Oracle® Cloud

Using Oracle Cloud Infrastructure Database Migration Service

F40519-31
December 2025

Oracle Cloud Using Oracle Cloud Infrastructure Database Migration Service,

F40519-31

Primary Authors: Roopam Jain, Virginia Beecher

Contributors: Alex Kotopoulis, Jean-Francois Verrier, Prakash Jashnani, Gaurav Taneja, Jorge Martinez, Sergey Demyanenko, Luis Chapa, Shariful Haque, Costa Siourbas , Abigail Martinez , Tom O'Shaughnessy, Christian Gomez, Humberto Garcia Robles, Jose Narvaez

Contents

4 Prepare

5 Create connections

6 Manage connections

7 Create migrations

8 Manage migrations

9 Manage jobs

10 Troubleshoot

11 Reference

Preface

Topics

Audience

Oracle Cloud Infrastructure Database Migration is intended for database administrators responsible for identifying migration strategies and timelines, setting up connections to source and target data sources, configuring and executing migrations, and validating and generating audit and compliance reports.

Related Resources

See these Oracle resources:

- Get Started with Oracle Cloud
- Oracle Public Cloud

  https://cloud.oracle.com

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

Getting started

Before you can start using Oracle Cloud Infrastructure Database Migration service and before migrating your databases, you must perform the following preparatory tasks:

1. Create Database Migration policies in your tenancy.
2. Create any dependent objects needed for the migration.
3. Configure the source and target databases as required.

- Configuring SUDO access
  You may need to grant certain users authority to perform operations using `sudo` on the source database servers.
- Access OCI Database Migration Service
  You can access Oracle Cloud Infrastructure Database Migration Service using the Oracle Cloud Interface Console (a browser based interface), REST APIs, or Oracle Cloud Infrastructure Software Development Kits and Command Line Interface.
- Basic taskflows
  The following taskflows guide you on how to get set up with OCI Database Migration Service quickly and easily. Depending on your use case, you may diverge from the path outlined below, in which case you can choose to follow one of the various quickstarts available.
- Quickstarts
  The following Quickstarts are common use cases that were mapped out into step by step instructions. Depending on your use case, you can choose to follow one of the various quickstarts available.

Configuring SUDO access

You may need to grant certain users authority to perform operations using `sudo` on the source database servers.

> **Note**
>
> This topic is only applicable for Oracle migrations.

To configure `sudo` access for source database servers:

If the source database server is accessed through SSH, then configure `sudo` operations to run without prompting for a password for the database installed user and the `root` user.

For example, if database installed user is `oracle`, then run `sudo su - oracle`.

Note that the `opc` user is a standard Oracle cloud user that is used to access database servers, but you can use any privileged user that has `sudo` privileges.

For the `root` user run `sudo su -`.

Also, note that because the target database server is on the cloud only any `sudo` operations are configured already.

Access OCI Database Migration Service

You can access Oracle Cloud Infrastructure Database Migration Service using the Oracle Cloud Interface Console (a browser based interface), REST APIs, or Oracle Cloud Infrastructure Software Development Kits and Command Line Interface.

To access Database Migration using the Console:

1. Use a supported browser to access the Console.

   See Signing In to the Console for details.
2. Enter your cloud tenant, user name, and password, when prompted.
3. Click **Sign in**.
4. In the upper-right corner of the window, select a region that offers the Database Migration service enabled; for example, **US East (Ashburn)**.

   Database Migration resources, such as database registrations, migrations, agents, and jobs, are region-specific. Therefore, you want to make sure that you select Database Migration in the region that contains the resources that you need.
5. From the navigation menu, select **Database Migration**.

   The Migrations page for the Database Migration service is displayed.

Basic taskflows

The following taskflows guide you on how to get set up with OCI Database Migration Service quickly and easily. Depending on your use case, you may diverge from the path outlined below, in which case you can choose to follow one of the various quickstarts available.

To access Database Migration using the Console:

Table 1-1 Database migration taskflow

Task	Description	More information
Planning your migration	• Identify the business requirements: Is downtime on the source database allowed? • Identify technical requirements: Source, target, database versions, database size, transfer medium. • Ensure that your use case is supported by the OCI Database Migration service.	Overview

Table 1-1 (Cont.) Database migration taskflow

Task	Description	More information
Creating OCI resources	• Before you can start using Oracle Cloud Infrastructure Database Migration service and before migrating your databases, you must perform the following preparatory tasks: • Create Database Migration policies in your tenancy. • Create any dependent objects needed for the migration	Getting started
Preparing databases	Before you can begin the migration of your data with Oracle Cloud Infrastructure Database Migration Service, you must configure your source and target databases.	Prepare
Creating connections	Oracle Cloud Infrastructure Database Migration database connection resources contain the connectivity details of the migration source and target databases.	Create connections
Creating migrations	When you create a migration with Oracle Cloud Infrastructure Database Migration, you specify how the migration should run, select the source and target databases, and then configure the data transport settings.	Create migrations
Running migrations	When you create a migration with Oracle Cloud Infrastructure Database Migration, you specify how the migration should run, select the source and target databases, and then configure the data transport settings.	Running a migration job
Monitoring migrations	In Oracle Cloud Infrastructure Database Migration, there are several places in the Console from which you can monitor the status of a migration job to varying degrees.	Monitoring job status
Completing migrations	On the Migration Details page you can view a list of the jobs a migration resource has performed in Oracle Cloud Infrastructure Database Migration.	Preparing for application switchover

Quickstarts

The following Quickstarts are common use cases that were mapped out into step by step instructions. Depending on your use case, you can choose to follow one of the various quickstarts available.

Table 1-2 Database migration quickstarts

Quickstart	More information
OCI Database Migration Service End-To-End Online Migration to Autonomous Databases	For migration scenarios where your application must remain online and your source database has a direct connection to OCI.
Configuring a Migration Resource with an Amazon RDS Source	To transfer the data from AWS, you have the following options: Amazon Simple Storage Service (Amazon S3) Bucket*: Amazon S3 integration needs to be completed. To learn more, see this link*.
OCI Database Migration Service Step by Step Guide: Online Migration for Autonomous Databases on Oracle Database@Azure	For migration scenarios where applications are online and source database is connected to OCI.
Migrate AWS RDS for MySQL to MySQL HeatWave using OCI Database Migration	For migration scenarios when you want to migrate an AWS RDS for MySQL database to MySQL HeatWave database with minimal downtime.
Migrating Databases from Amazon Web Services RDS to Supported Oracle Targets	For migration scenarios when you want to migrate an Oracle Database from Amazon Web Services (AWS) RDS to the supported Oracle targets using the Oracle Cloud Infrastructure Database Migration offline and online migration methods.
OCI Database Migration Service End-To-End Online Migration to Oracle Database@Google Cloud Autonomous Database	For migration scenarios where your application must remain online and your source database has a direct connection to OCI.

For instance, for a quick reference, here is one quickstart to get you started:

- Migrating Databases from Amazon Web Services RDS to Supported Oracle Targets
  You can migrate an Oracle Database from Amazon Web Services (AWS) RDS to the supported Oracle targets using the Oracle Cloud Infrastructure Database Migration offline and online migration methods.

Migrating Databases from Amazon Web Services RDS to Supported Oracle Targets

You can migrate an Oracle Database from Amazon Web Services (AWS) RDS to the supported Oracle targets using the Oracle Cloud Infrastructure Database Migration offline and online migration methods.

Configuring Secure Connections

Ensure that the subnet Amazon RDS security policy allows connections from Database Migration to the DB instance on the specified secure port. See the AWS documentation for details:

Scenarios for accessing a DB instance in a VPC

Scenarios for accessing a DB instance not in a VPC

Allowing Database Migration to connect to Amazon RDS Oracle DB instance using SSL/TLS

1. Enable Secure Socket Layer (SSL) or Transport Layer Security (TLS) in the Amazon RDS Oracle Instance to secure the connection from Database Migration to Amazon RDS Oracle Instance. See Encrypting client connections with SSL for details.
2. Create an orapki wallet as detailed in Updating applications to use new SSL/TLS certificates.

Configuring a Connection for an Amazon RDS Source

- Find the Amazon RDS Oracle Instance endpoint (DNS name) and port number in the RDS console DB Instance Connectivity & security tab.

  See Finding the endpoint of your Oracle DB instance for detailed help.
- In OCI Database Migration, create the Connection resource for the Amazon RDS Oracle source database, using the following guidelines.
  - In the **Database Connections** wizard, **Database details** step, select **Manually configure database**, choose Amazon RDS in the **Database type** list, and enter the full connect string with host, port, and service name in the following format:

    `host:port/db-service-name`
  - In the **Connection details** step, enter the database administrator credentials for the Amazon RDS Oracle source database. The user must have full Data Pump Export privileges.

    If you intend to use a database link to transfer the data, also set the TLS parameters.

Configuring a Migration Resource with an Amazon RDS Source

To transfer the data from AWS, you have the following options:

- Amazon Simple Storage Service (Amazon S3) Bucket
- Database link

When you create the Migration resource, in the **Migration options** step configure one of the initial load settings as follows.

- **Datapump via database link:** Enable this option to use a direct SQL*Net connection between the source and target databases. Note that using Data Pump with a database link to Autonomous Database targets requires that the source database be set up with SSL encryption.

  To use a database link to migrate Amazon RDS Oracle Database schema to Oracle Autonomous Database (ADB), you must have direct network connectivity between the Amazon RDS Oracle instance and the ADB target.
- **Data Pump via Amazon Simple Storage Service:** This option lets you select the **Amazon S3 bucket** option to let Data Pump temporarily store the exported database in an Amazon S3 bucket.

  Enter the details for the Amazon S3 bucket. This option is only shown if the source Connection is of type Amazon RDS.

  The bucket **Name** must be between 3 and 63 characters, and can consist only of lower case letters, numbers, dots (.), and hyphens (-). It must begin and end with a letter or number.

The **Region** must be in the same region as the RDS Oracle database. For example `us-east-1`

Note that you must also configure the OCI Object Storage bucket so that Database Migration can store Cloud Pre-migration Advisor Tool reports, Database Migration logs, and Data Pump logs there.

2

Overview

Learn about the Oracle Cloud Infrastructure Database Migration service.

The following topics explain what Oracle Cloud Infrastructure Database Migration can do and describe the concepts you need to know about the service.

To learn more about the service, see OCI Database Migration.

- About Oracle Cloud Infrastructure Database Migration
  Oracle Cloud Infrastructure Database Migration is a fully-managed service that provides you a high performing, self-service experience for migrating databases to Oracle Cloud Infrastructure (OCI).
- Database Migration terminology
  The following concepts are essential for working with Oracle Cloud Infrastructure Database Migration service.
- What's new in Database Migration
- About Zero Downtime Migration
  Oracle Cloud Infrastructure Database Migration service is internally driven by the Zero Downtime Migration Server, which is an integral part of the Oracle product, Zero Downtime Migration.
- Resource identifiers
  Database Migration resources have a unique, Oracle-assigned identifier called an Oracle Cloud ID (OCID).

About Oracle Cloud Infrastructure Database Migration

Oracle Cloud Infrastructure Database Migration is a fully-managed service that provides you a high performing, self-service experience for migrating databases to Oracle Cloud Infrastructure (OCI).

You can perform homogeneous Oracle and MySQL migrations using the Oracle Cloud Infrastructure Database Migration service. Database Migration runs as a managed cloud service separate from your tenancy and resources. The service operates as a multitenant service in a Database Migration service tenancy and communicates with your resources using Private Endpoints (PEs). PEs are managed by Database Migration.

Database Migration includes the following capabilities:

- Oracle Migration of data from on-premises, Oracle Cloud, and Amazon RDS Oracle databases into co-managed, Autonomous AI Lakehouse, or Autonomous AI Transaction Processing services on Oracle Cloud Infrastructure
- MySQL sources into OCI HeatWave.
- Simple offline migration option or enterprise-level logical migration with minimal downtime option
- Based on industry-leading Oracle GoldenGate replication and powered by the Zero Downtime Migration engine.

> **Note**
>
> ZDM is applicable only for Oracle migrations.

- Compliant with Oracle Maximum Availability Architecture (MAA) and supports Oracle Database 11g Release 2 (11.2.0.4), MySQL 5.7 and 8 versions, and later database releases.
- Seamless transition from initial load to streamed replication
- Performs change data capture on the source database and replicates these changes to the target
- Job subsystem lets you perform and manage database migrations at a fleet scale.
- Pause and resume functionality lets you pause and resume your migration job if needed, which is useful to conform to a maintenance window, for example
- Job termination lets you terminate a running migration job, rather than waiting for it to complete
- Re-run (resume) migration jobs from a point of failure
- Job pre-checks for migration tasks to prevent errors during database migration

Database Migration terminology

The following concepts are essential for working with Oracle Cloud Infrastructure Database Migration service.

Migration
Represents a single migration operation and contains the specifications by which the migration should run. Migration specifications include whether or not to perform bulk data copy, and/or capture ongoing changes, and the source and target database selections.

Migration Job
Represents an active or past migration execution. A migration job is created implicitly when you start a migration. A migration job is a snapshot with runtime information about the migration. You use this information to audit logs and investigate failures.

Validation Job
Validates the prerequisites and connectivity for source and target databases, Oracle GoldenGate instances, and Oracle Data Pump. A validation job is created when you evaluate the migration.

Connections
Represents a database instance, containing the database metadata and connection details. A data asset can have one or many connections to include all schemas within a database that need to be migrated.

Agent
The OCI Database Migration Agent is only required in rare, offline scenarios where your on-premises database cannot be accessed directly from OCI.
Typical use cases include:

- When OCI cannot initiate inbound connections due to strict firewall rules.

- When no VPN or FastConnect exists between your on-premises environment and OCI.

The agent runs in your local environment and securely initiates outbound communication to OCI, acting on behalf of the OCI Database Migration Service. It executes Data Pump exports locally and transfers the resulting data back to OCI under OCI's direction.

> **Note**
>
> In most scenarios, agent installation is not necessary. You should only use it when direct connectivity to the source database from OCI is not possible.

Private Endpoint
Private Endpoint (PE) is a networking feature by which you can allow OCI Services to access resources (database instances for example) in your VCN which are configured with only Private IP addresses. Oracle Cloud Infrastructure Database Migration service makes it easy to enable this ability though the use Database Connection Resources.
Gives hosts within your virtual cloud network (VCN) and your on-premises network access to a single resource within the Oracle service of interest (for example, one Autonomous Database with shared Exadata infrastructure). Connection to either a source or target database in the migration is currently supported by the service. Make sure security rules or network security groups allow traffic required for database migration jobs. Learn more at Port requirements.

Schema
Organizational concepts of databases to hold database objects such as tables, views, stored procedures, and so on.

What's new in Database Migration

For information about new features and enhancements, check What's New for Oracle Cloud Infrastructure Database Migration Service and Database Migration Release Notes on OCI.

About Zero Downtime Migration

Oracle Cloud Infrastructure Database Migration service is internally driven by the Zero Downtime Migration Server, which is an integral part of the Oracle product, Zero Downtime Migration.

Zero Downtime Migration configuration is handled automatically by Database Migration, so you don't have to perform any Zero Downtime Migration set up.

To learn more about Zero Downtime Migration see Zero Downtime Migration on Oracle Help Center and Oracle Zero Downtime Migration on Oracle's Database Technologies web site.

Resource identifiers

Database Migration resources have a unique, Oracle-assigned identifier called an Oracle Cloud ID (OCID).

Database Migration resources are `OdmsAgent`, `OdmsConnection`, `OdmsMigration`, and `OdmsJob`.

For example, the OCID format for `OdmsJob` is `ocid1.odmsjob.oc1.[REGION][.FUTURE USE].<UNIQUE ID>`.

For information about the OCID format and other ways to identify your resources, see Resource Identifiers.

3

Plan

Before you can begin the migration of your data with Oracle Cloud Infrastructure Database Migration Service, you must configure your source and target databases as described here.

- Before you begin
  Before you begin working with Oracle Cloud Infrastructure Database Migration, you must have an Oracle Cloud Infrastructure account with administrator privileges.
- Learn Oracle Cloud
  If you're completely new to Oracle Cloud, then you should review Getting Started with Oracle Cloud to learn about types of accounts, terminology, and how to sign up.
- Region availability
  OCI Database Migration is available to all Oracle Cloud customers using **Universal Credits** and **Pay As You Go**.
- Service limits
  Oracle Cloud Infrastructure Database Migration service limits you to 10 Connections, 5 migrations, and 5 agents.
- Compartment quotas
  In Oracle Cloud Infrastructure Database Migration service, creating a quota lets you limit the number of migration resources in a compartment.
- Integrated services
  The Database Migration service is integrated with various Oracle Cloud Infrastructure services and features.
- Migration type
  Select the type of migration, offline or online.
- What is migrated during initial load
  The initial load phases of an Oracle Cloud Infrastructure Database Migration service migration job work flow moves the contents of all selected schemas from the source database to schemas of the same name in the target database.
- Data replication
  During the Oracle Cloud Infrastructure Database Migration service migration job work flow replication phase, all data and metadata operations in transactions committed after the initial load are replicated until you resume the migration job after the Monitor Replication Lag phase.
- Supported technologies
  Following are the source and target database requirement for the supported technologies.

Before you begin

Before you begin working with Oracle Cloud Infrastructure Database Migration, you must have an Oracle Cloud Infrastructure account with administrator privileges.

> **Note**
>
> If you are planning to use the Marketplace GoldenGate instance, then refer to the Securing OCI GoldenGate topic.

See Add a User with Oracle Cloud Administrator Permissions for details.

- Creating resources
  Use the following instructions to create the resources that Oracle Cloud Infrastructure Database Migration operations depend on.
- Giving permissions to Database Migration users
  Use IAM policies to grant certain capabilities to the Oracle Cloud Infrastructure Database Migration user group.

Creating resources

Use the following instructions to create the resources that Oracle Cloud Infrastructure Database Migration operations depend on.

Configuring Policies

Your tenancy administrator provisions policies for OCI Database Migration resources along with the other OCI resources that are created and managed in your tenancy. The tenancy administrator can provision policies to create, manage, and use the service resources depending on your usage. To make the provision of policies for users easier, OCI Database Migration Service provides you pre-built policy templates for you to provision policies depending on the usage. See Creating a Policy for more information.

Create a compartment

If you don't already have a compartment, create a compartment in your tenancy.

For more information, see Working with Compartments.

Create a Virtual Cloud Network

Create a Virtual Cloud Network (VCN) with at least one subnet in the compartment.

The subnet must be regional, spanning all availability domains.

For more information, see VCNs and Subnets.

> **Note**
>
> If you don't see your subnet listed, go back and check that it was created as a regional subnet. By default, the VCN wizard creates regional subnets.

Create or reuse an existing Database Migration user or a dynamic group

Create a user group to manage database connections, migrations, and jobs, and then add users in charge of database migrations to the group.

Take note of the group name. You will create policies for the group in Creating Resource Policies. For more information, see Managing Groups.

Create an OCI API key pair

Create an OCI API key pair if you intend to directly use the REST API, OCI Software Development Kits and Command Line Interface, or if you are installing the Database Migration agent.

Follow the instructions in Required Keys and OCIDs.

Select a data transfer medium

The following data transfer mediums are available:

- Data Pump via database link
- Data Pump via object storage
- Data Pump via file storage

If you are not using a database link to transfer files directly from the source to the target database server, you can set up an Object Storage Service bucket for temporary storage of the Data Pump export dumps. Alternatively, use file storage as the data transfer medium which is the fastest method for transferring data.

> **Note**
>
> Currently, MySQL migrations support using the Data Pump via object storage data transfer medium.

See Creating a Bucket for details.

Make sure that the file system used for the Data Pump export directory has sufficient space to store Data Pump dump files.

Select or create a vault

Create a Vault and Create a Secret in a Vault.

Alternatively, select a vault when creating database connections. Create a key in the Master Encryption Keys to use with Database Migration. Optionally, you can create vault during database migrations.

Giving permissions to Database Migration users

Use IAM policies to grant certain capabilities to the Oracle Cloud Infrastructure Database Migration user group.

Previously, in Creating resources you created a user group for Oracle Cloud Infrastructure Database Migration. Now you will configure group permissions so that members can manage Database Migration resources.

Learn Oracle Cloud

If you're completely new to Oracle Cloud, then you should review Getting Started with Oracle Cloud to learn about types of accounts, terminology, and how to sign up.

You should also familiarize yourself with Oracle Cloud Infrastructure concepts such as tenancies, compartments, VCNs and subnets, and policies, before you create any OCI Database Migration resources. See:

- Setting up your tenancy
- Managing compartments
- VCNs and subnets
- How policies work

Region availability

OCI Database Migration is available to all Oracle Cloud customers using **Universal Credits** and **Pay As You Go**.

OCI Database Migration is available in:

- Commercial Regions
- Oracle Cloud Infrastructure United Kingdom Government Cloud
- Oracle EU Sovereign Cloud
- Oracle Alloy and Dedicated Regions

OCI Database Migration is not available for users of the Oracle Cloud Free Tier. The service is currently available in the following regions:

Table 3-1 OCI Database Migration Commercial Regions

Region Name	Region Identifier	Region Location	Region Key	Realm Key	Availability Domains
Australia East (Sydney)	ap-sydney-1	Sydney, Australia	SYD	OC1	1
Australia Southeast (Melbourne)	ap-melbourne-1	Melbourne, Australia	MEL	OC1	1
Brazil East (Sao Paulo)	sa-saopaulo-1	Sao Paulo, Brazil	GRU	OC1	1

Table 3-1 (Cont.) OCI Database Migration Commercial Regions

Region Name	Region Identifier	Region Location	Region Key	Realm Key	Availability Domains
Brazil Southeast (Vinhedo)	sa-vinhedo-1	Vinhedo, Brazil	VCP	OC1	1
Canada Southeast (Montreal)	ca-montreal-1	Montreal, Canada	YUL	OC1	1
Canada Southeast (Toronto)	ca-toronto-1	Toronto, Canada	YYZ	OC1	1
Chile (Santiago)	sa-santiago-1	Santiago, Chile	SCL	OC1	1
Chile West (Valparaiso)	sa-valparaiso-1	Valparaiso, Chile	VAP	OC1	1
Colombia Central (Bogota)	sa-bogota-1	Bogota, Colombia	BOG	OC1	1
France Central (Paris)	eu-paris-1	Paris, France	CDG	OC1	1
France South (Marseille)	eu-marseille-1	Marseille, France	MRS	OC1	1
Germany Central (Frankfurt)	eu-frankfurt-1	Frankfurt, Germany	FRA	OC1	3
India South (Hyderabad)	ap-hyderabad-1	Hyderabad, India	HYD	OC1	1

Table 3-1 (Cont.) OCI Database Migration Commercial Regions

Region Name	Region Identifier	Region Location	Region Key	Realm Key	Availability Domains
India West (Mumbai)	ap-mumbai-1	Mumbai, India	BOM	OC1	1
Israel Central (Jerusalem)	il-jerusalem-1	Jerusalem, Israel	MTZ	OC1	1
Italy Northwest (Milan)	eu-milan-1	Milan, Italy	LIN	OC1	1
Japan Central (Osaka)	ap-osaka-1	Osaka, Japan	KIX	OC1	1
Japan East (Tokyo)	ap-tokyo-1	Tokyo, Japan	NRT	OC1	1
Mexico Central (Queretaro)	mx-queretaro-1	Queretaro, Mexico	QRO	OC1	1
Mexico Northeast (Monterrey)	mx-monterrey-1	Monterrey, Mexico	MTY	OC1	1
Netherlands Northwest (Amsterdam)	eu-amsterdam-1	Amsterdam, Netherlands	AMS	OC1	1
Saudi Arabia Central (Riyadh)	me-riyadh-1	Riyadh, Saudi Arabia	RUH	OC1	1

Table 3-1 (Cont.) OCI Database Migration Commercial Regions

Region Name	Region Identifier	Region Location	Region Key	Realm Key	Availability Domains
Saudi Arabia West (Jeddah)	me-jeddah-1	Jeddah, Saudi Arabia	JED	OC1	1
Singapore (Singapore)	ap-singapore-1	Singapore,Singapore	SIN	OC1	1
Singapore West (Singapore)	ap-singapore-2	Singapore,Singapore	XSP	OC1	1
South Africa Central (Johannesburg)	af-johannesburg-1	Johannesburg, South Africa	JNB	OC1	1
South Korea Central (Seoul)	ap-seoul-1	Seoul, South Korea	ICN	OC1	1
South Korea North (Chuncheon)	ap-chuncheon-1	Chuncheon, South Korea	YNY	OC1	1
Spain Central (Madrid)	eu-madrid-1	Madrid, Spain	MAD	OC1	1
Sweden Central (Stockholm)	eu-stockholm-1	Stockholm, Sweden	ARN	OC1	1
Switzerland North (Zurich)	eu-zurich-1	Zurich, Switzerland	ZRH	OC1	1

Table 3-1 (Cont.) OCI Database Migration Commercial Regions

Region Name	Region Identifier	Region Location	Region Key	Realm Key	Availability Domains
UAE Central (Abu Dhabi)	me-abudhabi-1	Abu Dhabi, UAE	AUH	OC1	1
UAE East (Dubai)	me-dubai-1	Dubai, UAE	DXB	OC1	1
UK South (London)	uk-london-1	London, United Kingdom	LHR	OC1	3
UK West (Newport)	uk-cardiff-1	Newport, United Kingdom	CWL	OC1	1
US East (Ashburn)	us-ashburn-1	Ashburn, VA	IAD	OC1	3
US Midwest (Chicago)	us-chicago-1	Chicago, IL	ORD	OC1	3
US West (Phoenix)	us-phoenix-1	Phoenix, AZ	PHX	OC1	3
US West (Salt Lake City)	us-saltlake-2	Salt Lake City		OC1	1
US West (San Jose)	us-sanjose-1	San Jose, CA	SJC	OC1	1

See Database Migration API endpoints to find the API endpoints for each region.

Related Topics

- Oracle Cloud Infrastructure Free Tier
- About Universal Credits
- Oracle Cloud Infrastructure United Kingdom Government Cloud
- Oracle EU Sovereign Cloud
- Dedicated Regions

- OCI Data Regions

Service limits

Oracle Cloud Infrastructure Database Migration service limits you to 10 Connections, 5 migrations, and 5 agents.

Your tenancy has limits on the maximum number of resources that you are allowed to use. To view your tenancy's limits for Oracle Cloud Infrastructure Database Migration service, see Limits by Service. If you are an administrator in an eligible account, you can request to increase the service limits in the OCI Console, see Requesting a Service Limit Increase.

Compartment quotas

In Oracle Cloud Infrastructure Database Migration service, creating a quota lets you limit the number of migration resources in a compartment.

For example:

```
set database-migration quota odms-migration-count to 10 in compartment
compartment_name
```

See Overview of Compartment Quotas for information.

Integrated services

The Database Migration service is integrated with various Oracle Cloud Infrastructure services and features.

- IAM
  Database Migration integrates with the Identity and Access Management (IAM) service for authentication and authorization for the Console, SDK, CLI, and REST API.
- Work requests
  The OCI Database Migration service uses its own API for Work Requests.
- Monitoring
  Oracle Cloud Infrastructure Monitoring lets you actively and passively monitor your Oracle Cloud Infrastructure Database Migration resources and alarms.

IAM

Database Migration integrates with the Identity and Access Management (IAM) service for authentication and authorization for the Console, SDK, CLI, and REST API.

To learn more about IAM, see IAM Overview.

Work requests

The OCI Database Migration service uses its own API for Work Requests.

See WorkRequest.

Monitoring

Oracle Cloud Infrastructure Monitoring lets you actively and passively monitor your Oracle Cloud Infrastructure Database Migration resources and alarms.

Database Migration Metrics capture CPU utilization, OCPU consumption, memory utilization, deployment health, and inbound and outbound lag. You can view these metrics using the Monitoring service.

See Troubleshooting Database Migration Service for topics about monitoring resource status and accessing logs.

Migration type

Select the type of migration, offline or online.

> **Note**
>
> **For online migrations**
>
> - To prevent Foreign Key constraint violation, all foreign keys are automatically disabled and then re-enabled before the switchover.
> - Any DDL performed on these constraints during this period are not supported and will lead to failures in GoldenGate replication. This DDL limitation only applies during Data Pump Export.
> - If DDLs on the disabled constraints are expected to happen during Data Pump Export then an Online migration with Snapshot Standby database is recommended.
> - Performing the initial load from a Snapshot Standby (static environment) ensures a consistent dataset on the target, and all DDL changes will be captured by GoldenGate and applied to the target.

- Offline migration
  When using the offline migration method, you must stop updates to the source database before you start a migration.
- Online migration
  An online migration enables you to perform database migration without any downtime of your source database.

Offline migration

When using the offline migration method, you must stop updates to the source database before you start a migration.

Using the offline migration method, Database Migration service transports the data from the source database using the preferred transfer medium, and then imports the data from the selected transfer medium to the target database on the Cloud using Oracle Data Pump.

Online migration

An online migration enables you to perform database migration without any downtime of your source database.

- Online migrations consist of the following steps;
  1. Initial load

     > **Note**
     >
     > For MySQL migrations, MySQL Shell is used for initial load.

  2. Real-time replication
- When using the online migration method, you do not need to stop updates to the source database before you start a migration.
- Online migrations are facilitated by Oracle GoldenGate's replication technology to allow zero downtime of your source database.

To take advantage of parallelism and achieve the best data transfer performance, Oracle recommends that you transfer data using Object Store for databases over 50GB in size. The database link transfer medium can be convenient for smaller databases, but this choice may involve uncertainty in performance because of its dependence on network bandwidth for the duration of the transfer.

As part of a migration job, Database Migration uses GoldenGate's replication technology to facilitate database replication between the source and target databases.

When the application switches over to the target database, Database Migration tears down the replication so that the target database in the Cloud can then be used as the production database. Note that bi-directional synchronization is not currently supported. Synchronization is always from the source database to the target database.

Note that Online migration using GoldenGate Marketplace instances require SQL*Net connectivity from the GoldenGate host running in the target tenancy to the source database.

- Using the Snapshot Standby database
  You can use the Snapshot Standby database as the source database for Data Pump Export.

Using the Snapshot Standby database

You can use the Snapshot Standby database as the source database for Data Pump Export.

> **Note**
>
> This feature is available only for online Oracle migrations, except for Amazon RDS.

Why use this option:

- Enables Data Pump export with a high degree of parallelism and zero impact on the production database.
- Allows optimal resource usage without affecting the performance of the production system.

- Makes it possible to disable the `DBMS_SCHEDULER` jobs during export, helping avoid export issues that cannot be addressed in a production environment.
- Helps prevent `ORA-01555: snapshot too old errors`, especially during export of large tables.

Prerequisites:

1. You must manually convert the physical standby database to a snapshot standby database.
2. You must pre-create the export directory and export directory object on both the primary and standby databases, using the same name and path.
3. For Object Storage Service (OSS) as the transfer medium: The export directory on both primary and standby databases must point to a shared filesystem.
4. For File Storage Service (FSS) as the transfer medium: The same FSS mount must be accessible from both the primary and standby databases.
5. You must create a database connection to the snapshot standby database in the OCI Database Migration service.

How to enable this feature:
During migration creation and for supported Oracle online migrations, toggle the **Use Snapshot Standby database as source for Data Pump export** option. Then, select the corresponding **snapshot standby database connection** that was previously created.

