Oracle® Database

Move to Oracle Cloud Using Zero Downtime Migration

Release 19c
F13927-05
April 2024

Oracle Database Move to Oracle Cloud Using Zero Downtime Migration, Release 19c

F13927-05

Primary Author: Virginia Beecher

Contributors: Ruggero Citton, Jonathan Creighton, Ricardo Gonzalez Acuna, Sam Ravindhran, Kamalesh Ramasamy, Jony Safi, Giridhar Reddy Sangala, Ravi Sharma, Sanjay Singh, Nitin Vira, Soo Huey Wong

Contents

3 Preparing for Database Migration

4 Migrating Your Database with Zero Downtime Migration

Preface

This book provides information about using Zero Downtime Migration to quickly and smoothly move your Oracle databases from on-premises environments and third-party clouds to the Oracle Cloud without incurring any significant downtime.

Audience

This book is intended for database administrators who want to migrate databases to Oracle Cloud Services with minimal downtime.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

See Zero Downtime Migration on the Oracle Help Center for all published Zero Downtime Migration documentation.

See Zero Downtime Migration Release Notes for the latest information about known issues, My Oracle Support notes, and runbooks.

See the README file included with the downloaded Zero Downtime Migration software for additional information about installation.

See Zero Downtime Migration Licensing Information User Manual

Conventions

The following text conventions are used in this document:

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

1

Introduction to Zero Downtime Migration

Learn about the problems Zero Downtime Migration can help you solve, and its architecture, requirements, and supported configurations.

The following topics contain information you need to know before you install and use Zero Downtime Migration.

- About Zero Downtime Migration
  Zero Downtime Migration gives you a quick and easy way to move on-premises databases and Oracle Cloud Infrastructure Classic instances to Oracle Cloud Infrastructure, Oracle Exadata Database Service on Dedicated Infrastructure, and Oracle Exadata Database Service on Cloud@Customer without incurring any significant downtime, by leveraging technologies such as Oracle Data Guard.
- Zero Downtime Migration Capabilities
  The Zero Downtime Migration service has many benefits and is highly customizable.
- Supported Migration Methods
  Depending on your database source and target destination, Zero Downtime Migration supports different migration methods.
- Supported Migration Paths
  Zero Downtime Migration supports a variety of migration paths to the Oracle Cloud Infrastructure, Exadata Cloud Service, and Exadata Cloud at Customer.
- Supported Configurations
  Learn about the configurations and deployments supported by Zero Downtime Migration in this release.
- Zero Downtime Migration Security Provisions
  Zero Downtime Migration permissions and ownership of files and directories, and handling of configurations for security features, are equivalent to those of Oracle Database.
- Zero Downtime Migration Database Server Access
  The Zero Downtime Migration service host needs to access the source and target database servers during a database migration.
- Target Placeholder Database Environment
  Zero Downtime Migration requires that a placeholder database target environment be configured using Oracle Cloud Infrastructure Console, and the placeholder database should exist in open mode.
- Zero Downtime Migration Operational Phases
  The Zero Downtime Migration service defines the migration process in units of operational phases.

About Zero Downtime Migration

Zero Downtime Migration gives you a quick and easy way to move on-premises databases and Oracle Cloud Infrastructure Classic instances to Oracle Cloud Infrastructure, Oracle Exadata Database Service on Dedicated Infrastructure, and Oracle Exadata Database

Service on Cloud@Customer without incurring any significant downtime, by leveraging technologies such as Oracle Data Guard.

Zero Downtime Migration uses mechanisms such as backing up the source database to Oracle Cloud Infrastructure Object Storage, creating a standby database (with Data Guard configuration, Oracle Data Guard Maximum Performance protection mode and asynchronous (ASYNC) redo transport mode) in the target environment from the backup, synchronizing the source and target databases, and switching over to the target database as the primary database.

