby database and then modify the oratab, listener.ora, and tnsnames.ora files.

Add Entries for the Database Instances

Update the /etc/oratab file on the Oracle RAC nodes and add an entry for your database instance as follows:

1. Use SSH to sign in to the first node of the source database (the primary database) to be migrated.
2. Update /etc/oratab to add the database instance ID to the database entry by doing the following:
    a. Edit /etc/oratab:

    ```
    sudo vi /etc/oratab
    ```

    b. Add an entry for your database in the following format:

    ```
    $ORACLE_SID:$ORACLE_HOME:N
    ```

    Example for node 1:

    ```
    orcl1:/u01/app/oracle/product/12.1.0.2/dbhome_1:N
    ```

    Example for node 2:

    ```
    orcl2:/u01/app/oracle/product/12.1.0.2/dbhome_1:N
    ```

3. Run the oraenv script to set environment variables of the database such as $ORACLE_HOME:

    ```
    . oraenv
    ```

4. Repeat the previous steps for the second Oracle RAC node.

Drop the Standby (Target) Database

1. Use SSH to log in to the standby (target) server.
2. Switch to the oracle user that is the database owner.
3. Sign in to the database instance as a user who has the SYSDBA administrator privilege.

    ```
    sqlplus / as sysdba
    Enter password: password
    ```

4. Uncluster the database as follows:

   ```
   alter system set cluster_database=false sid='*' scope=spfile;
   System altered.
   ```

5. Exit SQL*Plus:

   ```
    exit
   ```

6. Stop the database using `srvctl`:

   ```
   srvctl stop database -db database_unqiue_name
   ```

7. Sign in to the database instance as a user who has the `SYSDBA` administrator privilege.

   ```
   sqlplus / as sysdba
   Enter password: password
   ```

8. Start up and mount the database in restricted mode:

   ```
    startup mount restrict
   ORACLE instance started.
   Total System Global Area 7516192768 bytes
   Fixed Size                  2941872 bytes
   Variable Size            1409289296 bytes
   Database Buffers         6073352192 bytes
   Redo Buffers               30609408 bytes
   Database mounted.
   ```

9. Drop the standby database:

   ```
    drop database;
   Database dropped.
   ```

Add Static Services to the Standby Database listener.ora File

After you add static services to the standby database (the source database) `listener.ora` file, you must restart the listener. .

1. Use SSH to sign in to the first Oracle RAC node of the standby database (the source database) server.
2. At the command line, connect as `grid` user.

   ```
   sudo su - grid
   ```

3. Modify the `$ORACLE_HOME/network/admin/listener.ora` file to include the `static listener`. The following example shows the format to use for one `static listener`:

   ```
   SID_LIST_LISTENER=
    (SID_LIST=
     (SID_DESC=
   ```

```
 (SDU=65535)
 (GLOBAL_DBNAME = standby_db_unique_name.standby_db_domain)
 (SID_NAME = standby_db_sid)
 (ORACLE_HOME= standby_oracle_home)
 (ENVS="TNS_ADMIN= standby_oracle_home/network/admin")
 )
)
```

4. Use the `srvctl` utility to stop the listener.

   ```
   srvctl stop listener -l LISTENER
   ```

5. Restart the `listener`.

   ```
   srvctl start listener -l LISTENER
   ```

6. Check the `listener` status.

   ```
   lsnrctl status
   ```

   > **Note:**
   >
   > output is the new listener in the status UNKNOWN

7. Repeat these steps for the second Oracle RAC node.

Copy TDE Wallets from the Primary Database to the Standby Database

You can manually copy the TDE wallet files from the primary database (the source database) system to the standby database (the target database) system by using Secure Copy Protocol (SCP).

Compress the TDE Wallet

You must perform this operation in the primary database (the source database).

1. Use SSH to sign in to the primary database (the source database) server.
2. At the command line, connect as the database software owner `oracle`.

   ```
   sudo su - oracle
   ```

3. Execute the oraenv script, which sets the $ORACLE_HOME environment variable.

   ```
   . oraenv
   ```

4. To find the wallet location, sign in to the primary database (the source database) instance with the `SYSDBA` administrator privilege.

   ```
   sqlplus / as sysdba
   ```

5. Query the WRL_PARAMETER column of the V$ENCRYPTION_WALLET dynamic view to find the directory where the wallet is located.

   ```
   SELECT * FROM V$ENCRYPTION_WALLET;
   ```

6. Exit SQL*Plus.

   ```
   exit
   ```

7. Go to the directory where the wallet files are located. For example:

   ```
   cd /u02/app/oracle/admin/source_db_unique_name
   ```

8. Use the tar command to compress the TDE wallet. For example:

   ```
   tar cvf tde_wallet.tar ./tde_wallet
   ```

   Output similar to the following appears:

   ```
   ./tde_wallet/
   ./tde_wallet/ewallet.p12
   ./tde_wallet/cwallet.sso
   ./tde_wallet/ewallet_2018021607225910.p12
   ```

Copy the TDE Wallet and Set Permissions on the Wallet Directory

After you back up the TDE wallet file, you must create a directory for the wallet and set permissions on this directory.

1. Copy the wallet tar file to a tmp directory. For example:

   ```
   cp tde_wallet.tar /tmp/
   ```

2. Exit to become the OPC user.

   ```
   $ exit
   ```

3. Copy the private key from your local host to the primary database.

   By default, the private keys aren't stored on the DBs

   ```
   scp -i /home/opc/.ssh/privateKey opc@<Primary DB IP>:/home/opc/.ssh/
   ```

4. Use SCP to copy the wallet files from the primary database (the source database) to the standby database (the target database), in the /opt/oracle/dcs/commonstore/wallets/tde/$ORACLE_UNQNAME directory. For example:

   ```
   scp -i /home/opc/.ssh/privateKey /tmp/tde_wallet.tar opc@<Standby DB
   IP>:/tmp/
   ```

Output similar to the following appears:

```
tde_wallet.tar
100% 20KB 20.0KB/s 00:00
```

5. Use SSH to sign in to the target database server.
6. Sign in as database software owner `oracle`.

   ```
   sudo su - oracle
   ```

7. Execute the oraenv script, which sets the $ORACLE_HOME environment variable.

   ```
   . oraenv
   ```

8. To find the wallet location, display the contents of `sqlnet.ora`:

   ```
   cat $ORACLE_HOME/network/admin/sqlnet.ora
   ```

9. The `ENCRYPTION_WALLET_LOCATION` parameter displays the location of the wallet. For example:

   ```
   ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=FILE)
   (METHOD_DATA=(DIRECTORY=/opt/oracle/dcs/commonstore/wallets/
   tde/$ORACLE_UNQNAME)))
   ```

10. Go to the target wallet directory. For example:

    ```
    cd /opt/oracle/dcs/commonstore/wallets/tde
    ```

11. Check that the correct wallet is in this directory.

    ```
    ls target_db_unique_name
    ```

12. Back up the wallet file. For example:

    ```
    mv target_db_unique_name target_db_unique_name.old
    ```

13. Create a directory in which to store the wallet. For example:

    ```
    mkdir target_db_unique_name
    ```

14. Check the permissions on the wallet directory.

    ```
    ls -ld target_db_unique_name
    ```

15. If necessary, give the database software owner oracle read, write, and execute permissions.

    ```
    chmod 700 target_db_unique_name
    ```

16. Check the permissions again.

    ```
    ls -ld target_db_unique_name
    ```

17. Copy the wallet tar file to the current directory.

    ```
    cp /tmp/tde_wallet.tar .
    ```

18. Check the permissions.

    ```
    ls -rlt
    ```

    Output similar to the following appears:

    ```
    total 124
    drwx------ 2 oracle oinstall 20480 Feb 16 09:25
    target_db_unique_name.old
    drwx------ 2 oracle oinstall 20480 Feb 16 10:16 target_db_unique_name
    -rw-r--r-- 1 oracle oinstall 20480 Feb 16 10:17 tde_wallet.tar
    ```

Complete the TDE Wallet Process

You must extract the TDE wallet file tar and then move its contents to the wallet directory on the standby database (the target database).

1. On the standby database (the target database), ensure that you are in the correct wallet directory. For example:

   ```
   pwd

   # Output similar to the following should appear:
   /opt/oracle/dcs/commonstore/wallets/tde
   ```

2. Extract the tar file.

   ```
   tar xvf tde_wallet.tar
   ```

   Output similar to the following should appear:

   ```
   ./tde_wallet/ewallet.p12
   ./tde_wallet/ewallet_2018050819024979.p12
   ./tde_wallet/cwallet.sso
   ```

3. Move the `tde_wallet` contents to the wallet directory on the standby database (the target database).

   ```
   mv ./tde_wallet/* ./target_db_unique_name
   ```

4. Remove the `tde_wallet` contents from the standby database (the target database).

   ```
   rm -rf ./tde_wallet
   ```

Configure the Standby Initialization Parameter File and Start the Instance in NOMOUNT Mode

After you configure the standby initialization file, then you can restart the database in NOMOUNT mode.

1. Use SSH to sign in to the standby database (the target database) server.
2. Sign in as the database software owner oracle.

   ```
   sudo su - oracle
   ```

3. Execute the oraenv script, which sets the $ORACLE_HOME environment variable.

   ```
   . oraenv
   ```

4. Go to the dbs directory.

   ```
   cd $ORACLE_HOME/dbs
   ```

5. Create a temporary initialization parameter file, named init_<sid>.orawhere <sid> is the target database SID as follows:

   ```
   echo "*.db_name='db_name'" > $ORACLE_HOME/dbs/init<sid>.ora
   echo "*.db_unique_name='target_db_unique_name'" >> $ORACLE_HOME/dbs/
   init<sid>.ora
   ```

6. Back up the existing password file, if one exists. For example:

   ```
   mv $ORACLE_HOME/dbs/orapw<sid> $ORACLE_HOME/dbs/orapw<sid>.old
   ```

7. Create a new password file. For example:

   ```
   orapwd file=$ORACLE_HOME/dbs/orapwtarget
   password=admin_password_for_primary entries=5
   ```

8. Change the password file in cluster to non-ASM password file:

   ```
   $ srvctl modify database -d <oci_db_unique_name> -
   pwfile $ORACLE_HOME/dbs/orapw<oci_oracle_sid>
   $ srvctl config database -d <oci_db_unique_name>
   ```

9. Connect to the standby database (the target database) instance as a user who has the sysdba administrator privilege. For example:

   ```
   sqlplus / as sysdba
   ```

10. Shut down the database. For example:

    ```
    shutdown immediate
    ```

11. Restart the database in NOMOUNT mode using the init<sid>.ora initialization parameter file.

    ```
    startup force nomount PFILE=?/dbs/init<sid>.ora
    ```

Duplicate the Target Database for the Standby from the Active Database

You can execute a script to duplicate the standby database (the target database). If the primary database (the source database) is large, then you can allocate additional channels to improve its performance. For a newly installed database, one channel typically runs the database duplication in a couple of minutes. Ensure that no errors occur after you run the Oracle Recovery Manager (Oracle RMAN) duplication operation. If errors occur, then restart the database by using the initialization parameter file (not spfile), in case it is generated under the $ORACLE_HOME/dbs directory as part of the Oracle RMAN duplication process.

1. Connect to the standby database (the target database) as the database software owner oracle.

   ```
   su - oracle
   ```

2. Execute the oraenv script, which sets the $ORACLE_HOME environment variable.

   ```
   . oraenv
   ```

3. Create dup.rcv

   ```
   vi dup.rcv
   ```

   Paste the contents of the script while updating the required variables.

4. With the standby database (the target database) in NOMOUNT mode, connect to Oracle RMAN.

   ```
   rman
   RMAN> connect target sys@<primary_db_tnsnames_name>
   target database Password: password
   RMAN> connect auxiliary sys@<standby_db_tnsnames_name>
   auxiliary database Password: password
   ```

5. Execute the following script to duplicate the target database for a standby database from an active database. The following example shows the dup.rcv script, which **must** be user-created, and is based on the My Oracle Support note 2369137. RMAN Active Duplicate Runs Into RMAN-06217 -- PUSH & PULL method Explanation (Doc ID 2369137.1). In this example, the dup.rcv script has been customized to push duplication (image copies) from the file system to Oracle Automatic Storage Management (Oracle ASM). Other options such as from the file system to Oracle Automatic Storage Management Cluster File System, or Oracle ASM to Oracle ASM, would require changes to the file destination parameters and the file name conversion parameters.

   ```
   @dup.rcv
   Output similar to the following appears:
   ```

```
RMAN> run {
2> allocate channel prmy1 type disk;
3> allocate channel prmy2 type disk;
4> allocate channel prmy3 type disk;
5> allocate channel prmy4 type disk;
6> allocate auxiliary channel stby1 type disk;
7> allocate auxiliary channel stby2 type disk;
8> allocate auxiliary channel stby type disk;
9> duplicate target database for standby from active database dorecover
10> spfile
11> parameter_value_convert '/u02/app/oracle/oradata/
source_db_name','+DATA'
12> Set CLUSTER_DATABASE='FALSE'
13> set db_unique_name='<target_db_unique_name>'
14> set db_create_file_dest='+DATA'
15> set db_create_online_log_dest_1='+RECO'
16> set db_recovery_file_dest='+RECO'
17> set audit_file_dest='/u01/app/oracle/admin/db_name/adump'
18> set control_files='+DATA','+RECO'
19> set dg_broker_config_file1='+DATA/<target_db_unique_name>/
dr1<target_db_unique_name>.dat'
20> set dg_broker_config_file2='+RECO/<target_db_unique_name>/
dr2<target_db_unique_name>.dat'
21> set dispatchers='(PROTOCOL=TCP) (SERVICE=<target_db_name>XDB)'
22> set instance_name='<target_db_name>'
23> set db_domain='<target_db_domain>'
24> set db_recovery_file_dest='+RECO'
25> ;
26> }
```

Post Oracle Recovery Manager Duplication Steps

After you complete the Oracle Recovery Manager (`Oracle RMAN`) duplication operation, you should perform these clean-up tasks on the standby database (the target database).

Update the Password File

Update the password file as follows:

1. Move the password file back to ASM:

```
[oracle@oci_node1 ~]$ cp $ORACLE_HOME/dbs/
orapw<oci_node1_oracle_sid> /tmp/orapw<oci_node1_oracle_sid>
```

2. Exit to the `opc` user:

```
[oracle@oci_node1 ~]$ exit
```

3. Switch to the `grid` user:

```
[opc@oci_node1 ~]$ sudo su - grid
```

4. Switch to ASMCMD prompt:

   ```
   [grid@oci_node1 ~]$ asmcmd
   ```

5. Copy the password in ASMCMD:

   ```
   ASMCMD> pwcopy --dbuniquename <oci_db_unique_name> '/tmp/
   orapw<oci_node1_oracle_sid>' '+DATA'
   ```

   > **Note:**
   >
   > The command may display errors as ASM is a different version than the database. The errors can be ignored as the registration issue is fixed later in this procedure.

6. Navigate to the +DATA/<OCI_DB_UNIQUE_NAME>/PASSWORD/ directory to find the system_generated_id to use in step **9**.

   ```
   ASMCMD> cd +DATA/<OCI_DB_UNIQUE_NAME>/PASSWORD/
   ASMCMD> ls -lt
   ```

7. Exit ASMCMD and the grid user:

   ```
   ASMCMD> exit
   [grid@oci_node1 ~]$ exit
   ```

8. Switch to the oracle user:

   ```
   [opc@oci_node1 ~]$ sudo su - oracle
   ```

9. Execute the oraenv script, which sets the $ORACLE_HOME environment variable.

   ```
   . oraenv
   ```

10. Modify password file in cluster to ASM:

    ```
    [oracle@oci_node1 ~]$ srvctl modify database -d <oci_db_unique_name> -
    pwfile +DATA/<OCI_DB_UNIQUE_NAME>/PASSWORD/
    pwd<oci_db_unique_name><system_generated_id>
    [oracle@oci_node1 ~]$ rm $ORACLE_HOME/dbs/orapw<oci_node1_oracle_sid>
    ```

Enable Oracle Flashback

You should enable Oracle Flashback.

1. Use SSH to sign in to the standby database (the target database) server.
2. Sign in as the database software owner oracle.

   ```
   sudo su - oracle
   ```

3. Sign in to the database instance as a user who has the SYSOPER administrator privilege. For example:

   ```
   sqlplus / as sysoper
   Enter password: password
   ```

4. Enable Oracle Flashback.

   ```
   ALTER DATABASE FLASHBACK ON;
   ```

5. Connect as a user with the SYSDBA administrator privilege.

   ```
   CONNECT / AS SYSDBA
   ```

6. Set the flashback retention target.

   ```
   ALTER SYSTEM SET DB_FLASHBACK_RETENTION_TARGET=120;
   ```

Move the spfile File to Oracle Automatic Storage Management

You should move the spfile file to Oracle Automatic Storage Management.

1. Use SSH to connect to the standby database (the target database) server.
2. Sign in to the database instance as a user who has the SYSOPER administrator privilege.
3. Create and move the spfile file to Oracle Automatic Storage Management.
   a. Create the pfile.

      ```
      create pfile='/tmp/init<sid>.ora' from spfile;
      ```

   b. Exit SQL*Plus:

      ```
      SQL> exit
      ```

   c. Edit the pfile:

      ```
      vi /tmp/init<sid>.ora
      ```

   d. Delete the following line from the file:

      ```
      *.instance_name=<sid_for_node1>
      ```

   e. Add the following lines for each node:

      ```
      <sid_for_node1>.instance_name='<sid_for_node1>'
      <sid_for_node2>.instance_name='<sid_for_node2>'
      ```

   f. Sign in to SQL*Plus as an administrator user. For example:

      ```
      sqlplus sys / as sysdba
      Password: password
      ```

g. Shut down the database

```
shutdown immediate
```

h. Restart the database in MOUNT mode by using the initdb_name.ora file that you just created.

```
startup mount pfile='/tmp/init<sid>.ora';
```

i. Create the spfile file.

```
create spfile='+DATA' from pfile='/tmp/init<sid>.ora';
```

4. Exit SQL*Plus.
5. As the grid user, find the spfile file on Oracle Automatic Storage Management by using the asmcmd command.

```
asmcmd
ASMCMD> cd +DATA/target_db_unique_name/PARAMETERFILE/
ASMCMD> ls -lt
```

Output similar to the following appears:

```
Type          Redund  Striped  Time             Sys  Name
PARAMETERFILE UNPROT  COARSE   APR 09 16:00:00  Y    spfile.
262.973010033
```

Make a note of the ASM name (spfile.262.973010033), which you will need in the next task.

Change the inittarget_db_name.ora File to Reference the spfile File

You can modify the init<sid>.ora file to reference the spfile file..

1. Use SSH to connect to the standby database (the target database) server.
2. Sign in as the database software owner oracle.

```
su - oracle
```

3. Execute the oraenv script to set the $ORACLE_HOME environment variable.

```
. oraenv
```

4. Go to the $ORACLE_HOME/dbs directory.

```
cd $ORACLE_HOME/dbs
```

5. Ensure that the init<sid>.ora file is in this directory.

```
ls *.ora
```

6. Change the inittarget_db_name.ora file to refer to the spfile file. For example:

   ```
   mv spfiletarget_db_name.ora spfiletarget_db_name.ora.stby
   mv init<sid>.ora init<sid>.ora.stby
   echo ''SPFILE='+DATA/target_db_unique_name/PARAMETERFILE/spfile.
   262.973010033'' > init<sid>.ora
   cat init<sid>inittarget_db_name.ora--To check the file
   ```

   In this output, spfile.262.973010033 is the name of the file that you generated when you moved the spfile file to Oracle Automatic Storage Management in the previous task.

   Output similar to the following appears:

   ```
   SPFILE='+DATA/target_db_unique_name/PARAMETERFILE/spfile.262.973010033'
   ```

Modify and Start the Standby Database in MOUNT Mode

You can use the srvctl to modify and start the standby database (the target database).

1. Use SSH to sign in to the standby database (the target database) server.
2. Sign in as the database software owner oracle.

   ```
   su - oracle
   ```

3. Execute the oraenv script, which sets the $ORACLE_HOME environment variable.

   ```
   . oraenv
   ```

4. Use srvctl to modify and start the standby database (the target database) in MOUNT mode. For example:

   ```
   srvctl modify database -db target_db_unique_name -role /
   PHYSICAL_STANDBY -s "READ ONLY"  -spfile /
   +DATA/target_db_unique_name/PARAMETERFILE/spfile.262.973010033

   srvctl config database -db target_db_unique_name
   ```

5. Sign in to the database instance as a user who has the SYSOPER administrator privilege.

   ```
   sqlplus / as sysoper
   ```

6. Shut down the database,

   ```
   SQL*Plus
   SHUTDOWN IMMEDIATE
   ```

7. tart the database in mount

   ```
   STARTUP MOUNT
   ```

8. alter the `cluster_database` parameter

   ```
   alter  system set cluster_database=True  sid='*' scope=spfile;
   ```

9. Shut down the database

   ```
   SHUTDOWN IMMEDIATE
   ```

10. Exit SQL*Plus.

    ```
    EXIT
    ```

11. Start the database in `MOUNT` mode by using `srvctl`.

    ```
    srvctl start database -db target_db_unique_name -o mount
    ```

12. Sign in to the database instance as a user who has the `SYSDBA` administrator privilege.

    ```
    sqlplus / as sysdba
    ```

13. Query the `V$DATABASE` dynamic view to ensure that the database is in `MOUNT` mode.

    ```
    SELECT NAME, OPEN_MODE FROM V$DATABASE;
    ```

    #Output similar to the following appears:

    ```
    NAME              OPEN_MODE
    --------------    ---------
    source_db_name    MOUNTED
    ```

Set the Database and Log File Name Conversion Parameters on the Primary Database

You must set the conversion parameters for the database and the log file name on the primary database (the source database).

1. Use SSH to sign in to the primary database (the source database) server.
2. Switch to the oracle user:

   ```
   sudo su - oracle
   ```

3. Sign in to the database instance as a user who has the `SYSDBA` administrator privilege. For example:

   ```
   sqlplus / as sysdba
   Enter password: password
   ```

4. Check the `CONVERT` parameter.

   ```
   SHOW PARAMETER CONVERT
   ```

Output similar to the following appears:

```
NAME                         TYPE        VALUE
------------------------------------ ----------- ------
db_file_name_convert                 string
log_file_name_convert                string
pdb_file_name_convert                string
```

The VALUE column should be empty (null). If there is a value, then make a note of this value for after the migration is complete. After the migration is complete, these values are set to null.

5. Use the information from this output to set the DB_FILE_NAME_CONVERT parameter.

   > **Note:**
   >
   > Note that in this step, the SOURCE_DB_NAME should be in upper case.