What the service will do:

1. Identify the standby SCN and report it in the progress log.
2. Perform the Data Pump export without setting the `FLASHBACK_SCN` or `FLASHBACK_TIME` parameters.

> **Note**
>
> While setting `FLASHBACK_SCN` ensures a consistent export, however it can reduce the performance.

3. As the snapshot standby is not being updated during export and no scheduler jobs are expected to run, the export is inherently consistent.
4. Once the export is complete, the service will:
   a. Proceed with the import.
   b. Start Replicat using the `AFTERCSN` value derived from the standby SCN.

> **Note**
>
> You can revert the snapshot standby database back to a physical standby at your convenience after the export phase is complete.

What is migrated during initial load

The initial load phases of an Oracle Cloud Infrastructure Database Migration service migration job work flow moves the contents of all selected schemas from the source database to schemas of the same name in the target database.

You can elect to include or exclude specific objects, and rename objects when you create a migration.

See Selecting objects for Oracle migration for information about how to configure object selection rules, and what objects are excluded by default.

Data replication

During the Oracle Cloud Infrastructure Database Migration service migration job work flow replication phase, all data and metadata operations in transactions committed after the initial load are replicated until you resume the migration job after the Monitor Replication Lag phase.

During the migration job it is recommended that your database avoid Data Definition Language (DDL) operations to provide the most optimal environment for fast database replication. When DDL is replicated, Oracle GoldenGate Replicat serializes data to ensure that there are no locking issues between DML and DDL on the same objects.

By default, the Oracle Cloud Infrastructure Database Migration service configures GoldenGate to replicate all DDL operations on the source database to the target database.

> **Note**
>
> DDL is applicable only for Oracle migrations.

The following objects are not supported:

- Changes to external tables
- Oracle GoldenGate Unsupported Types (see Understanding What's Supported)

Supported technologies

Following are the source and target database requirement for the supported technologies.

- Source database requirements
  Your source database environment must meet these requirements to use Oracle Cloud Infrastructure Database Migration.
- Target database requirements
  Your target database environment must meet these requirements to use Oracle Cloud Infrastructure Database Migration.

Source database requirements

Your source database environment must meet these requirements to use Oracle Cloud Infrastructure Database Migration.

Supported source database versions

The following Oracle Database versions can be migrated using Database Migration, and the source database can be at any configuration.

- Oracle Database 11g Release 2 (11.2.0.4)
- Oracle Database 12c Release 1 (12.1.0.2)
- Oracle Database 12c Release 2 (12.2.0.1)

- Oracle Database 19c
- Oracle Database 21c
- All subsequent Oracle Database releases
- Oracle Autonomous Database Serverless
- Oracle Autonomous Database on Dedicated Exadata Infrastructure

The following source databases are currently supported for MySQL connections and migrations:

- OCI HeatWave: MySQL 5.7 and 8 versions
- MySQL Database Server
- Amazon RDS for MySQL
- Azure Database for MySQL
- Amazon Aurora MySQL
- Google Cloud SQL for MySQL

Supported source environments

- Oracle Cloud Infrastructure co-managed databases or on-premises environments
- Amazon Web Services RDS Oracle Database (both offline and online migrations)

> **Note**
>
> Amazon Web Services RDS Oracle Database Multitenant architecture (CDB) is currently not supported for online migrations.

- Linux-x86-64, IBM AIX, and Oracle Solaris.

> **Note**
>
> Currently, using Microsoft Windows based Oracle Databases as source is not supported.

- Oracle Database edition support
  Oracle Cloud Infrastructure Database Migration service supports the migrations of Standard and Enterprise Edition Oracle databases for source and targets.

Oracle Database edition support

Oracle Cloud Infrastructure Database Migration service supports the migrations of Standard and Enterprise Edition Oracle databases for source and targets.

Target database requirements

Your target database environment must meet these requirements to use Oracle Cloud Infrastructure Database Migration.

Supported migration targets

Database Migration supports the following Oracle Cloud Infrastructure Database Service offerings as migration targets.

- Oracle Autonomous Database Serverless
- Oracle Autonomous Database on Dedicated Exadata Infrastructure
- Oracle Database@Google Cloud

> **Note**
>
> You need a standard OCI Tenancy that is linked to your Google cloud account. For more details, see Oracle Database@Google Cloud

- Oracle Cloud Infrastructure co-managed Oracle Base Database service (Oracle Base Database (VM, BM) and Exadata on Oracle Public Cloud)
- Oracle Database@Azure

> **Note**
>
> You need a standard OCI Tenancy that is linked to your Azure account. For more details, see Oracle Database@Azure.

- The OCI HeatWave is the only target database that is currently supported for MySQL migrations.

> **Note**
>
> A target co-managed database can be either a pluggable database (PDB) in a multitenant container database (CDB), or a traditional non-CDB Oracle database. For Bare Metal and Virtual Machine Database Systems, the user is responsible for securing, patching, and hardening the environment. To learn more about this, see Bare Metal and Virtual Machine DB Systems.

4

Prepare

Before you can begin the migration of your data with Oracle Cloud Infrastructure Database Migration Service, you must configure your source and target databases as described here.

- Preparing Oracle Databases for migration
  Prepare your databases using either of the following methods:
- Preparing MySQL databases for migration
  If you want to perform online MySQL migrations, prepare your source and target databases for replication as follows:

Preparing Oracle Databases for migration

Prepare your databases using either of the following methods:

- Prepare your database by running scripts generated by the database preparation utility (Recommended option).
- Manually configure your Oracle databases by following the documentation and running the SQL commands.

Preparing your databases using the database preparation utility:

To prepare your databases for migration:

1. Refer to this MOS note.
2. Download the database preparation utility which is a shell script file.
3. Follow the instructions to proceed.
4. Run the script locally.

The database preparation utility:

1. Accepts the inputs that are specific to your migration and generates a SQL script that you can run for your source and target databases.
2. Analyzes your databases for any missing required configurations or privileges.
3. Checks the current status of the database and provides information on the operations that will be performed on your databases.
4. Generates a final script that performs the required operations on your databases to prepare them for the migration.

> **Note**
>
> - You must review and make necessary corrections to the scripts generated by the database preparation utility before you run them for your database.
> - You must run the utility script twice, once for the source database and then for the target database.

Consequently, the configuration SQL script prepares the database for migration.

Manually configuring your databases for migration:

To prepare your source and target database manually using the SQL commands, see the following topics:

- Preparing the source database for migration
  Before you can migrate data with Oracle Cloud Infrastructure Database Migration, manually configure your source database as described here.
- Preparing the target database for migration
  Before you can migrate data with Oracle Cloud Infrastructure Database Migration, manually configure your target database as described here.

Preparing the source database for migration

Before you can migrate data with Oracle Cloud Infrastructure Database Migration, manually configure your source database as described here.

- To configure a single-tenant (Non CDB) as a source for migration, run the following SQL commands:

```
-- Archive Log Mode
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
ALTER DATABASE ARCHIVELOG;
ALTER DATABASE OPEN;

-- Global Names
ALTER SYSTEM SET GLOBAL_NAMES=FALSE;

-- Stream Pool Size
ALTER SYSTEM SET STREAMS_POOL_SIZE=256M;

-- Force Logging
ALTER DATABASE FORCE LOGGING;

-- Supplemental Logging
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;

-- User system for Datapump
ALTER USER SYSTEM IDENTIFIED BY system_pwd ACCOUNT UNLOCK;
```

- To configure a multi-tenant (CDB) as a source for migration, run the following SQL commands:

```
-- Connect to CDB and run:

-- Archive Log Mode
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
ALTER DATABASE ARCHIVELOG;
ALTER DATABASE OPEN;

-- Global Names
ALTER SYSTEM SET GLOBAL_NAMES=FALSE;
```

```
-- Stream Pool Size
ALTER SYSTEM SET STREAMS_POOL_SIZE=256M SCOPE=BOTH;

-- Force Logging
ALTER DATABASE FORCE LOGGING;

-- Supplemental Logging
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;

-- User system for Datapump
ALTER USER SYSTEM IDENTIFIED BY system_pwd ACCOUNT UNLOCK;
ALTER SESSION SET CONTAINER=PDB;
ALTER USER SYSTEM ACCOUNT UNLOCK;
```

- To configure Amazon RDS (non-CDB) as a source for migration, run the following SQL commands:

```
-- Remember to set the following parameters thru the Parameter groups
functionality:
-- STREAMS_POOL_SIZE=2147483648
-- GLOBAL_NAMES=FALSE
-- To see how Parameter groups work refer to https://docs.aws.amazon.com/
AmazonRDS/latest/UserGuide/parameter-groups-overview.html

-- Archive Log Mode
EXEC RDSADMIN.RDSADMIN_UTIL.SET_CONFIGURATION('ARCHIVELOG RETENTION
HOURS',72);

-- Force Logging
EXEC RDSADMIN.RDSADMIN_UTIL.FORCE_LOGGING(P_ENABLE => TRUE);

-- Supplemental Logging
EXEC RDSADMIN.RDSADMIN_UTIL.ALTER_SUPPLEMENTAL_LOGGING('ADD');
```

- Additional configurations
  Before you can migrate data with Oracle Cloud Infrastructure Database Migration, perform additional configurations for your source database for online migration as described here.
- Use case
  Following is a sample use case to prepare your source database for migration. To configure a PDB as a source for your migration, the steps are similar to setting up a classic database as a source, but there are requirements for using the `CDBROOT` as `ggaliassrc`.

Additional configurations

Before you can migrate data with Oracle Cloud Infrastructure Database Migration, perform additional configurations for your source database for online migration as described here.

- To configure a single-tenant (Non CDB) as a source for online migration, run the following SQL commands:

```
-- Archive Log Mode
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
ALTER DATABASE ARCHIVELOG;
```

```
ALTER DATABASE OPEN;

-- Global Names
ALTER SYSTEM SET GLOBAL_NAMES=FALSE;

-- Stream Pool Size
ALTER SYSTEM SET STREAMS_POOL_SIZE=2G;

-- Force Logging
ALTER DATABASE FORCE LOGGING;

-- Enable GoldenGate
ALTER SYSTEM SET ENABLE_GOLDENGATE_REPLICATION=TRUE;

-- Supplemental Logging
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;

-- User system for Datapump
ALTER USER SYSTEM IDENTIFIED BY system_pwd ACCOUNT UNLOCK;

-- Create GoldenGate nonCDB user
CREATE TABLESPACE GG_ADMIN DATAFILE '+DATA/ggadmin_data.dbf' SIZE 100m
AUTOEXTEND ON NEXT 100m;
CREATE USER GGADMIN IDENTIFIED BY ggadmin_pwd DEFAULT TABLESPACE GG_ADMIN
TEMPORARY TABLESPACE TEMP QUOTA 100M ON GG_ADMIN;
GRANT CONNECT TO GGADMIN;
GRANT RESOURCE TO GGADMIN;
GRANT CREATE SESSION TO GGADMIN;
GRANT SELECT_CATALOG_ROLE TO GGADMIN;
GRANT DV_GOLDENGATE_ADMIN TO GGADMIN;
GRANT DV_GOLDENGATE_REDO_ACCESS TO GGADMIN;
GRANT ALTER SYSTEM TO GGADMIN;
GRANT ALTER USER TO GGADMIN;
GRANT DATAPUMP_EXP_FULL_DATABASE TO GGADMIN;
GRANT DATAPUMP_IMP_FULL_DATABASE TO GGADMIN;
GRANT SELECT ANY DICTIONARY TO GGADMIN;
GRANT SELECT ANY TRANSACTION TO GGADMIN;
GRANT INSERT ANY TABLE TO GGADMIN;
GRANT UPDATE ANY TABLE TO GGADMIN;
GRANT DELETE ANY TABLE TO GGADMIN;
GRANT LOCK ANY TABLE TO GGADMIN;
GRANT CREATE ANY TABLE TO GGADMIN;
GRANT CREATE ANY INDEX TO GGADMIN;
GRANT CREATE ANY CLUSTER TO GGADMIN;
GRANT CREATE ANY INDEXTYPE TO GGADMIN;
GRANT CREATE ANY OPERATOR TO GGADMIN;
GRANT CREATE ANY PROCEDURE TO GGADMIN;
GRANT CREATE ANY SEQUENCE TO GGADMIN;
GRANT CREATE ANY TRIGGER TO GGADMIN;
GRANT CREATE ANY TYPE TO GGADMIN;
GRANT CREATE ANY SEQUENCE TO GGADMIN;
GRANT CREATE ANY VIEW TO GGADMIN;
GRANT ALTER ANY TABLE TO GGADMIN;
GRANT ALTER ANY INDEX TO GGADMIN;
GRANT ALTER ANY CLUSTER TO GGADMIN;
GRANT ALTER ANY INDEXTYPE TO GGADMIN;
```

```
GRANT ALTER ANY OPERATOR TO GGADMIN;
GRANT ALTER ANY PROCEDURE TO GGADMIN;
GRANT ALTER ANY SEQUENCE TO GGADMIN;
GRANT ALTER ANY TRIGGER TO GGADMIN;
GRANT ALTER ANY TYPE TO GGADMIN;
GRANT ALTER ANY SEQUENCE TO GGADMIN;
GRANT CREATE DATABASE LINK TO GGADMIN;
GRANT EXECUTE ON dbms_lock TO GGADMIN;
EXEC DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE('GGADMIN');
```

- To configure a multi-tenant (CDB) as a source for online migration, run the following SQL commands:

```
-- Connect to CDB and run:

-- Archive Log Mode
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
ALTER DATABASE ARCHIVELOG;
ALTER DATABASE OPEN;

-- Global Names
ALTER SYSTEM SET GLOBAL_NAMES=FALSE;

-- Stream Pool Size
ALTER SYSTEM SET STREAMS_POOL_SIZE=2G SCOPE=BOTH;

-- Force Logging
ALTER DATABASE FORCE LOGGING;

-- Enable GoldenGate
ALTER SYSTEM SET ENABLE_GOLDENGATE_REPLICATION=TRUE SCOPE=BOTH;

-- Supplemental Logging
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;

-- User system for Datapump
ALTER USER SYSTEM IDENTIFIED BY system_pwd ACCOUNT UNLOCK CONTAINER=ALL;

-- Create GoldenGate users
-- CDB user
ALTER SESSION SET CONTAINER = CDB$ROOT;
CREATE USER C##GGADMIN IDENTIFIED BY cggadmin_pwd CONTAINER=ALL DEFAULT
TABLESPACE USERS TEMPORARY TABLESPACE TEMP QUOTA UNLIMITED ON USERS;
GRANT CONNECT TO C##GGADMIN CONTAINER=ALL;
GRANT RESOURCE TO C##GGADMIN CONTAINER=ALL;
GRANT CREATE TABLE TO C##GGADMIN CONTAINER=ALL;
GRANT CREATE VIEW TO C##GGADMIN CONTAINER=ALL;
GRANT CREATE SESSION TO C##GGADMIN CONTAINER=ALL;
GRANT SELECT_CATALOG_ROLE TO C##GGADMIN CONTAINER=ALL;
GRANT DV_GOLDENGATE_ADMIN TO C##GGADMIN CONTAINER=ALL;
GRANT DV_GOLDENGATE_REDO_ACCESS TO C##GGADMIN CONTAINER=ALL;
GRANT ALTER SYSTEM TO C##GGADMIN CONTAINER=ALL;
GRANT ALTER USER TO C##GGADMIN CONTAINER=ALL;
GRANT SELECT ANY DICTIONARY TO C##GGADMIN CONTAINER=ALL;
```

```
GRANT SELECT ANY TRANSACTION TO C##GGADMIN CONTAINER=ALL;
GRANT EXECUTE ON dbms_lock TO C##GGADMIN CONTAINER=ALL;
EXEC
DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE('C##GGADMIN',CONTAINER=>'ALL');

-- PDB User
ALTER SESSION SET CONTAINER = v_pdb_name;
CREATE TABLESPACE GG_ADMIN DATAFILE '+DATA/ggadmin_data.dbf' SIZE 100m
AUTOEXTEND ON NEXT 100m;
CREATE USER GGADMIN IDENTIFIED BY ggadmin_pwd CONTAINER=CURRENT DEFAULT
TABLESPACE GG_ADMIN TEMPORARY TABLESPACE TEMP QUOTA UNLIMITED ON GG_ADMIN;
GRANT CONNECT TO GGADMIN CONTAINER=CURRENT;
GRANT RESOURCE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE SESSION TO GGADMIN CONTAINER=CURRENT;
GRANT SELECT_CATALOG_ROLE TO GGADMIN CONTAINER=CURRENT;
GRANT DV_GOLDENGATE_ADMIN TO GGADMIN CONTAINER=CURRENT;
GRANT DV_GOLDENGATE_REDO_ACCESS TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER SYSTEM TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER USER TO GGADMIN CONTAINER=CURRENT;
GRANT DATAPUMP_EXP_FULL_DATABASE TO GGADMIN CONTAINER=CURRENT;
GRANT DATAPUMP_IMP_FULL_DATABASE TO GGADMIN CONTAINER=CURRENT;
GRANT SELECT ANY DICTIONARY TO GGADMIN CONTAINER=CURRENT;
GRANT SELECT ANY TRANSACTION TO GGADMIN CONTAINER=CURRENT;
GRANT INSERT ANY TABLE TO GGADMIN CONTAINER=CURRENT;
GRANT UPDATE ANY TABLE TO GGADMIN CONTAINER=CURRENT;
GRANT DELETE ANY TABLE TO GGADMIN CONTAINER=CURRENT;
GRANT LOCK ANY TABLE TO GGADMIN CONTAINER=CURRENT;
GRANT SELECT ON V_$SESSION TO GGADMIN CONTAINER=CURRENT;
GRANT SELECT ON V_$TRANSACTION TO GGADMIN CONTAINER=CURRENT;
GRANT SELECT ON V_$DATABASE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY TABLE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY INDEX TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY CLUSTER TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY INDEXTYPE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY OPERATOR TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY PROCEDURE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY SEQUENCE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY TRIGGER TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY TYPE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY SEQUENCE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY VIEW TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY TABLE TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY INDEX TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY CLUSTER TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY INDEXTYPE TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY OPERATOR TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY PROCEDURE TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY SEQUENCE TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY TRIGGER TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY TYPE TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY SEQUENCE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE DATABASE LINK TO GGADMIN CONTAINER=CURRENT;
GRANT EXECUTE ON dbms_lock TO GGADMIN CONTAINER=CURRENT;
EXEC
DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE('GGADMIN',CONTAINER=>'CURRENT');
```

```
-- Privileges to be granted to the Initial Load user.
ALTER SESSION SET CURRENT_SCHEMA = GGADMIN;
GRANT CREATE ANY TABLE TO <INITIAL_LOAD_USERNAME> WITH ADMIN OPTION;
GRANT ALTER ANY TABLE TO <INITIAL_LOAD_USERNAME> WITH ADMIN OPTION;
GRANT DROP ANY TABLE TO <INITIAL_LOAD_USERNAME> WITH ADMIN OPTION;
GRANT CREATE ANY TRIGGER TO <INITIAL_LOAD_USERNAME> WITH ADMIN OPTION;
GRANT DROP ANY TRIGGER TO <INITIAL_LOAD_USERNAME> WITH ADMIN OPTION;
GRANT ALTER ANY TRIGGER TO <INITIAL_LOAD_USERNAME> WITH ADMIN OPTION;
GRANT DELETE ANY TABLE TO <INITIAL_LOAD_USERNAME> WITH ADMIN OPTION;
GRANT INSERT ANY TABLE TO <INITIAL_LOAD_USERNAME> WITH ADMIN OPTION;
GRANT UPDATE ANY TABLE TO <INITIAL_LOAD_USERNAME> WITH ADMIN OPTION;
-- Create EVENT_TABLE for GGADMIN user
-- Table EVENT_TABLE is required to handle the switchover during the online
migration.
CREATE TABLE GGADMIN.EVENT_TABLE (
    event_pk             NUMBER,
    event_desc           VARCHAR2(200),
    src_event_date       DATE,
    trg_received_date    DATE,
    src_commit_scn       NUMBER,
    src_commit_timestamp DATE,
    session_name         VARCHAR2(200),
    session_serial#      NUMBER,
    session_process      VARCHAR2(50),
    session_program      VARCHAR2(90),
    current_scn          NUMBER,
    number_of_open_txn   NUMBER,
    oldest_open_txn_scn  NUMBER
);
CREATE OR REPLACE TRIGGER GGADMIN.TRG_INSERT_EVENT_TABLE
BEFORE INSERT ON GGADMIN.EVENT_TABLE
FOR EACH ROW
DECLARE
    v_code  NUMBER;
    v_errm  VARCHAR2(64);
BEGIN
    :NEW.src_event_date := SYSDATE;
    :NEW.session_name := USER;
   SELECT sys_context('USERENV','SID') INTO :NEW.session_serial# FROM dual;
   SELECT min(start_scnb) INTO :NEW.oldest_open_txn_scn FROM v$transaction;
   SELECT count(*) INTO :NEW.number_of_open_txn FROM v$transaction;
   SELECT current_scn INTO :NEW.current_scn FROM v$database;
   SELECT process INTO :NEW.session_process FROM v$session a
       WHERE a.SID = sys_context('USERENV','SID');
   SELECT program INTO :NEW.session_program FROM v$session a
       WHERE a.SID = sys_context('USERENV','SID');

   EXCEPTION
      WHEN OTHERS THEN
        v_code := SQLCODE;
        v_errm := SUBSTR(SQLERRM, 1, 64);
        DBMS_OUTPUT.PUT_LINE('Error on GGADMIN.TRG_INSERT_EVENT_TABLE');
        DBMS_OUTPUT.PUT_LINE(v_code || ': ' || v_errm);
        RAISE;
END;
```

- To configure Amazon RDS (non-CDB) as a source for online migration, run the following SQL commands:

```
-- Remember to set the following parameters thru the Parameter groups
functionality:
-- STREAMS_POOL_SIZE=2147483648
-- ENABLE_GOLDENGATE_REPLICATION=TRUE
-- GLOBAL_NAMES=FALSE
-- To see how Parameter groups work refer to https://docs.aws.amazon.com/
AmazonRDS/latest/UserGuide/parameter-groups-overview.html

-- Archive Log Mode
EXEC RDSADMIN.RDSADMIN_UTIL.SET_CONFIGURATION('ARCHIVELOG RETENTION
HOURS',72);

-- Force Logging
EXEC RDSADMIN.RDSADMIN_UTIL.FORCE_LOGGING(P_ENABLE => TRUE);

-- Supplemental Logging
EXEC RDSADMIN.RDSADMIN_UTIL.ALTER_SUPPLEMENTAL_LOGGING('ADD');

-- Create GoldenGate user
CREATE USER GGADMIN IDENTIFIED BY ggadmin_pwd DEFAULT TABLESPACE USERS
TEMPORARY TABLESPACE TEMP QUOTA 100M ON USERS;
GRANT UNLIMITED TABLESPACE TO GGADMIN;
GRANT CONNECT, RESOURCE TO GGADMIN;
GRANT SELECT ANY DICTIONARY TO GGADMIN;
GRANT CREATE VIEW TO GGADMIN;
GRANT EXECUTE ON DBMS_LOCK TO GGADMIN;
GRANT SELECT ON SYS.CCOL$ TO GGADMIN;
GRANT SELECT ON SYS.CDEF$ TO GGADMIN;
GRANT SELECT ON SYS.COL$ TO GGADMIN;
GRANT SELECT ON SYS.CON$ TO GGADMIN;
GRANT SELECT ON SYS.DEFERRED_STG$ TO GGADMIN;
GRANT SELECT ON SYS.ICOL$ TO GGADMIN;
GRANT SELECT ON SYS.IND$ TO GGADMIN;
GRANT SELECT ON SYS.LOB$ TO GGADMIN;
GRANT SELECT ON SYS.LOBFRAG$ TO GGADMIN;
GRANT SELECT ON SYS.OBJ$ TO GGADMIN;
GRANT SELECT ON SYS.SEG$ TO GGADMIN;
GRANT SELECT ON SYS.TAB$ TO GGADMIN;
GRANT SELECT ON SYS.TABCOMPART$ TO GGADMIN;
GRANT SELECT ON SYS.TABPART$ TO GGADMIN;
GRANT SELECT ON SYS.TABSUBPART$ TO GGADMIN;
EXEC RDSADMIN.RDSADMIN_DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE
(GRANTEE=>'GGADMIN',PRIVILEGE_TYPE=>'CAPTURE',GRANT_SELECT_PRIVILEGES=>TRUE
,DO_GRANTS=>TRUE);
```

Use case

Following is a sample use case to prepare your source database for migration. To configure a PDB as a source for your migration, the steps are similar to setting up a classic database as a source, but there are requirements for using the `CDBROOT` as `ggaliassrc`.

The steps differ slightly if you're using a PDB as your source database, so make sure you follow the recommendations if your database is in a multitenant environment.

1. Configure the streams pool with the initialization parameter `STREAMS_POOL_SIZE`.
   - For offline logical migrations, for optimal Data Pump performance, it is required that you set `STREAMS_POOL_SIZE` to a minimum of 256MB-350MB, to have an initial pool allocated, otherwise you might see a significant delay during start up.
   - For online logical migrations, set `STREAMS_POOL_SIZE` to at least 2GB.

     For the explanation of 1GB `STREAMS_POOL_SIZE` per integrated extract + additional 25 percent recommendation, see Integrated Extract / Replicat and STREAMS_POOL_SIZE (Doc ID 2078459.1).
2. Check the `GLOBAL_NAMES` parameter. If it's set to `true`, change it to `false`.

   ```
   sqlplus > show parameter global
   NAME                                 TYPE    VALUE
   ------------------------------------ ------- ------------------------------
   global_names                         boolean TRUE

   sqlplus > alter system set global_names=false
   ```

3. Enable `ARCHIVELOG` if it is not already enabled.
   a. Check whether `archivelog` is enabled:

      ```
      sqlplus > archive log list
      ```

      Sample output returned:

      ```
      Database log mode Archive log Mode
      Automatic archival Enabled
      Archive destination USE_DB_RECOVERY_FILE_DEST
      Oldest online log sequence 33
      Next log sequence to archive 35
      Current log sequence 35
      ```

   b. Enable `archivelog` mode:

      ```
      sqlplus > shutdown immediate
      sqlplus > startup mount
      sqlplus > alter database archivelog;
      sqlplus > alter database open;
      ```

   c. Disable `archivelog` mode (for clean up later)

      ```
      sqlplus > shutdown immediate
      sqlplus > startup mount
      ```

```
sqlplus > alter database noarchivelog;
sqlplus > alter database open;
```

4. Enable logging:
   a. Check if logging is enabled:

      ```
      sqlplus > SELECT supplemental_log_data_min, force_logging FROM
      v$database;
      ```

   b. Enable logging:

      ```
      sqlplus > ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
      sqlplus > ALTER DATABASE FORCE LOGGING;
      ```

   c. Disable logging (for cleanup later)

      ```
      sqlplus > ALTER DATABASE DROP SUPPLEMENTAL LOG DATA;
      sqlplus > ALTER DATABASE NO FORCE LOGGING;
      ```

5. Create a database administrator user that has full Oracle Data Pump privileges for initial load to be performed. A user that has the `DATAPUMP_EXP_FULL_DATABASE` role is required for the export operation at the source database. This user is selected as database administrator when you create Database connections with the source databases.

   See Oracle Data Pump in the *Oracle Database Utilities* guide for more information.

6. In the PDB being exported, if there is any dependency created on local objects in the `C##` user's schema, then they would fail to be imported in the target Autonomous Database. Exclude the problematic schema from the migration job.
7. If you are using Object Storage as a data transfer medium, ensure that an export Directory Object exists and is usable by Data Pump to store generated dump files.
   - The directory object is a file path on the source database server file system. The name needs to comply with Oracle Database directory object rules. See CREATE DIRECTORY in *Oracle Database SQL Language Reference* for details.
   - The export Directory Object must be owned by same OS user who owns the database Oracle home.
   - This step is not required if you are using a database link transfer medium.
8. If you plan to transfer data using a database link, then you must set up SSL encryption on the source database. Using Data Pump with a database link to an Autonomous Database target requires that the source database have SSL encryption enabled. Creating a database link from an Autonomous Database Shared Infrastructure target to a source database with a private IP requires assistance from Oracle Support.

   See Configuring Transport Layer Security Authentication in *Oracle Database Security Guide* for more information.

9. For online logical migrations, if you plan to run migrations with replication, enable GoldenGate Replication:
   a. In a multitenant environment, if you are migrating a PDB, enable GoldenGate Replication on the CDB.

      ```
      sqlplus > ALTER SYSTEM SET ENABLE_GOLDENGATE_REPLICATION=TRUE
      SCOPE=BOTH;
      ```

b. Apply the mandatory RDBMS patches on the source database, based on your source database version:

- **Oracle Database 11.2:**

  My Oracle Support note Oracle GoldenGate -- Oracle RDBMS Server Recommended Patches (Doc ID 1557031.1) recommends the following updates:

  Database PSU 11.2.0.4.210720 includes a fix for Oracle GoldenGate performance bug 28849751 - IE PERFORMANCE DEGRADES WHEN NETWORK LATENCY BETWEEN EXTRACT AND CAPTURE IS MORE THAN 8MS

  OGG RDBMS patch 32248879 MERGE REQUEST ON TOP OF DATABASE PSU 11.2.0.4.201020 FOR BUGS 32048478 20448066 - This patch contains mandatory fix for Oracle GoldenGate Microservices bug 20448066 DBMS_XSTREAM_GG APIS SHOULD BE ALLOWED FOR SCA PROCESSES

- **Oracle Database 12.1.0.2 or later**

  My Oracle Support note Latest GoldenGate/Database (OGG/RDBMS) Patch recommendations (Doc ID 2193391.1) lists the additional RDBMS patches needed on top of the latest DBBP/RU for Oracle Database 12c and later if using Oracle GoldenGate.

Preparing the target database for migration

Before you can migrate data with Oracle Cloud Infrastructure Database Migration, manually configure your target database as described here.

- To configure an Autonomous database as a target for migration, run the following SQL commands:

```
-- Global Names
ALTER SYSTEM SET GLOBAL_NAMES=FALSE;
```

- To configure a non-Autonomous, single-tenant (non-CDB) as a target for migration, run the following SQL commands:

```
-- Global Names
ALTER SYSTEM SET GLOBAL_NAMES=FALSE;

-- User system for Datapump
ALTER USER SYSTEM IDENTIFIED BY system_pwd ACCOUNT UNLOCK;
```

- To configure a non-Autonomous, multi-tenant (CDB) as a target for migration, run the following SQL commands:

```
-- Connect to CDB and run:

-- Global Names
ALTER SYSTEM SET GLOBAL_NAMES=FALSE;

-- User system for Datapump
ALTER USER SYSTEM IDENTIFIED BY system_pwd ACCOUNT UNLOCK SCOPE=BOTH;
```

- Additional configurations
  Before you can migrate data with Oracle Cloud Infrastructure Database Migration, perform additional configurations for your target database for online migration as described here.
- Use case
  Following is a sample use case for preparing a target database for migration.

Additional configurations

Before you can migrate data with Oracle Cloud Infrastructure Database Migration, perform additional configurations for your target database for online migration as described here.

You must unlock the `ggadmin` user from the Oracle Cloud Infrastructure Console by performing the following steps:

1. Follow step 1 through step 3 mentioned in Manage Users and User Roles on Autonomous Database - Connecting with Database Actions.
2. Turn off the **Account is locked** toggle.
3. Provide a password with its corresponding confirmation.