Zero Downtime Migration enables and allows fallback capability after database migration is complete. Upon switchover, the target database running in the Oracle Cloud Infrastructure, Exadata Cloud at Customer, or Exadata Cloud Service becomes the primary database, and the on-premises becomes the standby. If there is SQL*Net connectivity between the new primary and the new standby after the switchover, the configuration continues to synchronize data from the new primary in the Oracle Cloud Infrastructure, Exadata Cloud at Customer, or Exadata Cloud Service to the new standby on-premises. However, if there is no SQL*Net connectivity there is no synchronization from new primary in the Oracle Cloud Infrastructure, Exadata Cloud at Customer, or Exadata Cloud Service to the new standby on-premises.

Zero Downtime Migration also supports offline (backup and recovery) migration. This approach uses mechanisms such as backing up the source database to Oracle Cloud Infrastructure Object Storage and instantiating a new database from this Object Storage backup to Oracle Cloud Infrastructure, Exadata Cloud at Customer, and Exadata Cloud Service. For this migration, no SQL*Net connectivity is needed between the source and target database servers

Zero Downtime Migration is compliant with Oracle Maximum Availability Architecture (MAA) and supports Oracle Database 11g Release 2 (11.2.0.4) and later database releases. Zero Downtime Migration provides a robust, flexible, and resumable migration process that is also easy to roll back. Zero Downtime Migration uses a controlled switchover method for dynamically moving database services to the new database environment in Oracle Cloud Infrastructure, Exadata Cloud at Customer, and Exadata Cloud Service. You can perform and manage a database migration of an individual database or perform database migrations at a fleet level using Zero Downtime Migration.

The server where the Zero Downtime Migration software is installed is called the Zero Downtime Migration service host. You can run one or more database migration jobs from the Zero Downtime Migration service host.

Zero Downtime Migration Capabilities

The Zero Downtime Migration service has many benefits and is highly customizable.

- **Audit capability** - All useractions are audited including actions performed by the migration job.
- **Work flow customization** - Work flow actions (marked by phases) can be customized with pre-useraction and post-useraction plug-ins.
- **Job subsystem** - You can perform and manage database migrations at a fleet scale.
- **Job scheduler** - You can schedule your migration job to run at a future point in time.

- **Pause and resume functionality** - You can pause and resume your migration job if needed, which is useful to conform to a maintenance window, for example.
- **Job rerun ability** - Your migration job can be re-run (resumed) from a point of failure.
- **Job pre-check** - You can run pre-checks for migration tasks to prevent errors during database migration.
- **Compliance** - Zero Downtime Migration is compliant with Oracle Maximum Availability Architecture best practices and supports Oracle Database 11g Release 2 (11.2.0.4.0) and later.

Supported Migration Methods

Depending on your database source and target destination, Zero Downtime Migration supports different migration methods.

Using Zero Downtime Migration to migrate your database from an on-premises or Oracle Cloud Infrastructure Classic source to Oracle Cloud Infrastructure, Exadata Cloud Service, or Exadata Cloud at Customer involves creating a backup of the source database and restoring it to the target database, followed by an Oracle Data Guard sync and switchover of the primary role from the source database to the target database

Zero Downtime Migration supports various migration methods, based on your chosen backup medium. The backup medium can be Oracle Cloud Infrastructure Object Storage, Zero Data Loss Recovery Appliance, or NFS storage.

When migrating a database to Oracle Cloud Infrastructure or Exadata Cloud Service, Object Storage is supported as the backup medium. Migration to an Exadata Cloud at Customer target can use Object Storage, Zero Data Loss Recovery Appliance, or NFS as the backup medium.

If you back up the database to Object Storage or to an NFS mount, then the Zero Downtime Migration service initiates the source database backup and restores it to Oracle Cloud Infrastructure, Exadata Cloud at Customer, and Exadata Cloud Service. The backup medium should be accessible from both the source and target environments. When backing up to Object Storage, the Zero Downtime Migration service host uses an SSH connection to the source and target database servers to install and configure the backup module software necessary to back up to and restore from Object Storage. The backup from the source database to Object Storage takes place over an RMAN channel.

If Zero Data Loss Recovery Appliance is chosen as backup medium, then you must ensure that the Zero Data Loss Recovery Appliance has a valid backup of the source database, because Zero Downtime Migration does not initiate a backup to Zero Data Loss Recovery Appliance as part of the workflow.