   ```
   ALTER SYSTEM SET DB_FILE_NAME_CONVERT='+DATA','/u02/app/oracle/oradata/
   SOURCE_DB_NAME/'' SID='*' SCOPE=SPFILE;
   ```

6. Set the LOG_FILE_NAME_CONVERT parameter. For example:

   ```
   ALTER SYSTEM SET LOG_FILE_NAME_CONVERT='+RECO','/u04/app/oracle/redo/'
   SID='*' SCOPE=SPFILE;
   ```

7. Restart the database.

   ```
   srvctl stop database -db source_db_name
   srvctl start database -db source_db_name
   ```

Set the Database and Log File Name Conversion Parameters on the Standby Database

You must set the conversion parameters for the database and the log file name on the standby database (the target database).

1. Use SSH to sign in to the standby database (the target database) server.
2. Switch to the oracle user:

   ```
   sudo su - oracle
   ```

3. Sign in to the database instance as a user who has the SYSDBA administrator privilege. For example:

   ```
   sqlplus / as sysdba
   Enter password: password
   ```

4. Check the CONVERT parameter.

   ```
   SHOW PARAMETER CONVERT
   ```

Output similar to the following appears:

```
 NAME                           TYPE        VALUE
------------------------------------ ----------- ------
db_file_name_convert                string
log_file_name_convert               string
pdb_file_name_convert               string
```

The VALUE column should be empty (null). If there is a value, then make a note of this value for after the migration is complete. After the migration is complete, these values are set to null.

5. Use the information from this output to set the DB_FILE_NAME_CONVERT parameter. For example:

   > **Note:**
   >
   > Note that in this step, the SOURCE_DB_NAME should be in upper case.

   ```
   ALTER SYSTEM SET DB_FILE_NAME_CONVERT='/u02/app/oracle/oradata/
   SOURCE_DB_NAME/','+DATA' SID='*' SCOPE=SPFILE;
   ```

6. Set the LOG_FILE_NAME_CONVERT parameter. For example:

   ```
   ALTER SYSTEM SET LOG_FILE_NAME_CONVERT='/u04/app/oracle/redo/','+RECO'
   SID='*' SCOPE=SPFILE;
   ```

7. Restart the database.

   ```
   srvctl stop database -db target_db_unique_name
   srvctl start database -db target_db_unique_name -o mount
   ```

Configure the Database with Oracle Data Guard Broker

You can use the dbmgrl utility to configure either the primary database (the source database) or the standby database (the target database) with Oracle Data Guard Broker.

1. Use SSH to sign in to the primary database (the source database) or the standby database (the target database) server.
2. Start the dgmgrl command line utility:

   ```
   dgmgrl
   ```

3. Connect as user SYS from either the primary or the standby database system. For example, to log in to a primary database whose TNS name is OCIC-ORCL:

   ```
   connect sys@ocic-orcl
   Enter password: password
   ```

4. Using the dbmgrl utility, create the Oracle Data Guard configuration and identity for the primary and standby databases. For example:

   ```
   create configuration configuration_name as primary database is
   source_db_unique_name connect identifier is OCIC-ORCL; -- Uses the
   source TNS name

   add database target_db_unique_name as connect identifier is OCI-ORCL; --
   Uses the target TNS name
   ```

5. Enable the configuration.

   ```
   enable configuration
   ```

6. Show the Oracle Data Guard configuration on the standby database.

   ```
   show configuration
   ```

   Output similar to the following appears:

   ```
   Configuration - configuration_name
     Protection Mode: MaxPerformance
     Members:
     source_db_unique_name            - Primary database
       target_db_unique_name          - Physical standby database

   Fast-Start Failover: DISABLED

   Configuration Status:
   SUCCESS   (status updated 12 seconds ago)
   ```

Validate Oracle Data Guard Broker on the Primary Database and the Standby Database

You can use SQL*Plus to validate Oracle Data Guard Broker on the primary database (the source database) and the standby database (the target database).

Validate Oracle Data Guard Broker on the Primary Database

You can use SQL*Plus to validate Oracle Data Guard Broker on the primary database (the source database).

1. Use SSH to sign in to the primary database (the source database) server.
2. Connect as a user who has the SYSDBA administrator privilege. For example, for a primary database whose TNS name is OCIC-ORCL:

   ```
   connect sys@ocic-orcl as sysdba
   Enter password: password
   ```

3. Query the V$DATABASE dynamic view.

   ```
   SELECT FORCE_LOGGING, FLASHBACK_ON, OPEN_MODE, DATABASE_ROLE,
   SWITCHOVER_STATUS, DATAGUARD_BROKER, PROTECTION_MODE FROM V$DATABASE;
   ```

4. Output similar to the following appears:

   ```
   FORCE_LOGGING                            FLASHBACK_ON       OPEN_MODE
   --------------------------------------- ------------------ -----------
   DATABASE_ROLE     SWITCHOVER_STATUS     DATAGUAR PROTECTION_MODE
   ---------------- -------------------- -------- --------------------
   YES                                      YES                READ WRITE
   PRIMARY           TO STANDBY           ENABLED  MAXIMUM PERFORMANCE
   ```

   In the output, the DATABASE_ROLE should be PRIMARY and OPEN_MODE should be READ WRITE.

Validate Oracle Data Guard Broker on the Standby Database

You can use SQL*Plus to validate Oracle Data Guard Broker on the standby database (the target database).

1. Use SSH to sign in to the standby database (the target database) server.
2. Connect as a user who has the SYSDBA administrator privilege.
3. Query the V$DATABASE dynamic view.

   ```
   SELECT FORCE_LOGGING, FLASHBACK_ON,
   OPEN_MODE, DATABASE_ROLE, SWITCHOVER_STATUS,
   DATAGUARD_BROKER, PROTECTION_MODE
   FROM V$DATABASE;
   ```

   Output similar to the following appears:

   ```
   FORCE_LOGGING                            FLASHBACK_ON       OPEN_MODE
   --------------------------------------- ------------------ -----------
   DATABASE_ROLE     SWITCHOVER_STATUS     DATAGUAR PROTECTION_MODE
   ---------------- -------------------- -------- --------------------
   YES                                      YES                MOUNTED
   PHYSICAL STANDBY NOT ALLOWED           ENABLED  MAXIMUM PERFORMANCE
   ```

   The output should show DATABASE_ROLE as PHYSICAL STANDBYand OPEN_MODE as MOUNTED.

4. Verify that the Oracle Data Guard processes are initiated in the standby database.

   ```
   SELECT PROCESS,PID,DELAY_MINS FROM V$MANAGED_STANDBY;
   ```

Output similar to the following appears:

```
PROCESS   PID                      DELAY_MINS
--------- ------------------------ ----------
ARCH      9207                              0
ARCH      9212                              0
ARCH      9216                              0
ARCH      9220                              0
RFS       1065                              0
RFS       1148                              0
RFS       1092                              0
MRP0      972                               0
RFS       1208                              0
```

The output should indicate that the processes are running with little or no delay. If the DELAY_MINS for MRP0, the databases are synchronized.

5. Check theLOG_ARCHIVE_DEST parameter.

```
SHOW PARAMETER LOG_ARCHIVE_DEST_
```

Output similar to the following appears:

```
NAME                          TYPE      VALUE
----------------------------- --------- -----------------------------
log_archive_dest_1            string
                                        location=USE_DB_RECOVERY_FILE_
                                        DEST, valid_for=(ALL_LOGFILES,
                                        ALL_ROLES)
log_archive_dest_10           string
log_archive_dest_11           string
log_archive_dest_12           string
log_archive_dest_13           string
log_archive_dest_14           string
log_archive_dest_15           string
...
log_archive_dest_2            string    service="oci-orcl", ASYNC
                                        NOAF FIRM delay=0 optional
                                        compression=disable
                                        max_failure=0 max_connections
                                        =1 reopen=300 db_unique_name=
                                        "source_db_unique_name"
                                        net_timeout=30, valid_for=
                                        (online_logfile,all_roles)
...
```

The output should be similar to the output for log_archive_dest_2, with the service pointing to the standby database (the target database), which in this example is oci-orcl.

6. Check the LOG_ARCHIVE_CONFIG parameter.

```
SHOW PARAMETER LOG_ARCHIVE_CONFIG#
```

Output similar to the following appears:

```
NAME                          TYPE      VALUE
---------------------------- ---------
-----------------------------------------------------------
log_archive_config            string
dg_config=(source_db_unique_name,target_db_unique_name)
```

7. Check the `FAL_SERVER` parameter.

```
SHOW PARAMETER FAL_SERVER
```

Output similar to the following appears:

```
NAME                         TYPE      VALUE
--------------------------- --------- ----------
fal_server                   string    <tns_entry_of_primary>
```

8. Check the `LOG_ARCHIVE_FORMAT` parameter.

```
SHOW PARAMETER LOG_ARCHIVE_FORMAT
```

Output similar to the following appears:

```
NAME                         TYPE      VALUE
--------------------------- --------- --------------
log_archive_format           string    %t_%s_%r.dbf
```

Complete the Validation on the Primary Database

You can use `dgmrgl` to complete the Oracle Data Guard Broker validation on the primary database (the source database).

1. Use SSH to sign in to the primary database (the source database) server.
2. Repeat steps 5 through 8 in the topic **Validate Oracle Data Guard Broker on the Standby Database** on the primary database (the source database).
3. Start the `dgmgrl` command line utility:

```
dgmgrl
```

4. Connect as user `SYS` from either the primary or the standby database system. For example, to log in to a primary database whose TNS name is `OCIC-ORCL`:

```
connect sys@primary_db_tnsnames_name
Enter password: password
```

5. Check the Oracle Data Guard configuration.

   ```
   show configuration verbose
   ```

   Output similar to the following appears:

   ```
   Configuration - configuration_name

     Protection Mode: MaxPerformance
     Members:
     source_db_unique_name           - Primary database
       target_db_unique_name         - Physical standby database

     Properties:
       FastStartFailoverThreshold      = '30'
       OperationTimeout                = '30'
       TraceLevel                      = 'USER'
       FastStartFailoverLagLimit       = '30'
       CommunicationTimeout            = '180'
       ObserverReconnect               = '0'
       FastStartFailoverAutoReinstate  = 'TRUE'
       FastStartFailoverPmyShutdown    = 'TRUE'
       BystandersFollowRoleChange      = 'ALL'
       ObserverOverride                = 'FALSE'
       ExternalDestination1            = ''
       ExternalDestination2            = ''
       PrimaryLostWriteAction          = 'CONTINUE'

   Fast-Start Failover: DISABLED
   ```

6. Check the status on the standby database (the target database). For example:

   ```
   show database verbose target_db_unique_name
   ```

   After you complete these steps, you must test that the Oracle Data Guard configuration is functioning as expected by performing switchover operations in both directions.

Perform the Migration

To complete the migration, you must perform a switchover operation from the primary database (the source database) to the standby database (the target database).

1. Use SSH to sign in to the primary database (the source database) server.
2. Start the `dgmgrl command line` utility.

   ```
   dgmgrl
   ```

3. Connect as user SYS from either the primary or the standby database system. For example, to log in to a primary database whose TNS name is OCIC-ORCL:.

   ```
   connect sys@ocic-orcl
   Enter password: password
   ```

4. Check the configuration.

   ```
   show configuration verbose
   ```

5. In the configuration verbose output, check that the Database Status setting says SUCCESS.
6. Check the configuration for the primary database (the source database).

   ```
   show database verbose source_db_unique_name
   ```

   The database verbose output should show that the role is primary and the setting for StaticConnectIdentifier is the same as DGConnectIdentifier.
7. Perform a switchover operation to the standby database (the target database).

   ```
   switchover to target_db_unique_name
   ```

   The output should indicate that the switchover operation is occurring between the two databases.
8. Run show configuration to verify that there are no errors or warnings:

   ```
   show configuration;
   ```

Post-Migration Steps

After you complete the migration of an Oracle database from an Oracle Cloud Infrastructure Compute Classic server to an Oracle Cloud Infrastructure server that uses a Virtual Machine Database system, you should validate the migration, and then remove the configuration from the primary database (the source database).

Test the Oracle Data Guard Configuration on the Standby Database

At this stage, the target database is now the primary database. The source database is now the standby database.

You can test the Oracle Data Guard connection on the target database, by performing a switchover operation with the source database. This switchover operation will make the target database take the standby role again. The purpose of this test is to prove that you can return to the original configuration in case the target database is not functional.

1. Use SSH to sign in to the standby database (the target database) server.

2. Start the `dgmgrl` utility. For example:

   ```
   dgmgrl sys@target_db
   Enter password: password
   ```

3. Perform a switchover operation to the source database, which will make the target database take the standby role.

   ```
   switchover to source_db_unique_name;
   ```

   The output should indicate that the switchover operation is occurring between the two databases.

4. (Optional) To prevent changes to the new standby database until the new primary database is determined to be fully functional, temporarily disable the `Redo Apply` feature.

   ```
   edit database source_db_unique_name set state = 'APPLY-OFF';
   ```

   If you perform another switchover operation so that the target database is now the standby database, you can perform an `APPLY-OFF` operation to prevent the source database from being updated. This enables the target database to be put in service, and keeps the source database as a point-in-time backup in case of a logical failure in the new configuration.

5. (Optional) To restart the apply feature:

   ```
   edit database source_db_unique_name set state = 'APPLY-ON';
   ```

6. Exit `dgmgrl`.

   ```
   exit
   ```

7. Perform a switchover operation to the target database, which will make the source database the standby role.

   ```
   switchover to target_db_unique_name;
   ```

   The output should indicate that the switchover operation is occurring between the two databases.

8. Test the connection to the new primary database. For example, after exporting the target unique name, connect as user `SYS` and select from an encrypted table space. In this example, the `HR.EMPLOYEES` table is encrypted.

   ```
   exit
   ```

9. Test the connection to the new primary database. For example, after exporting the target unique name, connect as user `SYS` and select from an encrypted table space. In this example, the `HR.EMPLOYEES` table is encrypted.

   ```
   export ORACLE_UNQNAME=target_db_unique_name
   ```

```
sqlplus sys@target_TNS_name
Password: password

SQL> ALTER SESSION SET CONTAINER = PDB1;
SQL> SELECT * FROM HR.EMPLOYEES;

SQL> EXIT
```

Clean Up the Standby Database

After you complete and test the migration, you can remove the Oracle Data Guard configuration from the standby database (the target database). You do not need to remove the original source database. At this stage, the standby database is the new source database.

1. Use SSH to sign in to the standby database (the target database) server and sign in to the Oracle Data Guard `dgmgrl utility`.
2. Check the configuration.

   ```
   show configuration
   ```

3. If the configuration does not show `Protection Mode: MaxPerformance`, then set Oracle Data Guard to use the `MaxPerformance` protection mode.

   ```
   edit configuration set protection mode as maxperformance
   ```

4. Disable and then remove the configuration.

   ```
   edit database source_db_unique_name set state = 'APPLY-OFF';

   disable configuration;

   remove configuration;

   exit
   ```

5. Connect to the database instance as a user who has the `SYSDBA` administrator privilege.For example:

   ```
   sqlplus / as sysdba
   ```

6. Check the `DG_BROKER_CONFIG_FILE` parameters.

   ```
   SHOW PARAMETER DB_BROKER_CONFIG_FILE
   ```

   The output should list the associated data and recovery files for this configuration, typically named `dg_broker_config_file1` and `dg_broker_config_file2`.
7. Start another terminal window, and sign in to `asmcmd` as the grid user.
8. Remove the Oracle Data Guard configuration files that were listed when you checked the `DG_BROKER_CONFIG_FILE` parameters.
9. Return to the window that is running SQL*Plus.

10. Execute the following `ALTER SYSTEM` statements:

    ```
    ALTER SYSTEM SET DG_BROKER_START=FALSE SID='*' SCOPE=BOTH;
    ALTER SYSTEM SET DG_BROKER_CONFIG_FILE1='' SID='*' SCOPE=SPFILE;
    ALTER SYSTEM SET DG_BROKER_CONFIG_FILE2='' SID='*' SCOPE=SPFILE;
    ALTER SYSTEM RESET LOG_ARCHIVE_CONFIG SID='*' SCOPE=SPFILE;
    ```

11. Check the following parameters:

    ```
    SHOW PARAMETER DB_FILE_NAME_CONVERT
    SHOW PARAMETER LOG_FILE_NAME_CONVERT
    SHOW PARAMETER LOG_ARCHIVE_DEST
    SHOW PARAMETER LOG_ARCHIVE_DEST_STATE
    SHOW PARAMETER STANDBY_ARCHIVE_DEST
    SHOW PARAMETER FAL
    ```

12. If any of the preceding parameters is set, then reset the parameters to use blank values. For example, for `STANDBY_ARCHIVE_DEST`:

    ```
    ALTER SYSTEM SET STANDBY_ARCHIVE_DEST='' SID='*' SCOPE=SPFILE;
    ```

13. Restart the database.

    ```
    SHUTDOWN IMMEDIATE
    STARTUP
    ```

14. Drop the standby logs from the primary database (the source database).

    a. Find the group numbers for the standby database redo logs that are on the new primary database (which was formerly the target database).

       ```
       SELECT GROUP# FROM V$STANDBY_LOG;Output similar to the following
       appears:    GROUP#
       ----------
                5
                6
                7
                8
                9
       ```

    b. Remove the standby logs. For example:

       ```
       ALTER DATABASE DROP STANDBY LOGFILE GROUP 5;
       ALTER DATABASE DROP STANDBY LOGFILE GROUP 6;
       ALTER DATABASE DROP STANDBY LOGFILE GROUP 7;
       ALTER DATABASE DROP STANDBY LOGFILE GROUP 8;
       ALTER DATABASE DROP STANDBY LOGFILE GROUP 9;
       ```

15. (Optional) Change the `DB_BLOCK_CHECKSUM` and `DB_BLOCK_CHECKING` parameters.

    The default values are `DB_BLOCK_CHECKSUM=TYPICAL` and `DB_BLOCK_CHECKING=FALSE`.

16. Exit SQL*Plus.

    ```
    EXIT
    ```

16 Migrate Virtual Machines and Block Storage to Oracle Cloud Infrastructure

If you have virtual machine instances and block storage volumes in your Oracle Cloud Infrastructure Compute Classic account, then you can use Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool to move your resources to an Oracle Cloud Infrastructure environment. All flavors of Linux VMs running in Oracle Cloud Infrastructure Compute Classic can be migrated. For Windows, VMs running Windows Server 2008, 2012, 2012 R2, or 2016 can be migrated.
Note that this migration tool doesn't help with discovering resources in your source environment. You can use Oracle Cloud Infrastructure Classic Discovery and Translation Tool to generate reports of all the resources in your Oracle Cloud Infrastructure Compute Classic account.

You can't use this tool for the following types of migration:

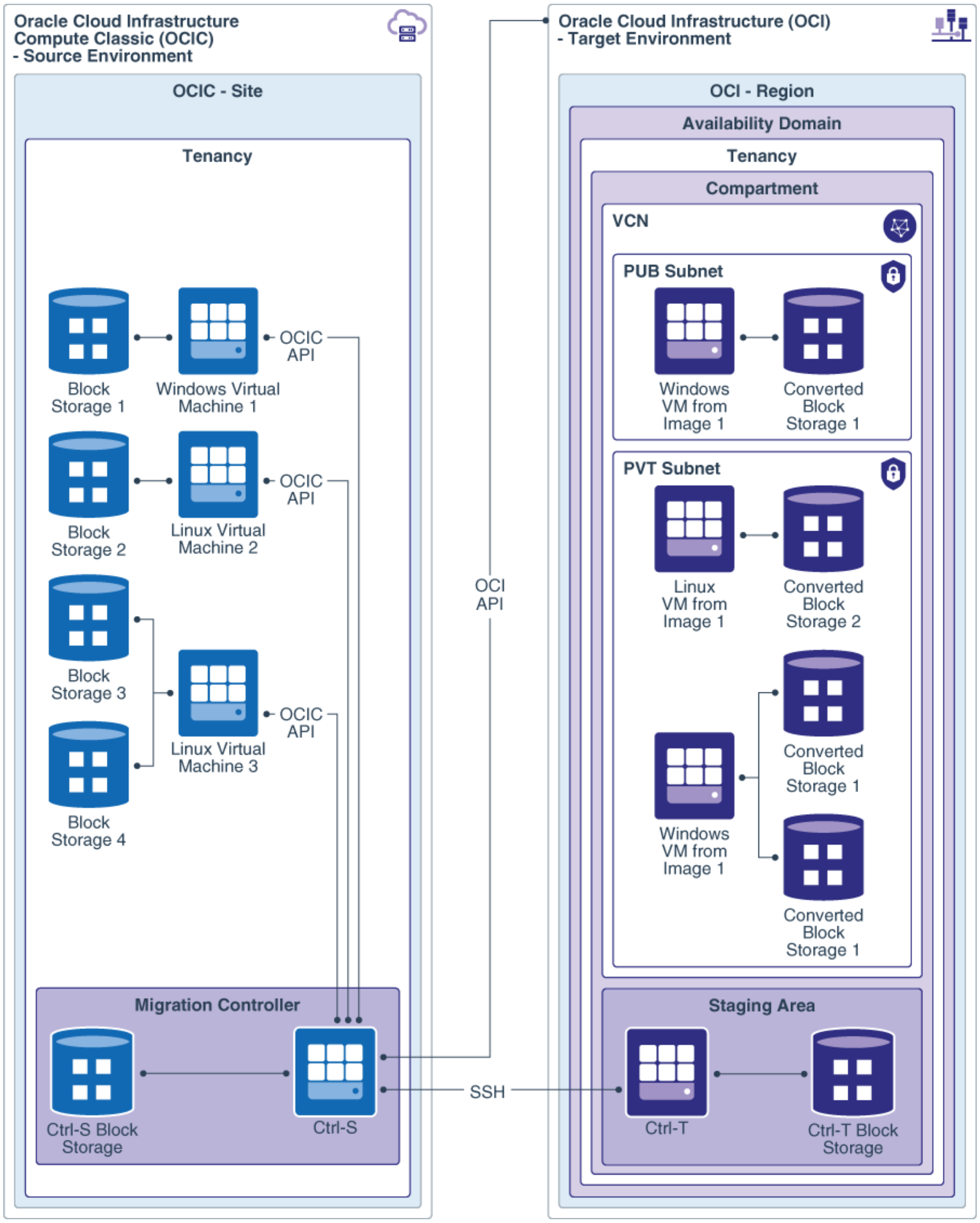
- Application-aware migration
- Object storage migration. Use rclone or CloudBerry to migrate data from Oracle Cloud Infrastructure Object Storage Classic to Oracle Cloud Infrastructure Object Storage.
- PaaS migration. Re-create the PaaS instances on Oracle Cloud Infrastructure and redeploy the applications.
- Oracle Database migration. Use native tools like RMAN, Data Pump, and GoldenGate or GoldenGate Cloud Service to migrate when possible. See Select a Method to Migrate Database Instances.