- Alternatively, to configure an Autonomous database as a target for online migration, run the following SQL commands:

```
-- Global Names
ALTER SYSTEM SET GLOBAL_NAMES=FALSE;

-- Create GoldenGate user if doesn't exist
CREATE TABLESPACE GG_ADMIN DATAFILE '+DATA/ggadmin_data.dbf' SIZE 100m
AUTOEXTEND ON NEXT 100m;
CREATE USER GGADMIN IDENTIFIED BY ggadmin_pwd CONTAINER=CURRENT DEFAULT
TABLESPACE GG_ADMIN TEMPORARY TABLESPACE TEMP QUOTA UNLIMITED ON GG_ADMIN;

-- Or unlock it if exists
ALTER USER GGADMIN IDENTIFIED BY ggadmin_pwd ACCOUNT UNLOCK;
```

- To configure a non-Autonomous, single-tenant (non-CDB) as a target for online migration, run the following SQL commands:

```
-- Global Names
ALTER SYSTEM SET GLOBAL_NAMES=FALSE;

-- User system for Datapump
ALTER USER SYSTEM IDENTIFIED BY system_pwd ACCOUNT UNLOCK;

-- Create GoldenGate nonCDB user
CREATE TABLESPACE GG_ADMIN DATAFILE '+DATA/ggadmin_data.dbf' SIZE 100m
AUTOEXTEND ON NEXT 100m;
CREATE USER GGADMIN IDENTIFIED BY ggadmin_pwd DEFAULT TABLESPACE GG_ADMIN
TEMPORARY TABLESPACE TEMP QUOTA 100M ON GG_ADMIN;
GRANT CONNECT TO GGADMIN;
GRANT RESOURCE TO GGADMIN;
GRANT CREATE SESSION TO GGADMIN;
GRANT SELECT_CATALOG_ROLE TO GGADMIN;
GRANT DV_GOLDENGATE_ADMIN TO GGADMIN;
GRANT DV_GOLDENGATE_REDO_ACCESS TO GGADMIN;
GRANT ALTER SYSTEM TO GGADMIN;
```

```
GRANT ALTER USER TO GGADMIN;
GRANT DATAPUMP_EXP_FULL_DATABASE TO GGADMIN;
GRANT DATAPUMP_IMP_FULL_DATABASE TO GGADMIN;
GRANT SELECT ANY DICTIONARY TO GGADMIN;
GRANT SELECT ANY TRANSACTION TO GGADMIN;
GRANT INSERT ANY TABLE TO GGADMIN;
GRANT UPDATE ANY TABLE TO GGADMIN;
GRANT DELETE ANY TABLE TO GGADMIN;
GRANT LOCK ANY TABLE TO GGADMIN;
GRANT CREATE ANY TABLE TO GGADMIN;
GRANT CREATE ANY INDEX TO GGADMIN;
GRANT CREATE ANY CLUSTER TO GGADMIN;
GRANT CREATE ANY INDEXTYPE TO GGADMIN;
GRANT CREATE ANY OPERATOR TO GGADMIN;
GRANT CREATE ANY PROCEDURE TO GGADMIN;
GRANT CREATE ANY SEQUENCE TO GGADMIN;
GRANT CREATE ANY TRIGGER TO GGADMIN;
GRANT CREATE ANY TYPE TO GGADMIN;
GRANT CREATE ANY SEQUENCE TO GGADMIN;
GRANT CREATE ANY VIEW TO GGADMIN;
GRANT ALTER ANY TABLE TO GGADMIN;
GRANT ALTER ANY INDEX TO GGADMIN;
GRANT ALTER ANY CLUSTER TO GGADMIN;
GRANT ALTER ANY INDEXTYPE TO GGADMIN;
GRANT ALTER ANY OPERATOR TO GGADMIN;
GRANT ALTER ANY PROCEDURE TO GGADMIN;
GRANT ALTER ANY SEQUENCE TO GGADMIN;
GRANT ALTER ANY TRIGGER TO GGADMIN;
GRANT ALTER ANY TYPE TO GGADMIN;
GRANT ALTER ANY SEQUENCE TO GGADMIN;
GRANT CREATE DATABASE LINK TO GGADMIN;
GRANT EXECUTE ON dbms_lock TO GGADMIN;
EXEC DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE('GGADMIN');
```

- To configure a non-Autonomous, multi-tenant (CDB) as a target for online migration, run the following SQL commands:

```
-- Connect to CDB and run:

-- Global Names
ALTER SYSTEM SET GLOBAL_NAMES=FALSE;

-- User system for Datapump
ALTER USER SYSTEM IDENTIFIED BY system_pwd ACCOUNT UNLOCK CONTAINER=ALL;

-- Create GoldenGate PDB User
ALTER SESSION SET CONTAINER = v_pdb_name;
CREATE TABLESPACE GG_ADMIN DATAFILE '+DATA/ggadmin_data.dbf' SIZE 100m
AUTOEXTEND ON NEXT 100m;
CREATE USER GGADMIN IDENTIFIED BY ggadmin_pwd CONTAINER=CURRENT DEFAULT
TABLESPACE GG_ADMIN TEMPORARY TABLESPACE TEMP QUOTA UNLIMITED ON GG_ADMIN;
GRANT CONNECT TO GGADMIN CONTAINER=CURRENT;
GRANT RESOURCE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE SESSION TO GGADMIN CONTAINER=CURRENT;
GRANT SELECT_CATALOG_ROLE TO GGADMIN CONTAINER=CURRENT;
```

```
GRANT DV_GOLDENGATE_ADMIN TO GGADMIN CONTAINER=CURRENT;
GRANT DV_GOLDENGATE_REDO_ACCESS TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER SYSTEM TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER USER TO GGADMIN CONTAINER=CURRENT;
GRANT DATAPUMP_EXP_FULL_DATABASE TO GGADMIN CONTAINER=CURRENT;
GRANT DATAPUMP_IMP_FULL_DATABASE TO GGADMIN CONTAINER=CURRENT;
GRANT SELECT ANY DICTIONARY TO GGADMIN CONTAINER=CURRENT;
GRANT SELECT ANY TRANSACTION TO GGADMIN CONTAINER=CURRENT;
GRANT INSERT ANY TABLE TO GGADMIN CONTAINER=CURRENT;
GRANT UPDATE ANY TABLE TO GGADMIN CONTAINER=CURRENT;
GRANT DELETE ANY TABLE TO GGADMIN CONTAINER=CURRENT;
GRANT LOCK ANY TABLE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY TABLE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY INDEX TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY CLUSTER TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY INDEXTYPE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY OPERATOR TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY PROCEDURE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY SEQUENCE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY TRIGGER TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY TYPE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY SEQUENCE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE ANY VIEW TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY TABLE TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY INDEX TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY CLUSTER TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY INDEXTYPE TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY OPERATOR TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY PROCEDURE TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY SEQUENCE TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY TRIGGER TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY TYPE TO GGADMIN CONTAINER=CURRENT;
GRANT ALTER ANY SEQUENCE TO GGADMIN CONTAINER=CURRENT;
GRANT CREATE DATABASE LINK TO GGADMIN CONTAINER=CURRENT;
GRANT EXECUTE ON dbms_lock TO GGADMIN CONTAINER=CURRENT;
EXEC
DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE('GGADMIN',CONTAINER=>'CURRENT');
```

Use case

Following is a sample use case for preparing a target database for migration.

1. Create an Autonomous Database. If the target autonomous database is already present you can skip this step.
2. Check the `GLOBAL_NAMES` parameter. If it's set to `true`, change it to `false`.

```
sqlplus > show parameter global
NAME                                 TYPE    VALUE
------------------------------------ ------- ------------------------------
global_names                         boolean TRUE

sqlplus > alter system set global_names=false
```

3. Create a database administrator user that has full Oracle Data Pump privileges for initial load to be performed. A user that has the `DATAPUMP_IMP_FULL_DATABASE` role is required for the export operation at the target database. This user is selected as database administrator when you create Database connections with the target databases.

   See Oracle Data Pump in the *Oracle Database Utilities* guide for more information.

Preparing MySQL databases for migration

If you want to perform online MySQL migrations, prepare your source and target databases for replication as follows:

- Source database requirements:
  Depending on your source server configuration, following modes are supported:
  - Single server: To enable binary logging, see The Binary Log.
  - Multiple replica servers: To setup replication using GTIDs, see Setting Up Replication Using GTIDs.

> **Note**
>
> The above steps are applicable only when you perform an online migration.

- Target database requirements:

> **Note**
>
> For online migrations, if you define a different replication user, then ensure that it has the privileges defined here. However, if you do not define the user, ensure that the main user for the connection has these privileges.

Related Topics

- Requirements and Restrictions

5

Create connections

Oracle Cloud Infrastructure Database Migration database connection resources contain the connectivity details of the migration source and target databases.

- Listing connections
  View the connections for a specific compartment. You have the options to search and filter the connections, sort, and apply filters on the table directly.
- Creating Oracle connections
  Oracle Cloud Infrastructure Database Migration database connection resources contain the connectivity details of the migration source and target databases. Create connection database resources in the Database Migration Databases Connections page for the following databases:
- Creating MySQL connections
  Create MySQL connections to connect to the MySQL databases.

Listing connections

View the connections for a specific compartment. You have the options to search and filter the connections, sort, and apply filters on the table directly.

1. Open the navigation menu, select **Migration & Disaster Recovery**. Under **Migration & Disaster Recovery** select **Database Migration**.
2. Under **Database Migration**, select **Database Connections**.

   The **Database connections** page opens. All the **Database connections** in the selected compartment are displayed in a table.
3. To view the **Database Connections** in a different compartment, use the **Compartment** filter to switch compartments.

> **Note**
>
> You must have permission to work in a compartment to see the resources in it. If you're not sure which compartment to use, contact an administrator. For more information, see Understanding Compartments.

Filtering List Results
Use filters to limit the **Database connections** in the list. Perform one of the following actions depending on the options that you see:

- From the **Search and Filter** box above the list table, select one or more filters and specify the values that you want to use to narrow the list. In general, the filters correspond to the columns shown in the list table, although some filters represent attributes that aren't shown in the table. The **Compartment** filter is always displayed next to **Applied filters**.
- On the left side of the list page, select a value from one of the available filters, such as compartment, state, or tags.

Change the order of the items in the list table by using the sort icons next to the column names.

For information about searching for resources and managing the columns in the list table, if those features are available, see Listing Resources.

Actions
In the list table, select the name of a **Database connection** to open its details page, where you can view its status and perform other tasks.

To perform an action on a **Database connection** directly from the list table, select an available option from the Actions menu in the row for that **Database connection**:

- **View details**: Viewing connection details.
- **Move resource**: Move the Database connection to another compartment.
- **Copy OCID**: Copy the OCID of the **Database connection** to the clipboard.
- **Manage tags**: Add one or more tags to the **Database connection**, or change the value of an existing tag. See Resource Tags.
- **Test connection**: Testing connectivity of a database connection
- **Delete**: Deleting a connection.

To create a **Database connection**, select Create connections.

Creating Oracle connections

Oracle Cloud Infrastructure Database Migration database connection resources contain the connectivity details of the migration source and target databases. Create connection database resources in the Database Migration Databases Connections page for the following databases:

- Oracle Autonomous Database
- Oracle Database
- Oracle Exadata
- Amazon RDS for Oracle

> **Note**
>
> Oracle Cloud Infrastructure Database Migration service runs network connectivity tests followed by database login tests (to validate credentials) using the information that you provided while creating database connections. See Testing Connectivity of a Database Connection.

> **Note**
>
> If the source database is a multitenant container database (CDB), you are performing an online migration, and if you have a pluggable database as a source with version earlier than Oracle Database 21c, you must create two connection entries for the source database: one for the PDB and one for the CDB. For offline migrations only the PDB connection is required.
> You will create only one database connection resource for the target database.

- Creating an Oracle Autonomous Database connection
  Create an Oracle Autonomous Database connection by performing the following steps:
- Creating an Oracle Database connection
  Create an Oracle Database connection by performing the following steps:
- Creating an Oracle Exadata connection
  Create an Oracle Exadata connection by performing the following steps:
- Creating an Amazon RDS for Oracle connection
  Create an Amazon RDS for Oracle connection by performing the following steps:

Creating an Oracle Autonomous Database connection

Create an Oracle Autonomous Database connection by performing the following steps:

1. On the **Database connections** list page, select **Create connection**. If you need help finding the list page, see **Listing connections**.
2. On the General information step, in the **Name** field, enter a display name for the database connection resource.

   This is not the actual database name, but a name that will appear in a list of databases connections on completion of this procedure. Note that the database connection resources for all source databases (CDB and PDB) and target databases will appear the same list.
3. Add **Description** for the database connection resource.
4. In **Compartment** select the compartment in which the Database Connection resource will be created.
5. In **Type**, select a connection type. Select the connection type as `Oracle Autonomous Database` .
6. In **Vault in *Compartment*** select the security vault.

   Database Migration uses the OCI Vault to store user secrets such as passwords, wallets, and keys, and encrypts them with the user-supplied encryption key.

   You can select a vault in a different compartment by clicking **Change Compartment**.
7. In **Encryption Key in *Compartment*** select the keystore that you configured in the vault.

   Only AES algorithm type keys are supported.
8. The **Advanced Options** section allows you to optionally create tags.
9. In the **Connection details** step, enter the connection details for the Oracle Autonomous Database. Provide the following information:
   - **Database name:** The database name
   - Enter the database administrator credentials in **Initial load database username** and **Initial load database password**.
10. Select **Use different credentials for replication** to enter the credentials for replication. Select this option if you want to use a separate database user for performing replication for your online migrations. The following options are available
    - Replication database username
    - Replication database password
11. In **Network Connectivity** details, **Create private endpoint to access this database** indicates whether the database is publicly accessible or if you want to create a private endpoint.

When you select the **Create private endpoint to access this database** option, then the **Subnet in *Compartment*** option lets you select the subnet containing the appropriate Virtual Cloud Network (VCN).

This is the subnet to which a private endpoint is created from the Oracle Cloud Infrastructure Database Migration service tenancy. This creates a network route for the Oracle Cloud Infrastructure Database Migration deployment to connect to the database within your customer tenancy.

Select the subnet containing the appropriate Virtual Cloud Network (VCN), then click **Next**. Click **Change Compartment** to select a subnet in a different compartment.

12. Click **Create**.

> **Note**
>
> After you click **Create**, the database connection name appears in the Connection list while the creating connection operation runs. The creating connection operation can take a few minutes.

You can monitor the operation status in the State column. When the state is **Active**, the database connection creation is complete and successful.

When the resource creation is complete and successful, check the Security Vault service to verify that the SSH private key file was uploaded and enabled in the vault you configured.

Creating an Oracle Database connection

Create an Oracle Database connection by performing the following steps:

1. On the **Database connections** list page, select **Create connection**. If you need help finding the list page, see **Listing connections**.
2. On the General information step, in the **Name** field, enter a display name for the database connection resource.

   This is not the actual database name, but a name that will appear in a list of databases connections on completion of this procedure. Note that the database connection resources for all source databases (CDB and PDB) and target databases will appear the same list.
3. Add **Description** for the database connection resource.
4. In **Compartment** select the compartment in which the Database Connection resource will be created.
5. In **Type**, select a connection type. Select the connection type as `Oracle Database`.
6. In **Vault in *Compartment*** select the security vault.

   Database Migration uses the OCI Vault to store user secrets such as passwords, wallets, and keys, and encrypts them with the user-supplied encryption key.

   You can select a vault in a different compartment by clicking **Change Compartment**.
7. In **Encryption Key in *Compartment*** select the keystore that you configured in the vault.

   Only AES algorithm type keys are supported.
8. The **Advanced Options** section allows you to optionally create tags.
9. In the **Connection details** step, enter the database details for the Oracle Database. Select one of the following options:

- Select database and provide the following information:
  - **Database System in *Compartment*:** Select a database.
  - **Database home:** The database home

    Database: The database name

    Pluggable Database: (Optional):

    Select the Pluggable Database.

    > **Note**
    >
    > If you have a database earlier than Oracle Database 19c (multiple instance), then you must create your PDB connection by providing the connection details.

  - Enter the database administrator credentials in **Initial load database username** and **Initial load database password**.
- Enter the database information and provide the following information:
  - **Database connection string:** Provide the full connect string with host, port, and service name, in the following format:

    ```
    host:port/db-service-name
    ```

    If a private endpoint is specified in the connection, the `host`entry should be a valid IP address.
  - Enter the database administrator credentials in **Initial load database username** and **Initial load database password**.

10. Select **Use different credentials for replication** to enter the credentials for replication. Select this option if you want to use a separate database user for performing replication for your online migrations. The following options are available
    - Replication database username
    - Replication database password
11. Optionally, upload the **Database wallet** drag and drop or select the wallet.zip for this database.

    > **Note**
    >
    > Optionally, upload your `cwallet.sso` file to use it as the wallet.

12. In **Network Connectivity** details, **Create private endpoint to access this database** indicates whether the database is publicly accessible or if you want to create a private endpoint.

    In **Subnet in *Compartment***, select the subnet to which a private endpoint is created from the Oracle Cloud Infrastructure Database Migration service tenancy. This creates a network route for the Oracle Cloud Infrastructure Database Migration deployment to connect to the database within your customer tenancy. Select the subnet containing the appropriate Virtual Cloud Network (VCN). Click **Change Compartment** to select a subnet in a different compartment.

> **Note**
>
> Oracle strongly recommends using the same subnet for both CDB and PDB connections.

13. (Optional) Under **Advanced Options**, enter the details for **SSH settings** if you prefer a Secure Shell (SSH) connection. Provide the SSH information for your database hosts if you wish to provide SSH access to the service to perform the migrations. Provide the SSH related information as follows:

> **Note**
>
> Ensure that the private SSH key file is an RSA key in PEM format. See Required Keys and OCIDs for more information.

- **SSH database server hostname**: Enter the IP address of the database host. This will be used by the service to connect to your host through/via SSH to perform the migration. Select the valid private key file used for database host access.

> **Note**
>
> Enter a valid SSH username that will be used by the service to create a ssh session to the database host. This user should have the `sudo` privilege to perform the necessary operations.

- **SSH private key**:
  Drop or select the private key file used to access the database server host.
- **SSH username**: Enter an OS user name for the database host. This user must be a privileged user allowed to run `sudo`.
- **SSH sudo location**: Enter the `sudo` binary location on the database host.

> **Note**
>
> If you do not provide SSH details and if you use Object Storage as your transfer medium for the migration, then review how to perform the steps to Download SSL wallet with Certificates and Set up Network ACL.

14. Click **Create**.

> **Note**
>
> After you click **Create**, the database connection name appears in the Connection list while the creating connection operation runs. The creating connection operation can take a few minutes.

You can monitor the operation status in the State column. When the state is **Active**, the database connection creation is complete and successful.

When the resource creation is complete and successful, check the Security Vault service to verify that the SSH private key file was uploaded and enabled in the vault you configured.

Creating an Oracle Exadata connection

Create an Oracle Exadata connection by performing the following steps:

1. On the **Database connections** list page, select **Create connection**. If you need help finding the list page, see **Listing connections**.
2. On the General information step, in the **Name** field, enter a display name for the database connection resource.

   This is not the actual database name, but a name that will appear in a list of databases connections on completion of this procedure. Note that the database connection resources for all source databases (CDB and PDB) and target databases will appear the same list.
3. Add **Description** for the database connection resource.
4. In **Compartment** select the compartment in which the Database Connection resource will be created.
5. In **Type**, select a connection type. Select the connection type as `Oracle Exadata`.
6. In **Vault in *Compartment*** select the security vault.

   Database Migration uses the OCI Vault to store user secrets such as passwords, wallets, and keys, and encrypts them with the user-supplied encryption key.

   You can select a vault in a different compartment by clicking **Change Compartment**.
7. In **Encryption Key in *Compartment*** select the keystore that you configured in the vault.

   Only AES algorithm type keys are supported.
8. The **Advanced Options** section allows you to optionally create tags.
9. In the **Connection details** step, enter the database details for the Oracle Exadata. Select one of the following options:
   - Select database and provide the following information:
     - **VM Cluster in *Compartment*:** Select a VM cluster.
     - **Database:** Select the database.

       Pluggable Database: (Optional):

       Select the Pluggable Database.
     - Enter the database administrator credentials in **Initial load database username** and **Initial load database password**.
   - Enter the database information and provide the following information:
     - **Database connection string:** Provide the full connect string with host, port, and service name, in the following format:

       ```
       host:port/db-service-name
       ```

       If a private endpoint is specified in the connection, the `host`entry should be a valid IP address.
     - Enter the database administrator credentials in **Initial load database username** and **Initial load database password**.

10. Select **Use different credentials for replication** to enter the credentials for replication. Select this option if you want to use a separate database user for performing replication for your online migrations. The following options are available
    - Replication database username
    - Replication database password
11. Optionally, upload the **Database wallet** drag and drop or select the wallet.zip for this database.

    > **Note**
    >
    > The `wallet.zip` must contain the `cwallet.sso` and `tnsnames.ora` files.

12. In **Network Connectivity** details, **Create private endpoint to access this database** indicates whether the database is publicly accessible or if you want to create a private endpoint.

    In **Subnet in *Compartment***, select the subnet to which a private endpoint is created from the Oracle Cloud Infrastructure Database Migration service tenancy. This creates a network route for the Oracle Cloud Infrastructure Database Migration deployment to connect to the database within your customer tenancy. Select the subnet containing the appropriate Virtual Cloud Network (VCN).Click **Change Compartment** to select a subnet in a different compartment.
13. (Optional) Under **Advanced Options**, enter the details for **SSH settings** if you prefer a Secure Shell (SSH) connection. Provide the SSH information for your database hosts if you wish to provide SSH access to the service to perform the migrations. Provide the SSH related information as follows:

    > **Note**
    >
    > Ensure that the private SSH key file is an RSA key in PEM format. See Required Keys and OCIDs for more information.

    - **SSH database server hostname**: Enter the IP address of the database host. This will be used by the service to connect to your host through/via SSH to perform the migration. Select the valid private key file used for database host access.

      > **Note**
      >
      > Enter a valid SSH username that will be used by the service to create a ssh session to the database host. This user should have the `sudo` privilege to perform the necessary operations.

    - **SSH private key**:
      Drop or select the private key file used to access the database server host.
    - **SSH username**: Enter an OS user name for the database host. This user must be a privileged user allowed to run `sudo`.
    - **SSH sudo location**: Enter the `sudo` binary location on the database host.

> **Note**
>
> If you do not provide SSH details and if you use Object Storage as your transfer medium for the migration, then review how to perform the steps to Download SSL wallet with Certificates and Set up Network ACL.

14. Click **Create**.

> **Note**
>
> After you click **Create**, the database connection name appears in the Connection list while the creating connection operation runs. The creating connection operation can take a few minutes.

You can monitor the operation status in the State column. When the state is **Active**, the database connection creation is complete and successful.

When the resource creation is complete and successful, check the Security Vault service to verify that the SSH private key file was uploaded and enabled in the vault you configured.

Creating an Amazon RDS for Oracle connection

Create an Amazon RDS for Oracle connection by performing the following steps:

1. On the **Database connections** list page, select **Create connection**. If you need help finding the list page, see **Listing connections**.
2. On the General information step, in the **Name** field, enter a display name for the database connection resource.

   This is not the actual database name, but a name that will appear in a list of databases connections on completion of this procedure. Note that the database connection resources for all source databases (CDB and PDB) and target databases will appear the same list.
3. Add **Description** for the database connection resource.
4. In **Compartment** select the compartment in which the Database Connection resource will be created.
5. In **Type**, select a connection type. Select the connection type as `Amazon RDS for Oracle`.
6. In **Vault in *Compartment*** select the security vault.

   Database Migration uses the OCI Vault to store user secrets such as passwords, wallets, and keys, and encrypts them with the user-supplied encryption key.

   You can select a vault in a different compartment by clicking **Change Compartment**.
7. In **Encryption Key in *Compartment*** select the keystore that you configured in the vault.

   Only AES algorithm type keys are supported.
8. The **Advanced Options** section allows you to optionally create tags.
9. In the **Connection details** step, enter the database details for Amazon RDS for Oracle. Enter the database information and provide the following information:

- **Database connection string:** Provide the full connect string with host, port, and service name, in the following format:

  ```
  host:port/db-service-name
  ```

  If a private endpoint is specified in the connection, the `host`entry should be a valid IP address.
- Enter the database administrator credentials in **Initial load database username** and **Initial load database password**.

10. Select **Use different credentials for replication** to enter the credentials for replication. Select this option if you want to use a separate database user for performing replication for your online migrations. The following options are available
    - Replication database username
    - Replication database password
11. Optionally, upload the **Database wallet** drag and drop or select the wallet.zip for this database.

    > **Note**
    >
    > The `wallet.zip` must contain the `cwallet.sso` and `tnsnames.ora` files.

12. In **Network Connectivity** details, **Create private endpoint to access this database** indicates whether the database is publicly accessible or if you want to create a private endpoint.

    In **Subnet in *Compartment***, select the subnet to which a private endpoint is created from the Oracle Cloud Infrastructure Database Migration service tenancy. This creates a network route for the Oracle Cloud Infrastructure Database Migration deployment to connect to the database within your customer tenancy. Select the subnet containing the appropriate Virtual Cloud Network (VCN).Click **Change Compartment** to select a subnet in a different compartment.
13. Click **Create**.

> **Note**
>
> After you click **Create**, the database connection name appears in the Connection list while the creating connection operation runs. The creating connection operation can take a few minutes.

You can monitor the operation status in the State column. When the state is **Active**, the database connection creation is complete and successful.

When the resource creation is complete and successful, check the Security Vault service to verify that the SSH private key file was uploaded and enabled in the vault you configured.

Creating MySQL connections

Create MySQL connections to connect to the MySQL databases.

- Creating an OCI HeatWave connection
  Create an OCI HeatWave connection by performing the following steps:
- Creating a MySQL Database Server connection
  Create a MySQL Database Server connection by performing the following steps:
- Creating an Amazon RDS for MySQL connection
  Create an Amazon RDS for MySQL connection by performing the following steps:
- Creating an Azure Database for MySQL connection
  Create an Azure Database for MySQL connection by performing the following steps:
- Creating an Amazon Aurora MySQL connection
  Create an Amazon Aurora MySQL connection by performing the following steps:
- Creating a Google Cloud SQL for MySQL connection
  Create a Google Cloud SQL for MySQL connection by performing the following steps:

Related Topics

- Source Database Requirements
- Target Database Requirements

Creating an OCI HeatWave connection

Create an OCI HeatWave connection by performing the following steps:

1. On the **Database connections** list page, select **Create connection**. If you need help finding the list page, see **Listing connections**.
2. On the General information step, in the **Name** field, enter a display name for the database connection resource.

   This is not the actual database name, but a name that will appear in a list of databases connections on completion of this procedure.
3. Add **Description** for the database connection resource.
4. In **Compartment** select the compartment in which the Database Connection resource will be created.
5. In **Type**, select a connection type. Select the connection type as `OCI HeatWave`.
6. In **Vault in *Compartment*** select the security vault.

   Database Migration Service uses the OCI Vault service to store your secrets such as passwords, wallets, and keys, and encrypts them with the encryption key that you provide. All of these are always stored in your tenancy and are in your full control.

   You can select a vault in a different compartment by clicking **Change Compartment**.
7. In **Encryption Key in *Compartment*** select the keystore that you configured in the vault.

   Only AES algorithm type keys are supported.
8. The **Advanced Options** section allows you to optionally create tags.
9. In the **Connection details** step, enter the connection details for the OCI HeatWave. Select one of the following options:
   - **Select MySQL database system:** Provide the following information:
     - **Database System in *Compartment*:** Select an OCI database.
     - **Database name:** The database name

Enter the database administrator credentials in **Initial load database username** and **Initial load database password**.

- **Enter MySQL database information**: Provide the following information:
  - **Database name:** The database name
  - **Host**: listener host IP address
  - **Port**: listener port number
  - Enter the database administrator credentials in **Initial load database username** and **Initial load database password**.

10. Select **Use different credentials for replication** to enter the credentials for replication. Select this option if you want to use a separate database user for performing replication for your online migrations. The following options are available
    - Replication database username
    - Replication database password
11. In the SSL details, select the security protocol.
    - Plain
    - TLS, and then select the SSL mode
    - MTLS, then select the SSL mode. Upload the Client certificate and Client private key.
12. In **Network Connectivity** details, **Create private endpoint to access this database** indicates whether the database is publicly accessible or if you want to create a private endpoint.

    By default, this option is selected while creating an OCI HeatWave connection. You must connect your database over a private IP address.

    In **Subnet in *Compartment***, select the subnet to which a private endpoint is created from the Oracle Cloud Infrastructure Database Migration service tenancy. This creates a network route for the Oracle Cloud Infrastructure Database Migration service to connect to the database within your tenancy. Select the subnet containing the appropriate Virtual Cloud Network (VCN).Click **Change Compartment** to select a subnet in a different compartment.
13. (Optional) Under **Advanced Options**, add **Connection attributes**. Click + **Another attribute** to add more.
14. Click **Create**.

> **Note**
>
> After you click **Create**, the database connection name appears in the Connection list while the creating connection operation runs. The creating connection operation can take a few minutes.

You can monitor the operation status in the State column. When the state is **Active**, the database connection creation is complete and successful.

Creating a MySQL Database Server connection

Create a MySQL Database Server connection by performing the following steps:

1. On the **Database connections** list page, select **Create connection**. If you need help finding the list page, see **Listing connections**.
2. On the General information step, in the **Name** field, enter a display name for the database connection resource.

   This is not the actual database name, but a name that will appear in a list of databases connections on completion of this procedure.
3. Add an optional **Description** enter a description that helps you distinguish for the database connection resource.
4. In **Compartment** select the compartment in which the Database Connection resource will be created.
5. In **Type**, select a connection type. Select the connection type as `MySQL Database Server`.
6. In **Vault in *Compartment*** select the security vault.

   Database Migration Service uses the OCI Vault service to store your secrets such as passwords, wallets, and keys, and encrypts them with the encryption key that you provide. All of these are always stored in your tenancy and are in your full control.

   You can select a vault in a different compartment by clicking **Change Compartment**.
7. In **Encryption Key in *Compartment*** select the keystore that you configured in the vault.

   Only AES algorithm type keys are supported.
8. The **Advanced Options** section allows you to optionally create tags.
9. In the **Connection details** page, enter the connection details for the MySQL Database Server as follows:
   - **Database name:** The database name
   - **Host**: listener host IP address
   - **Port**: listener port number
   - Enter the database administrator credentials in **Initial load database username** and **Initial load database password**.
10. Select **Use different credentials for replication** to enter the credentials for replication. Select this option if you want to use a separate database user for performing replication for your online migrations. The following options are available
    - Replication database username
    - Replication database password
11. In **Network Connectivity** details, **Create private endpoint to access this database** indicates whether the database is publicly accessible or if you want to create a private endpoint.

    Check this box if you plan to connect your database over a private IP address. Do not check it if the database has a public IP address.

    In **Subnet in *Compartment***, select the subnet to which a private endpoint is created from the Oracle Cloud Infrastructure Database Migration service tenancy. This creates a network route for the Oracle Cloud Infrastructure Database Migration service to connect to the database within your tenancy. Select the subnet containing the appropriate Virtual Cloud Network (VCN). Click **Change Compartment** to select a subnet in a different compartment.
12. In the SSL details, select the security protocol.
    - Plain

- TLS, and then select the SSL mode
- MTLS, then select the SSL mode. Upload the Client certificate and Client private key.