You must also ensure that all instances of the database are up before initiating a backup to Zero Data Loss Recovery Appliance. The duplicate database operation might fail if the backup is initiated when an instance is down.

The Zero Downtime Migration service accesses the backup in Zero Data Loss Recovery Appliance and restores it to Exadata Cloud at Customer. The Zero Data Loss Recovery Appliance access credentials and wallet location are mandatory input arguments, so that Zero Downtime Migration can handle the Zero Data Loss Recovery Appliance wallet setup at the target database.

Any transfer of redo stream between the source and the target database server, in either direction, takes place over a SQL*Net link.

Supported Migration Paths

Zero Downtime Migration supports a variety of migration paths to the Oracle Cloud Infrastructure, Exadata Cloud Service, and Exadata Cloud at Customer.

The following topics discuss supported migration paths:

- Migrating an On-Premises Database to Oracle Cloud Infrastructure
  You can migrate an Oracle on-premises database to Oracle Cloud Infrastructure (either virtual machine or bare metal) with Zero Downtime Migration.
- Migrating an On-Premises Database to Oracle Exadata Cloud at Customer
  You can migrate on-premises databases to Oracle Exadata Cloud at Customer environments with Zero Downtime Migration.
- Migrating an Oracle Cloud Infrastructure Classic Database to Oracle Cloud Infrastructure
  You can migrate a database in Oracle Cloud Infrastructure Classic to the Oracle Cloud Infrastructure (either virtual machine or bare metal) with Zero Downtime Migration.
- Migrating an On-Premises Database to Exadata Cloud Service
  You can migrate an Oracle on-premises database to Exadata Cloud Service with Zero Downtime Migration.

Migrating an On-Premises Database to Oracle Cloud Infrastructure

You can migrate an Oracle on-premises database to Oracle Cloud Infrastructure (either virtual machine or bare metal) with Zero Downtime Migration.

Zero Downtime Migration requires that you use Oracle Cloud Infrastructure Object Storage service as the intermediate backup medium to migrate on-premises databases to Oracle Cloud Infrastructure.

Migrating an On-Premises Database to Oracle Exadata Cloud at Customer

You can migrate on-premises databases to Oracle Exadata Cloud at Customer environments with Zero Downtime Migration.

Zero Downtime Migration requires that you use Object Storage Service (OSS), Zero Data Loss Recovery Appliance (ZDLRA), or a Network File System (NFS) as the intermediate backup medium to migrate on-premises databases to Oracle Exadata Cloud at Customer environments.

Migrating an Oracle Cloud Infrastructure Classic Database to Oracle Cloud Infrastructure

You can migrate a database in Oracle Cloud Infrastructure Classic to the Oracle Cloud Infrastructure (either virtual machine or bare metal) with Zero Downtime Migration.

Zero Downtime Migration requires that you use Oracle Cloud Infrastructure Object Storage service as the intermediate backup medium to migrate a database in Oracle Cloud Infrastructure Classic to the Oracle Cloud Infrastructure.

Migrating an On-Premises Database to Exadata Cloud Service

You can migrate an Oracle on-premises database to Exadata Cloud Service with Zero Downtime Migration.

Zero Downtime Migration requires that you use Oracle Cloud Infrastructure Object Storage service as the intermediate backup medium to migrate on-premises databases to Oracle Cloud Infrastructure.

Supported Configurations

Learn about the configurations and deployments supported by Zero Downtime Migration in this release.

Zero Downtime Migration currently supports the platforms, database architectures, and database versions discussed in the following topics.

- Supported Platforms
  Zero Downtime Migration supports the following platforms for the service host and the migration source and target database servers.
- Supported Database Versions for Migration
  Zero Downtime Migration supports most Oracle Database versions available on Oracle Cloud Infrastructure, Exadata Cloud at Customer, and Exadata Cloud Service.
- Supported Database Architectures for Migration
  Zero Downtime Migration supports Oracle Database Single-Instance, Oracle RAC One Node, and Oracle Real Application Clusters (RAC) databases.
- Supported Backup Locations
  Zero Downtime Migration supports the best performing and most popular backup mediums.