It is assumed that you are familiar with both Oracle Cloud Infrastructure Compute Classic as well as Oracle Cloud Infrastructure and that you have access to both services.

The Terraform configurations generated by Oracle Cloud Infrastructure Classic Discovery and Translation Tool can be used to automate the set up of resources in your Oracle Cloud Infrastructuretenancy. It is assumed that you are familiar with installing and using Terraform.

Architecture

This architecture shows resources in your Oracle Cloud Infrastructure Compute Classic account being migrated to your Oracle Cloud Infrastructure tenancy using Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool.



Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool uses the migration controller instance in your source environment, Control-S, to migrate your VMs and block storage volumes. The Control-S instance contains tools and scripts that you can use to configure your source VMs for migration. The Control-S instance also includes scripts that set up another migration controller instance, Control-T, in the target environment. These migration controllers are used to collect information about the source environment, copy data, create images from boot volumes, and create the required images and data volumes in the target environment.

Workflow

Here's an overview of the high-level steps required to migrate your VMs and block storage from Oracle Cloud Infrastructure Compute Classic to Oracle Cloud Infrastructure.

This workflow assumes that you'll use Oracle Cloud Infrastructure Classic Discovery and Translation Tool along with Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool to automate this process.

1. Create an instance in Oracle Cloud Infrastructure Compute Classic using the Oracle Cloud Infrastructure Classic Migration Tools image. If you've already created this instance earlier in your migration process, you can use the same instance for this procedure. You don't need to create it again.
2. Log in to the migration controller instance, Control-S, using SSH.
3. Set up and use Oracle Cloud Infrastructure Classic Discovery and Translation Tool to do the following. For more information about using this tool, see Identify and Translate Resources in Your Source Environment. For information about the commands, options, and permitted values, run the tool with the `--help` option. To view help on all commands and options, use: `opcmigrate --full-help`.
    a. Run `opcmigrate discover` to get a list of resources in your source environment.
    b. Run `opcmigrate plan create` to create a migration plan. You can use several options with this command to filter the output of the `opcmigrate discover` command.
    c. Edit the migration plan to specify migration attributes for individual objects, if required.
    d. Run `opcmigrate generate` to create Terraform configuration files. If you want to migrate your network using network topology mapping, use the `--with-security-rule-union` option to include security lists and security rules in the generated Terraform file.
    e. Run `opcmigrate instances-export --plan <plan_name>` to generate a list of instances to migrate.
4. Review the Terraform configuration carefully and edit it as required. When you're satisfied with the network definition, apply the Terraform plan in your Oracle Cloud Infrastructure tenancy. At this stage, the Terraform configuration creates only the networking objects. Compute instances aren't created yet, as all the information required to launch instances isn't yet available in the configuration file.
5. Use Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool to complete the following steps. Detailed instructions to complete these steps are provided in the following sections.
    a. Set up the Control-S instance to perform the migration.
    b. Use the list of instances generated by Oracle Cloud Infrastructure Classic Discovery and Translation Tool to create one or more job files. Alternatively, provide this list as input in the `secret.yml` file when you set up Control-S. You can also specify a list of unattached volumes that you want to migrate.
    c. Use the `opcmigrate migrate instance` set of commands to configure the source instances, set up the migration controller instance, Control-T, in the target environment, and run the migration job.
6. After volume migration is complete, launch your compute instances in the target environment using the appropriate boot volume and then attach the appropriate data volumes to each instance. You can complete this procedure either by using the Oracle Cloud Infrastructure, or by updating the Terraform configuration with the required information.

Migration Overview

Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool performs a number of tasks to copy boot and data volumes from your Oracle Cloud Infrastructure Compute Classic account to your Oracle Cloud Infrastructure tenancy.

This tool does the following:

1. Runs as a service daemon on the Control-S instance. You can start or stop the service, or view the status of the service. You can also run multiple migration jobs at the same time to migrate resources in parallel.
2. Configures Control-S with details of the resources in the source environment to be migrated, based on user input or input from Oracle Cloud Infrastructure Classic Discovery and Translation Tool.
3. Configures Control-S with information required to access the target Oracle Cloud Infrastructure environment, based on user input.
4. Configures the source instances to collect information about the attached storage volumes. When the boot images and storage volumes have been migrated to the Oracle Cloud Infrastructure environment, this information is used to attach the storage volumes to the appropriate virtual machines at the appropriate mount points.
5. Starts the migration controller VM, Control-T, in Oracle Cloud Infrastructure.
6. The migration process performs the following steps for each source instance and all attached storage volumes, for both boot and data volumes. These steps are carried out in parallel for up to eight boot and data volumes at a time.
    a. Creates colocated snapshots of the source storage volumes to be migrated. If you specify volumes restored from colocated snapshots, the tool creates a remote snapshot of those volumes. Creating a remote snapshot of a volume restored from a colocated snapshot can take a longer time, compared to creating a colocated snaphost of a storage volume.
    b. Creates new volumes from the snapshots.
    c. Attaches the new storage volumes to Control-S.
    d. Creates the corresponding storage volumes in Oracle Cloud Infrastructure, attached to the migration controller VM, Control-T.
    e. Copies the data from all volumes attached to Control-S to volumes attached to Control-T over a secure SSH pipe.
    f. Detaches the migrated volumes from Control-T, so they can be used to launch VMs or attached to new VMs as data volumes.
    g. Detaches the storage volumes from Control-S in Oracle Cloud Infrastructure Compute Classic.
    h. Deletes the new storage volumes that were created for migration in Oracle Cloud Infrastructure Compute Classic.
    i. Deletes the colocated snapshots in Oracle Cloud Infrastructure Compute Classic.

After the boot and data volumes have been migrated to the target environment, launch your VMs in Oracle Cloud Infrastructure using the migrated boot volumes and attach the migrated data volumes to the appropriate VMs.

Considerations for Migration

Before you start your migration, consider the following factors that could have an impact on your migration process.

- Perform a proof-of-concept migration with VMs running applications that are as close to the configurations as possible.
- Quiesce applications on your source VMs and don't make any changes on the source VMs while migration is in progress.
- The maximum size of the boot volumes of VMs that can be imported is approximately 1 TB – assuming the boot volume has 60% used space and 50% compression ratio.
- A block storage volume in Oracle Cloud Infrastructure Compute Classic can have a maximum of five colocated snapshots. If a storage volume has more than five snapshots, the tool generates an error and fails.
- A single Control-S instance can migrate up to eight storage volumes at a time. To migrate a larger number of volumes, you can launch multiple Control-S instances.
- You can create and specify multiple job files on a migration controller instance. These jobs use the same source and target environments; only the list instances and storage volumes specified for migration is different.
- Up to four migration jobs can be run in parallel. If you submit more than four jobs, the other jobs are queued until some jobs finish. Note that if you run multiple jobs in parallel, the total number of storage volumes being migrated must not exceed eight across all the currently running migration jobs.
- The steps to migrate data for all the storage volumes are carried out in parallel. So the overall time taken for the migration depends mainly on the size of the largest storage volume that needs to be migrated.
- In a single run of the migration tool, you can migrate VMs and storage volumes from a single source identity domain and site. To migrate VMs and storage volumes from a different site or identity domain, create and configure another Control-S instance.
- In a single run of the migration tool, you can migrate VMs and storage volumes to a single target tenancy, region, and availability domain. To migrate resources to a different tenancy, region, or availability domain, create and configure another Control-S instance.
- A boot volume is migrated to a specified availability domain in Oracle Cloud Infrastructure and it can be used to launch a VM in the same availability domain only. Ensure that you migrate each boot volume to the availability domain where you want to launch the VM.
- When possible, the private IP addresses of the target instances should be the same as the private IP addresses of the source instances. This should be taken into consideration when setting up the network in your Oracle Cloud Infrastructure tenancy before you start migrating VMs and block volumes. In some cases, you might not be able to re-create the Oracle Cloud Infrastructure Compute Classic

private IP addresses in your Oracle Cloud Infrastructure VCNs. In these cases, you might need to change application configurations to make things work.

Map Oracle Cloud Infrastructure Compute Classic Instance Shapes to Oracle Cloud Infrastructure Shapes

While some of the instance shapes in your Oracle Cloud Infrastructure Compute Classic account correspond to similar shapes in Oracle Cloud Infrastructure, in other cases you may not find an exact equivalent.

Here are some suggestions for which shapes in the target environment are the best fit for shapes you've used in your source environment.

Oracle Cloud Infrastructure Compute Classic			**Oracle Cloud Infrastructure**		
Shape	OCPU/GPU	RAM	Shape	OCPU/GPU	RAM
oc3	1	7.5	VM.Standard2.1	1	15
oc4	2	15	VM.Standard2.2	2	30
oc5	4	30	VM.Standard2.4	4	60
oc6	8	60	VM.Standard2.8	8	120
oc7	16	120	VM.Standard2.16	16	240
oc8	24	180	VM.Standard2.24	24	320
oc9	32	240	BM.Standard2.52	52	768
oc1m	1	15	VM.Standard2.1	1	15
oc2m	2	30	VM.Standard2.2	2	30
oc3m	4	60	VM.Standard2.4	4	60
oc4m	8	120	VM.Standard2.8	8	120
oc5m	16	240	VM.Standard2.16	16	240
oc8m	24	360	VM.Standard2.24	24	320
oc9m	32	480	BM.Standard2.52	52	768
ocio1m	1	15	VM.DenseIO2.8	8	120
ocio2m	2	30	VM.DenseIO2.8	8	120
ocio3m	4	60	VM.DenseIO2.8	8	120
ocio4m	8	120	VM.DenseIO2.8	8	120
ocio5m	16	240	VM.DenseIO2.16	16	240
ocsg2-k80	6 / 2	120	VM.GPU3.2	12 / 2	180
ocsg2-m60	6 / 2	120	VM.GPU3.2	12 / 2	180
ocsg1-k80	3 / 1	60	VM.GPU3.1	6 / 1	90
ocsg1-m60	3 / 1	60	VM.GPU3.1	6 / 1	90

If instances in your Oracle Cloud Infrastructure Compute Classic account have multiple virtual NICs (vNICs), then you might need to select a larger shape in Oracle Cloud Infrastructure, to ensure that the appropriate number of vNICs is supported.

Oracle Cloud Infrastructure Compute Classic Shape	Oracle Cloud Infrastructure Shape for 1 or 2 vNICs	Oracle Cloud Infrastructure Shape for 3 or 4 vNICs	Oracle Cloud Infrastructure Shape for 5 or more vNICs
oc3	VM.Standard2.1	VM.Standard2.4	VM.Standard2.8
oc4	VM.Standard2.2	VM.Standard2.4	VM.Standard2.8
oc5	VM.Standard2.4	VM.Standard2.4	VM.Standard2.8
oc1m	VM.Standard2.1	VM.Standard2.4	VM.Standard2.8
oc2m	VM.Standard2.2	VM.Standard2.4	VM.Standard2.8
oc3m	VM.Standard2.4	VM.Standard2.4	VM.Standard2.8

End-to-End Procedure

Here's an example of an end-to-end procedure for migrating instances and storage volumes using the migration tools. For more detailed information, see the relevant sections of this document.

Before you start, ensure that you have:

- The required permissions in your source and target environments.
- SSH or RDP access to the source VMs.
- The SSH key required to access the Control-S instance.
- The API PEM key for Oracle Cloud Infrastructure.

Perform the following steps to complete the migration. Note that the steps here are useful as a quick reference if you're already familiar with the migration process. If you're performing a migration for the first time, it's recommended that you follow the more detailed instructions provided in the relevant sections of this document.

Step	More Information
Create a Control-S instance in Oracle Cloud Infrastructure Compute Classic using the Oracle Cloud Infrastructure Classic Migration Tools image.	Launch the Migration Controller Instance (Control-S) in the Source Environment
Log in to the Control-S instance using SSH. On the Control-S instance, copy the PEM key required for the API connection to the file `/home/opc/.oci/oci_api_key.pem`. Modify permissions on the key file to restrict access.	Configure the Migration Controller Instance (Control-S)
Use the file `/home/opc/ansible/secret.yml.sample` to create your `secret.yml` file in the same path and enter the required information.	Configure the Migration Controller Instance (Control-S)
If you're migrating Linux instances, create the bucket `ocic-oci-sig` in Oracle Cloud Infrastructure Object Storage and generate a pre-authenticated request (PAR) for writes to this bucket. Enter this PAR in the `secret.yml` file on the Control-S instance.	Complete the Prerequisites

Step	More Information

On the Control-S instance, run:
`chmod 600 ansible/secret.yml`
`opcmigrate migrate instance service setup`
`opcmigrate migrate instance service start`
	Configure the Migration Controller Instance (Control-S)

Update the default profile file or create a new profile file in the directory `/home/opc/.opc/profiles`, to provide credentials for all the services in your Oracle Cloud Infrastructure Compute Classic account.	Set Up Your Profile
Discover resources in your source environment. `opcmigrate discover`	Generate a Summary and JSON Output

Generate a list of resources to be migrated.
`opcmigrate plan create --output migration-plan.json`
`opcmigrate instances-export --plan migration-plan.json --format json > instances.json`
	Generate a List of Instances to Migrate

Edit the list of resources to be migrated and create the required job files.	Specify the Instances and Storage Volumes to be Migrated
Set up the VCN, subnets, and other networking components in your Oracle Cloud Infrastructure tenancy to launch your migration controller instance, Control-T, as well as for your migrated VMs and block volumes.	Create a Virtual Cloud Network in Oracle Cloud Infrastructure
Enable Logical Volume Manager (LVM) on Control-S, if required.	Enable Logical Volume Manager (LVM) on Control-S

Step	More Information
SSH into each Linux VM that you want to migrate. Install: • `iscsi-initiator-utils` • `cronie` • `util-linux-ng` or `util-linux` Ensure that all secondary attached volume mounts have `_netdev` and `nofail` specified in `/etc/fstab`. It is mandatory to specify these options for the secondary attached storage volumes. Ensure that you **don't** specify `_netdev` and `nofail` in `/etc/fstab` for file systems that are stored in bootable storage volumes. You are required to specify the `_netdev` and `nofail` options in `/etc/fstab` only for the non-bootable storage volumes entries, such as `/dev/xvdb` and `/dev/xvdc`. Do not specify `_netdev` and `nofail` options in `/etc/fstab` for any volume groups and other non `/dev/` entries.	Prepare Your Linux Source Instances for Migration
On the Control-S instance, create or update the `/home/opc/ansible/hosts.yml` file with information about the Linux VMs that you want to migrate.	Prepare Your Linux Source Instances Using Tools on Control-S
Use the scripts on Control-S to set up your Linux source VMs. `opcmigrate migrate instance source setup`	Prepare Your Linux Source Instances Using Tools on Control-S
Log in to each Windows VMs that you want to migrate. Copy the file `/home/opc/src/windows_migrate.ps1` from Control-S to each source instance. Run `windows_migrate.ps1`.	Prepare Your Windows Source Instances for Migration
Launch the migration controller in the target environment. `opcmigrate migrate instance ctlt setup`	Launch the Migration Controller Instance (Control-T) in the Target Environment
Start a migration job. `opcmigrate migrate instance job run --job_file <full_path/job_file_name>`	Start the Migration

Step	More Information
Monitor the migration job. `opcmigrate migrate instance job status <job_name>`	Monitor the Migration

For an incremental migration or a base migration job that has the shutdown policy `wait`, resume the migration job when you are ready and the status of the migration job displays `ready`.
`opcmigrate migrate instance job resume <job_name>`
For the base migration phase or when your migration job that has the shutdown policy `wait`, do not use the `--do-not-finalize` option. In such a scenario, run the resume command without the `--do-not-finalize` option.
In the second or subsequent phase of incremental migration, you can use the `--do-not-finalize` option.
`opcmigrate migrate instance job resume <job_name> --do-not-finalize`
	Resume a Migration Job

When the migration job is complete, use the migrated block volumes to launch instances in your Oracle Cloud Infrastructure tenancy.	Launch VMs in the Target Environment
Attach block volumes to your instances. Then, on the Control-S instance, run: `opcmigrate migrate instance attachment_ready <instance_ocid>`	Attach and Mount Block Storage on Compute Instances in the Target Environment
Validate the target environment after the migration is complete.	Validate the Target Environment

Plan for the Migration

It's important to plan your migration carefully, to ensure the process is smooth and requires minimal down time.

Before you start the migration, you should:

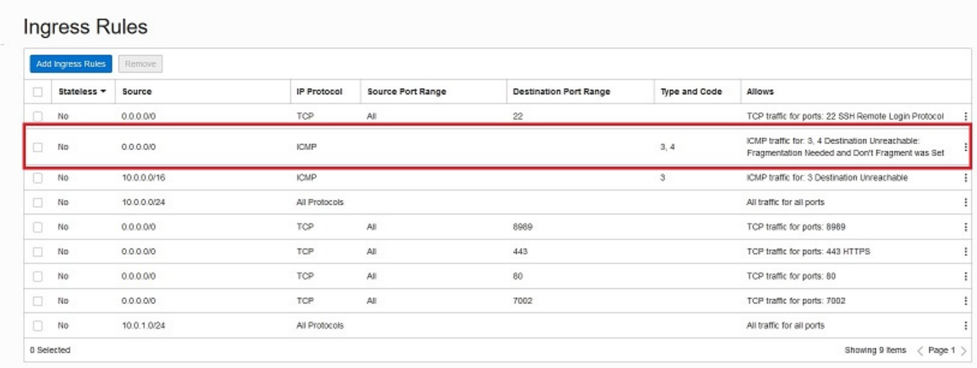
- Collect information about the source instances that you want to migrate.
- Ensure that you have the required SSH and PEM keys to access the source and target environments.

- Configure the source environment.
- Set up the network in the target environment.
- Collect information from the target environment, such as the tenancy, user, and compartment Oracle Cloud IDs (OCIDs).

Complete the Prerequisites

Before you begin your migration, complete the following prerequisites.

- Launch the migration controller instance, Control-S, in your Oracle Cloud Infrastructure Compute Classic account using the Oracle Cloud Infrastructure Classic Migration Tools image. For information about creating your Control-S instance, see Complete the Prerequisites and Launch the Migration Controller Instance (Control-S) in the Source Environment. If you've already created this instance earlier in your migration process, you can use the same instance for this procedure. You don't need to create it again.
- Verify that you have sufficient quota in Oracle Cloud Infrastructure to perform the migration.
  - For the migration controller instance, you'll need one VM of 1.2 shape and block storage of 50 GB of attached storage or double the size of all attached volumes of sources you intend to migrate. The minimum size of a storage volume in Oracle Cloud Infrastructure is 50 GB. Ensure that you have sufficient storage capacity for the number of storage volumes that you intend to migrate.
  - When your block volumes have been migrated, you'll need to launch the VMs that you're migrating. Ensure that you have sufficient quota to launch the required number of VMs. To check your service limits, in the Oracle Cloud Infrastructure Console, from the menu, select **Governance** and then click **Service Limits.**
- In Oracle Cloud Infrastructure, ensure that you have a virtual cloud network (VCN) with subnets and any other networking components that may be required. Consider creating a separate VCN and subnet for the migration controller instance, Control-T. This ensures that the private IP address assigned to Control-T doesn't conflict with the private IP address that you need to assign to an instance after migration. If you create the VCN using the Oracle Cloud Infrastructure Console, select the option **CREATE VIRTUAL CLOUD NETWORK PLUS RELATED RESOURCES**.
- Ensure that you have an ingress security rule in place for the Oracle Cloud Infrastructure VCN, to allow ICMP traffic with type 3, code 4, and with the source 0.0.0.0/0. This is required to enable the Control-S instance to access resources in the Oracle Cloud Infrastructure tenancy. To view or add a security rule, in the Oracle Cloud Infrastructure:
  1. Click **Networking** and then **Virtual Cloud Networks**.
  2. Click the network that you want to use.
  3. From the Resources list on the left, click **Security Lists**.
  4. On the Security Lists page, click the security list you want to view or edit. The Security List Details page displays the ingress and egress rules in this security list.



Ingress Rules

□	Stateless ▾	Source	IP Protocol	Source Port Range	Destination Port Range	Type and Code	Allows
□	No	0.0.0.0/0	TCP	All	22		TCP traffic for ports: 22 SSH Remote Login Protocol
□	No	0.0.0.0/0	ICMP			3, 4	ICMP traffic for: 3, 4 Destination Unreachable: Fragmentation Needed and Don't Fragment was Set
□	No	10.0.0.0/16	ICMP			3	ICMP traffic for: 3 Destination Unreachable
□	No	10.0.0.0/24	All Protocols				All traffic for all ports
□	No	0.0.0.0/0	TCP	All	8989		TCP traffic for ports: 8989
□	No	0.0.0.0/0	TCP	All	443		TCP traffic for ports: 443 HTTPS
□	No	0.0.0.0/0	TCP	All	80		TCP traffic for ports: 80
□	No	0.0.0.0/0	TCP	All	7002		TCP traffic for ports: 7002
□	No	10.0.1.0/24	All Protocols				All traffic for all ports

- If you're migrating Linux instances:
  - Create the bucket `ocic-oci-sig` in Oracle Cloud Infrastructure Object Storage and generate a pre-authenticated request (PAR) for writes to this bucket. This write-only PAR allows the tool to write a signal object to the `ocic-oci-sig` bucket when it is launched in the target environment, indicating that it is ready to handle storage attachments for the volumes post migration. Set the expiration date suitably far out, so that you can complete the migration before the PAR expires. If you need to run the migration tool multiple times, ensure that the PAR is still valid or create a new PAR for the bucket.
  - Ensure that you have `yum` installed on each Linux instance to be migrated, that a `yum` repository has been configured, and that the instance can access the `yum.oracle.com` repository. When you set up your Linux source instances for migration, `yum` is used to install the required tools.
  - Ensure that Python 2.6 or higher is installed on each Linux instance to be migrated.

Get Details of the Target Environment

Log in to the Oracle Cloud Infrastructure Console and collect the information required for the migration controller, Control-S, to connect to the target environment.

- You'll need the following information about your Oracle Cloud Infrastructure tenancy:
  - The user OCID. From the menu, choose **Identity** and then **Users**.
  - The API PEM key fingerprint. Click the user to view user details. The API Keys section displays the PEM key fingerprint.
  - The compartment OCID and the compartment name. From the menu, choose **Identity** and then **Compartments**.
  - The tenancy OCID and the tenancy region. From the menu, choose **Administration** and then choose **Tenancy Details**.
  - The Availability Domain and the subnet OCID. From the menu, choose **Networking** and then **Virtual Cloud Networks**. Click the VCN that you've created for this migration. The Subnets section displays the subnet OCID as well as the availability domain.