13. (Optional) Under **Advanced Options**, add **Connection attributes**. Click + **Another attribute** to add more.
14. Click **Create**.

> **Note**
>
> After you click **Create**, the database connection name appears in the Connection list while the creating connection operation runs. The creating connection operation can take a few minutes.

You can monitor the operation status in the State column. When the state is **Active**, the database connection creation is complete and successful.

Creating an Amazon RDS for MySQL connection

Create an Amazon RDS for MySQL connection by performing the following steps:

1. On the **Database connections** list page, select **Create connection**. If you need help finding the list page, see **Listing connections**.
2. On the General information step, in the **Name** field, enter a display name for the database connection resource.

   This is not the actual database name, but a name that will appear in a list of databases connections on completion of this procedure.
3. Add an optional **Description** enter a description that helps you distinguish for the database connection resource.
4. In **Compartment** select the compartment in which the Database Connection resource will be created.
5. In **Type**, select a connection type. Select the connection type as `Amazon RDS for MySQL`.
6. In **Vault in *Compartment*** select the security vault.

   Database Migration Service uses the OCI Vault service to store your secrets such as passwords, wallets, and keys, and encrypts them with the encryption key that you provide. All of these are always stored in your tenancy and are in your full control.

   You can select a vault in a different compartment by clicking **Change Compartment**.
7. In **Encryption Key in *Compartment*** select the keystore that you configured in the vault.

   Only AES algorithm type keys are supported.
8. The **Advanced Options** section allows you to optionally create tags.
9. In the **Connection details** page, enter the connection details for the Amazon RDS for MySQL as follows:
   - **Database name:** The database name
   - **Host**: listener host IP address
   - **Port**: listener port number

- Enter the database administrator credentials in **Initial load database username** and **Initial load database password**.

> **Note**
>
> To allow the connectivity between Database Migration Service and the database, add the IP address from which the connections will be sent to the AWS VPC security group associated with the RDS instance.

10. Select **Use different credentials for replication** to enter the credentials for replication. Select this option if you want to use a separate database user for performing replication for your online migrations. The following options are available
    - Replication database username
    - Replication database password
11. In the SSL details, select the security protocol.
    - Plain
    - TLS, and then select the SSL mode
    - MTLS, then select the SSL mode. Upload the Client certificate and Client private key.
12. (Optional) Under **Advanced Options**, add **Connection attributes**. Click + **Another attribute** to add more.
13. Click **Create**.

> **Note**
>
> After you click **Create**, the database connection name appears in the Connection list while the creating connection operation runs. The creating connection operation can take a few minutes.

You can monitor the operation status in the State column. When the state is **Active**, the database connection creation is complete and successful.

Creating an Azure Database for MySQL connection

Create an Azure Database for MySQL connection by performing the following steps:

1. On the **Database connections** list page, select **Create connection**. If you need help finding the list page, see **Listing connections**.
2. On the General information step, in the **Name** field, enter a display name for the database connection resource.

   This is not the actual database name, but a name that will appear in a list of databases connections on completion of this procedure.
3. Add an optional **Description** enter a description that helps you distinguish for the database connection resource.
4. In **Compartment** select the compartment in which the Database Connection resource will be created.

5. In **Type**, select a connection type. Select the connection type as `Azure Database for MySQL`.
6. In **Vault in *Compartment*** select the security vault.

   Database Migration Service uses the OCI Vault service to store your secrets such as passwords, wallets, and keys, and encrypts them with the encryption key that you provide. All of these are always stored in your tenancy and are in your full control.

   You can select a vault in a different compartment by clicking **Change Compartment**.
7. In **Encryption Key in *Compartment*** select the keystore that you configured in the vault.

   Only AES algorithm type keys are supported.
8. The **Advanced Options** section allows you to optionally create tags.
9. In the **Connection details** page, enter the connection details for the Azure Database for MySQL as follows:
   - **Database name:** The database name
   - **Host**: listener host IP address
   - **Port**: listener port number
   - Enter the database administrator credentials in **Initial load database username** and **Initial load database password**.
10. Select **Use different credentials for replication** to enter the credentials for replication. Select this option if you want to use a separate database user for performing replication for your online migrations. The following options are available
    - Replication database username
    - Replication database password
11. In the SSL details, select the security protocol.
    - Plain
    - TLS, and then select the SSL mode
    - MTLS, then select the SSL mode. Upload the Client certificate and Client private key.
12. (Optional) Under **Advanced Options**, add **Connection attributes**. Click + **Another attribute** to add more.
13. Click **Create**.

> **Note**
>
> After you click **Create**, the database connection name appears in the Connection list while the creating connection operation runs. The creating connection operation can take a few minutes.

You can monitor the operation status in the State column. When the state is **Active**, the database connection creation is complete and successful.

Creating an Amazon Aurora MySQL connection

Create an Amazon Aurora MySQL connection by performing the following steps:

1. On the **Database connections** list page, select **Create connection**. If you need help finding the list page, see **Listing connections**.
2. On the General information step, in the **Name** field, enter a display name for the database connection resource.

   This is not the actual database name, but a name that will appear in a list of databases connections on completion of this procedure.
3. Add an optional **Description** enter a description that helps you distinguish for the database connection resource.
4. In **Compartment** select the compartment in which the Database Connection resource will be created.
5. In **Type**, select a connection type. Select the connection type as `Amazon Aurora MySQL`.
6. In **Vault in *Compartment*** select the security vault.

   Database Migration Service uses the OCI Vault service to store your secrets such as passwords, wallets, and keys, and encrypts them with the encryption key that you provide. All of these are always stored in your tenancy and are in your full control.

   You can select a vault in a different compartment by clicking **Change Compartment**.
7. In **Encryption Key in *Compartment*** select the keystore that you configured in the vault.

   Only AES algorithm type keys are supported.
8. The **Advanced Options** section allows you to optionally create tags.
9. In the **Connection details** page, enter the connection details for the Azure Database for MySQL as follows:
   - **Database name:** The database name
   - **Host**: listener host IP address
   - **Port**: listener port number
   - Enter the database administrator credentials in **Initial load database username** and **Initial load database password**.
10. Select **Use different credentials for replication** to enter the credentials for replication. Select this option if you want to use a separate database user for performing replication for your online migrations. The following options are available
    - Replication database username
    - Replication database password
11. In the SSL details, select the security protocol.
    - Plain
    - TLS, and then select the SSL mode
    - MTLS, then select the SSL mode. Upload the Client certificate and Client private key.
12. (Optional) Under **Advanced Options**, add **Connection attributes**. Click + **Another attribute** to add more.
13. Click **Create**.

> **Note**
>
> After you click **Create**, the database connection name appears in the Connection list while the creating connection operation runs. The creating connection operation can take a few minutes.

You can monitor the operation status in the State column. When the state is **Active**, the database connection creation is complete and successful.

Creating a Google Cloud SQL for MySQL connection

Create a Google Cloud SQL for MySQL connection by performing the following steps:

1. On the **Database connections** list page, select **Create connection**. If you need help finding the list page, see **Listing connections**.
2. On the General information step, in the **Name** field, enter a display name for the database connection resource.

   This is not the actual database name, but a name that will appear in a list of databases connections on completion of this procedure.
3. Add an optional **Description** enter a description that helps you distinguish for the database connection resource.
4. In **Compartment** select the compartment in which the Database Connection resource will be created.
5. In **Type**, select a connection type. Select the connection type as `Google Cloud SQL for MySQL`.
6. In **Vault in *Compartment*** select the security vault.

   Database Migration Service uses the OCI Vault service to store your secrets such as passwords, wallets, and keys, and encrypts them with the encryption key that you provide. All of these are always stored in your tenancy and are in your full control.

   You can select a vault in a different compartment by clicking **Change Compartment**.
7. In **Encryption Key in *Compartment*** select the keystore that you configured in the vault.

   Only AES algorithm type keys are supported.
8. The **Advanced Options** section allows you to optionally create tags.
9. In the **Connection details** page, enter the connection details for the Google Cloud SQL for MySQL as follows:
   - **Database name:** The database name
   - **Host**: listener host IP address
   - **Port**: listener port number
   - Enter the database administrator credentials in **Initial load database username** and **Initial load database password**.
10. Select **Use different credentials for replication** to enter the credentials for replication. Select this option if you want to use a separate database user for performing replication for your online migrations. The following options are available
    - Replication database username

- Replication database password

11. In the SSL details, select the security protocol.
    - Plain
    - TLS, and then select the SSL mode
    - MTLS, then select the SSL mode. Upload the Client certificate and Client private key.
12. (Optional) Under **Advanced Options**, add **Connection attributes**. Click + **Another attribute** to add more.
13. Click **Create**.

> **Note**
>
> After you click **Create**, the database connection name appears in the Connection list while the creating connection operation runs. The creating connection operation can take a few minutes.

You can monitor the operation status in the State column. When the state is **Active**, the database connection creation is complete and successful.

6

Manage connections

To perform migration, create connections to your source and target databases by creating database connection resources. Database Connection resources enable network connectivity to the source and target databases.

Testing connectivity of a database connection

You can test the connectivity of a database connection before you start or create a migration. You can get information about the connection and fix any configuration issues before running the migration.

You can diagnose issues with a database connection such as:

- Incorrect IP address and/or port.
- Incorrectly declaring a connection public or private.
- Incorrect, expired, or locked database credentials.
- Missing entries in security lists or NSGs to allow communication with database IP or port.
- Connection failures through FastConnect, VPN, or any other network connectivity issues for your on-premises database.
  Oracle Cloud Infrastructure Database Migration service runs a network connectivity check followed by JDBC Connection or Socket Connectivity using the Database Connection data that you provide.

To test the connectivity of a database connection use either of the following methods:

- From the action menu (three dots), select **Test connection**.
- Select the database connection, which opens the Database connection details page and in the **Actions** menu select the **Test connection** button.

> **Note**
>
> If you are connecting to an AWS database, make sure that your VPC security group is configured to allow connections from the IP address listed in the connection details.

The **Test connection** dialog is displayed.

The **Test connection** dialog displays the following details:

- Result of the connection test.
- Error codes and the accompanying error messages.
- **Cause**: The issue causing the connection failure.
- **Action**: The action you can perform to resolve the error.

Viewing connection details

Database connection details page

On the Database connection details page you can view a list of your Connections in a table, which includes the following information:

- **Name** of the Connection resource
- **State** of the Connection resource, which can be any one of the following:
  - Creating: The new Connection resource is being created in OCI.
  - Updating: Changes to the Connection resource are being registered in OCI.
  - Active: The Connection resource has finished being created or updated and is ready for use.
  - Inactive: A fallback state for unexpected errors.
  - Deleting: This state appears when you delete a Connection resource. The resource remains in this state until deletion is completed, at which point the resource is no longer listed in the console.
  - Failed: There are problems with the Connection resource. You can review the Connection resource work requests to investigate the issue.
- **Created** date and time

Database connection details page

Select a Connection from the Database connection details to view its details.

On the Database connection details page you can view the Connection information, including:

General information:

- **OCID**: The resource's unique Oracle Cloud ID.
- **Compartment**: The compartment where the Connection resides.
- **Created**: The date and time when the Connection was created.
- **Updated**: The date and time when the Connection was updated.
- **Encryption vault**: The link takes you to the Vault Details page.
- **Encryption key**: The link takes you to the Key Details page
- **Subnet**: The link takes you to the Subnet Details page.

Technology information:

- **Type**: Autonomous Database, Database (BareMetal, VM, Exadata), VM Cluster Database (Exadata), Oracle, or Amazon RDS
- **Database**: For OCI co-managed databases--Autonomous Database, Database (BareMetal, VM, Exadata), VM Cluster Database (Exadata)--the display name of this Connection is also shown. The link takes you to the Database Details page in OCI.

- **Initial load database username**
- **Replication database username**

Network security groups

On the Database connection details page, you can find the **Network security groups** that can be associated, with this database connection.

> **Note**
>
> - You can associate NSGs available in your VCN to the connection. The NSGs that are listed for a subnet are only applicable to your current VCN.
> - You can add network security groups in Database Migration Service to control traffic, if you have connected over private endpoints while registering databases. The advantage of network security groups (NSGs) is that rules can be limited to individual resources within a subnet, whereas Security Lists apply to all resources within a subnet.
> - Associating NSGs to database connections provides you fine grained control over the access to your database connection resources that are involved in the migration process (Source and Target). See Network Security Groups for more information.

1. Select **Add network security groups** to open the **Add network security groups** panel.
2. Select a network security group from the compartment and click **Add network security groups**.
3. You can add up-to five unique network security groups by clicking on **Another network security group**.

You can view the following details associated with the **Network security groups** Resource:

- **Name** : The name of the added network security group.
- **State**: The state of the network security group.
- **Compartment** : The compartment where the network security group resides.
- **Created** : The date and time when the network security group was created.

Select single or multiple network security groups to remove them by clicking **Remove** in the **Remove network security groups** confirmation dialog.

Select **View details** from the actions menu (three dots) for a specific NSG to view information related to VCN.

Work Requests

On the Database connection details page, you can find the Work requests list. Work Requests lists any work requests sent to OCI to facilitate the creation, update, or deletion of this resource. Click the work request to go to the Work Request Details page for more information about the work request.

Editing a connection

To edit a connection:

1. In the list of databases on the Databases connection details page, select the Name of the Connection you want to edit.
2. In the Database connection details page, select Rename to change the name of the Connection.
3. Select the **Edit** option for a connection on top of the page to update the settings:
   - Enter **Name** and **Description** to change the name of the Connection.
   - **Encryption Key**: You can change the selected vault, encryption key, and compartment in which to create a secret. Only AEP algorithm type keys are supported.
   - **Subnet**: You can update the subnet and private endpoint compartment network connectivity settings.
   - **Database**: you can update the database administrator user name and password used to connect to the database. When editing a non-Autonomous database connection, you can also edit connect string, SSH details, and TLS details. The following options are available:
     - **Connect string**
     - **Initial load database username**
     - **Initial load database password**
     - If you select **Use different credentials for replication** , enter the following details:
       * **Replication database username**
       * **Replication database password**
     - **Keep existing certificates/key pair configuration**
     - **Remove certificate/key pair configuration**
     - **Update certificate/key pair configuration**
     - In the **Advanced options**, in optional **SSH settings** provide the SSH information for your database hosts if you wish to provide SSH access to the service to perform the migrations. Provide the SSH related information.
4. Click **Update**.

Moving a connection

You can move a Connection from one compartment to another.

To move a Connection:

1. In the list of databases on the Database connection details page, select **Move resource** from the Actions (three dots) menu for the database you want to move.

   You can also select **Move resource** on the Database connection details page from the **Actions** menu.
2. In the **Move Resource to a Different Compartment** dialog, select the compartment to move the Connection to from the dropdown.
3. Select **Move Resource**.

After you move the Connection to the new compartment, inherent policies apply immediately and may affect access to the Connection through the Console. For more information, see Managing Compartments.

Deleting a connection

Before you delete a Connection, ensure that you carefully review any resources that reference the Connection. It is not possible to delete a Connection if it is references by a migration. You must delete the migration before deleting the associated Connections.

Deleting a Connection also deletes the private connection and database credentials, so it will no longer be accessible to migrations. After you delete a Connection, it *cannot* be restored.

> **Note**
>
> Connections also capture and synchronize database credentials to Database Migration. Any change made to the credential, such as updating or deleting, synchronizes to Database Migration. You will encounter issues when the Replicat or Extract attempts to reconnect to a deleted Connection.

To delete a Connection:

1. In the list of databases on the Database connection details page, select **Delete** from the Actions (three dots) menu of the database you want to delete.

   You can also click **Delete** on the Database connection page from the **Actions** menu.
2. In the Delete dialog, click **Delete**.

Managing tags for connections

Tags help you locate resources within your tenancy. You can add and view a connection's tags from the Database Connections page and from the Database connection details page.

On the Database connection details page, from the Connection's Actions (three dots) menu, select **Add Tags** or **View Tags**.

On the Database connection details page, from the **Actions** menu, you can select **Manage tags**, or click the **Tags** tab to view and edit tags.

See Managing Tags and Tag Namespaces to learn more about tagging.

7

Create migrations

When you create a migration with Oracle Cloud Infrastructure Database Migration, you specify how the migration should run, select the source and target databases, and then configure the data transport settings.

- Listing migrations
  View the migrations for a specific compartment. You have the options to search and filter the migrations, sort, and apply filters on the table directly.
- Creating Oracle migrations
  A migration resource is a definition of your migration containing all settings and parameters that are eventually used when performing a migration.
- Creating MySQL migrations
  A migration resource is a definition of your migration containing all settings and parameters that are eventually used when performing a migration.
- Configuring validation options
  Oracle Cloud Infrastructure Database Migration is integrated with the Oracle Cloud Pre-Migration Advisor tool. The tool analyzes the source database during validation and advises you about database features and constructs that are problematic.
- Configuring optional replication advanced options
  In Oracle Cloud Infrastructure Database Migration, for online migrations using Oracle GoldenGate, you can configure some Oracle GoldenGate performance settings in the Migration resource Advanced Settings, Replication tab.

Listing migrations

View the migrations for a specific compartment. You have the options to search and filter the migrations, sort, and apply filters on the table directly.

1. Open the navigation menu, select **Migration & Disaster Recovery**. Under **Migration & Disaster Recovery** select **Database Migration**.
2. Under **Database Migration**, select **Migration**.

   The **Migration** page opens. All the **Migrations** in the selected compartment are displayed in a table.
3. To view the **Migrations** in a different compartment, use the **Compartment** filter to switch compartments.

> **Note**
>
> You must have permission to work in a compartment to see the resources in it. If you're not sure which compartment to use, contact an administrator. For more information, see Understanding Compartments.

Filtering List Results

Use filters to limit the **Migrations** in the list. Perform one of the following actions depending on the options that you see:

- From the **Search and Filter** box above the list table, select one or more filters and specify the values that you want to use to narrow the list. In general, the filters correspond to the columns shown in the list table, although some filters represent attributes that aren't shown in the table. The **Compartment** filter is always displayed next to **Applied filters**.
- On the left side of the list page, select a value from one of the available filters, such as compartment, state, or tags.

Change the order of the items in the list table by using the sort icons next to the column names.

For information about searching for resources and managing the columns in the list table, if those features are available, see Listing Resources.

Actions
In the list table, select the name of a **Migration** to open its details page, where you can view its status and perform other tasks.

To perform an action on a **Migration** directly from the list table, select an available option from the Actions menu in the row for that **Migration**:

- **View details**: Open the details page for the migration.
- **Rename**: Rename the migration
- **Validate**: Validating a migration.
- **Start**: Start the migration.
- **Clone**: Cloning a migration.
- **Copy OCID**: Copy the OCID of the **Migration** to the clipboard.
- **Move resource**: Move the Migration to another compartment.
- **Manage tags**: Add one or more tags to the **Migration**, or change the value of an existing tag. See Resource Tags.
- **Delete**: Delete a migration.

To create a **Migration**, select Create migrations.

Creating Oracle migrations

A migration resource is a definition of your migration containing all settings and parameters that are eventually used when performing a migration.

The following procedure explains how to create migrations, which contain the settings for running migration jobs with Database Migration. You can create multiple migration resources with different parameter settings to test different scenarios.

Creating a migration

1. On the **Migrations** list page, select **Create migration**. If you need help finding the list page, see **Listing migrations**.
   This opens the Create migration page.
2. Configure the following settings:
   - **Name:** Enter a unique name for the migration.

On completion of the Create Migration page, the name you enter here is displayed in the list of migrations on the Migrations page.

- Add an optional **Description:** for the migration resource.
- **Compartment:** Select the compartment in which the Database Migration service is hosted.

3. Enter the following information in the **Source database** box.
   - **Database connection in *Compartment*:** Select the source database connection entry.

     If the source database is a PDB, make sure you selected the PDB database connection in the drop-down, not the CDB connection.
   - **Database is pluggable database (PDB):** If the source database is a PDB, check this box so you can also enter the CDB details.

     Container database connection in *Compartment*: If the source database is a PDB, select the CDB you selected here. The CDB connection is not required if you are doing an offline migration.
4. Enter the following information in the **Target database** box. Select the target database connection for your *Compartment*.
5. In the **Transfer medium for initial load** step, select one of the following transfer mediums based on your requirement for your migration:
   - Select an Initial load option:

     Data Pump via database link: Enable this option to use a direct SQL*Net connection between the source and target databases. Note that using Data Pump with a database link to Autonomous Database targets requires that the source database be set up with SSL encryption.

     > **Note**
     >
     > If your source database is Oracle Database Standard Edition 2, select the **Datapump via database link:** option as the transfer medium. Encryption for the exported Datapump dumps is not available for the object storage or file storage transfer mediums.

     Data Pump via object storage: This option lets Data Pump temporarily store the exported database in an Object Storage bucket. If this option is enabled, also configure the following settings in the **Source Data Pump settings**.

     – **Amazon S3 bucket:** Enter the details for the Amazon S3 bucket. This option is only shown if the source database connection is of type Amazon RDS.

       The bucket **Name** must be between 3 and 63 characters, and can consist only of lower case letters, numbers, dots (.), and hyphens (-). It must begin and end with a letter or number.

       The **Region** must be in the same region as the RDS Oracle database. For example `us-east-1`

       For more information about the Amazon RDS source database use case, see Migrating Databases from Amazon Web Services RDS to Supported Oracle Targets.

- **Export directory object:** Enter the file **Name** and **Path** to the directory object that will be used by Data Pump export on the source database server file system. Database Migration handles the directory object creation for you.

  The name must comply with Oracle Database directory object rules. See CREATE DIRECTORY in *Oracle Database SQL Language Reference*.

> **Note**
>
> Oracle recommends using Data Pump via object storage as the preferred transfer medium. Data Pump via database link is only recommended for small databases as you cannot enable parallelism which affects the migration performance.

> **Note**
>
> When you select Object storage as the transfer medium, ensure to provide a valid directory path where the dumps will be stored. Specify a new directory object that points to a valid and accessible path or specify an existing directory object that is mapped to a valid directory path, and then submit the job if your job fails. Ensure that the specified initial load user has the necessary READ and WRITE privileges on the directory object.

Data Pump via file storage: Enable this option to use a shared File Storage Service (FSS) mount as the transfer medium between source and target databases.

- Network access to your file system is provided through a mount target (OCID provided in the OCI Console).
- Your file system must have at least one export in the mount target used to mount and use the file system.
- When creating the file system using the OCI Console, the workflow also creates a mount target and an associated export.
- Exports control how NFS clients access file systems when they connect to a mount target.

> **Note**
>
> Following are the prerequisites to use this option:
>
> – You must Create a File System and specify the mount target and the export path.
> – You must have the shared file system mounted on both the source and target database hosts.
> – Supported source database: This transfer medium is supported for all the supported on-premises database cases excluding Amazon RDS.
> – Supported target database : This transfer medium is supported for both non-ADB and ADB target databases (All Oracle Cloud Infrastructure co-managed and Exadata).
> – To perform the Data Pump export, you must have the read or write access to the shared mounts on the source host.
> – On the target database host, you must at least have read access to the shared mount to perform the Data Pump import.
> – See Mounting File Systems and Creating a Mount Target for more information.

Enter the following information in the **Source Data Pump settings**:

– **Export directory object name:** Name of the export directory object
– **Export directory object path:** The shared file system mount path on the source database with read and write access.
– **Source Database file system SSL Wallet Path**

> **Note**
>
> This field is displayed only when the SSH details are not provided during source database connection while performing migration to Autonomous database.

If you enable this option when the target database is Autonomous database, then configure the following settings. Enter the following information in the **Target Data Pump settings**:

– **Shared storage mount target in *compartment***: The file system mount target to be used in the target database.

> **Note**
>
> You can use the mount targets only when the target database is Autonomous database. For any other databases, use the shared file system based mount point.

– **Import directory object name:** The Data Pump import directory object name used to mount the shared file system.

If you enable this option when the target database is non Autonomous database, then configure the following settings. Enter the following information in the **Target Data Pump settings** box:

- **Import directory object name:** The Data Pump import directory object name used to mount the shared file system.
- **Import directory object path:** The Data Pump import directory object name used to mount the shared file system.

- **Object storage bucket in *Compartment*:** Select the object storage bucket. This bucket is used for any Cloud Premigration Advisor Tool reports, Database Migration, and Data Pump log storage, and Data Pump dump files.

  If the bucket is in a different compartment, click **Change Compartment** to look in another compartment.

6. If the source or the target database is non-ADB, then the following fields are shown when the initial Data Pump load is performed via object storage or file storage:

   Enter the following information in the **Source Data Pump settings** box:

   - **Export directory object name:**
   - **Export directory object path:**
   - **Source Database file system SSL Wallet Path**

   > **Note**
   >
   > This field is displayed only when the SSH details are not provided during source database connection.

   Enter the following information in the **Target Data Pump settings** box:

   - **Import directory object name**
   - **Import directory object path**
   - **Target Database file system SSL Wallet Path**

   > **Note**
   >
   > This field is displayed only when the SSH details are not provided during target database connection.

   If your source or target is non-ADB and you did not provide the SSH details for both source and target database when creating your database connection, and your transfer medium is object storage, you must perform the following steps:

   - Download SSL Wallet with Certificates
   - Set up Network ACL

   You can download a pre-created wallet.

   To download a wallet:

   a. Download the wallet file.
   b. Unzip the certificate files to a directory on the file system of your database host.
   c. Enter this location in **SSL Wallet Path** when creating the migration.

The user performing the export or import requires the necessary network ACL to be granted to access the network from the source and target database host.

In the following example, run the following commands as SYS if the export or import user is SYSTEM. If your database is multitenant, then perform the following actions in CDB$ROOT. Restrict the host as required.

Security consideration: Do not allow a complete network access from the database. Restrict the host access to the required OCI object storage region. For example, https://objectstorage.us-ashburn-1.oraclecloud.com and ACL can be time restricted with relevant start_date and end_date arguments in DBMS_NETWORK_ACL_ADMIN.CREATE_ACL. For example:

```
@$ORACLE_HOME/rdbms/admin/sqlsessstart.sql
define clouduser=<user performing export at src or import at target e.g.,
SYSTEM>
define sslwalletdir=< OCI wallet path e.g., /opt/oracle/dcs/commonstore/
import_dmp/nossh_wallet>
begin
dbms_network_acl_admin.append_host_ace(
        host =>'*',
        lower_port => 443,
        upper_port => 443,
        ace => xs$ace_type(
                privilege_list => xs$name_list('http', 'http_proxy'),
                principal_name => upper('&clouduser'),
                principal_type => xs_acl.ptype_db));
dbms_network_acl_admin.append_wallet_ace(
    wallet_path => 'file:&sslwalletdir',
        ace => xs$ace_type(privilege_list =>
                xs$name_list('use_client_certificates', 'use_passwords'),
                principal_name => upper('&clouduser'),
                principal_type => xs_acl.ptype_db));
end;

/
COMMIT;
@$ORACLE_HOME/rdbms/admin/sqlsessend.sql
```

Once the connect privilege is granted, connect as the relevant user such as, SYSTEM and verify if the privilege is granted using the following query:

```
COLUMN host FORMAT A30
SELECT host, lower_port, upper_port, privilege, status FROM
user_network_acl_privileges;
```

For more information, see How To Set Network ACLs.

7. If you want to create an online migration, check the **Use online replication** option to enable the replication of all data and metadata transactions from the source to the target database, committed after the initial load has begun. For additional optional configurations, see the **Replication** tab in the **Advanced options**. Optionally, you can set some additional properties which can affect the performance of your online migration.

> **Note**
>
> Oracle recommends using the default **Use online replication** option to perform an online replication.

> **Note**
>
> Skip this step for offline (Data Pump only) migrations.

8. If you want to **Use snapshot standby database as the source for the Data Pump export** toggle the switch and choose the appropriate **compartment** and select the **Snapshot Standby database connection** that you previously created. This option is available for supported Oracle online migrations. For more information, see the Using the Snapshot Standby database topic.

   > **Note**
   >
   > The Snapshot Standby database is a clone of the database in production and thus it is leveraged to avoid performance issues in the production database.

9. The **Advanced options** section allows you to optionally create tags.
10. Optionally, select **Advanced Options** to configure advanced Data Pump, validation, and Oracle GoldenGate settings.

    For details about these settings see Selecting objects for Oracle migration, Configuring initial load advanced options for Oracle migrations, Configuring validation options, and Configuring optional replication advanced options.

11. Click **Create**.

    The migration is loaded, and a new Migration Details page opens showing the information, metrics, and operations for the migration.

    The status of the creation operation is shown on top of the page along with the name of the migration. When the status is Active, you can run migration jobs with the migration.

    Alternatively click **Save as stack** option to save the migrations as a stack.

    The **Save as stack** pane is displayed. You can save the migration resource configurations as a stack. Use the stack to install, configure, and manage the resource through the Resource Manager service.

    The following options are available:

    a. **Name** (Optional)
    b. **Description** (Optional)
    c. **Save in *your* compartment**
    d. Optionally, add **Tags** to organize your resources.
    e. Click **Save** to save as a stack.
    f. Click **Cancel** if you do not want to save the stack.

    After you save the Stack, it displays a link in the upper right corner that directs you to the newly created Stack in the Resource Manager. See the following topics for the next steps:

- Selecting objects for Oracle migration
  As part of the migration creation, you can specify objects to include or exclude for Oracle Migrations. Alternatively, you can also perform the act of inclusion or exclusion of objects after a migration has been created, using the **Selected Objects** menu option.
- Configuring initial load advanced options for Oracle migrations
  Oracle Cloud Infrastructure Database Migration automatically sets optimal defaults for Oracle Data Pump parameters to achieve better performance and ensure security of data.

Related Topics

- Creating a Stack from a Resource Creation Page

Selecting objects for Oracle migration

As part of the migration creation, you can specify objects to include or exclude for Oracle Migrations. Alternatively, you can also perform the act of inclusion or exclusion of objects after a migration has been created, using the **Selected Objects** menu option.

When creating a migration, specify rules for selecting objects in the **Advanced Settings** on the **Selected Objects** tab.

Select the **Use advanced editor** toggle to add the objects you want to include or exclude in bulk as follows:

```
schema_name1,object_name1,TABLE,EXCLUDE
schema_name2,object_name2,TABLE,EXCLUDE
schema_name3,object_name3,TABLE,EXCLUDE
```

Add all the objects to include or exclude by listing the **Object Owner**, **Object Name**, **Object Type** , and Action (**Include** or **Exclude**), as shown in the above format (comma separated).

To exclude a table from replication, enter information in the following comma separated format:

```
schema_name1,object_name1,TABLE,EXCLUDE,EXCLUDEFROMREPLICATION
```

> **Note**
>
> In the advanced editor:
>
> - Use a comma separator character (,) to separate each item for every inclusion/ exclusion definition.
> - Use the escape character (\) if your schema or object name has a comma (,) character as part of its name.
> - You can add multibyte character (Unicode) names for schema or object names. For example, `ꜫꜫschema,ꝋbject,TABLE,EXCLUDE`.
> - The maximum input size is 500 KB.

Special Characters and Reserved Words
To avoid issues, do not use special characters such as (`& @ * $ | % ~ -`) in object names, except for underscore (_). Also do not use Oracle SQL or PL/SQL reserved words in object names, especially the Oracle name.