Supported Platforms

Zero Downtime Migration supports the following platforms for the service host and the migration source and target database servers.

Zero Downtime Migration Service Host - Supported Platforms

The Zero Downtime Migration service host can be configured on Oracle Linux 7 or later releases.

You can deploy the Zero Downtime Migration service on a standalone server on-premises or on a standalone Linux server (bare metal or virtual machine) in the Oracle Cloud. Oracle Linux is the supported platform for the Zero Downtime Migration service node.

Note that the Zero Downtime Migration service host can be shared with other applications for other purposes; however, no Oracle Grid Infrastructure instance should be running on the Zero Downtime Migration service host.

Source and Target Database Servers - Supported Platforms

Linux-x86-64 is the supported platform for migration source and target database servers.

Supported Database Versions for Migration

Zero Downtime Migration supports most Oracle Database versions available on Oracle Cloud Infrastructure, Exadata Cloud at Customer, and Exadata Cloud Service.

The following Oracle Database versions can be migrated using Zero Downtime Migration.

- 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 18 Release 3 (18.3)
- All subsequent Oracle Database releases

Because Zero Downtime Migration leverages Oracle Data Guard, you must have the same operating system and database version on both source and target.

> **Note:**
>
> Zero Downtime Migration does not support cross-edition migration. Zero Downtime Migration cannot be used to migrate an Enterprise edition database to a Standard edition database, and vice versa.

Supported Database Architectures for Migration

Zero Downtime Migration supports Oracle Database Single-Instance, Oracle RAC One Node, and Oracle Real Application Clusters (RAC) databases.

Zero Downtime Migration supports the following database architecture implementations.

- Oracle Database Single-Instance, which can be migrated to a single-instance or Oracle RAC database target
- Oracle RAC One Node, which can be migrated to an Oracle RAC database target
- Oracle RAC, which can be migrated to an Oracle RAC database target

Supported Backup Locations

Zero Downtime Migration supports the best performing and most popular backup mediums.

Depending on your target environment, Zero Downtime Migration supports the following backup mediums.

- Object Storage Service (OSS)
- Zero Data Loss Recovery Appliance (ZDLRA)
- External Backup Location (NFS)

Refer to the product documentation for your chosen backup medium for information about creating backups and object stores.

Zero Downtime Migration Security Provisions

Zero Downtime Migration permissions and ownership of files and directories, and handling of configurations for security features, are equivalent to those of Oracle Database.

Zero Downtime Migration installs in a location, named `ZDM_HOME`, that is structured similarly to the Oracle home directory, `ORACLE_HOME`, for Oracle Database. The permissions and ownership of files and directories in the `ZDM_HOME` follow the same conventions as that of a database `ORACLE_HOME`.

Zero Downtime Migration also creates a base directory structure for storing Zero Downtime Migration configuration files, logs, and other artifacts, named `ZDM_BASE`, that is similar to an Oracle base directory, `ORACLE_BASE`, that is associated with an Oracle home. The structure, owners, and permissions of directories and files in `ZDM_BASE` are similar to that of an `ORACLE_BASE`.

You do not need to do any additional steps to ensure security the of the Zero Downtime Migration configuration because the Zero Downtime Migration configuration is designed to be secure out of the box.

Zero Downtime Migration is configured to accept JMX connections only from the local host, and to listen on the loopback address for HTTP connections. Zero Downtime Migration operations can only be performed by the operating system user that installed the product.

SSH connectivity from the Zero Downtime Migration host to the on premise source node and the cloud target node is required. You must provide the SSH key file location as an input for a migration job, and the existence of this file is expected for the duration of the migration job. You must manage the security of the directories and files where these key files are located.

You can modify the communication ports when there is a port conflict with another application. Note that access to these ports are configured only from within the Zero Downtime Migration host. You can change the RMI and HTTP port properties in the file `$ORACLE_BASE/crsdata/<hostname>/rhp/conf/standalone_config.properties`.

The properties are:

- RMI port - `oracle.jwc.rmi.port=8895`
- HTTP port - `oracle.jwc.http.port=8896`

Bounce the Zero Downtime Migration server after changing the properties.