Name	Sample Value
User OCID	`ocid1.user.oc1..aaaaaaaahvtv5qo...`
API PEM key fingerprint	`81:45:aa:2e:de:39:ac:a3:c2:6f:...`
Compartment OCID	`ocid1.compartment.oc1..aaaaaaaaz...`
Tenancy OCID	`ocid1.tenancy.oc1..aaaaaaaaju6k54i7...`
Region	`us-ashburn-3`
Availability Domain	`kWVD:US-ASHBURN-AD-3`
Subnet OCID	`ocid1.subnet.oc1.iad.aaaaaaaarz7...`

Configure the Migration Controller Instance (Control-S)

Once the Control-S instance has started, connect to the instance using SSH. All of the tools required for the migration are already on the machine, but additional configuration is required to provide details of the source and target environments.

Remember that a single Control-S instance must be used to migrate resources from a single specified Oracle Cloud Infrastructure Compute Classic account and site to a single specified Oracle Cloud Infrastructure tenancy, region, and availability domain. After you've configured and used your Control-S instance with a specified source and target environment, to migrate resources from a different source environment or to a different target environment, set up a separate Control-S instance.

1. All of the configuration settings are in a file called `secret.yml`. You can use the sample file available at `/home/opc/ansible/secret.yml.sample` to create your `secret.yml` file. Enter the details of your Oracle Cloud Infrastructure Compute Classic account and your Oracle Cloud Infrastructure OCIDs.

   Here's an example of a `secret.yml` file, with sample values and information about each field. For help on YAML syntax, see https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html.

   ```
   # OCI info
   compartment_id: ocid1.compartment.oc1..aaaaaaaa...
   ```

```
user_id: ocid1.user.oc1..aaaaaaaa...
fingerprint: a0:a0:a0:a0:a0...
tenancy_id: ocid1.tenancy.oc1..aaaaaaaa...
region: us-ashburn-1
availability_domain: kWVD:US-ASHBURN-AD-3

# version and shape used to the Control-T instance
# 'Oracle Linux' is the only supported operating_system
oracle_linux_version: '7.6'
shape: 'VM.Standard2.1'

# subnet must be from the availability_domain you specified
subnet_id: ocid1.subnet.oc1.iad.aaaaaaaa...
# optional passphrase if used for OCI PEM file
pass_phrase:
# The ocic_oci_sig_par should point to the PAR (pre-authorized request) 
for the ocic-oci-sig
# bucket in the OCI object storage. The write-only PAR allows the 
system to write a signal
# object to the ocic-oci-sig when it is launched in the OCI side 
indicating it is ready to
# handle storage attachments for the volumes post migration.
ocic_oci_sig_par: PAR URL HERE

# OCI-Classic info
opc_profile_endpoint: compute.uscom-central-1.oraclecloud.com # or 
another one
opc_password:
container: /Compute-tenancy/user@email.com

# OCI-C Object Storage Classic REST Endpoint, user name, password.
# refer to https://docs.oracle.com/en/cloud/iaas/storage-cloud/ssapi/
SendRequests.html
# for instructions how to find the endpoint.
# If you comment them away (or remove them) from this secret.yml, then 
we will take
# your tenancy/opc_password given above as the username/password of 
Object Storage;
# and the endpoint of Object Storage is default to:
#     https://tenancy.storage.oraclecloud.com/v1/Storage-tenancy

# Object Storage Classic REST Endpoint, username, password.
opc_object_storage_endpoint:
opc_object_storage_username:
opc_object_storage_password:

# Control-S Instance settings
targetControllerAvailableStorageInGB: 2048
```

> **Note:**
>
> When you specify your passwords, ensure that you use a single quote `'` at the beginning and end of the string, if the string contains special characters.

2. The `secret.yml` file contains sensitive information about your account. Modify permissions on this file to restrict access.

   ```
   chmod 600 ansible/secret.yml
   ```

3. Apply the configuration to the system.

   ```
   opcmigrate migrate instance service setup
   ```

4. Start the migration service.

   ```
   opcmigrate migrate instance service start
   ```

5. Copy the PEM key required for the API connection to the file `/home/opc/.oci/oci_api_key.pem` on Control-S. Modify permissions on the key file to restrict access.

Enable Logical Volume Manager (LVM) on Control-S

On the migration controller instance, Control-S, Logical Volume Manager (LVM) is turned off by default for all attached volumes except the boot disk. If you want to use LVM with attached volumes, you must enable it.

This step is required *only* if you want to use LVM volumes to store the qcow2 images of boot volumes that you are migrating. Use LVM volumes only if you don't otherwise have sufficient space for the qcow2 images and you can't use non-LVM volumes. Remember that the LVM volume must be mounted on `/images`.

To enable LVM for attached volumes, log in to the Control-S instance using SSH and modify the global filter in the file `/etc/lvm/lvm.conf`. For example, if you want to enable LVM for the volume `/dev/xvdc`, then add `"a|^/dev/xvdc|"` to the global filter before `"r|.*|"`, as follows:

```
 global_filter = ["a|^/dev/xvdb|", "a|^/dev/xvdc|", "r|.*|" ]
```

Enable LVM only for attached volumes that you want to migrate. Don't enable LVM for any other devices, as it could interfere with the migration workflow. Don't turn off the global filter.

Specify the Instances and Storage Volumes to be Migrated

You can specify the resources to be migrated by creating one or more job files and specifying the relevant job file when you start a migration job.

Creating multiple job files to specify the instances and storage volumes to be migrated allows you to run multiple migration jobs in parallel.

Sample Input for Specifying Resources

Job files must be created in JSON format. A job file can include a list of instances as well as a list of unattached storage volumes. However, it's recommended that you specify either a list of instances or a list of unattached storage volumes in a job file to make the objects in a migration job easier to track. Here's an example of instances and an unattached storage volume with sample values in JSON format, which you can use to create your job files.

```
{
  "version": "1",
  "incremental": false,
  "instances": [
    {
      "name": "MULTIPART_INSTANCE_NAME_HERE",
      "os": "linux",
      "osKernelVersion": "4.1.12",
      "osSku": "",
      "attached_only": false,
      "specified_volumes_only": [],
      "shutdown_policy": "wait",
      "specified_launch_mode": "PARAVIRTUALIZED"
    },
    {
      "name": "MULTIPART_INSTANCE_NAME_HERE",
      "os": "windows",
      "osSku": "Server 2012 R2 Standard",
      "attached_only": false,
      "specified_volumes_only": [],
      "shutdown_policy": "shutdown",
      "specified_launch_mode": "EMULATED"
    }
  ],
  "volumes": [
    {
       "name": "MULTIPART_STORAGE_VOLUME_NAME_HERE",
     "os": "linux",
     "osKernelVersion": "4.1.12",
     "osSku": "",
     "specified_launch_mode": "PARAVIRTUALIZED"
     }
   ]
 }
```

Generate a List of Instances for Migration Using the `opcmigrate instances-export` Command

You can use the output of Oracle Cloud Infrastructure Classic Discovery and Translation Tool to generate your job files. For example, to generate a list of instances in your Oracle Cloud Infrastructure Compute Classic environment, run the following commands:

```
opcmigrate discover
opcmigrate plan create --output migration-plan.json

opcmigrate instances-export --plan migration-plan.json --format json >
instances.json
```

For more information about using these commands, see Run Oracle Cloud Infrastructure Classic Discovery and Translation Tool to Generate Reports.

From the output of this command, identify the instances that you want to migrate. Remember to add the `osKernelVersion` attribute for instances in the job file. This required attribute isn't included in the output generated by the `opcmigrate instances-export` command. Also add any other instance attributes described below, if required.

Specify Attributes of Instances for Migration

The following information is required about instances and block volumes that you want to migrate.

- `name:` Specify the full name of the instances that you want to migrate. Instance names have the following format:

  ```
  /Compute-<account_id>/<user_email_id>/<user_specified_instance_name>/
  <autogenerated_instance_name>
  ```

  You can find out the instance names from the Oracle Cloud Infrastructure Compute Classic console.
- `os:` Specify the operating system of the instance. For Linux instances, specify `linux` and for Windows instances, specify `windows`.
- `osKernelVersion:` This value is required for deciding whether an image will be imported in emulated mode or paravirtualized mode. The tool determines the best virtualization mode for an image based on the boot volume's image information and its default OS kernel version. The default virtualization modes are as follows:
  - For Linux instances with a kernel version less than 3.4 in the 3.x series, images are imported in emulated mode. Linux instances with kernel version from 3.4 to 4.x are imported in paravirtualized mode.
  - For all Windows instances, the boot image is imported in emulated mode.
  - For instances where the guest OS kernel version can't be determined, the boot image is imported in the emulated mode.

  This value is a *required* attribute for Linux instances. For other instances, this value is optional. However, it is recommended that you specify this value to ensure that boot volumes are imported in the appropriate mode if you have:
  - VMs created from custom images.
  - Bootable volumes restored from snapshots.
  - VMs created using Oracle Cloud Infrastructure Compute Classic images, where the guest OS has been updated after launching the instance.

  For any instance, if you use the `specified_launch_mode` attribute, then the virtualization mode specified there is used and the virtualization mode derived from `osKernelVersion` is ignored.

- `osSku`: For Windows instances, fill in the `osSku` attribute. Valid values for the `osSku` attribute are:
  - `Server 2008 R2 Enterprise`
  - `Server 2008 R2 Standard`
  - `Server 2008 R2 Datacenter`
  - `Server 2012 Standard`
  - `Server 2012 Datacenter`
  - `Server 2012 R2 Standard`
  - `Server 2012 R2 Datacenter`
  - `Server 2016 Standard`
  - `Server 2016 Datacenter`

  For Linux instances, leave the `osSku` attribute blank.
- `shutdown_policy`: Use `shutdown_policy` to specify if the instances to be migrated should be shut down during migrations, and if so, how. Valid values for this attribute are `ignore, shutdown`, and `wait` (the default).
  - `ignore`: The tool doesn't wait for the instances to shut down. It proceeds with the migration right away.
    For Windows instances that use RAID or mirrored volumes, don't set the shut down policy to `ignore`. If you do, then Windows will fail to recognize the volumes due to consistency checks after migration. In this case, you must specify the RAID or mirror volumes manually. To avoid this issue, select one of the other shut down policies.
  - `shutdown`: The tool shuts down all instances with this policy specified, before proceeding with the migration.
  - `wait`: You must shut down all instances before the migration starts. Use the Oracle Cloud Infrastructure Compute Classic console or any other interface to shut down the instances. When any instance in a job has the policy `wait`, the state of all instances and volumes in that job is set to `ready` after the attachment and volume information is captured. After all instances to be migrated have been shut down, use the `opcmigrate migrate instance job resume` command to resume the migration. See Stop and Restart the Migration Service.

  Use the `shutdown` or the `wait` mode to ensure that all data has been written to the boot volume before the volume is migrated.

  Note that instances must not be shut down *before* you start a migration job. Even if you specify the shut down policy `wait` or `shutdown`, all instances that are part of a migration job must be running when you start the migration job.
- `specified_launch_mode`: This attribute is optional. By default, the launch mode for an instance is determined based on the value of the `osKernelVersion` attribute. The default virtualization modes are as follows:
  - For Linux instances with a kernel version less than 3.4 in the 3.x series, images are imported in emulated mode. Linux instances with kernel version from 3.4 to 4.x are imported in paravirtualized mode.
  - For all Windows instances, the boot image is imported in emulated mode.

  - For instances where the guest OS kernel version can't be determined, the boot image is imported in the emulated mode.

  Specify the launch mode for the migrated image if you wish to override the default launch mode determined by the tool based on the value of `osKernelVersion`. If the default launch mode is appropriate for an instance, you can skip this attribute.

  You can specify a value for the `specified_launch_mode` attribute for instances or for unattached boot volumes. The valid values for this attribute are `PARAVIRTUALIZED` and `EMULATED`.

  If you specify the mode as `PARAVIRTUALIZED` for Windows images, ensure that you download and install the required virtualization drivers on the Windows source instances. See Prepare Your Windows Source Instances for Migration.

Specify Attributes of Attached Boot and Data Volumes for Migration

- `attached_only`: Specify whether you want to skip migrating the boot volume of an instance. You can set the `attached_only` attribute to `true` to indicate that you want to skip migrating the boot volume of an instance. In this case, only attached data volumes are migrated. This is useful if many instances use an identical image and you don't want to migrate identical boot volumes many times over. This is an optional attribute with the default value `false`, so by default the boot volume of an instance is migrated.
- `specified_volumes_only`: Specify the attached storage volumes that you want to migrate. You can use the `specified_volumes_only` attribute to specify a list of attached storage volumes you want to migrate. If specified, only the volumes in the `specified_volumes_only` list are migrated. This is an optional attribute. If an empty list is specified (the default value), all volumes are migrated.
  For example, consider an instance with a boot volume and three data volumes.
  - To migrate all the block volumes, use:

    ```
    {
       "name": "MULTIPART_INSTANCE_NAME_HERE",
       "os": "linux",
       "osKernelVersion": "4.1.12",
       "osSku": "",
       "attached_only": "false",
       "specified_volumes_only": []
    }
    ```

  - To migrate data volumes 1 and 2, but not 3, use:

    ```
    {
       "name": "MULTIPART_INSTANCE_NAME_HERE",
       "os": "linux",
       "osKernelVersion": "4.1.12",
       "osSku": "",
       "attached_only": "false",
       "specified_volumes_only": ["/Compute-590693805/
    jack.jones@example.com/vol1", "/Compute-590693805/
    jack.jones@example.com/vol2"]
    }
    ```

    Here the boot volume isn't explicitly included in the list of `specified_volumes_only`, so it won't be migrated, even though `attached_only` is set to `false`.

- When both `attached_only` and `specified_volumes_only` are used, both filters are applied. This means that only volumes satisfying both conditions are migrated.
  For example, consider an instance with a boot volume and three data volumes.
  - To migrate the boot volume and data volumes 1 and 2, but not 3, use:

    ```
    {
       "name": "MULTIPART_INSTANCE_NAME_HERE",
       "os": "linux",
       "osKernelVersion": "4.1.12",
       "osSku": "",
       "attached_only": "false",
       "specified_volumes_only": ["/Compute-590693805/
    jack.jones@example.com/boot_vol", "/Compute-590693805/
    jack.jones@example.com/vol1", "/Compute-590693805/
    jack.jones@example.com/vol2"]
    }
    ```

  - To migrate data volumes 1 and 2, but not 3 and not the boot volume, use:

    ```
    {
       "name": "MULTIPART_INSTANCE_NAME_HERE",
       "os": "linux",
       "osKernelVersion": "4.1.12",
       "osSku": "",
       "attached_only": "false",
       "specified_volumes_only": ["/Compute-590693805/
    jack.jones@example.com/vol1", "/Compute-590693805/
    jack.jones@example.com/vol2"]
    }
    ```

    Here the boot volume won't be migrated because it isn't included in the list of `specified_volumes_only`.

  - To migrate all the data volumes but not the boot volume, use:

    ```
    {
       "name": "MULTIPART_INSTANCE_NAME_HERE",
       "os": "linux",
       "osKernelVersion": "4.1.12",
       "osSku": "",
       "attached_only": "true",
       "specified_volumes_only": []
    }
    ```

    Here the boot volume won't be migrated because `attached_only` is set to `true`. In this case, regardless of the value of the `specified_volumes_only` list, the boot volume won't be migrated.

Specify Attributes of Unattached Boot and Data Volumes for Migration

- `volumes:` Specify the unattached storage volumes that you want to migrate. You can use this list to specify storage volumes restored from colocated snapshots as well. Note that, while you can migrate *remote* snapshots by restoring those snapshots as a block volume in Oracle Cloud Infrastructure, you can't directly migrate *colocated* snapshots. To migrate a colocated snapshot, first restore it to a volume in your Oracle Cloud Infrastructure Compute Classic account, then migrate that volume by including it in the list of unattached volumes to be migrated. Remember that it takes a longer time to migrate a volume restored from a colocated snapshot, than it does to migrate the original volume.
  For unattached boot volumes, specify values for `os`, `osKernelVersion` and `osSku`, if required.

Migrate Data Incrementally to Reduce Down Time

You can use the attribute `"incremental": "true"` in a job file to specify that data on all attached storage volumes in a migration job should be copied incrementally.

This attribute applies to all storage volumes attached to any instance in a job file. Incremental migration doesn't apply to unattached storage volumes defined in the `"volumes"` section of the job file.

> **Note:**
>
> This attribute is supported in recent versions of the Oracle Cloud Infrastructure Classic Migration Tools image. If you created your Control-S instance prior to May 2019, you won't be able to perform incremental migrations. Note also that this feature isn't supported in all sites. If you need to perform an incremental migration in a site where this feature isn't currently supported, please contact Oracle Support.

Specifying that data should be copied incrementally reduces the down time of source instances. With this option specified, the migration process is completed in two phases.

1. In the first phase, Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool generates base snapshots of all the specified boot and data volumes while instances are still running.
2. The tool uses these base snapshots to perform the first phase of the migration.
3. At the end of the first phase, the corresponding boot and data volumes are available in the target environment with the base data. At this stage, the tool displays the job status as ready. Whenever convenient, you can proceed to the second phase of migration.
4. In the second phase, source VMs must be shut down according to the specified shutdown policy.
   - With shutdown policy `ignore,` you must shut down the source VMs before resuming the migration job. Note that for jobs where incremental isn't explicitly specified as `true,` the entire migration procedure completes without shutting down the source VM. However, when an incremental migration is performed,

you must shut down source instances even if the specified shutdown policy is `ignore`, and you must resume the migration after the VMs are shut down.

- With shutdown policy `wait`, you must shut down the source VMs before resuming the migration job. The tool will continue to wait until all source VMs with this policy have been shut down. If any source VMs with this policy specified are still running when you resume the migration job, the tool will throw an error.
- With shutdown policy `shutdown`, you can resume the migration job while VMs with this policy are still running. The tool will shut down the running VMs and resume the migration job. If you shut down VMs with this policy specified, after the first phase completes and before you resume the migration, the tool will simply verify that the VMs are shut down and will resume the migration job.

5. Resume the migration job by using the `opcmigrate migrate instance job resume <job_name>` command.
6. A second snapshot is created for each of the boot and data volumes being migrated.
7. Data that has changed between the base snapshot and the second snapshot is then copied over to the target volumes. As this incremental data would typically be much smaller in size than the base data, this process can be completed much faster than the first phase of the migration.

When an incremental migration is performed, the second phase of the migration uses the associated Oracle Cloud Infrastructure Object Storage Classic account to copy data to the target volumes. You must specify the required credentials and access point in the `secret.yml` file.

Copying data incrementally is disabled by default. To enable it, specify it in the job file:

```
{
  "version": "1",
  "incremental": true,
  "instances": [...
```

When specified, this approach is applied to all attached boot and data volumes in the migration job.

> **Caution:**
>
> When you perform an incremental migration, at the end of the first phase, migrated volumes are available in the target environment. However, ensure that you don't attempt to attach and use the migrated volumes until the entire migration process is completed. If you access data on the migrated volumes after the first phase of migration is completed and before the second phase of migration has started, it could cause corruption of data in the migrated volume.

Prepare Your Linux Source Instances for Migration

You need to configure your source instances so that they can be re-initialized correctly in the target environment.

You can configure your instances either by using tools provided on the Control-S instance or manually using custom tooling or fleet managers. The following OS versions can be configured using the tools on Control-S:

- Oracle Linux 7.2
- Oracle Linux 6.8
- Ubuntu Server 18

For instances running other versions of Linux, the tools available on Control-S can be used to configure the instances if all dependencies, such as python, `lsblk`, `iscsi client utils` and so on are satisfied.

Instances running Oracle Linux 5.11 can't be configured using these tools. However, you can still proceed with migrating these instances using this tool. After migration is completed, launch the instances in Oracle Cloud Infrastructure and attach the volumes to each instance manually. Then run the iSCSI commands to mount the volumes on each instance and update the mount point information on the instance.

To prepare your Linux source instances, start by completing the following steps:

1. Ensure that you have SSH access to each Linux source instance.
2. On each source instance, install via `yum` or `apt`:
   - `iscsi-initiator-utils` – This is required for `iscsiadm`
   - `cronie`
   - `util-linux-ng` on Oracle Linux 5.x or Oracle Linux 6.x instances, and `util-linux` on Oracle Linux 7.x instances. This is required for `lsblk`.
3. Verify that all secondary attached volume mounts have `_netdev` and `nofail` specified in `/etc/fstab`, so that when the instance is launched on Oracle Cloud Infrastructure it can start before the required volumes are attached.
   Ensure that you **don't** specify `_netdev` and `nofail` in `/etc/fstab` for file systems that are stored in bootable storage volumes.

   You are required to specify the `_netdev` and `nofail` options in `/etc/fstab` only for the non-bootable storage volumes entries, such as `/dev/xvdb` and `/dev/xvdc`. Do not specify `_netdev` and `nofail` options in `/etc/fstab` for any volume groups and other non `/dev/` entries.
4. Ensure that, on each source instance, NOPASSWD is set for the user that you will connect to the instance as. If you will connect to the instance as the `opc` user, then verify the following entry in the `/etc/sudoers` file:

   ```
   opc ALL=(ALL) NOPASSWD: ALL
   ```
5. Verify that the `70-persistent-net.rules` file doesn't exist in the `/etc/udev/rules.d` directory. The network rules in this file can cause network devices to be renamed after migration due to changes in the MAC address of the VM. This could make the network interface unreachable after migration. To prevent such network issues, before you migrate the instance remove or rename this file if it exists.
6. Ensure that all kernel updates have been completed successfully and that the instance has been rebooted after the most recent kernel update. Check that the default grub boot selection is the same as the latest kernel version found under `/boot` and the current running kernel version.

Next, to use the tools on the Control-S instance to complete the process, see Prepare Your Linux Source Instances Using Tools on Control-S.

To complete the process using the manual steps, see Prepare Your Linux Source Instances Manually.

Prepare Your Linux Source Instances Using Tools on Control-S

To prepare the source instances using tools on the Control-S instance, do the following.

1. Copy the private SSH key for each source instance to the Control-S instance and modify permissions on the key file to restrict access. Alternatively, you can generate an SSH key pair on Control-S and then add the public key to each Linux source instance that you want to migrate. After you've got the SSH keys set up, validate that you can connect from Control-S to each of the Linux source instances.

2. Make a note of the IP address, sudo user, and the path to the SSH private key for each Linux source instance. If you used the network and resource discovery tool to identify resources in your source environment, you can find this information in the `instances.json` file generated by that tool.