For more information for excluding the table objects with special characters, see Database Object Naming Rules.

For information about Reserved words, characters and nonquoted identifiers, see Oracle SQL Reserved Words.

For the latest information, you can generate a list of all keywords and reserved words with the `V$RESERVED_WORDS` view, described here.

Alternatively, you can choose either **Include** or **Exclude** from the **Action** list to specify if a rule should include or exclude the specified database objects in the migration. You can either include or exclude objects in a migration, but you cannot do both.

If no rule is defined, all schemas and objects of the source database will be migrated, with exceptions explained in **Objects and Schemas Excluded by Default** below.

If you specify **Include** rules, the migration will only move the specified objects and their dependent objects; all other objects are automatically excluded.

When specifying **Exclude** rules, the migration will exclude the specified objects and their dependent objects; all other objects are included in the migration.

To create a rule, enter values for each of the following fields:

- **Object Owner** specifies the owner of the selected database objects. When using Include rules, all rules must be for the same owner, and wild characters are not allowed.
- **Object Name** specifies the name of selected database objects
- **Object Type** specifies the type of selected database objects. You can select **ALL** to select objects of all types.
- **Replication only**: You can select this toggle when you want to exclude the tables from replication. This option is enabled when the action is **Exclude** and the **Object Type** is **TABLE**. This ensures that object types such as `ROWID` columns, unsupported by Oracle GoldenGate, are not replicated during online migration.

> **Note**
>
> Although this excludes the tables from replication; these tables are migrated to the target database twice. Once during the initial load and another during the switchover phase. Data Pump export and import is performed again (reload) on these tables that were not part of replication to bring these tables on the target up to date with the source.

You can filter **Object Owner** and **Object Name** fields using any valid pattern in Java class Pattern. For example, you can enter **.** * in the **Object Name** field to select objects of any name.

The objects included in a migration are also influenced by the **Job Mode** of the initial load, as explained in Configuring initial load advanced options for Oracle migrations.

Please note the following restrictions:

- When excluding an object in a specified schema, and an object of the same name exists in a different schema that is also part of the migration, the objects will not be excluded (that is, the rule is ignored). The exclusion can be accomplished by migrating the schemas in separate migrations.
- When creating Include rules in Full job mode, only schema-level rules (Object Name is .* and Object Type is ALL) are allowed.

- If an Include rule has .* in Object Name, no other rule for the same Object Type is allowed. If the rule has ALL as Object Type, no other rule for any type is allowed.
- The Object type ALL is only allowed for schema-level rules (Object Name is .*).
- If you define a rule with an Object owner pattern other than .* and the Object Name is .* then the Object type TABLE is not allowed.
- Object-level rules (Object Name is any pattern other than .*) can only be used for the following object types: `DIRECTORY`, `FUNCTION`, `JOB`, `MATERIALIZED_VIEW`, `PACKAGE`, `PROCEDURE`, `TRIGGER`, `SEQUENCE`, `TABLE`. All other object types must be either included or excluded using the .* pattern in Object Name, and in addition for exclude, the owner should be .*

Examples

Example 1: Include all objects of schema MySchema

Action = **Include**

Object Owner	Object Name	Object Type
MySchema	.*	ALL

Example 2: Include all tables starting with PROD and procedure MYPROC of schema MySchema, including all dependent objects.

Action = **Include**

Object Owner	Object Name	Object Type
MySchema	PROD.*	TABLE
MySchema	MYPROC	PROCEDURE

Example 3: Exclude schemas starting with Experimental, the table `MySchema.OldTable` (also excluding all dependent objects) and all objects of type `DB_LINK`.

Note that MySchema.OldTable will not be excluded if a table called OldTable is present in a different schema that is also migrated.

Action = **Exclude**

Object Owner	Object Name	Object Type
Experimental.*	.*	ALL
MySchema	OldTable	TABLE
.*	.*	DB_LINK

Objects and schemas excluded by default

The following object types are always excluded:

- **GoldenGate administrators:** identified in `DBA_GOLDENGATE_PRIVILEGES`, including `ggadmin` and `c##ggadmin` users
- **If target is Autonomous Data Warehouse Shared Infrastructure:** `CLUSTER`, `DB_LINK`, `INDEXTYPE`, `STATISTICS`
- **If target is Autonomous Data Warehouse Dedicated Infrastructure, Autonomous Transaction Processing Shared or Dedicated Infrastructure:** `CLUSTER`, `DB_LINK`, `STATISTICS`

- **All other targets:** `STATISTICS`

The following schemas are excluded by default:

- Schema is marked as `ORACLE_MAINTAINED` in `SYS.DBA_USERS` on the source or target database
- Schema is marked as excluded from export in `SYS.KU_NOEXP_VIEW` on the source database
- Schema `GGADMIN` and `C##GGADMIN`

Configuring initial load advanced options for Oracle migrations

Oracle Cloud Infrastructure Database Migration automatically sets optimal defaults for Oracle Data Pump parameters to achieve better performance and ensure security of data.

To further tune performance, change the export modes, or rename database objects, there are several Data Pump settings that you can configure in the Migration resource Advanced Settings, Initial Load tab.

- **Source data transfer mechanism:** Type of dump transfer to use during Data Pump Export. The options are CURL or OCI_CLI. The default is CURL.
- **Target data transfer mechanism** Type of dump transfer to use during Data Pump Import. The options are CURL or OCI_CLI. The default is OCI_CLI.
- **Job mode:**
  - **Full** performs a full database export.
  - **Schema** (default) lets you specify a set of schemas to export.

  Specify schema objects for inclusion or exclusion in the Advanced Settings, Selected Objects tab. See Selecting objects for Oracle migration for details.

  See Oracle Data Pump Export Modes in *Oracle Database Utilities* guide for more information about the job modes.
- **Table exists action** sets the Data Pump `TABLE_EXISTS_ACTION` parameter, which specifies the action to be performed when data is loaded into a preexisting table.
  - **Skip** no changes to the preexisting table.
  - **Truncate** removes rows from a preexisting table before inserting rows from the Import. Note that if Truncate is specified on tables referenced by foreign key constraints, the truncate operation is changed to Replace.
  - **Replace** replaces preexisting tables with new definitions. Before creating the new table, the old table is dropped.
  - **Append** - new rows are added to the existing rows in the table
  - **UNSPECIFIED**: (default) Use default settings.
- **Cluster** is enabled by default. When enabled, Data Pump workers are distributed among the instances (nodes) in a cluster (Oracle RAC) architecture.

  If this setting is not checked, all Data Pump workers are started on either the current instance or on an instance usable by the job.
- **Export parallelism degree** sets the Data Pump export `SET_PARALLEL degree` parameter. This setting determines the maximum number of worker processes that can be used for the migration job. You use this parameter to adjust the amount of resources used for a job.

  By default, Database Migration sets source database export parallelism to (Sum of (2 x (no. of physical CPU) per node)) with Max 32 cap.

See SET_PARALLEL Procedure in *Oracle Database PL/SQL Packages and Types Reference* for more details.

> **Note**
>
> The export parallelism degree for MySQL migrations has a value of four threads. This is also applicable for the **Import parallelism degree**.

- **Import parallelism degree**, similar to Export Parallelism Degree, sets the Data Pump import `SET_PARALLEL degree` parameter.

  By default, Database Migration sets import parallelism for Autonomous Database to the number of OCPUs.

- **Auto-create tablespaces:** For ADB-Dedicated (ADB-D) and co-managed/non-ADB database targets, automatic tablespace creation is enabled by default. Database Migration validates whether automatic tablespace creation is supported on the specified target database. Oracle Autonomous Database Serverless targets are not supported.

  Database Migration automatically discovers the source database tablespaces associated with user schemas that are being migrated, and automatically creates them in the target database before the Data Pump import phase. Database Migration generates the DDL required to pre-create the tablespaces, creates the tablespaces on the target, and runs the generated DDL.

  With automatic tablespace creation enabled, Database Migration skips automatic creation for any tablespaces that are specified in the **Metadata remaps** section, or that already exist in the target database.

  Use big file: Autonomous Database systems support only BIGFILE tablespaces, so Database Migration enforces BIGFILE tablespace by default on Autonomous Database targets, and reports an error if SMALLFILE tablespaces are found. You can explicitly remap any SMALLFILE tablespaces instead.

  Extend size: enables tablespaces to AUTOEXTEND to avoid extend errors, with a default extend size of 500MB.

- **Remap target:** When migrating to an Oracle Autonomous Database Serverless target, all tablespaces are automatically mapped to DATA. You can override this by explicitly mapping tablespaces to a different target in **Metadata remaps**.

- **Block size of target database:** Optionally, when creating or updating a migration for ADB-Dedicated (ADB-D) and co-managed/non-ADB database targets, you can select the database block size for the tablespace as automatic tablespace creation is enabled by default.
  Currently, there are two possible values to select the target database block size: 8K or 16K.

- **Metadata remaps** lets you rename database objects during a migration job. Select the object to rename under **Type**, then enter the **Old Value** and **New Value**.

  Supported objects are Datafile, Schema, Table, and Tablespace.

  When migrating to an Oracle Autonomous Database Serverless target, all tablespaces are automatically mapped to DATA. You can override this by explicitly mapping tablespaces to a different target.

  Quota grants for individual users to tablespaces are not remapped, so you must manually create these grants for tablespace DATA.

  To rename multiple objects, click **+ Another Metadata Remap**.

- **Advanced parameters** Advanced parameters section lets you specify the advanced parameters pertaining to the initial load.
  While creating migration, select a list of parameters for migration. Select the **Parameter name** and **Parameter value** from the list. Click **+ Another parameter** to add the parameters.

  For more information, see Zero Downtime Migration Logical Migration Response File Parameters Reference .

Creating MySQL migrations

A migration resource is a definition of your migration containing all settings and parameters that are eventually used when performing a migration.

The following procedure explains how to migrate data from a MySQL source database to a MySQL target database, which contain the settings for running migration jobs with Database Migration. You can create multiple migration resources with different parameter settings to test different scenarios.

Creating a migration

1. On the **Migrations** list page, select **Create migration**. If you need help finding the list page, see **Listing migrations**.
   This opens the Create migration page.
2. Configure the following settings:
   - **Name:** Enter a unique name for the migration.

     On completion of the Create Migration page, the name you enter here is displayed in the list of migrations on the Migrations page.
   - Add an optional **Description:** for the migration resource.
   - **Compartment:** Select the compartment in which the Database Migration service is hosted.
3. Select the **Source database** connection entry for your compartment.
4. Select the **Target database** connection entry for your compartment.

   > **Note**
   >
   > Currently, OCI HeatWave service is supported as the target database. For a list of supported databases, see Creating MySQL connections.

5. In the **Migration options** step, select the **Object storage bucket in *compartment*:** transfer medium for initial load option.

   If the bucket is in a different compartment, click **Change Compartment** to look in another compartment.
6. If you want to create an online migration, check the **Use online replication** option to enable the replication of all data and metadata transactions from the source to the target database, committed after the initial load has begun. For additional optional configurations, see the **Replication** tab in the **Advanced options**. Optionally, you can set some additional properties which can affect the performance of your online migration.

> **Note**
>
> Currently, for MySQL to MySQL migrations, Oracle Cloud Infrastructure Database Migration supports GoldenGate Service Integration for replication.

> **Note**
>
> Skip this step for offline migrations.

7. Optionally, select **Advanced Options** to configure advanced replication, validation, and Oracle GoldenGate settings. The **Advanced Options** section allows you to optionally create tags.

   For details about these settings see Selecting objects for MySQL migration, Configuring initial load advanced options for MySQL migrations, Configuring validation options, and Configuring optional replication advanced options.

8. Click **Create**.

   The migration is loaded, and a new Migration Details page opens showing the information, metrics, and operations for the migration.

   The status of the creation operation is shown on top of the page along with the name of the migration. When the status is Active, you can run migration jobs with the migration.

   Alternatively click the **Save as stack** option to save the migrations as a stack.

   The **Save as stack** pane is displayed. You can save the migration resource configurations as a stack. Use the stack to install, configure, and manage the resource through the Resource Manager service.

   The following options are available:

   a. **Name** (Optional)
   b. **Description** (Optional)
   c. **Save in *your* compartment**
   d. Optionally, add **Tags** to organize your resources.
   e. Click **Save** to save as a stack.
   f. Click **Cancel** if you do not want to save the stack.

   After you save the Stack, it displays a link in the upper right corner that directs you to the newly created Stack in the Resource Manager. See the following topics for the next steps:

- Selecting objects for MySQL migration
  In Oracle Cloud Infrastructure Database Migration service, you can specify objects that are to be included or excluded from a MySQL migration.
- Configuring initial load advanced options for MySQL migrations
  To further tune performance, change the export modes, or select compatibility options, there are several settings that you can configure in the Migration resource Advanced Settings, Initial Load tab for MySQL migrations.

Selecting objects for MySQL migration

In Oracle Cloud Infrastructure Database Migration service, you can specify objects that are to be included or excluded from a MySQL migration.

When creating a migration, specify rules for selecting objects in the **Advanced Settings** on the **Selected Objects** tab.

You can choose either **Include** or **Exclude** from the **Action** list to specify if a rule should include or exclude the specified database objects in the migration.

If you do not define any rules for a **Full** migration type, then all the server schemas and objects are migrated. If you select a **Schema** migration type, then only the schema defined in the **Database name** when creating the source connection is migrated. For both the migration types, the **Objects and Schemas Excluded by Default** apply.

If you specify **Include** rules, the migration will only move the specified objects and their dependent objects; all other objects are automatically excluded.

When specifying **Exclude** rules, the migration will exclude the specified objects and their dependent objects; all other objects are included in the migration.

The following five schemas are always excluded by default, regardless of who generated the content:

- `information_schema`
- `mysql`
- `perormance_schema`
- `sys`
- `ndbinfo`

When performing an online migration, the Database Migration service creates a dynamic schema named `ggadmin_${dbName}`, where `dbName` is the name of the database that was defined for the source connection (for example, `ggadmin_employees`). This schema is used by Oracle GoldenGate to create the checkpoint and heartbeat tables, and is created on both the source and target databases. When the migration is complete, these schemas are deleted.

You can exclude the following object types for MySQL migrations: `USER, SCHEMA, TABLE, TRIGGER, EVENT, ROUTINE`

For `TRIGGER`, the object name can be defined in either of the following different ways:

- `tableName.triggerName`: This will limit the rule to just that one specific trigger.
- `tableName`: This will include all triggers from the defined table.

To create a rule, enter values for each of the following fields:

- **Schema** specifies the schema of the selected database objects. You can include/exclude objects from different schemas, you are not limited to having all rules defining the same schema. Moreover, wild characters are allowed in the form of a Java regular expression in this field.
- **Object name** specifies the name of selected database objects
- **Object Type** specifies the type of selected database objects.

You can filter **Schema** and **Object Name** fields using any valid regular expression as documented in Pattern. For example, you can enter `.*` in the **Object Name** field to select objects of any name.

The objects included in a migration are also influenced by the **Job Mode** of the initial load, as explained in Configuring initial load advanced options for MySQL migrations.

Note the following restriction: For `TRIGGER` and `USER` object types, regular expressions are not supported.

Configuring initial load advanced options for MySQL migrations

To further tune performance, change the export modes, or select compatibility options, there are several settings that you can configure in the Migration resource Advanced Settings, Initial Load tab for MySQL migrations.

- **Job mode:**
  - **Full** performs a full database export. It performs an export of all the schemas in the server.
  - **Schema** (default) exports the schema defined in the **Database name** option of the source connection.

  Specify schema objects for inclusion or exclusion in the Advanced Settings, Selected Objects tab. See Selecting Objects for MySQL migration for details.
- The following settings are available to fine-tune your MySQL migration:
  - **Enable consistent data dumps**: Enable consistent data dumps by locking the instance for backup during the dump.

    > **Note**
    >
    > This option is selected by default.

  - **Set the dump time zone to UTC** : Include a statement at the start of the dump to set the time zone to UTC.

    > **Note**
    >
    > This option is selected by default.

  - **Handle grant or revoke errors** : In the event of errors related to GRANT or REVOKE errors, select any of the following option:
    - **abort**: (default) stops the load process and displays an error.
    - **drop_account**: deletes the account and continues the load process.
    - **ignore**: ignores the error and continues the load process.
  - **Ignore existing target objects**: Import the dump even if it contains objects that already exist in the target schema in the MySQL instance.
- In **Compatibility options**, select the specified requirements for compatibility with OCI HeatWave Service for all tables in the dump output, altering the dump files as necessary:
  - **Primary key compatibility**: Select the primary key compatibility. The following options are available:

    - **None**: The default option.
    - **Create invisible primary keys**: Add a flag in the dump metadata to add primary keys in invisible columns, for each table that does not contain a primary key.
    - **Ignore missing primary keys**: Make the instance, schema, or table dump utility ignore any missing primary keys when the dump is carried out, so that the ocimds option can still be used without the dump stopping due to this check.
- **Force innodb**: Change CREATE TABLE statements to use the InnoDB storage engine for any tables that do not already use it.
- **Skip invalid accounts** : Remove user accounts created with external authentication plugins that are not supported in OCI HeatWave Service.
- **Strip definers**: Remove the DEFINER clause from views, routines, events, and triggers, so these objects are created with the default definer (the user invoking the schema), and change the SQL SECURITY clause for views and routines to specify INVOKER instead of DEFINER.
- **Strip restricted grants**: Remove specific privileges that are restricted by OCI HeatWave Service from GRANT statements, so users and their roles cannot be given these privileges (which would cause user creation to fail).
- **Strip tablespaces** : Remove the TABLESPACE clause from CREATE TABLE statements, so all tables are created in their default tablespaces.
- **Ignore wildcard grants**: If enabled, ignores errors from grants on schemas with wildcards, which are interpreted differently in systems where the partial_revokes system variable is enabled.
- **Strip invalid grants**: If enabled, strips grant statements which would fail when users are loaded.

Configuring validation options

Oracle Cloud Infrastructure Database Migration is integrated with the Oracle Cloud Pre-Migration Advisor tool. The tool analyzes the source database during validation and advises you about database features and constructs that are problematic.

The tool provides the following benefits:

- Warns you about any features used by your database that aren't supported in the target environment
- Makes suggestions for remedial changes and/or parameters to use for the Data Pump export and import operations

To configure premigration advisor settings:

When you are creating a migration you can configure premigration advisor settings in the Migration resource Advanced Settings, Validation tab.

Run premigration advisor during validation: Enables premigration advisor to run during a migration validation job, in the Validate Pre-migration Advisor phase

Continue premigration advisor validation on error: By default, a validation job stops running when premigration advisor finds an issue. When checked, if premigration advisor finds an error, the premigration advisor validation will continue to its conclusion.

This setting is useful if you want to proceed with a migration if error conditions have already been reviewed and the problematic objects have been excluded, because since premigration

advisor does not review the exclusions list, it will still report any blocking issues on objects even if they are excluded.

These settings can be changed after the migration is created. See Editing a migration.

See Cloud Premigration Advisor Tool Support in the Zero Downtime Migration documentation for more information about premigration advisor.

Configuring optional replication advanced options

In Oracle Cloud Infrastructure Database Migration, for online migrations using Oracle GoldenGate, you can configure some Oracle GoldenGate performance settings in the Migration resource Advanced Settings, Replication tab.

- **Acceptable lag (in seconds)** specifies the amount of lag. The lag is the time taken to extract or apply the data from the time it was created on the source database. This parameter specifies the amount of lag, in seconds, that triggers Oracle GoldenGate end-to-end latency monitoring. Monitoring continues until the lag time is lower than the specified value. The maximum value is 30 seconds, and the minimum is 2 seconds. The default value is 30 seconds.
- **Extraction settings**
  - **Performance profile** Sets the Oracle GoldenGate `PERFORMANCEPROFILE` parameter. Valid for GoldenGate Extract in Integrated Capture mode.
    * HIGH (default) for high volume use cases
    * MEDIUM
    * LOW RES to minimize resource usage for memory or resource constrained deployment

    This setting helps achieve better performance by tuning the group of Oracle GoldenGate parameters that affect performance. Once the performance profile is set up, this option automatically configures the relevant parameters to achieve the desired throughput and latency.
  - **Transaction maximum duration** specifies the length of time, in seconds, that a transaction can be open before Extract generates a warning message that the transaction is long-running. You can remove the value from this field if you don't want these error messages generated.
- **Replication settings**
  - **Performance profile** simplifies the Replicat performance.
    * Use HIGH when you have no concurrent workload on target. When HIGH is set, set Replicat Mappers to 5 and Appliers to 2 * PDB CPU_COUNT .
    * Use LOW when you have a concurrent workload on target. When LOW is set, set Replicat Mappers to 4 and Appliers to PDB CPU_COUNT / 2 on Target system.
- **Advanced parameters** lets you specify the advanced parameters for Replication settings for online migrations.
  While creating migration, select a list of parameters for migration. Select the **Parameter name** and **Parameter value** from the list. Click **+ Another parameter** to add the parameters.

  For more information, see see Zero Downtime Migration Logical Migration Response File Parameters Reference .

> **Note**
>
> This option is currently applicable for Oracle to Oracle migrations only.

- **GoldenGate instance**(**Optional**) **Use Marketplace GoldenGate instance**: Select this option if you want to perform replication using your own Marketplace GoldenGate instance provisioned by you in your tenancy.

  > **Note**
  >
  > Currently, the support for the **Use Marketplace GoldenGate instance** option is limited to Oracle to Oracle online migrations.

  > **Note**
  >
  > Oracle recommends using the **Use online replication** default option. Select the **Use Marketplace GoldenGate instance** option only when you want to use your own Marketplace GoldenGate compute instance.

  Enter the following details:

  – **GoldenGate instance OCID**: The instance ID of the compute that is hosting the Marketplace GoldenGate.

  – **GoldenGate hub URL**: Enter a URL containing only the public host name or IP address of your Marketplace GoldenGate instance.

  – **GoldenGate administrator username**: Enter the user name for connecting to your Marketplace GoldenGate instance.

  – **GoldenGate administrator password**: Enter the password for connecting to your Marketplace GoldenGate instance.

  > **Note**
  >
  > You must use `Marketplace` as the default name for your GoldenGate deployment.

8

Manage migrations

When you create a migration with Oracle Cloud Infrastructure Database Migration, you specify how the migration should run, select the source and target databases, and then configure the data transport settings. Optionally, you can configure advanced GoldenGate and Data Pump settings in the migration using the Database Migration console.

- Viewing migration details
  On the Migrations page of Oracle Cloud Infrastructure Database Migration service console, you can view a list of your migrations in a table, which includes the following information:
- Editing a migration
  You can modify some of the settings in a migration resource configuration in Oracle Cloud Infrastructure Database Migration.
- Cloning a migration
  To clone a migration:
- Moving a migration
  You can move a migration resource from one compartment to another.
- Deleting a migration
  Before you delete a migration, ensure that you carefully review any resources that reference the migration. If not, you could encounter errors.
- Managing tags for migrations
  Tags help you locate resources within your tenancy. In Oracle Cloud Infrastructure Database Migration, you can add and view a migration's tags from the Migrations page and from the Migration Details page.

Viewing migration details

On the Migrations page of Oracle Cloud Infrastructure Database Migration service console, you can view a list of your migrations in a table, which includes the following information:

- **Name**
- **State** of the migration resource, which can be any one of the following:
  - Creating: The new migration resource is being created in OCI.
  - Updating: Changes to the migration resource are being registered in OCI.
  - Active: The migration resource has finished being created or updated and is ready for validation. A migration resource in this state can be validated but cannot run a migration job.
  - In Progress: A validation job or migration job is currently running on this migration resource.
  - Accepted: The migration resource has been validated and can run another validation job or a migration job.
  - Succeeded: A migration job using this resource has completed successfully. Once a migration resource has reached this state, jobs can no longer be run with it.

  - Canceled: A migration job using this resource was canceled. You can run a new job on a migration resource in this state.
  - Waiting: A migration job using this resource is waiting for user input. This state appears when a migration job is paused.
  - Needs Attention: A validation job or migration job using this resource has failed and is blocked. Note that you must cancel a job before you can rerun it.
  - Inactive: A fallback state for unexpected errors.
  - Deleting: This state appears when you delete a migration resource. The resource remains in this state until deletion is completed, at which point the resource is no longer listed in the console.
  - Failed: There are problems with the migration resource. This can happen during creation, update, and any issues other than job failures. You can review the migration resource work request to investigate the issue.
- **Last Migration** shows the time stamp of the last job run with the migration
- **Created** time stamp when the migration was created

Select a migration from the Migrations page to view its details.

Viewing migration details: On the Migration Details page you can view the migration information, including:

- **OCID**: The resource's unique Oracle Cloud ID
- **Compartment**: The compartment where the migration resource resides
- **Created**: The date and time when the migration was created
- **Encryption Vault**: The link takes you to the Vault Details page
- **Encryption Key**: The link takes you to the Key Details page
- **Source Database**: The link takes you to the Database Details page. You can select **Test connection** to test the connectivity of the database connection.
- **Target Database**: The link takes you to the Database Details page. You can select **Test connection** to test the connectivity of the database connection.
- **Migration Type**: Online or Offline
- **Replication**: Enabled or Disabled
- **Validation**: Premigration advisor Enabled or Premigration advisor Disabled
- **Database type**

Excluded objects

The Excluded objects list displays objects that are excluded from migration.

Oracle Maintained: objects owned by Oracle-maintained users (ORACLE_MAINTAINED = Y) are excluded from migration

Unsupported: objects not supported for migration by Oracle GoldenGate, such as those owned by the ggadmin and c##ggadmin users, are excluded from migration

User Excluded: objects explicitly excluded by rules configured in your migration Selected Objects.

See Selecting objects for Oracle migration for details about objects excluded by default and explicitly selecting objects for migration.

Work requests

On the Migration Details page, under the migration information box you can find the Work Requests list. Work Requests lists any work requests sent to OCI to facilitate the creation, update, validation, cloning, or deletion of this resource. Click the work request to go to the Work Request Details page for more information about the work request.

Related Topics

- Overview of Notifications

Editing a migration

You can modify some of the settings in a migration resource configuration in Oracle Cloud Infrastructure Database Migration.

In the list of migrations on the Migrations page, select the Name of the migration you want to edit.

Select **Edit** next to any of the following modifiable settings:

Source Database

In the Edit Source Database dialog, you can choose a different source database.

Valid selection of source databases is the same as for when you create a new migration resource; Non-autonomous non-CDBs or PDB/CDB combinations are supported for the source.

Target Database

In the Edit Target Database dialog, you can choose a different target database.

Valid selection of target databases is the same as for when you create a new migration resource; Autonomous Databases are supported for the target.

Migration Type

The Migration Type (Offline or Online) cannot be changed, but you can change the settings that are valid for the migration type originally configured.

In the Edit Initial Load Settings dialog, you can choose to change the following settings.

- **Initial Load:** You can change the data transfer method to use Object Storage bucket, file storage, or database link.
- **Object Storage Bucket:** When your initial load data transfer method is Object Storage, you can change the bucket in which to store the Data Pump dumps. Enter the **Source Data Pump settings** and **Target Data Pump settings** if applicable.
- **Export Directory Object:** When your initial load data transfer method is Object Storage, you can change the export directory object by specifying a new name and path.
- **Advanced Options:** You can change the initial load advanced options. See Configuring initial load advanced options for Oracle migrations for information about these settings.

Replication

In the **Edit replication settings** dialog, you can enable or disable online replication, and you can change the following GoldenGate settings.

- **Use online replication**: Select this option if you want to enable the replication of all data and metadata transactions from the source to the target database committed after the initial load.
- **Acceptable lag (in seconds)**
- **Extraction settings**
- **Replication settings**
- **GoldenGate instance**:
  - (**Optional**). **Use Marketplace GoldenGate instance**: If you select this option, then enter the following details:
  - **GoldenGate instance OCID**: The instance ID of the compute that is hosting the Marketplace GoldenGate.
  - **GoldenGate hub URL**
  - **GoldenGate administrator username**
  - **GoldenGate administrator password**
- See Configuring optional replication advanced options for information about these settings.

Encryption Key

In the Edit Encryption Key dialog, you can choose a different Vault, Encryption Key, and the compartment in which to create a Secret.

Validation

In the Edit Validation Settings dialog, you can enable or disable the use of Pre-Migration Advisor Tool during migration validation, and you can change whether Pre-Migration Advisor validation continues on error.

Selected Objects

You can add or remove Include or Exclude database object rules in the **Resources** section, under **Selected Objects**, which is located below the Migration Information box.

To remove a rule, select the rule's checkbox and click **Remove**, or select the **Remove** action from the actions list.

To add or edit rules, click **Add Objects**. See Selecting objects for Oracle migration for information about configuring selected objects.

Cloning a migration

To clone a migration:

1. In the list of migrations on the Migrations page, select **Clone** from the Actions (three dots) menu of the migration you want to clone.

   You can also select **Clone** in the **More actions** menu on the Migration details page.
2. In the Clone Migration dialog, enter a unique **Name**, then click **Next** to update any of the source or target database details for the clone, and click **Clone** on the final page of the dialog.

Moving a migration

You can move a migration resource from one compartment to another.

To move a migration:

1. In the list of migrations on the Migrations page, select **Move Resource** from the Actions (three dots) menu for the migration you want to move.

   You can also select **Move Resource** on the Migration Details page.
2. In the **Move Resource to a Different Compartment** dialog, select the compartment to move the migration to from the dropdown.
3. Click **Move Resource**.

After you move the migration to the new compartment, inherent policies apply immediately and may affect access to the migration through the Console. For more information, see Managing Compartments.

Deleting a migration

Before you delete a migration, ensure that you carefully review any resources that reference the migration. If not, you could encounter errors.

To delete a migration:

1. In the list of migrations on the Migrations page, select **Delete** from the Actions (three dots) menu of the database you want to delete.

   You can also click **Delete** from the **More Actions** menu on the Migration Details page.
2. In the Delete dialog, click **Delete**.

Managing tags for migrations

Tags help you locate resources within your tenancy. In Oracle Cloud Infrastructure Database Migration, you can add and view a migration's tags from the Migrations page and from the Migration Details page.

On the Migrations page, from the migration's Actions (three dots) menu, select **Add Tags** or **View Tags**.

On the Migration Details page, you can select **Manage tags** in the **More actions** menu above the Migration Information box, or click the **Tags** tab to view and edit tags.

Learn more about tagging in Managing Tags and Tag Namespaces.

9

Manage jobs

In Oracle Cloud Infrastructure Database Migration, a migration job is the process of moving data from a source database to a target database. Run a validation pre-check on the migration before you run a job to ensure that it is configured properly. You can manage the jobs with several operations.