When Zero Downtime Migration operations require passwords, prompts are given for password entry. Passwords are encrypted and stored in the Zero Downtime Migration database. Provided passwords are not expected to change for the duration of a migration job.

From an operation perspective, Zero Downtime Migration follows the guidelines in *Oracle Database Security Guide* for handling source and target database configurations for migration, such as Oracle Wallets, Transparent Data Encryption, and so on.

> **See Also:**
>
> Configuring Connectivity Prerequisites
>
> Oracle Database Security Guide

Zero Downtime Migration Database Server Access

The Zero Downtime Migration service host needs to access the source and target database servers during a database migration.

To perform the migration, the Zero Downtime Migration service host requires either root user or SSH key-based access to one of the source database servers, and the Zero Downtime Migration service host requires SSH key-based access to one of the target database servers. If you are migrating an Oracle RAC database, providing access to one of the Oracle RAC nodes is adequate. The Zero Downtime Migration service host copies the software needed for migration to the source and target servers and cleans it up at the end of the operation.

An SSH private key is required to establish SSH connections. This generated key must not use a passphrase. You can create and add a new SSH key to your existing deployment using the Oracle Cloud Service Console.

Target Placeholder Database Environment

Zero Downtime Migration requires that a placeholder database target environment be configured using Oracle Cloud Infrastructure Console, and the placeholder database should exist in open mode.

You have complete control over the configuration of the placeholder database target environment, so you can set up and configure it as required for your needs. The Zero Downtime Migration service host restores the source database to this placeholder database target environment. Note that the placeholder database must be set up with the same database version as the source database and the same or higher patch level as the source database.

The database parameters for the target database, including SGA parameters, should be set as needed. These settings are maintained during the migration, and the migrated database runs with this same configuration. Any modifications to database parameters can be performed after the migration.

If the target database environment is at a higher patch level than the source database, you must run the datapatch utility on the target database as a post-migration task. For example, if your source database is at Oct 2018 PSU/BP and the target is at Jan 2019 PSU/BP), you must run the datapatch utility.

The SYS password specified should match that of source database.

The name of the provisioned database must match one of the following patterns, based on the target platform.

- For Exadata Cloud at Customer or Exadata Cloud Service targets, Zero Downtime Migration drops the placeholder database and recreates a database in the target environment with the same `db_name` as that of source database, using the backup

and the `db_unique_name` parameter specified in the response file parameter, `TGT_DB_UNIQUE_NAME`. The target database `db_unique_name` parameter value must be unique to ensure that Oracle Data Guard can identify the target as a different database from the on-premises primary database. Therefore, the value specified in `TGT_DB_UNIQUE_NAME` should be different from that of source database's `db_unique_name`.

- For virtual machine or bare metal targets, when using the Oracle Data Guard migration method, create the target database with a *different* `db_name` from that of the source database. This value must be specified as the parameter `TGT_DB_UNIQUE_NAME` in the response file. During standby creation, Zero Downtime Migration replaces the placeholder database data files with the data files restored from the object store backup. Zero Downtime Migration updates the `db_name` of the target to be the same as that of the source.

Once the migration is complete, the target database is accessible using Oracle Database Cloud Service console, and you can manage the database with SRVCTL commands using the `db_unique_name` value for the target database.

Any modifications to database parameters can be performed after the migration.

Zero Downtime Migration Operational Phases

The Zero Downtime Migration service defines the migration process in units of operational phases.

Zero Downtime Migration auto computes the migration workflow using defined operational phases based on configured input parameters, such as the target platform, backup medium, and so on. You can customize the workflow by inserting custom plug-ins on each of the operational phases. The Zero Downtime Migration service lets you pause and resume the migration workflow at any chosen operational phase.

Migration workflow-associated phases for a given operation can be listed. Phases that are performed on the source database server are listed with a _SRC suffix, and the phases associated with the target database server are listed with a _TGT suffix.

2

Setting Up Zero Downtime Migration Software

Follow prerequisite tasks and procedures for installing and uninstalling Zero Downtime Migration software.