3. On Control-S, create or update the `/home/opc/ansible/hosts.yml` file. You can use the provided `hosts.yml.sample` file to create your `hosts.yml` file.

   ```
   source:
     hosts:
       1.1.1.1:
         label: label_here
         remote_user: opc
         ansible_ssh_private_key_file: ~/.ssh/private_key_here
         ansible_python_interpreter: /usr/bin/python3
       2.2.2.2:
         label: label_here
         remote_user: opc
          ansible_ssh_private_key_file: ~/.ssh/private_key_here
          ansible_python_interpreter: /usr/bin/python3
   ```

   In this file, `1.1.1.1` and `2.2.2.2` represent the IP address of each source instance. By default, the port used is 22. If you want to use another port to establish the connection, you can specify it along with the IP address. For example, to use the IP address 192.168.31.91 along with port 19600, enter: `192.168.31.91:19600`

   The label for each source instance must be unique. You can find the instance label in the Oracle Cloud Infrastructure Compute Classic web console.

   If you are migrating a Linux instance with Python 3, uncomment the path to the python interpreter. Otherwise, ensure that that line is commented out.

4. On Control-S, run the following command to configure your Linux source instances. This command installs the required scripts and creates a cron job to run a script. These scripts ensure that the tool has the information it needs to mount the attached volumes after migration.

   ```
   opcmigrate migrate instance source setup
   ```

Review the output from this command. In addition to other tasks, this script checks the Linux kernel to determine whether it is eligible for para-virtualization after migration and whether the necessary kernel modules are available. The output recommends the missing kernel modules to be added. Follow the instructions provided by the script, if any.

If this command displays a warning about missing kernel modules, you might need to install the `virtio_console` driver and the `virtio_scsi` driver. The output of this command provides the `dracut` commands to install these drivers. After installing the `virtio_console` driver using the suggested `dracut` command, run `opcmigrate migrate instance source setup` again to obtain the recommended command for installing the `virtio_scsi` driver. Make sure that all the required drivers are successfully installed before proceeding. Otherwise, after migration, the instance will fail to boot.

Ensure that you back up the file `/boot/initramfs-<kernel_version>.img` before running any `dracut` command modifying the image.

5. If you made any changes to the source instance to install drivers using the `dracut` commands, then run the setup command again, to check if any further changes are required.

   ```
   opcmigrate migrate instance source setup
   ```

   If any further warnings are displayed, complete the recommended steps and then run the command again.

Prepare Your Linux Source Instances Manually

To prepare your Oracle Linux source instances manually, do the following.

1. Copy these files from Control-S to the `/ocic_oci_mig` folder on the source instance:
   - `/home/opc/src/create_manifest.py`
   - `/home/opc/src/inject_script.sh`
   - `/home/opc/src/iscsiattach.sh`
   - `/home/opc/src/process_manifest.py`
   - `/home/opc/src/validate_guest_os.py`
2. Change the owner of these files to `root/root`.
3. On each source instance, as the `root` user, run `create_manifest.py`. It creates the files `source_manifest.json` and `breadcrumb.json` on each attached volume. These files contain the information required to determine which volumes need to be mounted on which target after migration. When your migration is complete, you can remove the `create_manifest.py` file from the source instances.
4. On each source instance, create the file `/ocic_oci_mig/hostname.txt`. Enter the instance label and the Object Storage PAR in this file:

   ```
   <my_instance_label>,https://<oci_par_url>
   ```

5. On each source instance, as the root user, run:

   ```
   sudo python /ocic_oci_mig/validate_guest_os.py
   ```

   This script checks the Linux kernel to determine whether it is eligible for para-virtualization after migration and whether the necessary kernel modules are available. The output recommends the missing kernel modules to be added. Follow the instructions provided by the script, if any.

   If this command displays a warning about missing kernel modules, you might need to install the `virtio_console` driver and the `virtio_scsi` driver. The output of this command provides the `dracut` commands to install these drivers. After installing the `virtio_console` driver using the suggested `dracut` command, run `validate_guest_os.py` again to obtain the recommended command for installing the `virtio_scsi` driver. Make sure that all the required drivers are successfully installed before proceeding. Otherwise, after migration, the instance will fail to boot.

   Ensure that you back up the `/boot/initramfs-<kernel_version>.img` file before running any `dracut` command modifying the image.

6. Add the following entry to `crontab` to ensure the successful automount of attached disks. This cronjob runs every 5 minutes.

   ```
   */5 * * * * /ocic_oci_mig/inject_script.sh
   ```

Prepare Your Windows Source Instances for Migration

Use the script provided in Control-S to set up your Windows instances for migration.

The following Windows versions can be configured for migration:

- Server 2008 R2 Enterprise
- Server 2008 R2 Standard
- Server 2008 R2 Datacenter
- Server 2012 Standard
- Server 2012 Datacenter
- Server 2012 R2 Standard
- Server 2012 R2 Datacenter
- Server 2016 Standard
- Server 2016 Datacenter

For each Windows instance that you want to migrate, do the following.

1. Ensure that you have RDP access to each instance as the Administrator.
2. Use RDP to log in to the instance as the Administrator.
3. Copy the file `/home/opc/src/windows_migrate.ps1` from Control-S to each source instance.

4. On each source instance, navigate to the folder where you've saved the file and run `windows_migrate.ps1`.

   ```
   .\windows_migrate.ps1
   ```

5. Windows instances are migrated in the `EMULATED` mode by default. If you want to migrate a Windows instance in the `PARAVIRTUALIZED` mode, then download and install the Windows paravirtualized drivers on the Windows instance before you begin the migration. Log in to your Oracle account and download Patch 27637937. Alternatively:

   a. Log in to the Oracle Software Delivery Cloud site.

   b. In the search fields, select **Release** and enter **Oracle Linux.** Click **Search.**

   c. From the search results, select **REL: Oracle Linux 7.6.0.0.0** and click **Add to Cart.**

   d. Click **Checkout.**

   e. From the **Platforms/Languages** list, select **x86 64 bit** and click **Continue.**

   f. Read and accept the license agreement and click **Continue.**

   g. From the list of files, select **V981734-01.zip Oracle VirtIO Drivers Version for Microsoft Windows 1.1.3, 68.0 MB** and download the file.

   After the download completes, go to the folder where you downloaded the file. You must perform a custom install to install the drivers correctly.

   a. Double-click `installer.exe`

   b. Follow the prompts in the installation wizard.

   c. On the Installation Type page, select **Custom.**

   d. Restart the instance to complete the installation, if required.

Launch the Migration Controller Instance (Control-T) in the Target Environment

Use the setup script provided in Control-S to launch the migration controller in the target environment.

The command to set up Control-T creates the `ocic-oci-msg` bucket in Oracle Cloud Infrastructure Object Storage, with visibility set to `public.` If this bucket already exists, ensure that its visibility is set to `public` before you run the command to set up Control-T.

- To set up the Control-T instance on Oracle Cloud Infrastructure, log in to the Control-S instance and run the following command:

  ```
  opcmigrate migrate instance ctlt setup
  ```

  This command takes several minutes to complete. Monitor the `/images/migration_logs/control_t_deploy.log` file to view the progress.

Migrate the Specified VMs and Block Volumes

When you've prepared the source instances and completed configuring the migration controller instances in the source and target environments, you're ready to start the migration.

Start the Migration

Use the migration script on the Control-S instance to start the migration.

- To start the migration, log in to Control-S and run:

  ```
  opcmigrate migrate instance job run
  ```

- By default, the migration job uses the list of instances provided in the `secret.yml` file to identify resources to be migrated. If you want to run multiple jobs in parallel, specify a job file for each job.

  > **Note:**
  >
  > You can run up to four migration jobs in parallel.

  ```
  opcmigrate migrate instance job run --job_file <full_path/job_file_name>
  ```

The migration begins and the display shows the ongoing status of the process. Depending on the size and number of storage volumes being migrated, the process could take a few hours to complete.

Monitor the Migration

As the migration proceeds, you can see the volumes being created and listed in the Oracle Cloud Infrastructure Console.

- To monitor the status of all migration jobs:

  ```
  opcmigrate migrate instance job list
  ```

  This command displays information about all running migration jobs, including the job name, start time, and percent completed. Here's a sample output of this command.

  ```
  opcmigrate migrate instance job list
  opcmigrate-instance: INFO: [
  opcmigrate-instance: INFO:   {
  opcmigrate-instance: INFO:     "creation_time": "2019-03-21
  09:46:38.352254+00:00",
  opcmigrate-instance: INFO:     "instances": [
  opcmigrate-instance: INFO:       {
  opcmigrate-instance: INFO:         "creation_time": "2019-03-21
  09:46:38.356428+00:00",
  ```

```
opcmigrate-instance: INFO:         "src_instance_name":
"<instance_name>",
opcmigrate-instance: INFO:         "start_time": "2019-03-21
09:46:38.420505+00:00",
opcmigrate-instance: INFO:         "status": "completed",
opcmigrate-instance: INFO:         "status_details": "The migration
completed successfully",
opcmigrate-instance: INFO:         "task_id": "bf1172b7-4393-490f-9532-
c76d024fa1c1",
opcmigrate-instance: INFO:         "update_time": "2019-03-21
10:09:59.948118+00:00",
opcmigrate-instance: INFO:         "volumes": [
opcmigrate-instance: INFO:           {
opcmigrate-instance: INFO:             "creation_time": "2019-03-21
09:46:38.361539+00:00",
opcmigrate-instance: INFO:             "size": 12884901888,
opcmigrate-instance: INFO:             "src_volume_name":
"<storage_volume_name>",
opcmigrate-instance: INFO:             "start_time": "2019-03-21
09:46:38.434408+00:00",
opcmigrate-instance: INFO:             "status": "completed",
opcmigrate-instance: INFO:             "status_details": "The migration
completed successfully",
opcmigrate-instance: INFO:             "task_id":
"584313e1-8d42-4a95-908f-5bdc2506fa5a",
opcmigrate-instance: INFO:             "tgt_name": "_instance-for-
migration2_5c60fe-9f4a-96ac3-8dfe-...",
opcmigrate-instance: INFO:             "tgt_ocid":
"ocid1.image.oc1.iad.aaaaaaaadvmgexz5iu...",
opcmigrate-instance: INFO:             "update_time": "2019-03-21
10:09:59.965311+00:00"
opcmigrate-instance: INFO:           }
opcmigrate-instance: INFO:         ]
opcmigrate-instance: INFO:       },
opcmigrate-instance: INFO:       {
opcmigrate-instance: INFO:         "creation_time": "2019-03-21
09:46:38.364923+00:00",
opcmigrate-instance: INFO:         "src_instance_name":
"<instance_name>",
opcmigrate-instance: INFO:         "start_time": "2019-03-21
09:46:38.426048+00:00",
.
.
.
opcmigrate-instance: INFO:             "task_id": "f6efe3b5-4099-4846-
b0ce-52efa44b3f3e",
opcmigrate-instance: INFO:             "tgt_name": "",
opcmigrate-instance: INFO:             "tgt_ocid": "",
opcmigrate-instance: INFO:             "update_time": "2019-03-21
10:00:45.786605+00:00"
opcmigrate-instance: INFO:           }
opcmigrate-instance: INFO:         ]
opcmigrate-instance: INFO:       }
opcmigrate-instance: INFO:     ],
opcmigrate-instance: INFO:     "name": "0e59b00c-05a0-4cc3-8dfe-
```

```
f2383f0ef6fd",
opcmigrate-instance: INFO:     "start_time": "2019-03-21
09:46:38.348893+00:00",
opcmigrate-instance: INFO:     "status": "completed",
opcmigrate-instance: INFO:     "status_details": "The migration
completed successfully",
opcmigrate-instance: INFO:     "update_time": "2019-03-21
10:24:03.482980+00:00"
opcmigrate-instance: INFO:   },
opcmigrate-instance: INFO:   {
opcmigrate-instance: INFO:     "creation_time": "2019-03-21
11:20:38.475810+00:00",
opcmigrate-instance: INFO:     "instances": [
opcmigrate-instance: INFO:       {
opcmigrate-instance: INFO:         "creation_time": "2019-03-21
11:20:38.479765+00:00",
opcmigrate-instance: INFO:         "src_instance_name":
"<instance_name>",

.
.
.
opcmigrate-instance: INFO:             "tgt_name": "_instance-for-
migration2_5c60c336-e298-4bfe...",
opcmigrate-instance: INFO:             "tgt_ocid":
"ocid1.image.oc1.iad.aaaaaaaaafjdc4dinxdmoj7mot...",
opcmigrate-instance: INFO:             "update_time": "2019-03-21
11:44:13.536100+00:00"
opcmigrate-instance: INFO:           }
opcmigrate-instance: INFO:         ]
opcmigrate-instance: INFO:       }
opcmigrate-instance: INFO:     ],
opcmigrate-instance: INFO:     "name": "9c539f0d-20eb-4e2f-a2f6-
fbae3d0bb636",
opcmigrate-instance: INFO:     "start_time": "2019-03-21
11:20:38.472754+00:00",
opcmigrate-instance: INFO:     "status": "completed",
opcmigrate-instance: INFO:     "status_details": "The migration
completed successfully",
opcmigrate-instance: INFO:     "update_time": "2019-03-21
11:44:13.525604+00:00"
opcmigrate-instance: INFO:   }
opcmigrate-instance: INFO: ]
```

- To monitor the status of a specified migration job, use the following command and specify the job name. You can use the `opcmigrate migrate instance job list` command to get the job name for a particular job.

  ```
  opcmigrate migrate instance job status <job_name>
  ```

  Here's a sample output of this command:

  ```
  opcmigrate migrate instance job status 9c539f0d-20eb-4e2f-a2f6-
  fbae3d0bb636
  ```

```
opcmigrate-instance: INFO:   {
opcmigrate-instance: INFO:     "creation_time": "2019-03-21
11:20:38.475810+00:00",
opcmigrate-instance: INFO:     "instances": [
opcmigrate-instance: INFO:       {
opcmigrate-instance: INFO:         "creation_time": "2019-03-21
11:20:38.479765+00:00",
opcmigrate-instance: INFO:         "src_instance_name":
"<instance_name>",

.
.
.
opcmigrate-instance: INFO:             "tgt_name": "_instance-for-
migration2_5c60c336-e298-4bfe...",
opcmigrate-instance: INFO:             "tgt_ocid":
"ocid1.image.oc1.iad.aaaaaaaaafjdc4dinxdmoj7mot...",
opcmigrate-instance: INFO:             "update_time": "2019-03-21
11:44:13.536100+00:00"
opcmigrate-instance: INFO:           }
opcmigrate-instance: INFO:         ]
opcmigrate-instance: INFO:       }
opcmigrate-instance: INFO:     ],
opcmigrate-instance: INFO:     "name": "9c539f0d-20eb-4e2f-a2f6-
fbae3d0bb636",
opcmigrate-instance: INFO:     "start_time": "2019-03-21
11:20:38.472754+00:00",
opcmigrate-instance: INFO:     "status": "completed",
opcmigrate-instance: INFO:     "status_details": "The migration
completed successfully",
opcmigrate-instance: INFO:     "update_time": "2019-03-21
11:44:13.525604+00:00"
opcmigrate-instance: INFO:   }
```

A migration job can have one of the following statuses:

- `preparing`: The job is preparing for instances to be shut down or it is preparing for the second phase of an incremental migration.
- `running`: The job is running and data is being migrated.
- `ready`: The job is waiting for instances with the shutdown policy `wait` to be shut down, or the job is waiting for the resume command to be issued to start the second phase of an incremental migration.
- `pending`: The job is pending for processing after being resumed.
- `completed`: The job has completed successfully.
- `deleting`: The job is being deleted.
- `error`: The job has failed due to errors migrating some volumes or instances.

- To monitor the status of the migration service, run:

```
opcmigrate migrate instance service status
```

This command displays information about Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool build date and version and the status of the service. It doesn't provide any information about migration jobs. Here's a sample output of this command:

```
opcmigrate migrate instance service status
opcmigrate-instance: INFO: Checking Migration service PID 2334
opcmigrate-instance: INFO: Migration service with PID 2334 is running
opcmigrate-instance: INFO: {
opcmigrate-instance: INFO:   "build_date": "2019-03-05T21:13:36Z",
opcmigrate-instance: INFO:   "hash": "dce83be497...",
opcmigrate-instance: INFO:   "status": "running",
opcmigrate-instance: INFO:   "version": "1.5.0"
opcmigrate-instance: INFO: }
```

- You can also check the log files in `/images/migration_logs` while a migration job is in progress or after it is complete. The `run_migration.log` file is the main migration log file. The `control_t_deploy.log` file is the log file created when launching Control-T. Most failure information can be found in these files.

Stop and Restart the Migration Service

To interrupt a migration job, you can stop and then restart the migration service. You can then resume interrupted jobs.

- To interrupt the migration service, log in to Control-S and run:

  ```
  opcmigrate migrate instance service stop
  ```

- If a migration job is interrupted, for example if the service was killed or the Control-S instance was rebooted, you can resume the migration by restarting the service. The service resumes from the point at which it was interrupted. To restart the migration service after it has been stopped, log in to Control-S and run:

  ```
  opcmigrate migrate instance service start
  ```

  All interrupted jobs, that is, all jobs that were in the pending or initializing states, are resumed.

> **Note:**
>
> When you resume a migration job, your migration environment must not change other than to fix any issues that prevented the migration job from completing. Specifically, the migration controller instances, both Control-S and Control-T, must be the same. Any scripts running or files created on the source VMs and storage volumes to be migrated must not be cleaned up.

Resume a Migration Job

You might need to resume a migration job in several scenarios.

- If a migration job fails, you can resume the migration job. In this case, if boot volumes have already been uploaded, the job resumes from the image import stage. For all other cases, the job resumes from the beginning.
- If you used the default shutdown policy `wait` for any instance in a migration job, then the job won't begin until you have shut down all instances that have this policy specified. After you have shut down the relevant instances, you must resume the migration job.
- If you have specified an incremental migration in the job file, then after the first phase of the migration is completed, the tool will wait for source instances to be shut down before proceeding with the second phase of the migration. In this case, when you are ready to proceed with the second phase, you must resume the migration job. The source instances must be shut down as per the specified shutdown policy. For instances with a shutdown policy of `ignore` or `wait`, you must shut down the instances before you resume the migration. For instances with a shutdown policy of `shutdown,` the tool shuts down the instances after you resume the migration job.

Note that, when an incremental migration is specified, you must resume the migration job even if the shutdown policy specified was `ignore` or `shutdown`. However, if the incremental attribute isn't explicitly specified as `true` in the job file, then an incremental migration isn't performed. In this case, if the shutdown policy specified was `ignore` or `shutdown`, then you don't need to resume the migration job. The job automatically proceeds to completion.

- To resume a migration job, log in to Control-S and run:

```
opcmigrate migrate instance job resume <job_name>
```

You can set the `--do-not-finalize` option to retain information about snapshots for incremental migrations so that you can perform another incremental migration in the future. This allows you to perform multiple incremental migrations between the base migration and the final migration which reduces downtime when compared to the two-phase migration approach. By default, the `--do-not-finalize` option is not set.

You can perform additional incremental migrations to reduce the down time.

For the base migration phase or when your migration job that has the shutdown policy `wait`, do not use the `--do-not-finalize` option. In such a scenario, run the resume command without the `--do-not-finalize` option.

In the second or subsequent phase of incremental migration, you can use the `--do-not-finalize` option.

Let's consider the following example to understand the scenarios in which you can use the `--do-not-finalize` option. In this example, John Doe starts a migration and sees that the base migration took 24 hours to complete. Based on this, John Doe estimates that incremental migration will take 45 minutes. If a downtime of 45 minutes is not acceptable, then John Doe can use the `--do-not-finalize` option while running the resume command in phase 2. After the phase 2 of the migration is complete, John Doe estimates that another incremental migration will take 5 minutes. John Doe finds a downtime of 5 minutes is acceptable. Now, John Doe can issue the resume command without using the `--do-not-finalize` option to finalize the migration. In this last phase, John Doe will have to take a down time and shutdown their instances.

- To resume a migration job, log in to Control-S and run:

  ```
  opcmigrate migrate instance job resume <job_name> --do-not-finalize
  ```

Abort a Migration Job

You can abort a job which is in the pending or running state, but you can't abort a job which is in the ready state.

- To abort a migration job, log in to Control-S and run:

  ```
  opcmigrate migrate instance job abort <job_name> <scope>
  ```

Depending on your scenario, specify one of the following values for the `scope` option.

- `nocleanup`: Use this value when you don't want the artifacts created by the migration to be cleaned up. When you resume a migration job after aborting a job with this scope, the migration is resumed from the point where the abort command was issued.
- `partial`: Use this value when you want to retain all artifacts that were successfully migrated but you want the resources to be cleaned up for artifacts that were partially migrated. When you resume a migration job after aborting the job with this scope, the tool retries migrating the artifacts that were not previously migrated successfully.
- `complete`: Use this value when you want all the artifacts created by the migration job to be destroyed, including the artifacts that were migrated successfully. When you resume a migration job after aborting the job with this scope, the tool restarts the entire migration from the beginning as all the resources that were previously migrated are destroyed.

After starting a migration job, you might need to abort it in several scenarios.

- If you have accidentally started a migration job, you might want to abort the job completely. In this case, set the `scope` value as `complete`.
- If you want to change some configurations in the source VM or install drivers after starting a migration job, you can abort the job partially or completely, and then resume the migration after making the required changes. In this case, set the `scope` value as `partial` or `complete` depending on whether you want to abort the job partially or completely.
- If you notice that the migration is proceeding slowly due to network issues, you can pause the migration and resume the job after fixing the network issues. In this case, set the `scope` value as `nocleanup`.

Let's consider the following example to further understand the differences between the scope values. In this example, the migration job consists of migrating three volumes: vol_1, vol_2, and vol_3. When you issue the abort command for this migration job, let's assume that the migration of vol_1 has completed successfully, migration of vol_2 has started and is in progress, and migration of vol_3 has not started yet and is enqueued for migration.

When you specify `nocleanup` as the scope, the tools stops the migration of the volumes. None of the artifacts created by the migration are cleaned up.

When you specify `partial` as the scope, the tools stops the migration of the volumes. Any volumes that are in-progress have corresponding temporary resources cleaned up, but any volumes that have successfully completed migration are left untouched. In this example, the temporary resources corresponding to vol_2 are cleaned up but there is no impact on the volume created in Oracle Cloud Infrastructure corresponding to vol_1's migration. The temporary resources that are cleaned up include snapshots taken on the source volume, clones, any OCI custom images and volumes created in Oracle Cloud Infrastructure, related attachments to Control-S and Control-T instances. When you resume this migration job, the tool creates all the required resources to migrate vol_2 and then vol_3 as temporary resources were cleaned up during the abort.