- Validating a migration
  Before you can run a job with a migration resource in Oracle Cloud Infrastructure Database Migration, the migration resource must be validated.
- Checking the Premigration Advisor Report
  Database Migration provides you with an interactive validation report with its integration with the Cloud Premigration Advisor Tool (CPAT). Using premigration advisor, Database Migration analyzes the source database during a migration job, and advises you about database features and constructs that are problematic, based on the specified Oracle Cloud target.
- Running a migration job
  After a migration resource is validated you can run migration jobs with it in Oracle Cloud Infrastructure Database Migration.
- Pausing and resuming a job
  When you start a migration job, you can configure it to pause at a specified phase, and then you can resume the job when you are ready.
- Viewing job details
  On the Migration Details page you can view a list of the jobs a migration resource has performed in Oracle Cloud Infrastructure Database Migration, which includes the following information:
- Viewing parameter files for online Oracle migrations
  When an online migration gets into **Needs Attention** or **Waiting** state due to a failed or a waiting migration job and the integrated GoldenGate is used, then you may choose to view and edit the Extract and Replicat parameter files and then resume the online migration.
- Editing parameter files for online Oracle migrations
  When an online migration gets into **Needs Attention** state or **Waiting** state due to a failed or a waiting migration job, using the GoldenGate Service integration, then you may choose to view and edit the Extract and Replicat parameter files and then resume the online Oracle migration.
- Monitoring job status
  In Oracle Cloud Infrastructure Database Migration, there are several places in the Console from which you can monitor the status of a migration job to varying degrees.
- Preparing for application switchover
  The following procedure ensures minimum data loss during a read-write application switchover.
- Abort a job
  You can abort or cancel a migration job while it is running or paused.
- Deleting a job
  To delete a job:

- Managing tags for jobs
  Tags help you locate resources within your tenancy. In Oracle Cloud Infrastructure Database Migration, you can add and view a migration job's tags from the Migration Details page and from the Job Details page.

Validating a migration

Before you can run a job with a migration resource in Oracle Cloud Infrastructure Database Migration, the migration resource must be validated.

1. In the list of migrations on the Migrations page, select **Validate** from the Actions (three dots) menu for the migration you want to validate.

   You can also select **Validate** on the Migration Details page.
2. In the **Validate Migration** dialog, click **Validate**.
3. Click **Jobs** on the details page to monitor the status of a validation job.

   In the Jobs table the validation job is listed with type **Evaluation**.

If you enabled the Premigration Advisor to run during the validation job, the Premigration Advisor report and details about any failed checks are found in the Job Details page. See Configuring validation options and Checking the Premigration Advisor Report for more information.

Checking the Premigration Advisor Report

Database Migration provides you with an interactive validation report with its integration with the Cloud Premigration Advisor Tool (CPAT). Using premigration advisor, Database Migration analyzes the source database during a migration job, and advises you about database features and constructs that are problematic, based on the specified Oracle Cloud target.

The premigration advisor runs by default and provides you with the following features and benefits:

- Warns you about any features used by your database that aren't supported in the target environment
- Makes suggestions for remedial changes and/or parameters to use for the Data Pump export and import operations
- Generates remedial scripts for failing checks that you can run against the source database

After a validation job runs, the job output displays the checks performed, descriptions of any problems, and actions you can take to resolve the issues.

To view or download the premigration advisor results

1. On a migration resource detail page, click **Jobs**, then the job name, then **Phases**.
2. Click the **Validate pre-migration advisor** phase name to open the Validation pre-migration advisor detail page.

   From this page you can download the premigration advisor report, view the report statistics, and drill down in the **Checks** list.

Checks List Operations

Filters: You can filter the checks listed using the **Filters** checkboxes on the left side.

Drill down on individual checks: Click a check name in the list to display details about that check from the premigration advisor report.

Check Details

The **View check details** panel shows you

- Information about the check, including the issue that caused the result shown, its potential impact on the migration, any action you can take to mitigate the issue, and if applicable, the location of a fixup script you can run on the source database.
- A **Reviewed** indicator, which lets you mark a check as "reviewed" so that you can see in the **Validate premigration advisor** list of checks whether you have completed whatever tasks you wanted to do with the check.

  Click the link in the check details to change it to a No or Yes value. The indicator does not have any impact on how the checks are processed; it is available for your convenience.
- A list of **Objects** that were flagged by the check as problematic.

  Some checks will show a read-only list of objects and some checks let you interactively update the objects listed.

Running Fixup Scripts

The location of the fixup script is shown on the **View check details** panel for an individual check.

The **Fixup script location** specifies where the script is located on the source database.

Running the fixup script against the source database requires sys admin privileges.

In multitenant architecture, fixup scripts should be run on the CDB. Running them on the PDB will produce an error.

Excluding Problematic Objects

After a validation run, every object listed in the **View check details** pane shows a No in the **Is excluded** column.

To exclude objects from the next validation run, you can check the boxes next to objects in the list and click **Exclude selected**, or you can choose **Exclude all** to exclude all objects that were listed in the check.

Any objects you choose to exclude will show a Yes in the **Is excluded** column. Objects marked Yes are ignored by premigration advisor in the next validation run, and they will not appear in the Objects list the next time you run the validation.

The migration's Selected Objects configuration is also updated to reflect this change. You will see a new row with the exclusion rule in the Selected Objects on the Migration details page. If you want to include this object again you must remove the rule in the Selected Objects list.

Note that **Exclude all** will exclude all of the objects on the page displayed, plus any objects from pages not displayed. However, if you check the first box in the checkbox column and click **Exclude all**, only the objects listed on the current page are excluded.

> **Note**
>
> Excluding tables does not exclude them from premigration advisor analysis. Schemas can be excluded from premigration advisor if the entire schema was excluded. The presence of an Oracle Cloud unsupported table can lead to Blocker status in the premigration advisor report.

See Selecting objects for Oracle migration for information about explicitly including and excluding objects.

Running a migration job

After a migration resource is validated you can run migration jobs with it in Oracle Cloud Infrastructure Database Migration.

1. In the list of migrations on the Migrations page, select **Start** from the Actions (three dots) menu for the migration you want to run.

   You can also select **Start** on the Migration Details page.
2. (Optional) In the confirmation dialog, select a phase after which to require user input to continue the migration job.
3. Click **Start**.

Pausing and resuming a job

When you start a migration job, you can configure it to pause at a specified phase, and then you can resume the job when you are ready.

When you start a migration job, a confirmation dialog opens, and there you can configure the job to pause at any point by selecting a phase in **Require User Input After**.

When the phase you selected to pause after completes, the job will enter a **Waiting** state until you resume (or terminate) the job.

If you pause after the phase Monitor Replication Lag, the transaction replication continues during the Waiting state. It will stop upon resume.

To start the job where it left off, do one of the following:

- In the Jobs table on the Migration Details page, select **Resume** from the Actions (three dots) menu for the job you want to resume.
- Click **Resume** on the Job Details page.

At this time you can select another phase after which to pause the job again.

Viewing job details

On the Migration Details page you can view a list of the jobs a migration resource has performed in Oracle Cloud Infrastructure Database Migration, which includes the following information:

- **Name**: Name of the job
- **State**: State of the job resource, which can be any one of the following:

  - Accepted: The job is pending execution.
  - In Progress: The validation job or migration job is currently running.
  - Unknown: The status cannot be retrieved and Database Migration is waiting for recovery to continue. No action is required.
  - Terminated: Unfeasible to reach, similar to Canceled.
  - Failed: For a validation job, the precheck failed, no user action required. For a migration job, the job failed and it is waiting for user action (abort or resume the job).
  - Succeeded: The job has completed successfully.
  - Waiting: A migration job is paused and waiting for user input.
  - Canceling: A migration job is in the process of being stopped following an Abort action.
  - Canceled: A migration job has stopped following an Abort action. You can run a new job on a migration resource once a job reaches this state.
- **Type**: Migration or Evaluation
- **Status Details:** Displays a supporting message with additional details regarding the State.

  For example, if a validation job fails, with State=Failed, Status Details informs you that the job failed and Database Migration auto-terminated the job.

  If the Unknown state is displayed, a Status Details informs you that Database Migration is unable to retrieve the latest status of the job and is waiting for downtime recovery.
- **Created**: Time stamp when the job was created

Select a job from the Jobs table to view its details.

On the job details page you can view the job information, including:

- **OCID**: The resource's unique Oracle Cloud ID
- **Created**: The date and time when the migration was created

On the job details page you can also view the job type information, including:

- **Migration**
- **Compartment**
- **Type**

On the job details page you can select the **Actions** menu to perform the following tasks:

- **Resume**
- **Abort**
- **Add tags**
- **Delete**

In the Job details page, you can also view information about resources associated with the job, such as Metrics (see Metrics), Phases (see Oracle migration phases), and Unsupported Objects.

> **Note**
>
> Migration jobs that enter in a **Failed** state can be resumed within 14 days of the failure. If they are not resumed within this period, they automatically transition to **Canceled** state and cannot be resumed.

Viewing parameter files for online Oracle migrations

When an online migration gets into **Needs Attention** or **Waiting** state due to a failed or a waiting migration job and the integrated GoldenGate is used, then you may choose to view and edit the Extract and Replicat parameter files and then resume the online migration.

> **Note**
>
> This feature is applicable only when you use the inbuilt GoldenGate.

Job Details - Phases

You can view the GoldenGate process (Extract, Replicat) parameter files if they have been created to set up replication for your online migration. On a migration resource detail page, you can click **Jobs**, then the job name, then **Phases**, and see the phases of the job that have **Failed** or **Wait** state.

Click the **View parameter files** option to view the current replication (Extract, Replicat) parameter files.

> **Note**
>
> You may or may not be able to view both the Extract, Replicat parameter files depending on the current state of the replication setup for your online migration. You can view the parameter file during or after the **Prepare** phase is concluded. If the parameter files are not created, then you cannot view them. Similarly, you can view the Replicat parameter files during or after the **Prepare replication target** phase.

The **Parameter files** panel displays the parameter files currently in use.

- Details such as Type and creation details.
- Click **View parameter file** to view the Extract or Replicat files in read-only mode.

You can modify the relevant GoldenGate parameter file for the **Failed** or **Wait** phase.

> **Note**
>
> This capability is applicable only for some phases wherein the file is displayed only if the GoldenGate Extract or Replicat is available.

After modifying the parameter files, resume the online migration in this **Failed** or **Wait** state.

Editing parameter files for online Oracle migrations

When an online migration gets into **Needs Attention** state or **Waiting** state due to a failed or a waiting migration job, using the GoldenGate Service integration, then you may choose to view and edit the Extract and Replicat parameter files and then resume the online Oracle migration.

> **Note**
>
> You can only view the Extract/Replicat parameter files, if the corresponding process Extract/Replicat is **Active**. You can edit only if the respective process Extract/Replicat is not active at that time and the migration is in the **Needs Attention** state and the migration job is in **Failed** state, or when the migration is in a **Waiting** state.
>
> This capability is applicable only for some phases such as **Prepare, Prepare replication target, Monitor replication lag, and Switchover**, wherein the file is displayed only if the GoldenGate Extract or Replicat is available.

The Oracle Cloud Infrastructure Database Migration service provides you a comprehensive feature to edit your replication parameter files whenever applicable during the migration process. It has the following features:

- You can view and edit the current Extract or Replicat parameter file belonging to the replication setup for your online migration.
- Any updates that you perform to the replication parameter files are versioned for your convenience.
- Whenever you edit on the Extract or Replicat parameter files, the original factory default version is always retained for you to revert for your reference.
- When you perform edits to either Extract or Replicat parameter files, after saving it, it is immediately applied to the replication setup.
- You can resume the migration after saving the file, from the failed process.

Job Details - Phases

On a migration resource detail page, you can click **Jobs**, then the job name, then **Phases**, and see the phase of the job that has failed.

Click the **Edit parameter file** option to view the current GoldenGate Extract and Replicat parameter files.

> **Note**
>
> You can edit the parameter depending on the phase and the current state of the GoldenGate process (Extract or Replicat).

The **Parameter files** panel displays the parameter files currently in use.

1. Details such as Type and creation details.
2. Click **Edit parameter file** to edit the Extract or Replicat files.
3. Edit the parameter file using a free-form editor.

4. If you perform an invalid parameter file edit, then GoldenGate rejects the edit with the error displayed on the screen.
5. Click **Save as new and make current** option to save this new version of the file and access it from the list available in the **Show all files** option. The new version is saved and applied to GoldenGate and marked as **In use**.
6. Click **Cancel** if you do not want to save the edited file.

Click the **Show all files** option to view the details of the files such as **Name**, **Type**, and **Created** details. The file currently being used is shown as **In use**. The following types of files are available in this option:

- You can switch between any of these two edited versions or the original (factory) parameter file.
- The original (factory) parameter file is provided by Oracle.
- From the action menu (three dots), select **Edit and make current** to switch between the files. You can view, edit, and save the file or you can view and save the file.
- You can switch back to the factory (original) parameter file.
- The original (factory) parameter file provided by Oracle is never deleted and you can use it when needed.

You can modify the relevant GoldenGate parameter file for the failed phase.

After modifying the parameter files, resume the online migration in this failed state.

Monitoring job status

In Oracle Cloud Infrastructure Database Migration, there are several places in the Console from which you can monitor the status of a migration job to varying degrees.

Job Details page

On the Migration Details page, you can click **Jobs**, and see information about the jobs run with this resource.

You can click **Download Log** to view the log generated by Database Migration Service.

Below the job information details you can click the **Metrics**, **Phases**, and **Unsupported Objects** to get more information about the migration job.

Job Details - Phases

On a migration resource detail page, you can click **Jobs**, then the job name, then **Phases**, and see the phases of the job that have completed, are pending, and are currently running (labeled **Started**).

For any phase that has some error or warning text (or log) to display, the phase name is displayed as a clickable hyperlink, and the phase's action items menu includes **View Details**. Clicking on the phase name link or **View details** action opens a **View phase details** panel which displays the details of the error or warning

If there is a log available for download, an option appears in the action menu to allow download. You can also download logs using a button in the **View phase details** panel. The **View phase details** panel displays the following information:

- **Name**: Name of the phase
- **Status**: Status of the phase. A failed status informs you that the job failed.

- **Duration**: Time elapsed from the beginning of the phase till the point where the error occurred.
- **Issue**: The issue causing the job failure.
- **Action**: The action you can perform to resolve the error.
- **Error messages**: The validation or migration errors causing the job failure.

Download Log downloads the Data Pump log. Applies to jobs run in migration mode in the Export Initial Load and Import Initial Load phases.

The **Validate pre-migration advisor** phase is a special case, because this link opens the Validation pre-migration advisor detail page, from which you can download the Pre-migration Advisor report, view the report statistics, and drill down in the **Checks** list.

> **Note**
>
> In case your migration job fails for the **Initialize replication infrastructure** phase, the job status is set as cancelled.

For more information about the report and interactive details page, see Checking the Premigration Advisor Report.

For details about the migration work flow phases, see Oracle migration phases.

For information about metrics, see Metrics.

- Oracle migration phases
  A migration job in Oracle Cloud Infrastructure Database Migration runs in operational phases as a work flow.
- MySQL migration phases
  A migration job in Oracle Cloud Infrastructure Database Migration runs in operational phases as a work flow.

Oracle migration phases

A migration job in Oracle Cloud Infrastructure Database Migration runs in operational phases as a work flow.

The phases in Database Migration are shown in the console with user-friendly names (DMS Phase), and in the REST API with the codes prefixed with "ODMS_", as shown in the table below.

Note that Database Migration harnesses the Zero Downtime Migration tool to run the migration job work flow, so in the logs the migration phase names will have a "ZDM_" prefix. Also note that one Database Migration phase corresponds to one or more Zero Downtime Migration phases, which will give you a more granular look at the work flow.

Table 9-1 Database Migration Process Phase Descriptions

DMS Phase Name Console (API Codes)	Description	ZDM Phase Name	Description
Validate (ODMS_VALIDATE)	Performs validation of the source and target database, the GoldenGate Hub, and Data Pump configuration.	Validate Source (ZDM_VALIDATE_SRC)	Validates the source database access credentials, database parameter settings, ggadmin user privileges, and GoldenGate capture support for objects in source database
		Validate Target (ZDM_VALIDATE_TGT)	Verifies that the target database exists and that the database type is Autonomous Database, and validates access credentials, security, and connectivity.
		Validate Pre-migration Advisor (ZDM_PRE_MIGRATION_ADVISOR)	Cloud Pre-Migration Advisor Tool is run.
		Validate GoldenGate Hub (ZDM_VALIDATE GG_HUB)	Verifies GoldenGate Microservices REST endpoints, software configuration, health, and connectivity to the source and target databases.
		Validate Datapump Source Settings (ZDM_VALIDATE_DATAPUMP_SETTINGS_SRC)	Validates the export directory object (if applicable), and checks for sufficient space and permission for specified user in the source database to export dumps. Checks if the specified Oracle Cloud Object Store buckets, data bucket, and wallet bucket are accessible from the source. Also validates the proxy configuration if applicable.
		Validate Datapump Target Settings (ZDM_VALIDATE_DATAPUMP_SETTINGS_TGT)	Verifies that the Data Pump import directory object exists. If a pre-existing DBLINK was specified, checks if it exists and is valid, and ensures that the ADB requirements for the DBLINK and wallet files are met.

Table 9-1 (Cont.) Database Migration Process Phase Descriptions

DMS Phase Name Console (API Codes)	Description	ZDM Phase Name	Description
Prepare (ODMS_PREPARE_REPLICATION)	Prepares for and starts the GoldenGate Extract process, and enables supplemental logging	Prepare GoldenGate Hub (ZDM_PREPARE_GG_HUB)	Registers database connection details and credentials with GoldenGate Microservices.
		ZDM_ADD_HEARTBEAT_SRC	Creates GoldenGate heartbeat table in the source database. If the table already exists, sets update frequency to 60 seconds.
		ZDM_ADD_SCHEMA_TRANDATA_SRC	Prepares the source database schemas for instantiation by enabling schema level supplemental logging.
		Create GoldenGate Source Extract (ZDM_CREATE_GG_EXTRACT_SRC)	Starts the GoldenGate Extract process at the source database
Prepare (ODMS_PREPARE_INITIAL_LOAD)	Creates any necessary directory objects for Data Pump, and creates a DBLINK, if applicable.	Prepare Source Datapump (ZDM_PREPARE_DATAPUMP_SRC)	Creates a new directory object for Data Pump, if required. Creates OCI Auth Token to access OCI OSS bucket if required.
		Prepare Target Datapump (ZDM_PREPARE_DATAPUMP_TGT)	Creates a new directory object for Data Pump, if required. Stores OCI Auth token in the database for secure OSS access. If migrating via DBLINK, and a DBLINK must be created, creates the necessary database credentials to access the source and create a new DBLINK. Ensures Autonomous Database security requirements are met using DBLINK over SSL.
Datapump Source Export (ODMS_INITIAL_LOAD_EXPORT)	Starts and monitors the Data Pump Export on the source database.	Datapump Source Export ZDM_DATAPUMP_EXPORT_SRC)	Starts and monitors the Data Pump Export on the source database.
Upload Source Dump Files (ODMS_DATA_UPLOAD)	Uploads Data Pump dump files from the source to OCI OSS.	Upload Source Dump Files (ZDM_UPLOAD_DUMPS_SRC)	Uploads Data Pump dump files from the source to OCI OSS.

Table 9-1 (Cont.) Database Migration Process Phase Descriptions

DMS Phase Name Console (API Codes)	Description	ZDM Phase Name	Description
Datapump Target Import (ODMS_INITIAL_LOAD_IMPORT)	Starts import of Data Pump Dumps to the target database, either from the OCI OSS bucket or via DBLINK, and monitors the Data Pump import progress.	Datapump Target Import (ZDM_DATAPUMP_IMPORT_TGT)	Starts import of Data Pump Dumps to the target database, either from the OCI OSS bucket or via DBLINK, and monitors the Data Pump import progress.
Post Datapump (ODMS_POST_INITIAL_LOAD)	Removes directory objects, access credentials, and DBLINK that were created for Data Pump by Database Migration.	Post Source Datapump (ZDM_POST_DATAPUMP_SRC)	Removes any Data Pump directory object created by Database Migration.
		Post Target Datapump (ZDM_POST_DATAPUMP_TGT)	Fixes any invalid objects in the target database. Removes the database access and OCI OSS access credentials that were created for the migration. Removes any DBLINK created by Database Migration. Optionally, removes source database dumps stored in OCI OSS bucket.
Prepare GoldenGate (ODMS_PREPARE_REPLICATION_TARGET)	Prepares for GoldenGate replication.	ZDM_ADD_HEARTBEAT_TGT	Creates the GoldenGate heartbeat table in the target database. If the table already exists, sets update frequency to 60 seconds.
		ZDM_ADD_CHECKPOINT_TGT	Creates GoldenGate checkpoint table in the target database to track Replicat progress.
		ZDM_CREATE_GG_REPLICAT_TGT	Creates GoldenGate Replicat process for the target database.
		ZDM_START_GG_REPLICAT_TGT	Starts GoldenGate Replicat process for the target database.
Monitor GoldenGate Lag (ODMS_MONITOR_REPLICATION_LAG)	Polls the GoldenGate checkpoint and heartbeat data to measure end-to-end apply lag until lag decreases below desired threshold.	Monitor GoldenGate Lag (ZDM_MONITOR_GG_LAG)	Polls the GoldenGate checkpoint and heartbeat data to measure end-to-end apply lag until lag decreases below desired threshold.

Table 9-1 (Cont.) Database Migration Process Phase Descriptions

DMS Phase Name Console (API Codes)	Description	ZDM Phase Name	Description
Switchover App (ODMS_SWITCHOVER)	If the source database is idle, stops GoldenGate Extract, waits for GoldenGate Replicat to complete apply, and stops GoldenGate Replicat.	Switchover App (ZDM_SWITCHOVER_APP)	If the source database is idle, stops GoldenGate Extract, waits for GoldenGate Replicat to complete apply, and stops GoldenGate Replicat.
Cleanup (ODMS_CLEANUP)	Performs cleanup operations such as deleting GoldenGate Extract and GoldenGate Replicat processes and connection details on source and target database respectively, removing Autonomous Database access to wallet, and so on.	Post Switchover (ZDM_POST_SWITCHOVER_TGT)	Performs post-switchover actions for the target database.
		ZDM_RM_GG_EXTRACT_SRC	Deletes GoldenGate Extract process on source database
		ZDM_RM_GG_REPLICAT_TGT	Deletes GoldenGate Replicat process on target database
		ZDM_DELETE_SCHEMA_TRANDATA_SRC	Disables schema level supplemental logging on source database
		ZDM_RM_HEARTBEAT_SRC	Drops the GoldenGate heartbeat table in source database, if the table was created by Database Migration. Otherwise, resets update frequency to original setting.
		ZDM_RM_CHECKPOINT_TGT	Drops the GoldenGate checkpoint table in the target database.
		ZDM_RM_HEARTBEAT_TGT	Drops the GoldenGate heartbeat table in the target database, if the table was created by Database Migration. Otherwise, resets the update frequency to the original value.
		Clean GoldenGate Hub (ZDM_CLEAN_GG_HUB)	Deletes the database connection details and credentials saved with GoldenGate Microservices
		Post Actions (ZDM_POST_ACTIONS)	Removes Autonomous Database access wallet from Database Migration.

Table 9-1 (Cont.) Database Migration Process Phase Descriptions

DMS Phase Name Console (API Codes)	Description	ZDM Phase Name	Description
		ZDM_CLEANUP_SRC	Deletes the Cloud Pre-Migration Advisor Tool related binaries on source database servers.

MySQL migration phases

A migration job in Oracle Cloud Infrastructure Database Migration runs in operational phases as a work flow.

The phases in Database Migration service are shown in the console with user-friendly names (DMS Phase), and in the REST API with the codes prefixed with "ODMS_", as shown in the table below.

Table 9-2 MySQL Validation/Evaluation Job Phases (offline and online)

Phase Name Console (API Codes)	Phase enum value	Description
Validate source	ODMS_VALIDATE_SRC	Validates the source database connectivity.
Validate target	ODMS_VALIDATE_TGT	Validates target database connectivity
Validate source metadata	ODMS_FETCH_METADATA_SRC	Validates source database basic configuration
Validate target metadata	ODMS_FETCH_METADATA_TGT	Validates target database basic configuration
Run premigration advisor	ODMS_VALIDATE_PREMIGRATION_ADVISOR	Runs MySQL Evaluator (Export/Import and GoldenGate checks)

Table 9-3 MySQL Migration Job Phases (offline)

Phase Name Console (API Codes)	Phase enum value	Description
Validate	ODMS_VALIDATE	Validates the source and target database connectivity.
Export initial load and upload data	ODMS_INITIAL_LOAD_EXPORT_DATA_UPLOAD	Perform MySQL source dump to object storage.
Import initial load	ODMS_INITIAL_LOAD_IMPORT	Perform MySQL target import from object storage
Validate target configuration	ODMS_FETCH_METADATA_TGT	Validates target database basic configuration
Cleanup	ODMS_CLEANUP	Cleanup temporary artifacts, object storage

Table 9-4 MySQL Migration Job Phases (online GGS)

Phase Name Console (API Codes)	Phase enum value	Description
Initialize replication infrastructure	ODMS_INITIALIZE_REPLICATION_INFRASTRUCTURE	Setup GGS deployment, connections, networking
Validate	ODMS_VALIDATE	validates source and target database connectivity.
Export initial load and upload data	ODMS_INITIAL_LOAD_EXPORT_DATA_UPLOAD	Perform MySQL source dump to object storage
Import initial load	ODMS_INITIAL_LOAD_IMPORT	Perform MySQL target import from object storage
Prepare	ODMS_PREPARE	Setup GoldenGate replicat to apply to the target
Monitor replication lag	ODMS_MONITOR_REPLICATION_LAG	Monitor GoldenGate replication lag. The replication lag needs to be a value equal or lower than the one defined when creating the migration before moving to the next phase.
Switchover	ODMS_SWITCHOVER	Stop the replication based on the switchover logic. Extract and Replicat are monitored to ensure that there are no pending transactions. After confirmation, Extract and Replicat are stopped.
Cleanup	ODMS_CLEANUP	Cleanup temporary artifacts, object storage

Preparing for application switchover

The following procedure ensures minimum data loss during a read-write application switchover.

> **Note**
>
> This process is currently applicable for Oracle to Oracle online migrations.

When both the source and target databases are open in a read-write mode during the logical migration work flow, the following conditions apply:

- For **read-only** applications, switchover can happen immediately after GoldenGate Replicat has applied all outstanding source transactions, allowing for zero application downtime for those services.
- **Read-write** applications require assurance that all transactions have been applied on the target before switching the application over to ensure minimum data loss.

If you have an active source database with a lot of active transactions and if your application is read-write, then you must do the following to ensure minimum data loss:

1. If your source database is occupied with the application usage, incoming transactions, you can decide when to do the application switchover. OCI Database Migration Service ensures that the data is replicated from where you resume the migration.

Pause the migration job after the **Monitor replication lag** phase.

This phase monitors Oracle GoldenGate Extract and Replicat operations until Replicat has caught up on the target database.

2. After the **Monitor replication lag** phase completes and the migration job pauses, stop the workload on the source database (start of downtime).
3. You can initiate a switchover by resuming a migration which is at a **Waiting** state after the monitor replication phase.

> **Note**
>
> OCI Database Migration Service pauses a migration after the monitor replication phase. This gives you an option to resume the migration. However, if you do not pause the migration, OCI Database Migration Service automatically performs the switchover. When you resume the migration after the **Monitor replication lag** phase, the Switchover phase will be initiated.

> **Note**
>
> For production loads, Oracle recommends you to pause the migration after the **Monitor replication lag** phase.

When you **Resume** the migration, OCI Database Migration Service performs the following tasks:

a. Inserts a barrier transaction in the source database.
b. Actively monitors the Extract and Replicat processes.
c. Shuts off the Extract & Replicat processes.
d. Checks if the barrier transaction was replicated to the Target database. This will also trigger the object types such as reload sequences, audit trails, materialized views, unsupported objects by Oracle GoldenGate, and the tables that you want to deliberately exclude from replication will be reloaded.

4. After phase **Switchover** has completed, you can start the workload on the target database (end of downtime).

Abort a job

You can abort or cancel a migration job while it is running or paused.

1. In the list of jobs on the Migration Details page, select **Abort** from the Actions (three dots) menu of the job you want to delete.

   You can also click **Abort** on the Job Details page.
2. In the Abort Job dialog, click **Abort**.

Deleting a job

To delete a job:

1. In the list of Jobs on the Migration Details page, select **Delete** from the Actions (three dots) menu of the job you want to delete.

   You can also click **Delete** on the Job Details page.

2. In the Delete dialog, click **Delete**.

Managing tags for jobs

Tags help you locate resources within your tenancy. In Oracle Cloud Infrastructure Database Migration, you can add and view a migration job's tags from the Migration Details page and from the Job Details page.

On the Migration Details page, select Jobs under Resources.

In the Jobs table, from the job's Actions (three dots) menu, select **Add Tags** or **View Tags**.

On the Job Details page, you can select **Manage tags** above the Job Information box, or click the **Tags** tab to view and edit tags.

Learn more about tagging at Managing Tags and Tag Namespaces.

10

Troubleshoot

Depending on the type of issue you may encounter while using Oracle Cloud Infrastructure Database Migration, you can use the Metrics graphs in the OCI Console, the log and trail files found in the manual backup, or logs from the Migrations page to help determine the root cause and update your migration configuration.

- Metrics
  Metrics are collected every five minutes for each deployment. The data produced can help you troubleshoot issues that you may encounter.
- Alarms
  For each metric on the Details page, you can create an alert to inform you when a condition is met. For example, you can create an alarm to notify you when OCPU consumption is less than 50%.
- Logs
  The Database Migration Jobs Details page provides detailed error information and access to logs for troubleshooting performance.
- Error messages
  Error message are reported in the Jobs output log in Oracle Cloud Infrastructure Database Migration.
- Work request details
  The Database Migration Jobs Details and Migration Details pages provide a resource work request list.
- Connection creation failures
  In Oracle Cloud Infrastructure Database Migration, if you have a Connection (odms-connection) resource in failed state use the following steps to gather more details about the failure.
- Connectivity issues
  Use this feature to test the connectivity before you create or start the migration.
- Migration creation failures
  In Oracle Cloud Infrastructure Database Migration, if you have a Migration (odms-migration) resource in failed state use the following steps to gather more details about the failure:

Metrics

Metrics are collected every five minutes for each deployment. The data produced can help you troubleshoot issues that you may encounter.

For more information, see Metrics.

Alarms

For each metric on the Details page, you can create an alert to inform you when a condition is met. For example, you can create an alarm to notify you when OCPU consumption is less than 50%.

To create an alarm:

1. From the **Options** dropdown of a metric chart, select **Create an Alarm on this Query**.
2. On the **Create Alarm** page, under **Define Alarm**, add the trigger.
3. For **Alarm Settings**, complete the following fields as needed:
   - **Alarm Name**: Enter the name that serves as the title for notifications related to this alarm. Avoid entering confidential information.
   - **Alarm Severity**: Select the perceived type of response required when the alarm is in the firing state.
   - **Alarm Body**: Enter the content of the notification to deliver.
   - **Tags** (optional): Select or enter free-form tags to apply to this resource.
   - **Metric description**: The metric to evaluate for the alarm condition.
     - **Compartment**: Select the compartment that contains the resources that emit the metrics evaluated by the alarm. The selected compartment is also where the alarm is stored.
     - **Metric Namespace**: Enter the service or application emitting metrics for the resources that you want to monitor.
     - **Resource Group** (optional): Select the group that the metric belongs to.
     - **Metric Name**: Enter the name of the metric. Only one metric can be specified.
     - **Interval**: Select the aggregation window, or the frequency at which data points are aggregated.
     - **Statistic**: Select the aggregate function.
4. Confirm the values for **Metric dimensions**. Optionally, click **+ Additional dimension** to add another dimension to the alarm.
5. For **Trigger rule**, complete the **Operator**, **Value**, and **Trigger delay minutes** fields. The graph displays the boundaries for which the alarm triggers a notification.
6. For **Notifications**, complete the fields as needed:
   - For **Destination service**, select **Notifications Service**.
   - For **Compartment**, select the compartment to store the topic used for this notification.
   - For **Topic**, click **Create topic** to set up a topic and subscription protocol in the designated compartment using the designated Destination service.
   - (Optional) Click **+ Additional destination service** to add another destination service.
   - (Optional) Enable **Repeat Notification** and select **Notification Interval** if you want the alarm to resend notifications at the specified intervals when the alarm is in the firing state.
   - (Optional) Enable **Suppress Notifications** to specify a window of time to suspend evaluations and notifications. This is useful for maintenance periods.
7. Click **Save alarm**.