The following topics detail the prerequisites and procedures for installing and uninstalling Zero Downtime Migration software.

- About Setting Up Zero Downtime Migration Software
  Zero Downtime Migration software is installed on a standalone Linux server.
- Prerequisites for Zero Downtime Migration Software Installation
  Complete the following prerequisites before installing Zero Downtime Migration software on the Zero Downtime Migration service host.
- Installing Zero Downtime Migration Software
  Download the Zero Downtime Migration software and install it on the Zero Downtime Migration service host.
- Uninstalling Zero Downtime Migration Software
  You can uninstall the Zero Downtime Migration software using this procedure.

About Setting Up Zero Downtime Migration Software

Zero Downtime Migration software is installed on a standalone Linux server.

Zero Downtime Migration software requires a Linux host running Oracle Linux 7 or later. Once the software is installed on the Linux host, that host is referred to as the Zero Downtime Migration service host. The Zero Downtime Migration service host should be a dedicated system, but it can be shared for other purposes; however, the Zero Downtime Migration service host should not have Oracle Grid Infrastructure running on it.

Prerequisites for Zero Downtime Migration Software Installation

Complete the following prerequisites before installing Zero Downtime Migration software on the Zero Downtime Migration service host.

- Ensure that the Linux host has 100 GB of free storage space.
- You may use an existing user, or, on the Zero Downtime Migration service host, as root user, create a `zdm` group and add `zdmuser` user to the group.

  For example,

  ```
  root> groupadd zdm
  root> useradd -g zdm zdmuser
  ```

- Verify that the `glibc-devel` and `expect` packages are installed.

  For Oracle Linux 7 installations with Base Environment "Minimal Install" you also need to install the packages `unzip libaio oraclelinux-developer-release-el7`.

- Verify that the `/etc/hosts` entry for the host name and IP address are configured as expected, so that the host selected for Zero Downtime Migration software installation resolves to the correct IP address and the IP address is reachable with `ping`.
- During the installation, the script might report any missing packages and instructions for setting appropriate values for kernel parameters. Be sure to install the missing packages and set the kernel parameters before the Zero Downtime Migration software installation.

Installing Zero Downtime Migration Software

Download the Zero Downtime Migration software and install it on the Zero Downtime Migration service host.

See the Zero Downtime Migration Release Notes for the latest information about known issues. Also, see the README file included with the downloaded Zero Downtime Migration software for any additional information about installation.

All commands are run as `zdmuser`.

1. Download the Zero Downtime Migration software kit from https://www.oracle.com/database/technologies/rac/zdm-downloads.html to the Zero Downtime Migration service host.
2. Install the Zero Downtime Migration software as a non-root user.

   In this example the installation user is `zdmuser`.

   a. Change to the directory to where Zero Downtime Migration software is downloaded and unzip the software.

   ```
   zdmuser> cd zdm_download_directory
   zdmuser> unzip zdm_home.zip
   ```

   b. Run the Zero Downtime Migration installation script.

   ```
   zdmuser>./zdminstall.sh setup oraclehome=zdm_oracle_home
   oraclebase=zdm_base_directory
           ziploc=zdm_software_location -zdm
   ```

   - `zmdinstall.sh` is the installation script
   - `oraclehome` is the Oracle Home where the Zero Downtime Migration kit will be installed
   - `oraclebase` is the base directory where all of the Zero Downtime Migration configuration files, logs, and other artifacts are stored
   - `ziploc` is the location of the compresed software file (zip) included in the Zero Downtime Migration kit

   For example,

   ```
   zdmuser>./zdminstall.sh setup oraclehome=/u01/app/zdmhome
           oraclebase=/u01/app/zdmbase ziploc=/u01/app/oracle/zdm/
   ```

```
    shiphome/zdm_home.zip
            -zdm
```

Hereafter, the `oraclehome` value is referred to as `ZDM_HOME`, and the `oraclebase` value is referred to as `ZDM_BASE`.