When you specify `complete` as the scope, the tools stops the migration of the volumes and all the artifacts created by the migration job are destroyed including any volumes that were successfully migrated to the target environment. In this example, the Oracle Cloud Infrastructure volume corresponding to vol_1's migration is deleted along with temporary artifacts created for vol_2's migration.

To resume a migration job after aborting it, see Resume a Migration Job.

Delete a Migration Job

If a job has completed successfully or if it has gone into an error state and you want to stop it, you can delete the job.

- To delete a migration job, log in to Control-S and run:

  ```
  opcmigrate migrate instance job delete <job_name>
  ```

Launch VMs in the Target Environment

When a migration job completes, the boot volumes that were migrated in that job are available as boot volumes in your tenancy and the data volumes that were migrated are listed on the block volumes page. Next, use the boot volumes to launch compute instances in your Oracle Cloud Infrastructure tenancy.

Remember that a compute VM must be launched in the same availability domain as the boot volume that you want to use. If you want to launch a VM in a different availability domain, you can create a backup of the boot volume that you want to use and restore it in the required availability domain, before using it to launch a VM. Launch your VMs by using the Oracle Cloud Infrastructure Console or any other interface. Use the migrated boot volumes to launch the required compute instances. When your instances are ready, attach the appropriate data volumes to each instance.

Alternatively, you can complete the following steps to launch your instances using the boot volumes and attach the appropriate data volumes to each instance using Terraform.

1. Import information about the OCIDs of the migrated volumes into the Terraform configuration file. Run:

   ```
   opcmigrate volumes-import
   ```

This command fetches volume OCIDs for all the volumes that have been migrated and provides this information in the form of Terraform variables. Add these variables to your `terraform.tfvars` file.

2. Edit the Terraform configuration as required, to ensure that you have all the information needed to launch the VMs.
3. Re-apply the Terraform configuration in your Oracle Cloud Infrastructure tenancy. VMs are launched using the appropriate boot volumes. After the VMs have started, the appropriate data volumes are attached to each VM.

Attach and Mount Block Storage on Compute Instances in the Target Environment

If you use the Terraform configuration generated by `opcmigrate generate`, then when you launch your compute instances, the appropriate storage volumes are automatically attached to each instance.

If you launch instances using any other interface, ensure that you attach the required storage volumes to each instance.
For Linux instances, after the migrated storage volumes have been successfully attached to each instance, the storage volumes must mounted be on each instance.
For each migrated Linux instance:

1. To mount the attached storage volumes on each migrated instance, log in to Control-S and run:

   ```
   opcmigrate migrate instance attachment_ready <instance_ocid>
   ```

2. Confirm that this process has completed successfully. Log in to the migrated instance and view the log file `/ocic_oci_mig/inject_script_logfile.log`. You should see the following message at the end of the file.

   ```
   cleanup the cron job for inject_script.sh
   ```

   This message indicates that volumes have been successfully mounted.

3. Alternatively, run the scripts on the boot volume of the migrated instance to mount the block volumes. Log in to the migrated instance and run the following commands:

   ```
   cd /ocic_oci_mig

   sudo ./iscsiattach.sh

   sudo python ./process_manifest.py
   ```

4. After these scripts complete, reboot the instance. When the instance restarts, use the `lsblk` command or view the `/etc/fstab` file to verify that the attached block volumes are mounted and all mounts are good.

   Disk device names will change on the migrated instance from `/dev/xvd*` to `/dev/sd*`. When the instance reboots, the device name `/dev/sd*` might not be associated with same block volume. To keep mounts consistent and data safe, it is recommended that you use UUIDs instead of device names in the `/etc/fstab` file.

When the instance reboots, verify that no /dev/xvd* devices are listed in /etc/fstab. Non-LVM devices /dev/xvd* should have been changed into UUID=**** form. If /dev/xvd* entries still exist in /etc/fstab, it indicates that automounting has not completed successfully.

5. If you are using a custom configuration that depends on the disk device names, you may need to perform some manual configuration. For example, the migrated instance's LVM filter will be automatically modified to accept all devices named /dev/sd*. If you are using a custom LVM filter configuration, you may need to manually edit the filter and/or global_filter entries in /etc/lvm/lvm.conf to accept only the appropriate devices.
6. If any block volume fails to be discovered or if the automated mount fails, follow the documented iSCSI procedure to mount the volumes.

Validate the Target Environment

After you've completed migration of your VMs and block storage, validate the setup of your target environment. Ensure that all VMs are running and can be accessed, that all storage volumes are attached and mounted, and that all network and firewall rules have been appropriately implemented.

Validate Your VMs and Block Storage in the Target Environment

After you have launched each of your migrated VMs, log in to each VM to ensure that you have access to the system and to verify that all the required block volumes are attached and mounted as expected.

Verify that your instances are launched in the emulated mode or the paravirtualized mode, as expected. Linux instances with a kernel version less than 3.4, are imported in emulated mode. Instances with a higher kernel version are imported in paravirtualized mode. If the guest OS kernel version couldn't be determined, the image is imported in emulated mode.

Validate Your Windows Licenses in the Target Environment

After launching your Windows VMs, check your Windows license.

- Log in to each Windows VM using RDP. In PowerShell or a command prompt window, enter:

  ```
  slmgr /dli
  ```

  The Windows Script Host dialog box appears. Verify that the new KMS address **169.254.169.253** is displayed and that the License status is **Licensed**.

Validate the Network Setup

When your instances are running, verify that network access to each instance is both permitted and restricted as intended.

Check each of the following, as applicable:

- VCNs and subnets have been created corresponding to the IP networks and the shared network in your source environment. Remember that the mapping of source environment networking to the target environment depends on whether you adopted the network topology mapping strategy or the security context mapping strategy.
- Instances are launched in the appropriate subnet.
- Appropriate security lists are applied to each subnet and appropriate security rules are created in each security list.
- Instances are accessible over the public Internet, where required.
- Instances *aren't* accessible over the public Internet, where such access isn't required. If you created all subnets as public subnets, ensure that instances in those subnets that should not be accessed over the public Internet don't have a public IP address.
- Instances in different subnets in the same VCN can communicate with each other as expected.
- Instances in different VCNs can't communicate with each other unless such communication has been explicitly enabled.
- If you used network topology mapping to set up your network, ensure that appropriate firewall rules are in place on instances to restrict network access, if required. The network topology mapping strategy might in some cases enable traffic that was restricted in the source environment.
- If you had IP networks connected by an IP network exchange in the source environment, ensure that there is connectivity between the corresponding VCNs and subnets in the target environment. If the IP networks were migrated to separate VCNs, ensure that local peering gateways (LPGs) have been set up to enable connectivity across those VCNs.
- If you migrated your FastConnect connection or if you set up a VPN connection, validate that those connections are working as expected.
- If you connected your Oracle Cloud Infrastructure Compute Classic network with your Oracle Cloud Infrastructure network, ensure that that connection is working as expected.

Troubleshooting

Here are a few tips for dealing with errors that might occur while installing and using Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool.

- **Setting up Control-S fails with one of the following errors:**
  - `secret.yml has insecure file permissions; please ensure they are correct by running `chmod 600 ansible/secret.yml``

    This error indicates that permissions on the file /home/opc/ansible/secret.yml aren't restricted. The `secret.yml` file contains sensitive information about your account. Ensure that you set the permissions appropriately using `chmod 600`.
  - `fatal: [127.0.0.1]: FAILED! => {"changed": false, "msg": "AnsibleUndefinedVariable: 'opc_profile_endpoint' is undefined"}`

This error indicates that there is an issue with the information provided in the `secret.yml` file. In this example, the AP end point for Oracle Cloud Infrastructure Compute Classic isn't provided. Errors in other fields in this file are similarly indicated. Fix the error and then run the command to set up Control-S again. Remember that whenever you make any changes in `secret.yml,` you have to run the command to set up Control-S again, to ensure that those changes get propagated.

  – `INFO The control-s id is b'401 Unauthorized\n'`

  This error indicates that the tool is unable to access the metadata service on Control-S. Reboot the Control-S instance and try again.

- **Setting up Control-T fails with one of the following errors:**

  – `The required information to complete authentication was not provided.`

  This error indicates that there is an issue with the credentials for the Oracle Cloud Infrastructure tenancy. This information is usually provided in the `secret.yml` file. Check the OCIDs entered in that file and ensure that they map to the OCIDs that you copied from the Oracle Cloud Infrastructure Console.

  – `oci.exceptions.ServiceError: {'opc-request-id': 'D4AFD39...','code': 'NotAuthorizedOrNotFound', 'message': 'subnet ocid1.subnet.oc1.phx.aaaaa... not found', 'status': 404}`

  This error indicates that the subnet OCID provided in the `secret.yml` file is incorrect. Check the subnet OCID entered in that file and ensure that it is the appropriate OCID for the subnet that you want to use.

  – `oci.exceptions.ServiceError: {'opc-request-id': '3375100...', 'code': 'InvalidParameter', 'message': "Parameter 'availabilityDomain' does not match. VNIC has 'phx-ad-3' while the subnet has 'phx-ad-2'", 'status': 400}`

  This error indicates that the subnet information provided in `secret.yml` conflicts with the availability domain, which is also specified in that file. From the Oracle Cloud Infrastructure Console, find the OCID for the subnet in the appropriate availability domain.

  – `no such identity: /home/opc/.ssh/private_key: No such file or directory`

  This error indicates that the SSH key required to connect to the Linux source instance wasn't found in the specified path. Verify that the private key is available in the path specified in the `hosts.yml` file on Control-S.

- **A migration job could not be submitted due to one of the following errors:**

  – `Running into connection issues: HTTPConnectionPool(host='127.0.0.1', port=8000): Max retries`

```
      exceeded with url: /api/v1/jobs (Caused by
      NewConnectionError('<urllib3.connection.HTTPConnection object
at 0x7fb4c08b7668>: Failed to
      establish a new connection: [Errno 111] Connection refused',))
```

This error indicates issues in connecting with your Oracle Cloud Infrastructure Compute Classic account. Check the `run_migration` log file for more information about the issue. For example, the log file might contain logs similar to the following:

```
File "/home/opc/oci-python-sdk/lib/python3.6/site-packages/mig/
run_migration.py", line 135, in run_migration_svc_internal

    ocic_compute_client.authenticate()

  File "/home/opc/oci-python-sdk/lib/python3.6/site-packages/mig/
retry_utils.py", line 52, in f_retry

    return f(*args, **kwargs)

  File "/home/opc/oci-python-sdk/lib/python3.6/site-
packages/mig/api/ocic_compute_client.py", line 142, in wrapper

    _raise_exception_for_http_error(e)

  File "/home/opc/oci-python-sdk/lib/python3.6/site-
packages/mig/api/ocic_compute_client.py", line 131, in
_raise_exception_for_http_error

    raise clazz(message)

mig.api.ocic_compute_client.ApiUnauthorizedError: Incorrect
username or password. Resource: authenticate.
```

This indicates an error in the user name or password specified in the `secret.yml` file. This information might have been changed inadvertently while editing the file after setting up Control-S.

-
  ```
  ERROR Migration terminated due to unexpected error.
  Traceback (most recent call last):
  File "/home/opc/oci-python-sdk/lib/python3.6/site-packages/mig/
  run_migration.py", line 97, in run_migration_as_daemon
  return run_migration_svc_internal()
  File "/home/opc/oci-python-sdk/lib/python3.6/site-packages/mig/
  run_migration.py", line 145, in run_migration_svc_internal
  raise Exception(msg)
  Exception: Cannot obtain instance with id {} in container {}.
  Double check your opc_profile_endpoint.
  ```

  This error indicates that there is an issue with the API end point specified in the `secret.yml` file. This could be caused if you updated your `secret.yml` file to set up migration from another Oracle Cloud Infrastructure Compute Classic site or region after you launched your Control-S instance. Make sure that the `secret.yml` file has the API end point for the site where your Control-S instance is created.

-
  ```
  mig.api.ocic_compute_client.ApiForbiddenError: <!DOCTYPE HTML
  PUBLIC "-//IETF//DTD HTML 2.0//EN">
  <html><head>
  <title>403 Forbidden</title>
  </head><body>
  <h1>Forbidden</h1>
  <p>You don't have permission to access /instanceIntHubAPIPre-1/
  a6122673-177c-475a-8869-e6519ab2a9a3
  on this server.</p>
  </body></html>
  ```

  This error indicates that the complete, multipart instance name hasn't been provided in the list of instances to be migrated in `secret.yml` or in the job file. The complete, multipart instance name has the format `/Compute-590693805/jack.jones@example.com/<instance_name>`

- **A migration job that was successfully submitted fails to complete with one of the following errors:**

  -
    ```
    Reached the limit of number of snapshots (5) per volume for
    <storage_volume_name>
    ```

    This error indicates that five colocated snapshots already exist for the specified storage volume, so the tool is unable to create a colocated snapshot for migration. Delete one of the existing colocated snapshots and retry the job.

  -
    ```
    Cannot initiate the import image for {}, as the max images limit is
    exceeded, please delete unused ones and retry the migration job.
    ```

    This error indicates that you have already reached the limit for the number of custom images you can have in Oracle Cloud Infrastructure. Delete some of the existing custom images or request an increase in the limit.

  -
    ```
    "Failed to execute remote cmd: sudo iscsiadm -m discovery -t
    sendtargets -p <ip_adderss:port>
          Err: b'Permission denied (publickey,gssapi-keyex,gssapi-with-
    mic).\\r\\n'
    ```

    This error indicates that the SSH key generated on Control-S to connect with Control-T doesn't match the public key on Control-T. This could happen if you have already set up Control-T once and you set up Control-T a second time and it fails. In this case, the tool tries to use the SSH key generated for the second Control-T instance with the Control-T instance that was generated first. Ensure that Control-T is set up successfully before you start a migration job.

  -
    ```
    Not enough local space for bootvolume migration, volume size...
    ```

    This error indicates that the local file system containing `/images` on Control-S isn't large enough to convert the qcow2 images. Attach larger data volumes to the Control-S instance and mount the larger volume on `/images`. Make sure the `opc` user has write access on `/images`.

– `Either the bucket named ‘ocic-oci-mig’ does not exist in the namespace <compartment_name> or you are not authorized to access it`

  This error indicates that the Oracle Cloud Infrastructure user whose credentials are provided in the secret.yml file doesn't have permission to create objects in the `ocic-oci-mig` bucket. Ensure that this user has write permission on this bucket.

17

Migrate Remote Snapshots and Scheduled Backups

If you have remote snapshots of storage volumes in your Oracle Cloud Infrastructure Compute Classic account, or if you've used backup schedules to back up storage volumes in your Oracle Cloud Infrastructure Compute Classic account, then you can restore these backups as block volumes in your Oracle Cloud Infrastructure tenancy. In Oracle Cloud Infrastructure Compute Classic, both scheduled backups as well as remote snapshots are stored in the associated Oracle Cloud Infrastructure Object Storage Classic account.

The migration of remote snapshots is important because storage volume backups stored in Oracle Cloud Infrastructure Object Storage Classic are encoded using a different format from objects stored in Oracle Cloud Infrastructure Object Storage. When you migrate *object storage*, the data stored in these snapshots is migrated over to Oracle Cloud Infrastructure. However, the encoding format used on the data isn't modified, so you can't directly access the data by restoring the snapshots to block volumes in Oracle Cloud Infrastructure. When you migrate *remote snapshots*, the encoding format is modified so that this data can be accessed by block volumes in Oracle Cloud Infrastructure.

After you restore your backups to block volumes in Oracle Cloud Infrastructure, you can attach these restored block volumes to VMs in Oracle Cloud Infrastructure, if required, or create backups of these volumes in Oracle Cloud Infrastructure Object Storage. Creating a backup of these migrated block volumes in Oracle Cloud Infrastructure allows you to complete the process of migrating your block volume backups from Oracle Cloud Infrastructure Object Storage Classic to Oracle Cloud Infrastructure Object Storage.

Note that, when you migrate storage backups, you can restore those backups only as data volumes. If you had a snapshot of a boot volume in Oracle Cloud Infrastructure Compute Classic, when you migrate that snapshot to Oracle Cloud Infrastructure, you can use that snapshot to create a data volume, but not to create a boot volume.

Workflow

Here's an overview of the high-level steps required to migrate your remote snapshots from Oracle Cloud Infrastructure Compute Classic to Oracle Cloud Infrastructure. You can use Oracle Cloud Infrastructure Classic Block Volume Backup and Restore Tool to automate this process.

1. Create an instance in Oracle Cloud Infrastructure Compute Classic using the Oracle Cloud Infrastructure Classic Migration Tools image.
2. Get the snapshot ids of the storage volume snapshots and the scheduled backups that you want to migrate.
   a. Run `opcmigrate discover` to get a list of resources in your source environment.

   b. Identify the remote snapshots that you want to migrate.

> **Note:**
>
> Alternatively, you can use the Oracle Cloud Infrastructure Compute Classic CLI to get a list of storage volume snapshots and to retrieve the snapshot ids. See Storage Volume Snapshot in *opc compute CLI Reference for Oracle Compute Cloud Service (IaaS)*.

3. Use Oracle Cloud Infrastructure Classic Block Volume Backup and Restore Tool to do the following:
   a. Set up the migration process.
   b. Restore each snapshot as a block volume in Oracle Cloud Infrastructure.
   c. Release each block volume, so that it can be attached to another VM.
4. Attach each block volume to a VM in Oracle Cloud Infrastructure, if required, and validate that you can access data on the restored volumes.
5. Create a backup of the block volumes, either manually or by using a backup policy.
6. When you are done migrating your remote snapshots and scheduled backups, use Oracle Cloud Infrastructure Classic Block Volume Backup and Restore Tool to tear down the resources created for this migration.

Complete the Prerequisites

Before you begin your migration, complete the following prerequisites.

- To use Oracle Cloud Infrastructure Classic Block Volume Backup and Restore Tool to migrate your storage volume backups, you must create an instance in your Oracle Cloud Infrastructure Compute Classic account using the Oracle Cloud Infrastructure Classic Migration Tools image. Ensure that you have sufficient quota for this instance to be created. For information about creating your Control-S instance, see Complete the Prerequisites and Launch the Migration Controller Instance (Control-S) in the Source Environment. If you've already created this instance earlier in your migration process, you can use the same instance for this procedure. You don't need to create it again.
- The Oracle Cloud Infrastructure Classic Block Volume Backup and Restore Tool creates a VM in an isolated compartment in your Oracle Cloud Infrastructure tenancy to manage the migration. Ensure that you have sufficient quota for this VM to be launched.
- The minimum size of a storage volume in Oracle Cloud Infrastructure is 50 GB. Ensure that you have sufficient storage capacity for the number of storage snapshots that you intend to migrate.
- Ensure that you have a virtual cloud network (VCN) with subnets set up in a compartment that you have access to.

Plan for the Migration

To use Oracle Cloud Infrastructure Classic Block Volume Backup and Restore Tool to migrate your storage volume backups, you must configure the Control-S instance in your Oracle Cloud Infrastructure Compute Classic account. This instance must have already been created using the Oracle Cloud Infrastructure Classic Migration Tools image.

You should also get the snapshot ids of the remote snapshots and scheduled backups that you want to migrate. Do the following:

1. Create or update the `/home/opc/.opc/profiles/default` file on Control-S. The `opcmigrate migrate rsm` commands used to migrate storage volume backups require access to the Oracle Cloud Infrastructure Object Storage Classic account. If you've used other `opcmigrate` commands earlier in your migration process, this section might already exist in your profile file. If not, add this section now, along with valid credentials.

   > **Note:**
   >
   > If you run the Control-S setup command for Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool, it generates the default profile using information provided by you in the `secret.yml` file. The Control-S setup command overwrites any existing default profile. Verify and update the existing profile, or create a new profile, if required.

   a. You'll need the user name and API end point for the Oracle Cloud Infrastructure Object Storage Classicservice. Look this up in your Oracle Cloud Dashboard.

   b. Create or locate your profile file. This file contains the credentials and other information required to access your source environment. The default location for this file is `~/.opc/profiles/default`. If required, you can create multiple profiles and use the `--profile` option to specify the profile you want to use when you run the tool. If you create profiles in a location other than `~/.opc/profiles`, provide the full path to the profile location by using the `--profile-directory` option, along with the `--profile` option to specify the profile file name. If no profile is specified, the `~/.opc/profiles/default` profile is used.

   c. Use the following template to add the "`object-storage`" section to your profile file. Replace the sample values with values specific to your account.

   ```
    "object_storage": {
       "auth-endpoint": "uscom-central-1.storage.oraclecloud.com/auth/
   v1.0",
       "user": "Storage-example:user@example.com",
       "endpoint": "uscom-central-1.storage.oraclecloud.com/v1/Storage-
   example"
     }
   ```

   Passwords aren't specified in the profile file for security reasons. You'll be prompted to provide the password for each service when you run the tool.

2. Get details of your target environment. Log in to the Oracle Cloud Infrastructure Console and collect the required information. You'll need the following:
   - The user OCID. From the menu, choose **Identity** and then choose **Users.**
   - The API PEM key fingerprint. Click the user to view user details. The API Keys section displays the PEM key fingerprint.
   - The compartment OCID. From the menu, choose **Identity** and then choose **Compartments.** The VM created by the tool to manage the migration of storage volume backups is created in a compartment under this compartment.
   - The tenancy OCID and the tenancy region. From the menu, choose **Administration** and then choose **Tenancy Details.**
   - The Availability Domain. From the menu, choose **Compute** and then choose **Instances.** The Instances page displays the availability domain. If you want to attach the restored volumes to existing VMs, ensure that the availability domain that you select has the VMs that you want to use. You can't attach block volumes to VMs in a different availability domain.
3. Create or edit the `/home/opc/.oci/config` file on Control-S. This file contains the credentials and other information required to access your target environment. If you've used Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool earlier in the migration process, this would already have been created. However, the file might not contain all of the information required by the `opcmigrate migrate rsm` commands. For example, you might have to add information about the availability domain.
   If you run the Control-S setup command for Oracle Cloud Infrastructure Classic VM and Block Storage Migration Tool, it generates the `.oci/config` file, using information provided by you in the `secret.yml` file. The Control-S setup command overwrites any existing `.oci/config file`.

   Use the following template to create or update the `.oci/config` file. Replace the sample values with values specific to your tenancy.