For more information, see Viewing Default Metric Charts.

Logs

The Database Migration Jobs Details page provides detailed error information and access to logs for troubleshooting performance.

Database Migration Service Job Log

Database Migration Service generates a log during every migration job.

On the Migration Details page, click **Jobs**, and then click **Download Log** to download the log.

Data Pump Logs

On the Migration Details page, click **Jobs**, and then click **Phases** under the Resources section.

Should a problem arise during the Data Pump Export or Data Pump Import phases of a migration job, the phase name is displayed as a clickable hyperlink.

There are two ways to access the log:

- Click the action menu (three dots) and click **Download Datapump Log**
- Click the phase name, which opens the **View Details** panel, on which you can find the **Download Datapump Log** button.

Error messages

Error message are reported in the Jobs output log in Oracle Cloud Infrastructure Database Migration.

If you see errors reported in Job output log, such as ORA and PR*, to understand the cause and action for these errors you can look up the error by code in Oracle Database Error Messages reference at Database Error Messages.

Work request details

The Database Migration Jobs Details and Migration Details pages provide a resource work request list.

The Work Requests list allows you to monitor long-running operations such as resource creation, update, validation, cloning, or deletion. Click the work request in the list to go to the Work Request Details page and view more detailed information.

For more information about OCI work requests, see Work Requests

Connection creation failures

In Oracle Cloud Infrastructure Database Migration, if you have a Connection (odms-connection) resource in failed state use the following steps to gather more details about the failure.

1. Get the connection OCID.
   - Using the OCI Web Console:

Go to the OCI Console, and open Database Migration Service/Database Connections (for example, https://console.us-phoenix-1.oraclecloud.com/odms/registrations).

Locate the failed Connection in the list, and select **Copy OCID** from the Actions (three dots) menu.

- Using Oracle Database Migration Service (database-migration) OCI Command line interface:

  The compartment OCID where the failed connection was created is required.

  – List all connection resources and find the failure:

    ```
    oci database-migration connection list -c `compartmentOCID`
    ```

  – If you know the connection display name, you can use it to filter results:

    ```
    oci database-migration connection list -c `compartmentOCID` --
    display-name `connectionDisplayName`
    ```

2. Get the Work Request OCID associated with the connection creation request.

   - Using Oracle Database Migration Service (database-migration) OCI Command line interface:

     The compartment OCID where the failed connection was created is required. You can find it using database-migration >> work-request >> list as shown in the following example.

     – Use the connection OCID to list the work requests:

       ```
       oci database-migration work-request list --resource-id odms-
       connection-OCID --compartment-id odms-connection-compartment-OCID
       ```

     – Use the compartment OCID to list the work requests of the compartment:

       ```
       oci database-migration work-request list --compartment-id odms-
       connection-compartment-OCID
       ```

     – Use the `sort-by` option to sort the results by `displayName` or `timeCreated`:

       ```
       oci database-migration work-request list --compartment-id odms-
       connection-compartment-OCID --sort-by displayName
       ```

     – Use the `sort-order` option with `asc` or `desc`:

       ```
       oci database-migration work-request list --compartment-id odms-
       connection-compartment-OCID --sort-oder ASC
       ```

       Only one sort order can be specified. Default order for `--sort-by timeCreated` is descending.

   - Using the Database Migration REST API:

     – Use the connection OCID to list the work requests:

See ListWorkRequests

```
GET /20210929/workRequests?resourceId=odms-connection-OCID&compartmentId=odms-connection-compartment-OCID
```

- Use the compartment OCID to list the work requests of the compartment:

  ```
  GET /20210929/workRequests?compartmentId=compartment-OCID-of-resource
  ```

- Use the `sort-by` option to sort the results by `displayName` or `timeCreated`:

  ```
  GET /20210929/workRequests?compartmentId=compartment-OCID-of-resource&sortBy=displayName
  ```

- Use the `sort-order`option with `asc` or `desc`:

  ```
  GET /20210929/workRequests?compartmentId=compartment-OCID-of-resource&sortOrder=ASC
  ```

  Only one sort order can be specified. Default order for `--sort-by timeCreated` is descending.

3. Use the work request OCID to get details, logs, and errors related to the failure:
   - Using Oracle Database Migration Service (database-migration) OCI Command line interface:
     - Use the work request identifier to get details:

       ```
       oci database-migration work-request get --work-request-id workRequestId
       ```

       See database-migration >> work-request >> get
     - Use the work request identifier to list the errors:

       ```
       oci database-migration work-request-error list --work-request-id workRequestId
       ```

       See database-migration >> work-request-error >> list
     - Use the work request identifier to list the logs:

       ```
       oci database-migration work-request-logs list --work-request-id workRequestId
       ```

       See database-migration >> work-request-logs >> list
   - Using the Database Migration REST API:
     - Use the work request identifier to get the details:

       ```
       GET /20210929/workRequests/{workRequestId}
       ```

       See GetWorkRequest

- Use the work request identifier to get the errors:

  ```
  GET /20210929/workRequests/{workRequestId}/errors
  ```

  See ListWorkRequestErrors
- Use the work request identifier to get the logs:

  ```
  GET /20210929/workRequests/{workRequestId}/logs
  ```

  See ListWorkRequestLogs

4. Inspect the logs and errors in the work request and resolve the issues reported.

For more information about using the API and signing requests, see REST APIs and Security Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

Connectivity issues

Use this feature to test the connectivity before you create or start the migration.

See Testing Connectivity of a Database Connection for more information.

Migration creation failures

In Oracle Cloud Infrastructure Database Migration, if you have a Migration (odms-migration) resource in failed state use the following steps to gather more details about the failure:

1. Get the migration OCID.
   - Using the OCI Web Console:

     Go to the OCI Console, and open Database Migration Service/Migrations (for example, https://console.us-phoenix-1.oraclecloud.com/odms/migrations).

     Locate the failed Migration in the list, and select **Copy OCID** from the Actions (three dots) menu.
   - Using Oracle Database Migration Service (database-migration) OCI Command line interface:

     The compartment OCID where the failed migration was created is required.
     - List all migration resources and find the failure:

       ```
       oci database-migration migration list -c `compartmentOCID`
       ```

       See database-migration >> migration >> list
     - If you know the migration display name, you can use it to filter results:

       ```
       oci database-migration migration list -c `compartmentOCID` --
       display-name `migrationDisplayName`
       ```

2. Get the Work Request OCID associated with the migration creation request.
   - Using Oracle Database Migration Service (database-migration) OCI Command line interface:

The compartment OCID where the failed migration was created is required.

- Use the migration OCID to list the work requests:

  ```
  oci database-migration work-request list --resource-id odms-migration-OCID --compartment-id
   odms-migration-compartment-OCID
  ```

  See database-migration >> work-request >> list

- Use the compartment OCID to list the work requests of the compartment:

  ```
  oci database-migration work-request list --compartment-id odms-migration-compartment-OCID
  ```

- Use the `sort-by` option to sort the results by `displayName` or `timeCreated`:

  ```
  oci database-migration work-request list --compartment-id odms-migration-compartment-OCID --sort-by displayName
  ```

- Use the `sort-order` option with `asc` or `desc`:

  ```
  oci database-migration work-request list --compartment-id odms-migration-compartment-OCID --sort-oder ASC
  ```

  Only one sort order can be specified. Default order for `--sort-by timeCreated` is descending.

- Using the Database Migration REST API:
  - Use the migration OCID to list the work requests:

    ```
    GET /20210929/workRequests?resourceId=odms-migration-OCID&compartmentId=odms-migration-compartment-OCID
    ```

    See ListWorkRequests

  - Use the compartment OCID to list the work requests of the compartment:

    ```
    GET /20210929/workRequests?compartmentId=compartment-OCID-of-resource
    ```

  - Use the `sort-by` option to sort the results by `displayName` or `timeCreated`:

    ```
    GET /20210929/workRequests?compartmentId=compartment-OCID-of-resource&sortBy=displayName
    ```

  - Use the `sort-order`option with `asc` or `desc`:

    ```
    GET /20210929/workRequests?compartmentId=compartment-OCID-of-resource&sortOrder=ASC
    ```

    Only one sort order can be specified. Default order for `--sort-by timeCreated` is descending.

3. Use the work request OCID to get details, logs, and errors related to the failure:

- Using Oracle Database Migration Service (database-migration) OCI Command line interface:
  - Use the work request identifier to get details:

    ```
    oci database-migration work-request get --work-request-id
    workRequestId
    ```

    See database-migration >> work-request >> get
  - Use the work request identifier to list the errors:

    ```
    oci database-migration work-request-error list --work-request-id
    workRequestId
    ```

    See database-migration >> work-request-error >> list
  - Use the work request identifier to list the logs:

    ```
    oci database-migration work-request-logs list --work-request-id
    workRequestId
    ```

    See database-migration >> work-request-logs >> list
- Using the Database Migration REST API:
  - Use the work request identifier to get the details:

    ```
    GET /20210929/workRequests/{workRequestId}
    ```

    See GetWorkRequest
  - Use the work request identifier to get the errors:

    ```
    GET /20210929/workRequests/{workRequestId}/errors
    ```

    See ListWorkRequestErrors
  - Use the work request identifier to get the logs:

    ```
    GET /20210929/workRequests/{workRequestId}/logs
    ```

    See ListWorkRequestLogs

4. Inspect the logs and errors in the work request and resolve the issues reported.

For more information about using the API and signing requests, see REST APIs and Security CredentialsSecurity Credentials. For information about SDKs, see Software Development Kits and Command Line Interface.

11

Reference

This section contains reference materials.

- Using Database Migration APIs
  Refer to the OCI Database Migration service APIs for more information.
- Metering and billing
  Metering and billing for Oracle Cloud Infrastructure Database Migration Service is based on the number of Migration Hours elapsed.
- Metrics
  Monitor the health, capacity, and overall performance of your Oracle Cloud Infrastructure Database Migration database registrations, migrations, agents, and jobs using metrics, alarms, and notifications.
- Creating notifications
  On the Migration details page, you can create rules to get notified about changes to your migrations based on events or metrics by navigating to the **Monitoring** tab. You can create notifications by following either of the two ways:
- Policies
  To control access to Oracle Cloud Infrastructure Database Migration and the type of access each user group has, you must create policies.
- Set up Oracle GoldenGate for online migrations
  To use your own GoldenGate instance, Oracle Cloud Infrastructure Database Migration service has a few additional prerequisite tasks, create GoldenGate users on the source database and unlock the GoldenGate user on the target database (optional).
- Database Migration Data Pump defaults
  Oracle Cloud Infrastructure Database Migration automatically sets optimal defaults for Data Pump parameters to achieve better performance and ensure security of data. There are also Data Pump errors that are ignored by default.
- Events
  Oracle Cloud Infrastructure Database Migration service emits events in Oracle Cloud Infrastructure (OCI), which are structured messages that indicate state changes in resources.
- Port requirements
  The ports required for communication when using Oracle Cloud Infrastructure Database Migration are described in the following table.
- Migrating between Oracle Autonomous Database instances with OCI Database Migration service
  Following are the prerequisites and procedures for migrating between two Oracle Autonomous Database instances using OCI Database Migration Service. The following data transfer methods can be used: Network File System (NFS) and Database Link (DBLINK).
- Known issues
  Refer to the Known Issues for OCI Database Migration.

Using Database Migration APIs

Refer to the OCI Database Migration service APIs for more information.

See REST APIs and Software Development Kits and Command Line Interface for more information about using REST APIs and the OCI Software Development Kits and Command Line Interface.

Related Topics

- Database Migration API

Metering and billing

Metering and billing for Oracle Cloud Infrastructure Database Migration Service is based on the number of Migration Hours elapsed.

A **Migration Hour** is defined as the amount of time that a Migration Job is running, where **running** is defined as a Migration Job being in an `IN_PROGRESS` or `WAITING` state. Partial Migration Hours consumed are billed as partial hours with a one-minute minimum.

Migration Jobs are only metered if either of the following is true:

- The Migration Job is running more than 183 days (6 months) after creation
- The Migration Job is running for more than 60 days idle (no data transferred)

Migration Hours are billed down to the Second level. Note that the minimum amount billed will be 1 minute. That is, if a resource is spun up for less than 60 seconds, the customer will still be charged for 1 minute. For any usage over 1 minute, all usage will be tracked at the Second level.

You can monitor the Migration Hours of a Migration Job in the Console under Governance & Administration, in Cost and Usage Reports. The migration billing meter should be included in the report as service name DATABASEMIGRATION.

Metrics

Monitor the health, capacity, and overall performance of your Oracle Cloud Infrastructure Database Migration database registrations, migrations, agents, and jobs using metrics, alarms, and notifications.

Resources: `odms-agent`, `odms-connection`, `odms-job`, `odms-migration`

- Overview
- Prerequisites
- Available metrics
- Using the console

Overview

Oracle Cloud Infrastructure Database Migration metrics help you measure the amount of data replicated between source and target databases.

> **Note**
>
> Capacity of target databases can be monitored through the target database service and used to trigger alerts. For more information, see Use Autonomous Database Metrics to Monitor Databases and Use Enterprise Manager to Manage and Monitor Databases.

Terminology

The following terms are helpful for understanding metrics:

- **Namespace**: A container for Database Migration metrics. The namespace for Database Migration is `oci_database_migration_service`.
- **Metrics**: The fundamental concept in telemetry and monitoring. Metrics define a time-series set of datapoints. Each metric is uniquely defined by namespace, metric name, compartment identifier, a set of one or more dimensions, and a unit of measure. Each datapoint has a timestamp, a value, and a count associated with it.
- **Dimensions**: A key-value pair that defines the characteristics associated with the metric. For example, `resourceId`, which is the Database Migration deployment OCID.
- **Statistics**: Metric data aggregations over specified periods of time. Aggregations are done using the namespace, metric name, dimensions, and the datapoint unit of measure within the time period specified.
- **Alarms**: Used to automate operations monitoring and performance. An alarm keeps track of changes that occur over a specific period of time. It also performs one or more defined actions, based on the rules defined for the metric.

Prerequisites

- IAM policies: To monitor resources, you must be given the required type of access in a policy written by an administrator, whether you're using the Console or the REST API with an SDK, CLI, or other tool. The policy must give you access to the monitoring services as well as the resources being monitored. If you try to perform an action and get a message that you don’t have permission or are unauthorized, confirm with your administrator the type of access you've been granted and which compartment you should work in. For more information on user authorizations for monitoring, see Monitoring or Notifications.
- The metrics listed on this page are automatically available for any Database Migration resource you create. You do not need to enable monitoring on the resource to get these metrics.

Available metrics

Oracle Cloud Infrastructure Database Migration metrics may include the following dimensions:

- `resourceId`: For all metrics, the `resourceId` is the Migration, Agent, or Job resource OCID.
- `resourceName`: Name of the Migration, Agent, or Job resource.

Table 11-1 OCI Database Migration Metrics

Metric	Metric Display Name	Unit	Description	Dimensions
`AgentHealth`	Agent Health	percent	Measures health of a registered agent. 0% - unhealthy 100% - healthy	`resourceId = agentOcid` `resourceName = agentDisplayName`
`MigrationHealth`	Migration Health	percent	Measures health of a migration. 0% - failed migration 100% - successful migration	`resourceId = migrationOcid` `resourceName = migrationProjectName`
`DbImportProgress`	Database Import Progress	percent	Percentage of data imported into OCI database.	resourceId = jobOcid resourceName = jobName
`DbExportProgress`	Database Export Progress	percent	Percentage of data exported from source database.	resourceId = jobOcid resourceName = jobName
`ReplicationLatency`	Replication Latency	seconds	End-to-end replication latency (using heartbeat table)	resourceId = jobOcid resourceName = jobName
`ReplicationThroughput`	Replication Throughput	GB per hour	Replication throughput in GB per hour	resourceId = jobOcid resourceName = jobName

Using the console

To view Oracle Cloud Infrastructure Database Migration metrics:

1. In the Console navigation menu, under **Solutions and Platform**, go to **Monitoring** and then select **Service Metrics**.
2. For **Compartment**, select the compartment that contains the Database Migration resources you're interested in.
3. For **Metric Namespace**, select **oci_database_migration_service**.

Refresh your browser to view the latest metrics emitted by the service.

Creating notifications

On the Migration details page, you can create rules to get notified about changes to your migrations based on events or metrics by navigating to the **Monitoring** tab. You can create notifications by following either of the two ways:

You can create notifications by following either of the two ways:

- Create notifications using the provided templates:
  - Select the hyperlink to open the template.
  - Alternatively, select from the quickstart templates to create notifications.
- If you click the template selection Menu, you can select the hyperlink, or in the **Create notifications** template selection option, select a template. The following options are available:
  - Evaluation or Migration job status has changed
  - Evaluation or Migration job completed successfully
  - Evaluation or Migration job failed to complete
  - Migration job went into a waiting state
  - A phase completed for an Evaluation or Migration job
  - Replication latency exceeds 5 seconds

  The rules that trigger notifications can be created using service events or metrics.
- If you want to create your own templates, you can select **Advanced event notifications** or **Advanced alarm notifications** based on the metrics you want to be alerted on.

You can create notifications based on the following options:

- Quickstarts notifications
- Advanced event notification
- Advanced alarm notification

To configure Quickstarts notifications, follows the steps:

1. Select a template from **Template selection**.
2. In Event rule, enter the **Rule name**.
3. In Topic and subscriptions,
   a. In the **Create new topic** , select the compartment that you want to create the topic in and enter a user-friendly name for the topic. The topic you created appears in the list of the **Topics**. Alternatively, you can **Select existing topic**.
   b. In the **Subscription** panel:
      i. Set **Subscription protocol** for email, Slack, or SMS notification:
         - **Email**:
           i. Set **Protocol** to **Email**.
           ii. Enter the email address that should receive the notifications.
         - **Slack**:
           i. Set **Protocol** to **Slack**.
           ii. For **URL**, enter the URL for your Slack Webhook.
         - **SMS**:
           i. Set the **Country**.
           ii. Set the **Phone number**.

      In **Advanced options** Specify tag information for one or more tag namespaces. Click **Another tag** to specify tag information for another tag namespace.

When you receive the subscription confirmation email, click the **Confirm Subscription** link in the body of the message.
An Oracle Cloud Infrastructure page appears in your browser, indicating that your subscription is confirmed.

4. Click **Create Notifications**.

To configure Advanced event notification, follows the steps:

1. In Event rule, enter the **Rule name**.
2. Select the **Event type** for which you want to be notified.
3. In Topic and subscriptions,
   a. In the **Create new topic** , select the compartment that you want to create the topic in and enter a user-friendly name for the topic. The topic you created appears in the list of the **Topics**. Alternatively, you can **Select existing topic**.
   b. In the **Subscription** panel:
      i. Set **Subscription protocol** for email, Slack, or SMS notification:
         - **Email**:
            i. Set **Protocol** to **Email**.
            ii. Enter the email address that should receive the notifications.
         - **Slack**:
            i. Set **Protocol** to **Slack**.
            ii. For **URL**, enter the URL for your Slack Webhook.
         - **SMS**:
            i. Set the **Country**.
            ii. Set the **Phone number**.
         - **HTTPS (custom URL):** Set the URL.
         - **PagerDuty**: Set the Integration Key.
         - **Function**:
            i. Set the Function compartment.
            ii. Set the Oracle Functions application.
            iii. Set the Function.

      In **Advanced options** specify tag information for one or more tag namespaces. Click **Another tag** to specify tag information for another tag namespace.

   When you receive the subscription confirmation email, click the **Confirm Subscription** link in the body of the message.
   An Oracle Cloud Infrastructure page appears in your browser, indicating that your subscription is confirmed.
4. Click **Create Notifications**.

You can create alarms to notify you when metrics meet alarm-specified triggers. To configure Advanced alarm notification, follows the steps:

1. In the **Alarms** pane, create a **Alarm name**. Enter a user-friendly name for the alarm. Avoid entering confidential information.

2. In or **Alarm severity**, select the perceived type of response required when the alarm is in the firing state.
3. In the **Metric description** area, enter values to specify the metric to evaluate for the alarm.
   - **Metric name**: Select the name of the metric that you want to evaluate for the alarm. You can select any OCI metric or custom metric if the data exists in the selected compartment and metric namespace.
   - **Interval**: Select the aggregation window, or the frequency at which the alarm needs to be triggered.
   - **Statistic**: Select the function to use to trigger the alarm.
     - **Mean** - The value of Sum divided by Count during the specified time period.
     - **Rate** - The per-interval average rate of change.
     - **Sum** - All values added together.
     - **Max** - The highest value observed during the specified time period.
     - **Min** - The lowest value observed during the specified time period.
     - **Count** - The number of observations received in the specified time period.
     - **P50** - The value of the 50th percentile.
     - **P90** - The value of the 90th percentile.
     - **P95** - The value of the 95th percentile.
     - **P99** - The value of the 99th percentile.
4. In the **Trigger rule** area, specify the condition that must be satisfied for the alarm to be in the firing state. The condition can specify a threshold, such as 90% for CPU utilization, or an absence.
   - **Operator**: Select the operator to use in the condition threshold.
   - **Value**: Enter the value to use for the condition threshold. For the **between** and **outside** operators, enter both values for the range.
   - **Trigger delay minutes**: Enter the number of minutes that the condition must be maintained before the alarm is in the firing state.
5. In Topic and subscriptions,
   a. In the **Create new topic** , select the compartment that you want to create the topic in and enter a user-friendly name for the topic. The topic you created appears in the list of the **Topics**. Alternatively, you can **Select existing topic**.
   b. In the **Subscription** panel:
      i. Set **Subscription protocol** for email, Slack, or SMS notification:
         - **Email**:
           i. Set **Protocol** to **Email**.
           ii. Enter the email address that should receive the notifications.
         - **Slack**:
           i. Set **Protocol** to **Slack**.
           ii. For **URL**, enter the URL for your Slack Webhook.
         - **SMS**:
           i. Set the **Country**.

ii. Set the **Phone number**.

- **HTTPS (custom URL):** Set the URL.
- **PagerDuty**: Set the Integration Key.
- **Function**:
  i. Set the Function compartment.
  ii. Set the Oracle Functions application.
  iii. Set the Function.

When you receive the subscription confirmation email, click the **Confirm Subscription** link in the body of the message.
An Oracle Cloud Infrastructure page appears in your browser, indicating that your subscription is confirmed.

6. In **Advanced options** Specify:
   - Alarm body
   - Notifications:
     - Message format: Select an option for the appearance of messages that you receive from this alarm (for **Notifications** only).
       * **Send formatted messages**: Simplified, user-friendly layout. To view supported subscription protocols and message types for formatted messages (options other than **Raw**), see Friendly formatting.
       * **Send Pretty JSON messages (raw text with line breaks)**: JSON with new lines and indents.
       * **Send raw messages**: Raw JSON blob.
     - Repeat notification: If you want to receive notifications at regular intervals when the alarm is firing. Specify the period of time to wait before resending the notification.
     - Suppress notification: To suppress evaluations and notifications for a specified length of time. Specify a start time, end time, and an optional description.
   - Tag information for one or more tag namespaces.
7. Click **Create Notifications**.

You can view the created notifications in the **Notifications** tab. The following details are available:

- **Type**
- **Name**
- **Topic**
- **Created**

You can select the Actions (three dots) menu and perform the following tasks:

- **Edit**
- **View tags**
- **Add tags**
- **Delete**

Below the migration details, you can view information about resources associated with the migration, such as Jobs (see Manage jobs), Excluded Objects, Work Requests, and Metrics.

Policies

To control access to Oracle Cloud Infrastructure Database Migration and the type of access each user group has, you must create policies.

The following topics explain how to create policies for Database Migration.

- About IAM policies
- Basic syntax for policies
- Details for verbs + resource-type combinations
  There are various Oracle Cloud Infrastructure verbs and resource-types that you can use when you create a policy. The topics in this section show the permissions and API operations covered by each verb for Database Migration.
- Creating a policy
- Required policies
- Permissions required for Database Migration API operations
- Supported variables
  When you add conditions to your policies, you can use either Oracle Cloud Infrastructure general or service specific variables.

About IAM policies

A tenancy administrator can create policies in Oracle Cloud Infrastructure Identity and Access Management (IAM) to grant permissions to groups on resources in compartments in a tenancy.

For example, you can create an Administrators group whose members can access all Database Migration resources. You can then create a separate group for everyone else who's involved with Database Migration, and create policies that restrict their access to Database Migration resources in different compartments.

For a complete list of Oracle Cloud Infrastructure policies, see policy reference.

Basic syntax for policies

A policy is a document that consists of one or more statements. A policy statement follows this basic syntax:

```
Allow group <group_name> to <verb><resource-type> in compartment
<compartment_name>
```

Policy language uses simple verbs like `inspect`, `read`, `use`, and `manage`.

- Database Migration resource-types

Database Migration resource-types

Database Migration offers individual resource-types for writing policies.

Resource-Type	Description
odms-agent	Software that allows migrations from sources databases not accessible from Oracle Cloud
odms-connection	Connection settings
odms-job	Migration job operations
odms-migration	Migration parameter settings

Details for verbs + resource-type combinations

There are various Oracle Cloud Infrastructure verbs and resource-types that you can use when you create a policy. The topics in this section show the permissions and API operations covered by each verb for Database Migration.

The level of access is cumulative as you go from inspect to read to use to manage.

- odms-connection
- odms-agent
- odms-migration
- odms-job

odms-connection

Permission	APIs Fully Covered
INSPECT	
ODMS_CONNECTION_INSPECT	ListConnection
READ	
INSPECT +	INSPECT+
ODMS_CONNECTION_READ	GetConnection
USE	
READ +	READ +
ODMS_CONNECTION_USE	N/A
MANAGE	
USE +	USE +
ODMS_CONNECTION_CREATE	CreateConnection
ODMS_CONNECTION_UPDATE	UpdateConnection
ODMS_CONNECTION_DELETE	DeleteConnection
ODMS_CONNECTION_MOVE	ChangeConnectionCompartment

odms-agent

Permission	APIs Fully Covered
INSPECT	
ODMS_AGENT_INSPECT	ListAgents
READ	
INSPECT +	INSPECT+
ODMS_AGENT_READ	GetAgent

Permission	APIs Fully Covered
USE	
READ +	READ +
N/A	N/A
MANAGE	
USE +	USE +
ODMS_AGENT_UPDATE	UpdateAgent
ODMS_AGENT_DELETE	DeleteAgent
ODMS_AGENT_MOVE	ChangeAgentCompartment

odms-migration

Permission	APIs Fully Covered
INSPECT	
ODMS_MIGRATION_INSPECT	ListAgentImages
ODMS_MIGRATION_INSPECT	ListMigrations
READ	
INSPECT +	INSPECT+
ODMS_MIGRATION_READ	GetMigration
ODMS_MIGRATION_READ	RetrieveSupportedPhases
USE	
READ +	READ +
ODMS_MIGRATION_USE	StartMigration
ODMS_MIGRATION_VALIDATE	EvaluateMigration
MANAGE	
USE +	USE +
ODMS_MIGRATION_CREATE + ODMS_CONNECTION_USE	CreateMigration
ODMS_MIGRATION_CLONE + ODMS_CONNECTION_USE	CloneMigration
ODMS_MIGRATION_UPDATE + ODMS_CONNECTION_USE	UpdateMigration
ODMS_MIGRATION_DELETE	DeleteMigration
ODMS_MIGRATION_MOVE	ChangeMigrationCompartment

odms-job

Permission	APIs Fully Covered
INSPECT	
ODMS_JOB_INSPECT	ListJobs
READ	
INSPECT +	INSPECT+
ODMS_JOB_READ	GetJob
USE	
READ +	READ +
ODMS_JOB_USE	ListJobOutputs

Permission	APIs Fully Covered
ODMS_JOB_USE	GetJobOutputContent
ODMS_JOB_ABORT	AbortJob
ODMS_JOB_RESUME	ResumeJob
MANAGE	
USE +	USE +
ODMS_JOB_UPDATE	UpdateJob
ODMS_JOB_DELETE	DeleteJob

Creating a policy

To create a policy:

1. In the Console navigation menu, under **Identity & Security**, go to **Identity**, and then click **Policies**.
2. Click **Create Policy**.
3. Enter a name and description for the policy.
4. Select the compartment. If you want to attach the policy to a compartment other than the one you're viewing, select it from list.
5. Select `Database Migration Service` in the **Policy use cases** option in the policy builder.
6. Select the template that best matches your requirements from the **Common policy templates** list. The policy builder displays the description of the chosen policy and lists the policy statements that it includes. The following policies are available for the `Database Migration Service`:
   - Let users manage required database migration resources along with all other required resources/networking- This is recommended for Database Migration Service and other required OCI resources.
   - Let users manage database migration resources with networking
   - Let users manage database migration resources without networking
   - Let users use database migration service resources
   - Let users manage database migration resources with networking if the manage virtual-network-family policy is restricted.
7. Select the group and the location.
8. Select **Show manual editor** if you already know how to write the statements you need and you want to enter them in a text box.
9. To add tags to this policy, click **Show advanced options**.
10. If you want to create another policy, select **Create another policy**.
11. Click **Create**.

For more information about policies, see Creating a Policy, how policies work, policy syntax, and policy reference.

Required policies

Following are the required set of policies for the Database Migration service. The tenancy administrator can easily provision these policies by using the templates available in the Policy Builder.

> **Note**
>
> This policy lets you perform all the actions for the Database Migration service such as creating connections, manage connections, and so on, depending on your usage.

> **Note**
>
> Oracle recommends using the first group of policies **To let users manage connections, migrations, jobs, and networks** as it has the required policies to go end to end.