   ```
   [DEFAULT]
   user=ocid1.user.oc1..aaaaa...
   fingerprint=81:45:aa:2...
   key_file=<path to api pem key>
   pass_phrase=
   tenancy=ocid1.tenancy.oc1..aaaaaaa...
   region=us-ashburn-1
   ```

   The compartment OCID and availability domain aren't specified in this file. However, make a note of those values. You'll need them later in this migration procedure.
4. Generate a list of resources in your source environment. If you've already done this earlier in your migration process, you don't need to do this again. However, if your environment has changed since the last time you ran this command, then run the command again to get the latest set of resources. On the Control-S instance, run:

   ```
   opcmigrate discover
   ```

You can specify a profile or a profile directory with this command. The output of this command is a file named `resources-*.json`, where `*` represents the profile used with this command. This file created in the directory where you run the command.

5. You can use other `opcmigrate` commands to filter the output of `opcmigrate discover` or to view the output in various formats. Use the `--help` option for help with `opcmigrate` commands and options. Select a method that works best for you, to identify the remote snapshots and scheduled backups in your source environment. For example, you can search for objects with the following property:

```
"property": "/oracle/public/storage/snapshot/default",
```

To find out whether a remote snapshot was generated manually or by a scheduled backup, look for the following tags. If there are no tags, no backup schedule is associated with the snapshot.

```
"tags": [
        "backupPrivateId=a0f1....",
        "backupConfigurationPrivateId=4a...."
      ],
```

For each of the snapshots in the resources file that you want to migrate, make a note of the snapshot id. For example:

```
"snapshot_id": "4465........751e-uscom",
```

Migrate the Specified Storage Volume Backups

Use Oracle Cloud Infrastructure Classic Block Volume Backup and Restore Tool to restore remote snapshots and scheduled backups from Oracle Cloud Infrastructure Compute Classicto block volumes in Oracle Cloud Infrastructure.

You can run this tool on an instance in Oracle Cloud Infrastructure Compute Classic that was created using the Oracle Cloud Infrastructure Classic Migration Tools image.

> **Note:**
>
> If you download and install Oracle Cloud Infrastructure Classic Discovery and Translation Tool on your local system, you won't be able to use Oracle Cloud Infrastructure Classic Block Volume Backup and Restore Tool because this tool isn't included in that distribution of the migration tools. To use Oracle Cloud Infrastructure Classic Block Volume Backup and Restore Tool, you must create an instance using the Oracle Cloud Infrastructure Classic Migration Tools image.

1. To set up the migration controller instance in the target environment, run:

```
opcmigrate migrate rsm setup <compartment_id> <availability_domain>
```

This command launches a VM in your Oracle Cloud Infrastructure tenancy, in the specified availability domain. This VM is launched in an isolated subcompartment named `rsm`, which is created in the specified compartment.

If you want to migrate backups to different availability domains, you can set up migration controller VMs in other availability domains by running this command again with a different availability domain specified. That way, you can restore backups to different availability domains concurrently.

For a given availability domain, it is recommended that you run the setup command only once.

Note that, if you want to attach the restored volumes to existing VMs, those VMs must be in the same availability domain that you specify for restoring volumes. You can't attach block volumes to VMs in a different availability domain.

2. The `opcmigrate migrate rsm setup` command returns a stack OCID. Make a note of this value. You'll need this later, when you want to tear down this migration environment.

3. To restore a storage volume from a snapshot to a block volume in Oracle Cloud Infrastructure, run:

   ```
   opcmigrate migrate rsm restore <snapshot_id> <name> <compartment_id>
   <availability_domain>
   ```

   Here, `<snapshot_id>` is the value of `snapshot _id` that you obtained from the resources file. You must provide the snapshot ID for each backup that you want to migrate. You can provide only one snapshot ID at a time. To migrate multiple backups, run this command multiple times.

   `<name>` is the name of the block volume that will be created in your Oracle Cloud Infrastructure tenancy.

   You should have already run the setup command for the same availability domain that you specify here.

   This command creates a target volume with the specified name. The size of the target volume is the same as the size of the backup in the source environment. However, note that the minimum size of a block volume in Oracle Cloud Infrastructure is 50 GB, so even when you migrate smaller backups, the target volume is created with a size of 50 GB.

   The target volume is attached to the migration controller instance in the specified availability domain in Oracle Cloud Infrastructure.

   It is recommended that you run a single restore job at a time in a specified availability domain. Wait for each job to complete before you launch another job in the same availability domain.

4. The `opcmigrate migrate rsm restore` command returns a job ID and a volume OCID. Make a note of both values. You'll need the job ID later to monitor the status of the job as well as to release the volume. You can use the volume OCID to attach a restored volume to a VM later on, if required.

5. To view the status of a migration job, run:

   ```
   opcmigrate migrate rsm status <job_id>
   ```

Here, `job_id` is the value returned by the `opcmigrate migrate rsm restore` command.

6. When the restore step completes, you must release the block volume to detach it from the migration controller instance. You can't attach a block volume to another VM until it has been released. To release a block volume, run:

   ```
   opcmigrate migrate rsm release <job_id>
   ```

7. For each remote snapshot or scheduled backup that you want to migrate, run the commands `opcmigrate migrate rsm restore` and `opcmigrate migrate rsm release` until all backups have been successfully migrated.
8. Validate the migrated volumes. After each restored volume has been released, attach it to a VM and use the iSCSI commands to mount it on the VM. Verify the data on the volume. If required, create a backup of each volume in your Oracle Cloud Infrastructure tenancy.
9. When all backups have been successfully migrated and verified, tear down the migration environment. This destroys resources created for the migration including the migration controller VM and releases any migrated volumes created by the specified stack id that haven't yet been released. To tear down the environment, run:

   ```
   opcmigrate migrate rsm destroy <stack_id>
   ```

   Here, `stack_id` is the value of the stack OCID returned by the `opcmigrate migrate rsm setup` command.
10. The subcompartment `rsm` that is created for this migration isn't deleted by the `opcmigrate migrate rsm destroy` command. You can delete this subcompartment by using the Oracle Cloud Infrastructure Console, if required.

Troubleshooting

Here are a few tips for dealing with errors that might occur while migrating remote snapshots and scheduled backups using Oracle Cloud Infrastructure Classic Block Volume Backup and Restore Tool.

- **The `opcmigrate migrate rsm setup` command errors out.**
  Errors in the setup could be due to resource limits in your Oracle Cloud Infrastructure tenancy. Ensure that you have sufficient quota in the specified availability domain to create the migration controller VM and the block volumes that you want to migrate. If the setup command returns an Apply Job OCID, use that value to search in the Oracle Cloud Infrastructure Console for the components that caused the error.
- **The setup command returns the following error: `Could not find config file at <home_dir>/.oci/config`**
  Ensure that you've created the `.oci/config` file in the appropriate path, or use the `-c` option to specify a different location for this file.
- **The `opcmigrate migrate rsm restore` command fails with the following error: `Could not retrieve snapshot metadata`**
  Ensure that the backup corresponding to the specified snapshot ID exists and that it is a remote snapshot or a scheduled backup. This error could indicate that the

specified snapshot ID belongs to a colocated snapshot. This tool can't be used to migrate colocated snapshots.

- **The restore command fails to connect to the migration controller VM in Oracle Cloud Infrastructure. Error:** `No valid service VM found`
  Ensure that you've run the setup command to launch the migration controller VM and that the migration environment hasn't subsequently been destroyed. Also check that the migration controller VM is in the same availability domain as the availability domain that you specified for the migration job.
- **The restored volume can't be attached to a VM in Oracle Cloud Infrastructure.**
  Ensure that you've run `opcmigrate migrate rsm release` to release the volume from the migration controller VM. Only after releasing a volume can you attach it to another VM.
- **The `rsm` subcompartment still exists after cleaning up the target environment using `opcmigrate migrate rsm destroy`.**
  The `rsm` compartment that is created by `opcmigrate migrate rsm setup` isn't destroyed by `opcmigrate migrate rsm destroy`. Verify that all the other resources have been removed by the destroy command, and the delete the `rsm` compartment manually by using the Oracle Cloud Infrastructure Console.

18
Set Up Load Balancers in Oracle Cloud Infrastructure

If you are currently using Oracle Cloud Infrastructure Load Balancing Classic, you create a similar load balancer in your Oracle Cloud Infrastructure environment.

Load Balancing in Oracle Cloud Infrastructure Load Balancing Classic

Load balancing in an Oracle Cloud Infrastructure Compute Classic network is handled by the Oracle Cloud Infrastructure Load Balancing Classic service. Instances of Oracle Cloud Infrastructure Load Balancing Classic can be created from the Oracle Cloud Infrastructure Compute Classic console and used to balance requests against a pool of Oracle Cloud Infrastructure Compute Classic VMs.

For more information, see About Oracle Cloud Infrastructure Load Balancing Classic.

Oracle Cloud Infrastructure Load Balancers

Oracle Cloud Infrastructure also offers a load balancing solution. The Oracle Cloud Infrastructure Load Balancing service provides automated traffic distribution from one entry point to multiple servers reachable from your virtual cloud network (VCN). The service offers a load balancer with your choice of a public or private IP address, and provisioned bandwidth.

For more information, see Overview of Load Balancing.

Features in Oracle Cloud Infrastructure Load Balancing Classic and Oracle Cloud Infrastructure Load Balancing

The following table compares load balancer features in Oracle Cloud Infrastructure Load Balancing Classic versus the Oracle Cloud Infrastructure Load Balancing service.

Feature	Oracle Cloud Infrastructure Load Balancing Classic	Oracle Cloud Infrastructure Load Balancing
Types of load balancers	• Internet-facing or internal load balancer in a given IP network. • When you create a load balancer in Oracle Cloud Infrastructure Load Balancing Classic, it allows you to select a scheme for the load balancer: – **Internet-facing** - This scheme enables you to add a load balancer to your own IP Network, while assigning a internet addressable IP address to the load balancer. – **Internal** - This scheme enables you to add a load balancer to your own IP network for the sole consumption of other clients inside the same network. See Creating a Load Balancer.	• Public or private load balancer in a virtual cloud network (VCN). • **Public** - A public load balancer has a public IP address that is accessible from the internet. To accept traffic from the internet, you create a public load balancer. • **Private** - A private load balancer has an IP address from the hosting subnet, which is visible only within your VCN. To isolate your load balancer from the internet and simplify your security posture, you can create a private load balancer. • Oracle Cloud Infrastructure load balancer is based on pre-provisioned bandwidth shape (100Mb, 400Mb and 8G). See How Load Balancing Works.
Origin servers/Backend servers	**Origin servers** A server or host computer to which the load balancer routes requests. In the context of the Oracle Cloud Infrastructure Load Balancing Classic, an origin server is an Oracle Cloud Infrastructure Compute Classic service instance.	**Backend servers** • When you implement a load balancer, you must specify the backend servers (Compute instances) to include in each backend set. The load balancer routes incoming traffic to these backend servers based on the policies you specified for the backend set. • The backend servers (Compute instances) associated with a backend set can exist anywhere, as long as the associated security lists and route tables allow the intended traffic flow.

Feature	Oracle Cloud Infrastructure Load Balancing Classic	Oracle Cloud Infrastructure Load Balancing
Backend Set/Server Pool	**Server pool** When you create a load balancer with Oracle Cloud Infrastructure Load Balancing Classic, you must define one or more servers (referred to as origin servers) to which the load balancer can distribute requests. A set of origin servers is called a server pool. In Oracle Cloud Infrastructure Load Balancing Classic, there is no concept of a backend set (which is defined by a list of backend servers, a load balancing policy, and a health check policy). However, Oracle Cloud Infrastructure Compute Classic has server pools (a set of origin servers). In Oracle Cloud Infrastructure Compute Classic, after the creation of a load balancer, users must finish the configuration of the load balancer by adding a server pool (where you can also define health checks for the origin servers), a listener, and optional policies (such as Load Balancing Mechanism Policy).	**Backend Set** A logical entity defined by a list of backend servers, a load balancing policy, and a health check policy. SSL configuration is optional. The backend set determines how the load balancer directs traffic to the collection of backend servers.
Certificates	You must obtain a digital certificate if you want to use a secure connection between the load balancer and the clients sending the request or between the load balancer and the origin servers in the server pool. Oracle Cloud Infrastructure Load Balancing Classic supports two types of digital certificates: • Server certificates • Trusted certificates	If you use HTTPS or SSL for your listener, you must associate an SSL server certificate (X.509) with your load balancer. A certificate enables the load balancer to terminate the connection and decrypt incoming requests before passing them to the backend servers.

Feature	Oracle Cloud Infrastructure Load Balancing Classic	Oracle Cloud Infrastructure Load Balancing
Health Check	The load balancer can perform regular health checks of the origin servers and route inbound traffic to the healthy origin servers. This feature is not enabled automatically when an origin server pool is created and must be enabled explicitly either during the origin server pool creation or update. Types of health check supported: • TCP • SSL • HTTP	A test to confirm the availability of backend servers. A health check can be a request or a connection attempt. Based on a time interval you specify, the load balancer applies the health check policy to continuously monitor backend servers. If a server fails the health check, the load balancer takes the server temporarily out of rotation. If the server subsequently passes the health check, the load balancer returns it to the rotation. You configure your health check policy when you create a backend set. Types of health check supported: • TCP-level • HTTP-level
Health Status	N/A	The Load Balancing service provides health status indicators that use your health check policies to report on the general health of your load balancers and their components. You can see health status indicators for load balancers, backend sets, and backend servers.

Feature	Oracle Cloud Infrastructure Load Balancing Classic	Oracle Cloud Infrastructure Load Balancing
Listeners	Before you use a load balancer, you must define at least one listener. A listener defines the virtual host, port, and protocol that the load balancer will use to listen for new requests. Supported protocols include: • **Balancer Protocol** - The transport protocol that will be accepted for all incoming requests to the selected load balancer listener. – **HTTP** - Use this protocol to listen for non-secure HTTP requests. – **HTTPS** - Use this protocol to listen only for secure HTTP requests sent over SSL or TLS. • **Server Protocol** - The protocol to be used for routing traffic to the origin servers in the server pool. – **HTTP** - Use this protocol to route HTTP or HTTPS requests to the origin servers using the non-secure HTTP protocol. – **HTTPS** - Use this protocol to route HTTP or HTTPS requests to the origin servers using the secure HTTPS protocol.	A logical entity that checks for incoming traffic on the load balancer's IP address. You configure a listener's protocol and port number, and the optional SSL settings. To handle TCP, HTTP, and HTTPS traffic, you must configure multiple listeners. Supported protocols include: • TCP • HTTP/1.0 • HTTP/1.1 You can have one SSL certificate bundle per listener. You can configure two listeners, one each for ports 443 and 8443, and associate SSL certificate bundles with each listener.

Feature	Oracle Cloud Infrastructure Load Balancing Classic	Oracle Cloud Infrastructure Load Balancing
Load Balancer Policies	Oracle Cloud Infrastructure Load Balancing Classic provides advanced features that you can configure by attaching specific policies to the load balancer. Supported policies include: • Application Cookie Stickiness Policy • CloudGate Policy • Load Balancer Cookie Stickiness Policy • Load Balancing Mechanism Policy This policy enables you to specify a load balancing mechanism for distributing client requests across multiple origin servers by using one of the following methods: – Round Robin – IP Hash – Least Connections • Rate Limiting Request Policy • Redirect Policy • Resource Access Control Policy • Set Request Header Policy • SSL Negotiation Policy • Trusted Certificate Policy See About Load Balancer Policies and Creating Policies for a Load Balancer.	• A load balancing policy tells the load balancer how to distribute incoming traffic to the backend servers. Supported policies include: – Round robin – Least connections – IP hash See How Load Balancing Policies Work. • Oracle Cloud Infrastructure load balancer supports HTTP cookie based Session Persistence. See Session Persistence. • Oracle Cloud Infrastructure load balancer supports SSL termination and SSL tunneling policies.

Create a Load Balancer in Oracle Cloud Infrastructure

This topic provides information on creating a load balancer in Oracle Cloud Infrastructure.

Topics:

- Considerations When Creating a Load Balancer as Part of Your Migration Project
- Prerequisites
- Creating the Load Balancer

Considerations When Creating a Load Balancer as Part of Your Migration Project

Note that certain load balancer features which were available in Oracle Cloud Infrastructure Load Balancing Classic may not be available in Oracle Cloud Infrastructure Load Balancing service. See Features in Oracle Cloud Infrastructure Load Balancing Classic and Oracle Cloud Infrastructure Load Balancing.

Prerequisites

This procedure assumes:

- you have already created a virtual cloud network (VCN) using the procedures described in Create a Virtual Cloud Network in Oracle Cloud Infrastructure.
- you have already created the required Oracle Cloud Infrastructure virtual machines and you have migrated your workloads to the new VMs.
- you have access to the URLs and endpoints where the load balancers will direct incoming network traffic.

Creating the Load Balancer

The following table describes the tasks involved in creating a load balancer in Oracle Cloud Infrastructure.

Task	Description	More Information
Add two subnets to your VCN to host your load balancer.	Your load balancer must reside in different subnets from your application instances. This configuration allows you to keep your application instances secured in subnets with stricter access rules, while allowing public internet traffic to the load balancer in the public subnets.	Add two subnets to your VCN to host your load balancer.
Create a load balancer.	When you create a public load balancer, you choose its shape (size) and you select two subnets, each in a different availability domain. This configuration ensures that the load balancer is highly available. It is active in only one subnet at a time. This load balancer comes with a public IP address and provisioned bandwidth corresponding to the shape you chose.	Create a load balancer.

Task	Description	More Information
Create a backend set with health check.	A backend set is a collection of backend servers to which your load balancer directs traffic. A list of backend servers, a load balancing policy, and a health check script define each backend set.	Create a backend set with health check.
Add backend servers to your backend set.	After the backend set is created, you can add compute instances (backend servers) to it. To add a backend server, you can enter the OCID for each instance and your application port. The OCID enables the Console to create the security list rules required to enable traffic between the load balancer subnets and the instance subnets. Tip	Add backend servers to your backend set.
Create a listener.	A listener is an entity that checks for connection requests. The load balancer listener listens for ingress client traffic using the port you specify within the listener and the load balancer's public IP.	Create a listener.
Update the load balancer subnet security list and allow internet traffic to the listener.	When you create a listener, you must also update your VCN's security list to allow traffic to that listener. The subnets where the load balancer resides must allow the listener to accept traffic. To enable the traffic to get to the listener, update the load balancer subnet's security list.	Update the load balancer subnet security list and allow internet traffic to the listener.
Verify your load balancer.	To test your load balancer's functionality, you can open a web browser and navigate to its public IP address (listed on the load balancer's detail page). If the load balancer is properly configured, you can see the name of one of the web server instances.	Verify your load balancer.
Update rules to protect your backend servers.	Update the default security list and the default route table to limit traffic to your backend servers.	Update rules to protect your backend servers.

Task	Description	More Information
Terminate your load balancer.	When your load balancer becomes available, you are billed for each hour that you keep it running. Once you no longer need a load balancer, you can delete it. When the load balancer is deleted, you stop incurring charges for it. Deleting a load balancer does not affect the backend servers or subnets used by the load balancer.	Terminate your load balancer.

19 Migrate Object Storage

Currently, migration of Oracle Cloud Infrastructure Object Storage container data is available only for the following scenarios.

- If you are using the Oracle Cloud Infrastructure Storage Software Appliance, you can use Cloud Sync to migrate data to Oracle Cloud Infrastructure Storage Gateway.
- If you are not using Oracle Cloud Infrastructure Storage Software Appliance and if the total amount of object storage data does not exceed the guidelines for this type of data migration, then you can use Rclone to migrate data from Oracle Cloud Infrastructure Object Storage Classic to Oracle Cloud Infrastructure Object Storage. For more information on how to determine if your environment is a candidate for this migration strategy, contact your Oracle support or consulting representative.

Migrate Storage Appliance Data to Oracle Cloud Infrastructure Storage Gateway

Oracle Cloud Infrastructure Storage Software Appliance provides a POSIX compliant file system that persists the file data to Oracle Cloud Infrastructure Object Storage Classic.
Oracle Cloud Infrastructure Storage Gateway is the recommended gateway for Oracle Cloud Infrastructure Object Storage. Oracle recommends that you migrate from Oracle Cloud Infrastructure Storage Software Appliance backed by containers in Oracle Cloud Infrastructure Object Storage Classic to Oracle Cloud Infrastructure Storage Gateway that persists data in object storage buckets in Oracle Cloud Infrastructure.

Migration steps can take a considerable amount of time depending upon the number of files/directories and size of files, available resources (CPU, memory, network bandwidth, disk throughput, etc.). While scheduling the migration process, consider the estimated time required to complete the migration. If you abort the migration process for any reason, you can restart the process.

To migrate data from an Archive container, you must first restore the archived objects. For more information, see Restoring Archived Objects in *Using Oracle Cloud Infrastructure Object Storage Classic*.

Before You Begin

Before you begin migrating your workloads, verify that you have a cloud account with Identity Cloud Service to access Oracle Cloud Infrastructure, and then complete the planning checklist. This section provides information to help you complete the planning checklist.

Identify the following information for each storage software appliance that you want to migrate.

Information Required	Details
Is the appliance is running in Compute Classic or in your on-premises environment?	
Version of the storage software appliance running in Compute Classic or in your on-premises environment	
Link to access the management console and password	
Size of the cache storage	
Total number of file systems	

Identify the following information for every file system in the storage software appliance that you want to migrate.

Information Required	Details
Number of files in the file system	
Total storage space used	
Is the file system is encrypted? If yes, note down the location where you have downloaded the encryption keys.	

Get Details of the Source Environment

Identify the user name, password, authentication URL, and REST Endpoint URL for the Oracle Cloud Infrastructure Object Storage Classic account from which you want to migrate data.

1. Note down the user name and password for your Compute Classic instance, which is case-sensitive. The account creation email from Oracle contains this information. If you don't have this information, contact your service administrator.
2. Sign in to your Cloud Account and navigate to the My Services Dashboard.

   The My Services dashboard is displayed. It lists the services that are assigned to your account.
3. Look for **Oracle Cloud Infrastructure Object Storage Classic**.
4. Select **View Details** from the **Actions** menu.

   On the resulting page, the details of your Oracle Cloud Infrastructure Object Storage Classic instance are displayed.
5. From the **Additional Information** section, note down the following information:
   - REST Endpoint URL from the **REST Endpoint** field. For example, `https://acme.storage.oraclecloud.com/v1/Storage-acme`.
   - The authentication URL from the **Auth V1 Endpoint** field. For example, `https://acme.storage.oraclecloud.com/auth/v1.0`.
6. Also identify the region where the Storage Classic containers are located. This is the region where your Compute Classic account has been provisioned.

Get Details of the Appliance Instance

Before gathering details of the instance on which Oracle Cloud Infrastructure Storage Software Appliance is installed, identify if the appliance is installed on a Compute Classic instance or in your on-premise environment. If you have set up multiple appliance instances, identify the following information for every appliance instance that you want to migrate: the public IP address of the instance that hosts the application, link to access the management console and the password. Use the public IP address to connect to the appliance instance using SSH and to log in to the management console of the appliance.

Get Details of an Appliance Installed on a Compute Classic Instance

If the appliance is installed on a Compute Classic host, then to gather the required information perform the following steps:

1. On the Compute Classic host go to the directory where you have downloaded the appliance provisioning tool and run the following command:

   ```
   ./fscs.sh -i config_file
   ```

   In this command, `config_file` is the full path and name of the appliance configuration file.

2. The tool prompts you to enter the password for the Compute Classic user specified in the configuration file. Enter the password.

   The tool validates the password and displays the IP addresses of the instance, as shown in the following example:

   ```
   Extracting IP addresses ...
   Private IP address: 10.196.35.245
   Public IP address: 203.0.113.48
   ```

3. The password for the `admin` user to access the management console of the appliance.

   If you don’t remember the password, you can reset it. See Changing the Admin Password for the Appliance in *Using Oracle Cloud Infrastructure Storage Software Appliance*.

4. Access the management console. In your web browser, go to `https://`*`public_IP_address_of_appliance_instance`*, and then enter the admin password.

   The available filesystems are displayed. Note down the number of the filesystems that you want to migrate.

Get Details of an Appliance Installed in the On-Premises Environment

If the appliance is installed on a Compute Classic host, use the `oscsa` command-line tool to manage Oracle Cloud Infrastructure Storage Software Appliance. `oscsa` commands might fail, if you don't run these commands as a root user. To gather the required information, perform the following steps:

1. If the appliance instance is in your on-premises environment, log in to the appliance host and then run the following commands:

   ```
   oscsa info
   ```

   Note down the link to access the management console from the output. Example:

   ```
   Management Console: https://myApplianceHost.example.com:32775
   If you have already configured an OSCSA FileSystem via the Management
   Console,you can access the NFS share using the following port.
   NFS Port: 32774
   ```

   Note down the link to access the management console and the NFS port from the output.

   If the appliance instance is not running, then run the following command to bring up the appliance and get the required information.

   ```
   oscsa up
   ```

2. Run the following command to identify the version of the storage appliance.

   ```
   oscsa version
   ```

   Note down the version. While migrating the on-premise storage appliance, you may install a storage appliance on a Compute Classic instance. The version of the storage appliance that you install on the Compute Classic instance must be same as the on-premise storage appliance that you want to migrate.

3. Access the management console. In your web browser, go to `https://myApplianceHost.example.com:32775`, and then enter the admin password.

   The available file systems are displayed. Note down the name and number of the file systems that you want to migrate.

Get Details of a File System

Each Oracle Cloud Infrastructure Storage Software Appliance instance can have multiple file systems. Gather the following details for every file system that you want to migrate.

To view the details of a file system, log in to the management console, and click the name of the file system:

1. The **Settings** tab displays the enabled file system properties, such as encryption, `Archive` storage class and deleting old file versions.
2. If the file system is encrypted, identify encryption key to be used for decrypting the data.
   a. Select the file system for which you want to download encryption keys, and go to the **Encryption Keys** tab.
   b. Click **Download** in the **Existing Encryption Keys** section.

   c. Save the `.tar.gz` file (that contains the encryption keys) in a location of your choice. Ensure that the file is stored at a secure location. You can delete this file when it is no longer required.

3. View all the contents in the file system to ensure that the file system contains only files and directories. Special devices, symlink, or hard link cannot be migrated. Run the following command in the NFS client on which you have mounted the appliance file system.

   ```
   find source_directory -print
   ```

4. From the details displayed, check if any file name has characters such as '\r' and '\n that are incompatible with Oracle Cloud Infrastructure Storage Gateway. If any file name has characters which are incompatible with storage gateway, you must rename the files before starting the migration process.
5. If the file system contains any special devices, symlink, or hard link, complete the following tasks:
   a. Create an archive file using `tar`, `cpio`, or `zip` to ensure that you can restore the special files when needed.
   b. Only after the archive file is created, delete all special devices, symlinks or hard links in the source environment before starting the migration.
6. Run the following command using the Oracle Cloud Infrastructure Object Storage Classic File Transfer Manager command-line interface (FTM CLI) to retrieve metadata of the Oracle Cloud Infrastructure Object Storage Classic container that is associated with the file system:

   ```
   java -jar ftmcli.jar describe container_name
   ```

   From the details displayed, note down the number of files in the container and the total storage space used by all the objects in the container.

For information about using FTM CLI, see Preparing to Use the FTM CLI in *Command-Line Reference for Oracle Cloud Infrastructure Object Storage Classic*.

Prepare the Target Environment

Before migrating the file systems, set up the target environment. To prepare the target environment, install and configure an instance of the Oracle Cloud Infrastructure Storage Gateway. You will use this instance after you complete the migration process. Test the target environment and resolve issues, if any. You have to test the environment only once before migrating the first file system. Mount a test file system on the storage gateway instance, and then check if you can successfully read and write to object storage.

1. Install and configure Oracle Cloud Infrastructure Storage Gateway in Oracle Cloud Infrastructure. See Installing Storage Gateway in *Oracle Cloud Infrastructure documentation*.

   If the source appliance instance runs in the cloud, create the storage gateway instance in the cloud. If the source appliance instance runs on-premises, create the storage gateway instance in your on-premises environment.
2. Create a storage gateway file system to test the connection. See Creating Your First File System in *Oracle Cloud Infrastructure documentation*.

You can also use the source file system that you are migrating from the Oracle Cloud Infrastructure Storage Software Appliance instance.

3. Connect the file system to an Oracle Cloud Infrastructure Object Storage bucket. See Connecting a File System in Oracle Cloud Infrastructure documentation.
4. Mount the test file system on a Linux NFS client. See Mounting File Systems on Clients in Oracle Cloud Infrastructure documentation.
5. Use the Oracle Cloud Infrastructure Console to verify that you can read and write files to this file system. Create files in the file system and then read the content of the uploaded files to check that the files were uploaded successfully to the object storage.
6. Unmount the test file system from NFS client. SSH to the client instance, and then run the following command:

   ```
   umount /mnt/myFileSystem1
   ```

   Where, `myFileSystem1` is the name of the filesystem that you want to unmount.
7. Disconnect the test file system. See Disconnecting a File System in Oracle Cloud Infrastructure documentation.
8. Delete the test file system. See Deleting a File System in Oracle Cloud Infrastructure documentation.

Additional Steps to Prepare the On-Premises Environments

If you are migrating object storage data from a storage appliance in the cloud to a storage gateway in the cloud, you don't need to perform any additional steps to prepare the environments. However, if you are migrating object storage data from a storage appliance in your on-premises environment to a storage gateway in your on-premises environment, consider performing the following additional steps to prepare the environment.

Even if the appliance instance is available on-premises, you don't need to perform these additional steps if you are connecting to Oracle Cloud through a low latency, high throughput connectivity, such as Oracle Cloud Infrastructure FastConnect. In such a scenario, you can use the appliance instance in your on-premises environment to perform the migration.

Prepare Source Appliance for Migration

1. Install and configure an Oracle Cloud Infrastructure Storage Software Appliance instance on a Compute Classic instance.
2. Create a file system with the same name and properties (such as storage tier, encryption, and other options) as the file system being migrated. Set appropriate NFS options, which is available in advanced options, to ensure security.
3. Do not connect the file system. Before connecting the file system you must ensure that all pending uploads to the appliance are complete. You'll also need to disconnect the existing appliance instance. You'll connect the file system later in the migration process.

Prepare Target Gateway for Migration

1. Install and configure Oracle Cloud Infrastructure Storage Gateway on a compute instance in Oracle Cloud Infrastructure.
   You 'll use this Oracle Cloud Infrastructure Storage Gateway instance during the migration process. Create this instance only if you want to use an Oracle Cloud Infrastructure instance to complete the migration instead of using an instance in your on-premises environment.
2. Create a file system with the same name and properties as the file system being migrated. Use standard tier bucket along with object life cycle policy and not the archive tier bucket.
3. Set appropriate NFS options, which is available in advanced options, to ensure security.
4. Connect the Oracle Cloud Infrastructure Storage Gateway file system.

Start Migrating a File System

The following steps are required for migrating a single file system. Repeat these steps for each file system that you want to migrate. Depending upon the availability of resources, you may migrate multiple file systems in parallel.

1. Quiesce the source file system that you want to migrate. Complete the following tasks in the source storage appliance.
   a. Access the management console of the storage appliance.
   b. Click the name of the file system to view its details.
   c. The **Activity** tab shows the ongoing and pending upload activity. If there is any ongoing activity, wait for the activity to complete.
   d. Unmount the filesystem from the client instances. SSH to the client instance, and then run the following command:

      ```
      umount /mnt/myFileSystem1
      ```

      Where, `myFileSystem1` is the name of the filesystem that you want to unmount.
   e. Wait for pending uploads, if any, to complete.
   f. On the **Dashboard** tab of the management console, select the filesystem that you want to disconnect, and then click **Disconnect**.
2. If you are using the source storage appliance instance to perform the migration, then perform the following tasks in the source storage appliance. If you have configured a Oracle Cloud Infrastructure Storage Gateway on a compute instance in Oracle Cloud Infrastructure to perform the migration, then perform the following steps in the instance that you have created.
   a. In the **Settings** tab, set the **NFS Export Options** to read-only (`ro`) for the file system, and then click **Save**.
   b. For the changes to take effect, reconnect the file system. On the **Dashboard** tab of the management console, select the filesystem that you want to connect, and then click **Connect**.

   c. If the **FileSystem: Claim Ownership** window is displayed, re-enter your Oracle Cloud Infrastructure Object Storage Classic password and then select **Claim Ownership**.

3. Set up the destination file system for the copy operation.

   a. On the host running the Oracle Cloud Infrastructure Storage Gateway, mount the source file system that is available on the appliance instance. To transfer the files using Cloud Sync, you must mount the file system under `/cloudsync/mounts`.

   ```
   # mount -t nfs <mount_options>
   <storage_appliance_server>:<filesystem> <mountpoint>
   ```

   b. Connect the new file system in the Oracle Cloud Infrastructure Storage Gateway instance. See Connecting a File System in Oracle Cloud Infrastructure documentation.

4. Create a Cloud Sync job that syncs all files and directories between the source and the destination file system. See Using Storage Gateway Cloud Sync in *Oracle Cloud Infrastructure documentation*.

   Use Storage Gateway management console to create, manage, and monitor Cloud Sync jobs.

5. Run the Cloud Sync job that you have created. See Using Storage Gateway Cloud Sync in *Oracle Cloud Infrastructure documentation*.

6. Get the status of the Cloud Sync job to monitor status of the process. See Using Storage Gateway Cloud Sync in *Oracle Cloud Infrastructure documentation*.

   The management console displays the status of the job (Created, Running, Completed, Failed, or Canceled) just below the job name. View the output of this command to determine if any files have changed on the storage appliance after you copied files to the storage gateway. If any files have been modified, rerun the job to copy only the modified files.

Validate

After object storage data migration is complete, perform the following steps to determine if the migration was completed successfully.

1. After the migration is complete, check if any errors occurred during the migration process. Cloud Sync records error messages in a file. You can download this file from the management console.
2. Compare the file inventory in the source and target environments. You can list the files in the source and target environment and compare them.

When you get the status of a Cloud Sync job, it also reports any mismatch in the files on the source and target. If there is a discrepancy between files in the source and target, try restarting the cloud sync job first. If that doesn't work, restart the migration process to reconcile the differences. If that does not resolve the issue, contact My Oracle Support to complete the migration.

Post Migration Requirements for On-premises Environment

Perform the following steps only if you are migrating an appliance instance which is installed in your on-premises environment and you have used a temporary storage gateway instance which is installed in the cloud for the purpose of migration. After migrating the file system to the temporary storage gateway instance in the cloud, connect the file system to an on-premises storage gateway instance.

1. Disconnect the file system on the temporary Oracle Cloud Infrastructure Storage Gateway instance that you had set up in the cloud and used for migration. See Disconnecting a File System in Oracle Cloud Infrastructure documentation.
2. Connect the file system to the Oracle Cloud Infrastructure Storage Gateway instance that you have installed in the on-premises environment. See Connecting a File System in Oracle Cloud Infrastructure documentation.

   This results in a file system takeover and requires you to provide credentials to access your Oracle Cloud Infrastructure account. It might take a while to connect the migrated file system. The time taken for this operation depends on the number of files and directories in the file system.
3. (Optional.) Perform additional validation by mounting the file system and performing a dry run with relevant application and workload.

Delete the Source File System and Appliance

After the migration is complete and you've validated the Oracle Cloud Infrastructure Storage Gateway file system, you can clean up unused resources.

1. Delete the file system from the Oracle Cloud Infrastructure Storage Software Appliance instance.
   1. Log in to the management console.
   2. On the **Dashboard** tab, identify the file system that you want to delete.
   3. Make sure that the file system is disconnected. If it’s still connected, then click **Disconnect**.
   4. After the file system is disconnected, click its name.
   5. On the page that displays the details of the file system, click **Delete**.
2. Delete the storage classic container after removing the objects in the container. The delete option is available only for empty containers.
   1. Sign in to the Oracle Cloud Infrastructure Object Storage Classic console.
   2. Identify the container that you want to delete.
   3. Click delete on the left side of the container name.
   4. Click **OK** to delete the container.
3. After migrating all the file systems, you can uninstall the Oracle Cloud Infrastructure Storage Software Appliance instance.

   To uninstall an Oracle Cloud Infrastructure Storage Software Appliance instance in the cloud, see Deleting the Appliance in *Using Oracle Cloud Infrastructure Storage Software Appliance*.

   To uninstall an Oracle Cloud Infrastructure Storage Software Appliance instance in your on-premises environment, see Uninstalling the Appliance in *Using Oracle Cloud Infrastructure Storage Software Appliance*.

You can also delete any instances that you have created for migrating the workloads as well as any associated resources, including block volumes.

Migrate Object Storage Using Rclone

If you don't use the Oracle Cloud Infrastructure Storage Software Appliance, then you can use Rclone to migrate data from Oracle Cloud Infrastructure Object Storage Classic to Oracle Cloud Infrastructure Object Storage.

> **Note:**
>
> Before you use the procedures in this document, review the amount of object storage data to be migrated. Consult with your Oracle representative to be sure the amount of data you want to migrate is supported by this migration method.

Considerations for Data Migration Using Rclone

Before you start your migration, consider the following factors that could have an impact on your migration process.

- You can't copy multiple containers at a time. You can copy only one container at a time. However, you can copy one or more objects at a time.
- Metadata and policies aren't copied. This includes:
  - Custom metadata on objects and containers
  - Cross-Origin Resource Sharing (CORS) settings
  - Object immutability
  - ACLs on containers
  - Container quotas
  - Container replication policies
- Server side encryption can be enabled on containers. Rclone can successfully copy containers where server side encryption is enabled.
- If you have large objects in your Oracle Cloud Infrastructure Object Storage Classic account, then you should check if the object as well as its segments are stored in the same container. When the same container has the object and as well as its segments, Rclone detects this and copies the object correctly without duplicating data. However, when the segments are in one container and the object manifest is in a different container (for example, if you upload a large object using the File Transfer Manager command-line interface), then Rclone can't detect the duplicate object. In such cases, ensure that you don't copy the container that has the segments or if you copy it, you delete it later to avoid duplicating data.
- There are some differences in the rules for naming containers in Oracle Cloud Infrastructure Object Storage Classic versus buckets in Oracle Cloud Infrastructure Object Storage. Container names can include UTF-8 characters subject to a maximum of 1061 bytes, while bucket names can include only upper or lower case letters, numbers, hyphens, underscores, and periods. Bucket names are also subject to a maximum of 256 characters. Due to these differences, you might need to rename some containers before you copy them.

- There are also some differences in the rules for naming objects in Oracle Cloud Infrastructure Object Storage Classic versus Oracle Cloud Infrastructure Object Storage. In Oracle Cloud Infrastructure Object Storage Classic you can specify object names with UTF-8 characters subject to a maximum of 1061 bytes, while in Oracle Cloud Infrastructure Object Storage you can specify object names with UTF-8 characters subject to a maximum of 1024 bytes and up to 1024 characters.

Consider using the following `rclone` options in the following scenarios:

- `dry-run:` Use this option to validate a migration before start to copy data. Amongst other benefits, this option allows you to check that the specified bucket name is valid.
- `includes, excludes,` or `filtering:` Use these options to include or exclude files to be copied, based on patterns or size.
- `s3-upload-cutoff:` Use this option to copy large objects.
- `progress:` Use this option to generate a real-time overview of the transfer.
- `transfers:` Use this option to utilize your network bandwidth better and increase throughput. You will need to tune this value based on the available bandwidth for your compute shape.

For more information about Oracle Cloud Infrastructure Object Storage Classic, see https://docs.oracle.com/en/cloud/iaas/storage-cloud/index.html and for information about Oracle Cloud Infrastructure Object Storage see https://docs.cloud.oracle.com/iaas/Content/Object/Concepts/objectstorageoverview.htm.

Before You Begin

Before you begin, complete the following prerequisites.

- Validate the existing object storage data and back up the object storage data in the source environment.
- Create a new object storage destination in your target environment. For information about creating a bucket, see Managing Buckets in Oracle Cloud Infrastructure documentation. You cannot change the name of the bucket after you create it, so carefully consider the naming conventions and if you want the bucket name to match the name of the container available at the source.
- To migrate data from an Archive container, you must first restore the archived objects. For more information, see Restoring Archived Objects in *Using Oracle Cloud Infrastructure Object Storage Classic*.

Get Details of Your Source Environment

Identify the user name, password, authentication URL, and REST Endpoint URL for the Oracle Cloud Infrastructure Object Storage Classic account from which you want to migrate data.

You can find out the user name and password from the New Account Information email that you received from Oracle Cloud when your account was set up. If you don't have your New Account Information email, ask your account administrator for your Oracle Cloud user name and password.
To identify the authentication URL and REST Endpoint URL:

1. Sign in to the Oracle Cloud My Services application.

The My Services dashboard is displayed. It lists the services that are assigned to your account.

2. Look for **Oracle Cloud Infrastructure Object Storage Classic**.
3. Select **View Details** from the **Actions** menu. Alternatively, click the **Oracle Cloud Infrastructure Object Storage Classic** link on the **Dashboard** page.

   On the resulting page, the details of your Oracle Cloud Infrastructure Object Storage Classic instance are displayed.
4. If your account was created after November 2017, then note down the following information that is displayed under the **Additional Information** section:
   - The authentication URL from the **Auth V1 Endpoint field** field. For example:

     ```
     https://acme.storage.oraclecloud.com/auth/v1.0
     ```

     If your authentication URL is not available, then you must construct the authentication URL. See Authenticating Access When Using the REST API in *Using Oracle Cloud Infrastructure Object Storage Classic*. You must specify this value while sending a request for an authentication token.
   - The REST Endpoint URL from the **REST Endpoint** field. For example:

     ```
     https://acme.storage.oraclecloud.com/v1/Storage-acme
     ```

Use Rclone to Migrate Your Object Storage Data

To migrate your object storage data, install and run `rclone`.

1. Install `rclone`.

   You can install rclone on any virtual machine or system that has network access to both the source and destination environments. Rclone supports Windows, Linux, and other operating systems. The procedure in this document has been tested using version 1.48 of the rclone tool against Windows and Macintosh operating systems. For more information about supported operating systems, downloading the installer, and installation instructions, see https://rclone.org/downloads/.
2. Create the `rclone.conf` file in the `~/.config/rclone` folder if the file doesn't already exist.
3. Add the following information to the `~/.config/rclone/rclone.conf` file to create the remote device configuration for the source.

   ```
   [classic-source]
   type = swift
   env_auth = false
   user = Storage-acme:myuserName
   key = pas$word
   storage_url = https://acme.storage.oraclecloud.com/v1/Storage-acme
   auth = https://acme.storage.oraclecloud.com/auth/v1.0
   auth_token = AUTH_xxxx
   ```

   Replace the values for the `user, key, storage_url, auth`, and `auth_token` parameters with the values specific to your source environment. Where:

- user: Specify the value you that you passed to the X-Storage-User header while requesting an authentication token to access Oracle Cloud Infrastructure Object Storage Classic.
- key: Specify the password to access your Oracle Cloud Infrastructure Object Storage Classic account.
- storage_url: Specify the REST Endpoint URL.
- auth: Specify the authentication URL that you had passed while requesting an authentication token to access Oracle Cloud Infrastructure Object Storage Classic.
- auth_token: Optionally, specify the value of a valid authentication token. This is not a required field when the user name, password, and authentication URL are specified.

4. To create the remote device configuration for the destination, add the following information to the ~/.config/rclone/rclone.conf file:

```
[oci-dest]
type = s3
env_auth = false
access_key_id = YOUR_ACCESS_KEY
secret_access_key = YOUR_ACCESS_SECRET_KEY
region = YOUR_REGION_IDENTIFIER
endpoint = https://
YOUR_NAMESPACE.compat.objectstorage.YOUR_REGION_IDENTIFIER.oraclecloud.c
om
```

   Replace the values for the access_key_id, secret_access_key, region and endpoint parameters with the values specific to your target environment. Where:

   - access_key_id and secret_access_key: To identify your access key and secret access key, see To create a Custom Secret key in Oracle Cloud Infrastructure documentation.
   - region: To identify the region, see Regions and Availability Domains in Oracle Cloud Infrastructure documentation. For example, us-ashburn-1.
   - endpoint: To identify the namespace, see Understanding Object Storage Namespaces in Oracle Cloud Infrastructure documentation..

5. Run the following commands to ensure that you can access the Oracle Cloud Infrastructure Compute Classic and the Oracle Cloud Infrastructure environments using the information that you have entered in the configuration file. The following command returns a list of containers that exist in your Oracle Cloud Infrastructure Object Storage Classic account.

```
rclone -v lsd classic-source:
```

   Note down the name of the container from which you want to copy the data.

6. The following command returns a list of buckets that exist in your Oracle Cloud Infrastructure account.

```
rclone -v lsd oci-dest:
```

Note down the name of the bucket to which you want to copy the data.

7. Start copying the data. The following command copies files from the source to destination and skips the files that have already been copied.

   ```
   rclone copy classic-source:<containername> oci-dest:<containername>
   ```

   To monitor the progress, you can add a debug option. For example:

   ```
   rclone -I --log-level DEBUG copy classic-source:<containername> oci-
   dest:<containername>
   ```

8. Optional. The following command modifies the destination to make it identical with source.

   ```
   rclone sync classic-source:<containername> oci-dest:<containername>
   ```