To let users manage connections, migrations, jobs, and networks:

```
Allow group {group name} to manage odms-connection in  {location}
Allow group {group name} to manage odms-migration in {location}
Allow group {group name} to manage odms-job in {location}
Allow group {group name} to manage goldengate-connections in {location}
Allow group {group name} to manage virtual-network-family in {location}
Allow group  {group name} to manage tag-namespaces in {location}
Allow group {group name}  to manage vaults in {location}
Allow group {group name}  to manage keys in {location}
Allow group {group name} to manage secret-family in {location}
Allow group {group name} to manage object-family in {location}
```

To complete an end-to-end migration, the tenancy administrator needs to provision certain policies, where you need to create, update, and use the resources (change the level or permission depending on your use case):

```
Allow group {group name} to manage virtual-network-family in compartment
{compartment name}
Allow group {group name} to manage vaults in compartment {compartment name}
Allow group {group name} to manage keys in compartment {compartment name}
Allow group {group name} to manage secret-family in compartment {compartment
name}
Allow group {group name} to manage object-family in compartment {compartment
name}
Allow group {group name} to manage odms-connection in compartment
{compartment name}
Allow group {group name} to manage odms-migration in compartment {compartment
name} Allow group {group name} to manage odms-job in compartment {compartment
name}
```

For scenarios where Manage virtual-network-family cannot be assigned, it can be substituted by:

```
Allow group {group name}  to inspect vcns in compartment {compartment name}
Allow group {group name}  to use subnets in compartment {compartment name}
Allow group {group name}  to manage vnic in compartment {compartment name}
```

Depending on whether or not you intend to use the following services, you will need to add policies to enable access to these services as well:

- Oracle Autonomous Databases for your target databases:

  ```
  Allow group {group name} to manage autonomous-database-family in
  compartment {compartment name} →Aggregate resource type
  ```

- Base database for your source or target:

  ```
  Allow group {group name} to manage database-family in compartment
  {compartment name} → Aggregate resource type
  ```

- If you need to access the connections created by GoldenGate integrated service:

  ```
  Allow group {group name} to manage GoldenGate-connections in compartment
  {compartment name}
  ```

- If you need to deploy your own GoldenGate Marketplace instance and use as an advanced replication option:

  ```
  Allow group {group name} to manage instance-family in compartment
  {compartment name}
  Allow group {group name} to manage volume-family in compartment
  {compartment name}
  Allow group {group name} to manage public-ips in compartment {compartment
  name}
  Allow group {group name} to use tag-namespaces in tenancy
  Allow group {group name} to inspect compartments in tenancy
  Allow group {group name} to manage orm-family in compartment {compartment
  name}
  Allow group {group name} to manage app-catalog-listing in compartment
  {compartment name}→ Required to launch the GG marketplace stack
  ```

- To access the HeatWave cluster data when creating a HeatWave connection using the Database Migration Service:

  ```
  Allow group {group name} to manage mysql-heatwave in compartment
  {compartment name} → Aggregate resource type
  ```

Policies define what actions members of a group can perform, and in which compartments. You create policies using the Oracle Cloud Console. In the Oracle Cloud Console navigation menu, under **Identity & Security** and then under **Identity**, click **Policies**. Policies are written in the following syntax:

```
Allow group <group-name> to <verb> <resource-type> in <location> where
<condition>
```

- <group-name>: The name of the user group you're giving permissions to
- <verb>: Gives the group a certain level of access to a resource-type. As the verbs go from inspect to read to use to manage, the level of access increases and the permissions granted are cumulative.
- <resource-type>: The type of resource you're giving a group permission to work with, such as odms-agent, odms-connection, odms-job, and odms-migration.
  For more information, see resource-types.
- <location>: Attaches the policy to a compartment or tenancy. You can specify a single compartment or compartment path by name or OCID, or specify tenancy to cover the entire tenancy.
- <condition>: Optional. One or more conditions for which this policy will apply.

Creating a Network Resource Policy

Database Migration requires you to provide VCN and subnet information when creating migrations and database registrations. In order to provide this information, you need to have the ability to view cloud network information. The following statement gives the group permission to inspect network resources in the compartment and select them when creating Database Migration resources:

```
allow group <group-name> to inspect virtual-network-family in compartment
<compartment-name>
```

Creating a Tagging Policy

The following statement gives a group permission to manage tag-namespaces and tags for workspaces:

```
allow group <group-name> to manage tag-namespaces in compartment <compartment-
name>
```

To add a defined tag, you must have permission to use the tag namespace.

Related Topics

Learn more about:

- Policies
- policy syntax
- Resource Tags
- Permissions
- Policy Details for Base Database Service
- Policy Details for Autonomous Database on Serverless

Permissions required for Database Migration API operations

Here's a list of the API operations for Oracle Cloud Infrastructure Database Migration in logical order, grouped by resource-type.

The resource-types are odms-agent, odms-connection, odms-job and odms-migration.

API Operation	Permission
GetAgent	ODMS_AGENT_READ
ListAgents	ODMS_AGENT_INSPECT
DeleteAgent	ODMS_AGENT_DELETE
UpdateAgent	ODMS_AGENT_UPDATE
ChangeAgentCompartment	ODMS_AGENT_MOVE
ValidateAgent	ODMS_AGENT_REGISTER
RegisterHeartbeat	ODMS_AGENT_REGISTER
GetActionGenerateToken	ODMS_AGENT_REGISTER
CreateConnection	ODMS_CONNECTION_CREATE
UpdateConnection	ODMS_CONNECTION_UPDATE
GetConnection	ODMS_CONNECTION_READ
ListConnections	ODMS_CONNECTION_INSPECT
DeleteConnection	ODMS_CONNECTION_DELETE
ChangeConnectionCompartment	ODMS_CONNECTION_MOVE
ListAgentImages	ODMS_MIGRATION_INSPECT
CreateMigration	ODMS_CONNECTION_USE and ODMS_MIGRATION_CREATE
CloneMigration	ODMS_CONNECTION_USE and ODMS_MIGRATION_CLONE
UpdateMigration	ODMS_CONNECTION_USE and ODMS_MIGRATION_UPDATE
GetMigration	ODMS_MIGRATION_READ
RetrieveSupportedPhases	ODMS_MIGRATION_READ
ListMigrations	ODMS_MIGRATION_INSPECT
DeleteMigration	ODMS_MIGRATION_DELETE
EvaluateMigration	ODMS_MIGRATION_VALIDATE
StartMigration	ODMS_MIGRATION_USE
ChangeMigrationCompartment	ODMS_MIGRATION_MOVE
AbortJob	ODMS_JOB_ABORT
ResumeJob	ODMS_JOB_RESUME
DeleteJob	ODMS_JOB_DELETE
GetJob	ODMS_JOB_READ
ListJobs	ODMS_JOB_INSPECT
UpdateJob	ODMS_JOB_UPDATE
ListJobOutputs	ODMS_JOB_USE
GetJobOutputContent	ODMS_JOB_USE

Supported variables

When you add conditions to your policies, you can use either Oracle Cloud Infrastructure general or service specific variables.

Database Migration supports all general variables. For more information, see general variables for all requests.

Set up Oracle GoldenGate for online migrations

To use your own GoldenGate instance, Oracle Cloud Infrastructure Database Migration service has a few additional prerequisite tasks, create GoldenGate users on the source database and unlock the GoldenGate user on the target database (optional).

> **Note**
>
> You can have a single user for database connection, if you have the required privileges. For the source database, the user for CDB and PDBs has all the privileges for GoldenGate and Data Pump.

Create GoldenGate Users on the Source Database

On the source database, you must create a GoldenGate administration user, for example `ggadmin`.

If the source database is multitenant, create the user in the PDB, and also create a different user in the CDB root, for example `c##ggadmin`.

To create `ggadmin`, connect to the PDB and run the following commands:

```
CREATE TABLESPACE gg_admin DATAFILE '+DATA/ggadmin_data.dbf' SIZE 100m
AUTOEXTEND ON NEXT 100m;
CREATE USER ggadmin IDENTIFIED BY ggadmin_pwd DEFAULT TABLESPACE gg_admin
TEMPORARY TABLESPACE TEMP QUOTA UNLIMITED ON gg_admin;
GRANT CONNECT TO ggadmin;
GRANT RESOURCE TO ggadmin;
GRANT CREATE TO ggadmin;
GRANT SELECT_CATALOG_ROLE TO ggadmin;
GRANT DV_GOLDENGATE_ADMIN TO ggadmin;
GRANT DV_GOLDENGATE_REDO_ACCESS TO ggadmin;
GRANT ALTER SYSTEM TO ggadmin;
GRANT ALTER USER TO ggadmin;
GRANT DATAPUMP_EXP_FULL_DATABASE TO ggadmin;
GRANT DATAPUMP_IMP_FULL_DATABASE TO ggadmin;
GRANT SELECT ANY DICTIONARY TO ggadmin;
GRANT SELECT ANY TRANSACTION TO ggadmin;
GRANT INSERT ANY TABLE TO ggadmin;
GRANT UPDATE ANY TABLE TO ggadmin;
GRANT DELETE ANY TABLE TO ggadmin;
GRANT LOCK ANY TABLE TO ggadmin;
GRANT CREATE ANY TABLE TO ggadmin;
GRANT CREATE ANY INDEX TO ggadmin;
GRANT CREATE ANY CLUSTER TO ggadmin;
GRANT CREATE ANY INDEXTYPE TO ggadmin;
GRANT CREATE ANY OPERATOR TO ggadmin;
GRANT CREATE ANY PROCEDURE TO ggadmin;
GRANT CREATE ANY SEQUENCE TO ggadmin;
GRANT CREATE ANY TRIGGER TO ggadmin;
GRANT CREATE ANY TYPE TO ggadmin;
GRANT CREATE ANY SEQUENCE TO ggadmin;
GRANT CREATE ANY VIEW TO ggadmin;
```

```
GRANT ALTER ANY TABLE TO ggadmin;
GRANT ALTER ANY INDEX TO ggadmin;
GRANT ALTER ANY CLUSTER TO ggadmin;
GRANT ALTER ANY INDEXTYPE TO ggadmin;
GRANT ALTER ANY OPERATOR TO ggadmin;
GRANT ALTER ANY PROCEDURE TO ggadmin;
GRANT ALTER ANY SEQUENCE TO ggadmin;
GRANT ALTER ANY TRIGGER TO ggadmin;
GRANT ALTER ANY TYPE TO ggadmin;
GRANT ALTER ANY SEQUENCE TO ggadmin;
GRANT CREATE DATABASE LINK TO ggadmin;
GRANT EXECUTE ON dbms_lock TO ggadmin;
EXEC DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE('ggadmin');
```

To create `c##ggadmin`, connect to the CDB and run the following commands:

```
CREATE USER c##ggadmin IDENTIFIED BY cggadmin_pwd CONTAINER=ALL DEFAULT
TABLESPACE USERS TEMPORARY TABLESPACE TEMP QUOTA UNLIMITED ON USERS;
GRANT CONNECT TO c##ggadmin CONTAINER=ALL;
GRANT RESOURCE TO c##ggadmin CONTAINER=ALL;
GRANT CREATE TABLE TO c##ggadmin CONTAINER=ALL;
GRANT CREATE VIEW TO c##ggadmin CONTAINER=ALL;
GRANT CREATE SESSION TO c##ggadmin CONTAINER=ALL;
GRANT SELECT_CATALOG_ROLE TO c##ggadmin CONTAINER=ALL;
GRANT DV_GOLDENGATE_ADMIN TO c##ggadmin CONTAINER=ALL;
GRANT DV_GOLDENGATE_REDO_ACCESS TO c##ggadmin CONTAINER=ALL;
GRANT ALTER SYSTEM TO c##ggadmin CONTAINER=ALL;
GRANT ALTER USER TO c##ggadmin CONTAINER=ALL;
GRANT SELECT ANY DICTIONARY TO c##ggadmin CONTAINER=ALL;
GRANT SELECT ANY TRANSACTION TO c##ggadmin CONTAINER=ALL;
GRANT EXECUTE ON dbms_lock TO c##ggadmin CONTAINER=ALL;
EXEC
DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE('c##ggadmin',CONTAINER=>'ALL');
```

Create or Unlock the GoldenGate User on the Target Database

On co-managed targets:

If the target is not Oracle Autonomous Database, create a `ggadmin` user in the target PDB. This user is similar to the `ggadmin` user you created on the source database, but will require more privileges. See Establishing Oracle GoldenGate Credentials for information about privileges required for a "Replicat all modes" user.

On Autonomous targets:

Autonomous Database has a pre-created `ggadmin` user that you must unlock. These commands need to be run on the GoldenGate Target instance.

1. Connect to the target database as `admin`.

   ```
   export TNS_ADMIN=/u02/deployments/Target/etc
   export ORACLE_HOME=/u01/app/client/oracle19
   $ $ORACLE_HOME/bin/sqlplus admin/ADW_password@ADW_name
   ```

   An example of the *ADW_name* would be `targetatp_high`.

2. Unlock `ggadmin`.

```
SQL> ALTER USER ggadmin IDENTIFIED BY ggadmin_password ACCOUNT UNLOCK;
```

3. Verify that `ggadmin` is unlocked.

```
export TNS_ADMIN=/u02/deployments/Target/etc
export ORACLE_HOME=/u01/app/client/oracle19
$ORACLE_HOME/bin/sqlplus ggadmin/ADW_password@ADW_name
```

> **Note**
>
> Oracle Cloud Infrastructure Database Migration Service supports only those scenarios where the Target database and Oracle GoldenGate, both run on private IP addresses.

Database Migration Data Pump defaults

Oracle Cloud Infrastructure Database Migration automatically sets optimal defaults for Data Pump parameters to achieve better performance and ensure security of data. There are also Data Pump errors that are ignored by default.

The following table lists the Data Pump parameters set by Database Migration, and the values they are set to. If there is a Database Migration migration setting available to override the default, it is listed in the Optional DMS Setting to Override column.

Table 11-2 Database Migration Data Pump Parameter Defaults

Data Pump Parameter	Default Value	Optional DMS Setting to Override
EXCLUDE	index (ADW-S) cluster (ADB-D, ADB-S) indextype (ADW-S) materialized_view (ADW-S) materialized_view_log (ADW-S) materialized_zonemap (ADW-S) db_link (ADB) statistics (User managed Target and ADB)	Excluded Objects: Object Owner and Object Name See Selecting objects for Oracle migration
PARALLEL	Database Migration sets PARALLEL parameter by default as follows For user-managed database (Sum of (2 x (no. of physical CPU) per node)) with Max 32 cap. For Autonomous Database, number of OCPUs	Import Parallelism Degree Export Parallelism Degree See Configuring initial load advanced options for Oracle migrations
CLUSTER	Database Migration always sets the Cluster mode as default	Cluster See Configuring initial load advanced options for Oracle migrations

Table 11-2 (Cont.) Database Migration Data Pump Parameter Defaults

Data Pump Parameter	Default Value	Optional DMS Setting to Override
COMPRESSION	COMPRESSION_ALGORITHM is set to BASIC(for Oracle Database 11.2) and MEDIUM (for Oracle Database 12.1 and later) COMPRESSION is set to ALL	N/A
ENCRYPTION	ENCRYPTION is set to ALL ENCRYPTION_ALGORITHM is set to AES128 ENCRYPTION_MODE is set to PASSWORD	N/A
FILESIZE	FILESIZE is set to 5G	N/A
FLASHBACK_SCN	For offline migrations, sets FLASHBACK_TIME System time now. For online migrations, uses neither FLASHBACK_SCN nor FLASHBACK_TIME	N/A
REUSE_DUMPFILES	Always set to YES	N/A
TRANSFORM	Always sets OMIT_ENCRYPTION_CLAUSE:Y for Oracle Database 19c and later target releases Always sets LOB_STORAGE:SECUREFILE For an Autonomous Database target, the following transform is set by default SEGMENT_ATTRIBUTES:N DWCS_CVT_IOTS:Y CONSTRAINT_USE_DEFAULT_INDEX:Y	Allows additional TRANSFORM to be specified
METRICS	Always set to Yes	N/A
LOGTIME	Always set to ALL	N/A
TRACE	Always set to 1FF0b00	N/A
LOGFILE	Always set to Data Pump job name and created under the specified export or import directory object. For example, if a Data Pump job is ZDM_2_DP_EXPORT_8417 and directory object used is DATA_PUMP_DIR, then the operation log is created by name ZDM_2_DP_EXPORT_8417.log under DATA_PUMP_DIR.	N/A

Delay Ref Constraints parameter

1. Oracle Cloud Infrastructure Database Migration service has enabled the advance parameter **Delay Ref Constraints** to mitigate foreign key issues.
2. Export is performed from the source using: **Delay Ref Constraints=TRUE**.
3. This ensures foreign key constraints are not migrated during the initial load, preventing dependency violations.
4. GoldenGate (GGS) replicates the changes during the migration window, keeping the target in sync.
5. During the downtime window, Oracle Cloud Infrastructure Database Migration service reload applies the constraints using an additional export/import.

Oracle Data Pump errors that are ignored by default are as follows:

- ORA-31684: XXXX already exists
- ORA-39111: Dependent object type XXXX skipped, base object type
- ORA-39082: Object type ALTER_PROCEDURE: XXXX created with compilation warnings

This is not configurable.

Events

Oracle Cloud Infrastructure Database Migration service emits events in Oracle Cloud Infrastructure (OCI), which are structured messages that indicate state changes in resources.

You can define rules in the OCI Event Service to get notified of events happening in an OCI native service and use the Notification Service (ONS) to send notifications or trigger events.

Table 11-3 Database Migration Service Event Types

Resource	Attributes (Common for Object)	Event name	Notes
Migration	• compartmentId • compartmentName • resourceId • resourceName • availabilityDomain • freeformTags • definedTags • Oracle-Tags • CreatedBy • CreatedOn • eventID	Migration Clone Begin	When a migration clone request is received.
		Migration Clone End	When a migration clone request is completed.
		Migration Create Begin	When a migration create request is received.
		Migration Create End	When a migration create request is completed.
		Migration Delete Begin	When a migration delete request is received.
		Migration Delete End	When a migration delete request is completed.
		Migration Evaluate Begin	When a migration evaluate request is received.
		Migration Evaluate End	When a migration evaluate request is completed.
		Migration Start Begin	When a migration start request is received.
		Migration Start End	When a migration start request is completed.
		Migration State Change	When a migration state change request is received.

Table 11-3 (Cont.) Database Migration Service Event Types

Resource	Attributes (Common for Object)	Event name	Notes
		Migration Update Begin	When a migration update request is received.
		Migration Update End	When a migration update request is completed.
Job	compartmentId compartmentName resourceName resourceIdavailabilityDomain eventID	Job Abort Begin	When a job abort request is received.
		Job Abort End	When a job is aborted.
		Job Delete Begin	When a job delete request is received.
		Job Delete End	When a job is deleted.
		Job Resume Begin	When a job resume request is received.
		Job Resume End	When a job resume is completed.
		Job State Changed	When a job state is changed.
		Job State Failed	When a job fails.
		Job State Successful	When a job state is successful.
		Job State Waiting	When a job update request is in waiting state.
		Job Update Begin	Called when the state of job is changed.
		Job Update End	When a job update request is completed.
Connection	• compartmentId • compartmentName • resourceId • resourceName • availabilityDomain • freeformTags • definedTags • Oracle-Tags • CreatedBy • CreatedOn • eventID	Connection Create Begin	When a connection create request is sent.
		Connection Create End	When a connection create request is completed.
		Connection Delete Begin	When a connection delete request is sent.
		Connection Delete End	When a connection delete request is complete.
		Connection Update Begin	When a connection update request is sent.
		Connection Update End	When a connection update request is completed.
Phase	compartmentName compartmentId migrationName migrationId jobName jobId phaseName phaseStatus	Phase Begin	Emitted when the phase starts.
		Phase End	Emitted when the phase ends.

Table 11-4 Example Use Cases

Use case	Event	Attribute filters	Action
To start a process when Migration XYZ is started	Migration - Start	migrationName=XYZ	Function or Streaming
To start a function for a given job after the Data Pump file is uploaded and before import starts (You must also configure a pause after phase `ZDM_UPLOAD_DATAPUMP_DUMP_FILES`, and the function must call the API to resume the job)	Job - State Change	jobId=*job_OCID* jobLifecycleState=PAUSED phaseName=ZDM_UPLOAD_DATAPUMP_DUMP_FILES	Function
To send a notification whenever a Migration in my compartment is failing	Job - State Change	compartmentId=*myCompartmentId*migrationLifecycleState=FAILED	Notification
To send a notification when Migration XYZ starts waiting on replication	Job - State Change	migrationName=XYZ jobLifecycleState=PAUSED phaseName=ZDM_MONITOR_GG_LAG	Notification
To send a notification when Migration XYZ starts Data Pump export	Phase - Begin	migrationName=XYZ phaseName=ZDM_DATAPUMP_EXPORT	Notification

For information about the migration job phases, see Oracle migration phases.

Port requirements

The ports required for communication when using Oracle Cloud Infrastructure Database Migration are described in the following table.

Table 11-5 Database Migration Communication Ports

Initiator	Target	Protocol	Port	Purpose
Source database servers	Oracle Cloud Object Store Service	SSL	443	This port allows Data Pump dumps to be uploaded to Oracle Cloud Storage
Database Migration	Oracle Autonomous Database Serverless On premises and co-managed Oracle Database servers	TCP	1521, 1522	Allow Oracle client connections to the database over Oracle's SQL*Net protocol

Table 11-5 (Cont.) Database Migration Communication Ports

Initiator	Target	Protocol	Port	Purpose
Database Migration	Oracle Autonomous Database on Dedicated Exadata Infrastructure On premises and co-managed Oracle Database servers	TCP	1521, 2484	Allow Oracle client connections to the database over Oracle's SQL*Net protocol

Table 11-5 (Cont.) Database Migration Communication Ports

Initiator	Target	Protocol	Port	Purpose
Database Migration agent service host	Source and target database servers	TCP	22	SSH Authentication-based operations to run Database Migration operational phases Source and target database servers should accept incoming connections from the Database Migration agent service host Not applicable to Autonomous Database targets

Note

Required only for SSH connection.

> **Note**
>
> If you are using a non-default port number (that is, something other than port 1521) for the local listener address, then the non-default port should allow connections.

Configuring Network Security Rules

If you have Oracle Database or Oracle GoldenGate compute instances in private subnets, ensure their subnet security rules or network security groups allow traffic required for Database Migration jobs.

Database Migration allows you to specify a subnet to create a Private Endpoint for Database Migration Connections (Connections). Refer to steps 9 and 10 in Manage connections. For Autonomous Database Connections, the Console pre-populates the subnet field using the Autonomous Database (ADB) subnet; however, you can use the dropdown list to select a different subnet. The corresponding Database Migration API is CreateConnection.

1. The following **EGRESS** security rules must be configured for **your subnet specified for privateEndpointDetails** when creating Database Migration connections:

Rule Type	Stateful	Direction	Source Port Range	Protocol	Destination	Destination Port Range
SecurityList	No	Egress	All	TCP	CIDR (Classless Inter-Domain Routing) of subnet hosting Co-managed database or Oracle Autonomous Database Serverless	1521-1523
SecurityList	No	Egress	All	TCP	CIDR of subnet hosting Oracle Autonomous Database on Dedicated Exadata Infrastructure	2484
SecurityList	No	Egress	All	TCP	CIDR of subnet hosting Co-managed database **Note** Required only for SSH connection.	22

Rule Type	Stateful	Direction	Source Port Range	Protocol	Destination	Destination Port Range
SecurityList	No	Egress	All	TCP	CIDR of subnet hosting Oracle GoldenGate compute instance	443

2. The following **INGRESS** security rules must be configured for the **subnets hosting your databases or Oracle GoldenGate compute instances**:

Subnet Hosting Co-managed System

Rule Type	Stateful	Direction	Source Port Range	Protocol	Destination	Destination Port Range
SecurityList	No	Ingress	CIDR of subnet specified for PrivateEndpoint for Database Migration Connection (Connection)	All	TCP	1521-1523
SecurityList	No	Ingress	CIDR of subnet specified for PrivateEndpoint for Database Migration Connection (Connection)	All	TCP	22

Note

Required only for SSH connection.

Subnet Hosting ADB-S

Rule Type	Stateful	Direction	Source Port Range	Protocol	Destination	Destination Port Range
SecurityList	No	Ingress	CIDR of subnet specified for PrivateEndpoint for Database Migration Connection (Connection)	All	TCP	1521-1523

Subnet Hosting ADB-D

Rule Type	Stateful	Direction	Source Port Range	Protocol	Destination	Destination Port Range
SecurityList	No	Ingress	CIDR of subnet specified for PrivateEndpoint for Database Migration Connection (Connection)	All	TCP	2484

Subnet Hosting Oracle GoldenGate Compute Instance

Rule Type	Stateful	Direction	Source Port Range	Protocol	Destination	Destination Port Range
SecurityList	No	Ingress	CIDR of subnet specified for PrivateEndpoint for Database Migration Connection (Connection) for target database	All	TCP	443

3. Additionally, if you have configured **Network Security Groups (NSGs) for ADB-S or Oracle GoldenGate compute instances**, then the following **INGRESS** rules must be set for the Network Security Groups:

   NSG Associated With ADB-S

Rule Type	Stateful	Direction	Source Port Range	Protocol	Destination	Destination Port Range
NSG rule	No	Ingress	CIDR of subnet specified for PrivateEndpoint for Database Migration Connection (Connection)	All	TCP	1521-1523

NSG Associated With Oracle GoldenGate Compute Instance

Rule Type	Stateful	Direction	Source Port Range	Protocol	Destination	Destination Port Range
NSG rule	No	Ingress	CIDR of subnet specified for PrivateEndpoint for Database Migration Connection (Connection) for target database	All	TCP	443

Network security groups (NSGs) are associated with individual virtual network interface cards (VNIC), ADBs, compute instances, and so on. You can configure INGRESS and EGRESS NSG rules.

Security lists apply to entire subnet.

You can use both security lists and NSGs. In this case, a union of security list rules and NSG rules is applied.

For more details, see Comparison of Security Lists and Network Security Groups and If You Use Both Security Lists and Network Security Groups

Private Endpoint Support

CreateConnection supports the following use cases for databases with private IP addresses:

1. Database is in subnetA and customer specifies subnetB to create a PrivateEndpoint:
   - SubnetA must allow INGRESS from subnetB for relevant ports
   - SubnetB must allow EGRESS to subnetA for relevant ports
2. Database is in subnetA and customer selects subnetA to create a PrivateEndpoint
   - SubnetA's INGRESS rules must not prohibit subnetA as source for relevant ports
   - SubnetA's EGRESS rules must not prohibit subnetA as destination for relevant ports

Migrating between Oracle Autonomous Database instances with OCI Database Migration service

Following are the prerequisites and procedures for migrating between two Oracle Autonomous Database instances using OCI Database Migration Service. The following data transfer methods can be used: Network File System (NFS) and Database Link (DBLINK).

Required Parameter for Online Migrations

For online migrations, you must use `GGADMIN` or `admin` with the following grants on both source and target database. For example:

```
GRANT SELECT ON SYS.V_$SESSION TO ggadmin;
GRANT SELECT ON SYS.V_$TRANSACTION TO ggadmin;
GRANT SELECT ON SYS.V_$DATABASE TO ggadmin;

ALTER SYSTEM SET GLOBAL_NAMES=FALSE;

ALTER PLUGGABLE DATABASE ADD SUPPLEMENTAL LOG DATA;

GRANT PDB_DBA to ggadmin;
```

In **Migration**, go to **Replication**.

In **Advanced parameters**, when performing an online migration, you must set the following advanced parameter in the migration configuration:

In `GOLDENGATESETTINGS`, set the value of `Skip Oracle GoldenGatereload_query_advisor` to `TRUE`.

Migration Method 1: Using a Network File System (NFS)

This method uses a shared OCI File Storage (NFS) file system to transfer the Data Pump files. The service orchestrates the mounting and unmounting of this file system directly on the ADB instances.

Before you begin, you must unmount the specific file system when you select in the Oracle Cloud Console.

1. Unmount File System: On both the source and target ADB instances, ensure the OCI File Storage (NFS) you plan to use for the migration is unmounted.
2. Clean the source directory: On the source ADB instance, you must clean up ALL files within the DATA_PUMP_DIR directory. The service automatically uses this directory path to mount the NFS share, and the mount operation will fail if the directory is not empty.
3. Prepare the target directory: On the target ADB instance, you can either specify an existing empty directory or let the service create one. Oracle recommends to let the service create the directory, as it will automatically handle mounting the file system for you.

After completing the prerequisites, you can run the migration job. The service manages the attachment of the NFS file system for the data transfer.

Migration Method 2: Using a Database Link (DBLINK)
Create Database Links: You must create a database link on both the source and target ADB Shared instances.

> **Note**
>
> The database links on the source and target must have the exact same name. For detailed instructions, see Use Database Links with Autonomous Database.

Known issues

Refer to the Known Issues for OCI Database Migration.

Known issue: If you have a source database as Standard Edition Oracle Database and if you are not using DB link as the transfer medium, then the export fails with the following error:
`PRGZ-1464 : PRGZ-1464 : Oracle Data Pump dump encryption feature is not available for Oracle Standard Edition Database`

Known issue:
Currently, editions for editionable objects are not migrated, the export is done according to the current object edition that you have in the source database.

Editionable views are not currently migrated and you need to manually create them in the target database.

Known issue
Issue: `PRGZ-3808 | The specified directory path {0} in the parameter "DATAPUMPSETTINGS.EXPORTDIRECTORYOBJECT.PATH" does not match with the path {0 } that the database directory object {0 } points to.`

Solution: You can either update the directory object on the database to match the directory object path that you provide in the migration. Alternatively, you can update the migration with the correct directory object path and the directory object name. Analyze the following query:

```
SELECT directory_name, directory_path
FROM dba_directories
WHERE directory_name = '<DIRECTORY_OBJECT_NAME>';
```

Known issue

Issue: While creating a migration from an Autonomous database to an Autonomous database, evaluation job fails whenever the `ADMIN` user is configured for initial load and has an `insufficient privileges` error.

Solution: As a workaround, when performing an online migration, you must set the value of the `GOLDENGATESETTINGS_SKIPRELOADQUERYADVISOR` advanced parameter in the migration configuration to `true`.

Ensure to add the following permissions:

```
GRANT SELECT ON SYS.V_$SESSION TO ggadmin;
GRANT SELECT ON SYS.V_$TRANSACTION TO ggadmin;
GRANT SELECT ON SYS.V_$DATABASE TO ggadmin;
ALTER SYSTEM SET GLOBAL_NAMES=FALSE;
ALTER PLUGGABLE DATABASE ADD SUPPLEMENTAL LOG DATA;
GRANT PDB_DBA to ggadmin;
```

- Migration Method 1: Using a Network File System (NFS):
  This method uses a shared OCI File Storage (NFS) file system to transfer the Data Pump files. The service performs the mounting and unmounting of this file system directly on the ADB instances.
  1. Before you begin, you must unmount the specific file system that you select for the migration.
  2. Unmount File System: On both the source and target ADB instances, ensure the OCI File Storage (NFS) you plan to use for the migration is unmounted.
  3. Clean Source Directory: On the source ADB instance, you must clean up ALL files within the DATA_PUMP_DIR directory. ZDM automatically uses this directory path to mount the NFS share, and the mount operation will fail if the directory is not empty.
  4. Prepare Target Directory: On the target ADB instance, you can either specify an existing empty directory or let ZDM create one. Oracle recommends to let ZDM create the directory, as it will automatically handle mounting the file system for you.
  5. After completing the prerequisites, run the migration job. ZDM manages the attachment of the NFS file system for the data transfer.
- Migration Method 2: Using a Database Link (DBLINK)
  1. Create Database Links: You must create a database link on both the source and target ADB shared instances. **Important**: The database links on the source and target must have the exact same name. For detailed instructions, see Create Database Links from Autonomous AI Database to Another Autonomous AI Database.
  2. When defining the migration job using the API or SDK, you must specify the name of the database link in the migration transfer medium parameters.
  3. After completing the prerequisites, you can run the migration job.

Known issue
Issue: The online migrations fail when using IPv4+IPv6 subnet. PE (Private Endpoint) and RCE (Remote Code Execution) do not yet support IPv6.

Solution: Currently, when you attempt an online migration and the connections are assigned to a subnet that contains IPv6 CIDR blocks, the migration fails because it would not be able to create the GGS Connection. This is because both PE and RCE do not yet support IPv6. PE has only rolled out IPv6 support to a few regions and the tenancies are yet to be enabled.

Related Topics

- Known Issues for Oracle Cloud Infrastructure Database Migration Service