Ignore the following messages which are displayed on the terminal at the end of installation. There is no need to run these scripts.

```
As a root user, execute the following script(s):
        1. /u01/app/zdmhome/inventory/orainstRoot.sh
        2. /u01/app/zdmhome/root.sh
```

3. Start the Zero Downtime Migration service as user `zdmuser`.

```
zdmuser> /u01/app/zdmhome/bin/zdmservice start
```

You must start `zdmservice` before you can migrate your databases using Zero Downtime Migration.

If you must stop the Zero Downtime Migration service, run the following command.

```
zdmuser> /u01/app/zdmhome/bin/zdmservice stop
```

4. Verify that the Zero Downtime Migration service installation is successful.

When you run the following command, the output should be similar to that shown here.

```
zdmuser> /u01/app/zdmhome/bin/zdmservice status
---------------------------------------
        Service Status
---------------------------------------

Running: true
Tranferport: 5000-7000
Conn String: jdbc:derby:/u01/app/base/derbyRepo;create=true
Repo Path: /u01/app/base/derbyRepo
RMI port: 8895
HTTP port: 8896
Wallet path: /u01/app/base/crsdata/fopds/security
```

Uninstalling Zero Downtime Migration Software

You can uninstall the Zero Downtime Migration software using this procedure.

All commands are run as `zdmuser`.

1. Stop the Zero Downtime Migration service.

```
zdmuser> /u01/app/zdmhome/bin/zdmservice stop
```

2. Run the following command to uninstall the software.

```
zdmuser> /u01/app/zdmhome/bin/zdmservice deinstall
```

3

Preparing for Database Migration

Before starting a Zero Downtime Migration database migration configure connectivity, prepare the database, and configure any required migration job customization.

See the Zero Downtime Migration Release Notes for the latest information about known issues, My Oracle Support notes, and runbooks.

- Configuring Connectivity Prerequisites
  Connectivity must be set up between the Zero Downtime Migration service host and the source and target database servers.
- Preparing the Database for Migration
  Prepare the source and target databases for the migration.
- Preparing for Automatic Application Switchover
  To minimize or eliminate service interruptions on the application after you complete the database migration and switchover, prepare your application to automatically switch over connections from the source database to the target database.
- Customizing a Migration Job
  You can customize the Zero Downtime Migration workflow by registering action scripts or plug-ins as pre-actions or post-actions to be performed as part of the operational phases involved in your migration job.

Configuring Connectivity Prerequisites

Connectivity must be set up between the Zero Downtime Migration service host and the source and target database servers.

The following topics describe how to configure the Zero Downtime Migration connectivity prerequisites before running a migration job.

- Configuring Connectivity From the Zero Downtime Migration Service Host to the Source and Target Database Servers
  Complete the following procedure to ensure the required connectivity between the Zero Downtime Migration service host and the source and target database servers.
- Configuring Connectivity Between the Source and Target Database Servers
  You can configure connectivity between the source and target database servers using one of two options.
- Generating a Private SSH Key Without a Passphrase
  If, on the Zero Downtime Migration service host, source database server, or target database server, the authentication key pairs are not available without a passphrase for the Zero Downtime Migration software installed user, you can generate a new SSH key using the following procedure.
- Setting Up the Transparent Data Encryption Wallet
  For Oracle Database 12c Release 2 and later, if the source database does not have TDE enabled, then it is mandatory that you configure the TDE wallet before migration begins. Enabling TDE on Oracle Database 11g Release 2 (11.2.0.4) and Oracle Database 12c Release 1 is not required.

Configuring Connectivity From the Zero Downtime Migration Service Host to the Source and Target Database Servers

Complete the following procedure to ensure the required connectivity between the Zero Downtime Migration service host and the source and target database servers.

1. On the Zero Downtime Migration service host, verify that the authentication key pairs are available without a passphrase for the Zero Downtime Migration software installed user.

   If a new key pair must be generated without the passphrase, then, as a Zero Downtime Migration software installed user, generate new key pairs as described in Generating a Private SSH Key Without a Passphrase.

2. Rename the private key file.

   Rename the *ZDM_installed_user_home*/.ssh/id_rsa file name to *ZDM_installed_user_home*/.ssh/*ZDM_service_node_name*.ppk.

3. Add the contents of the *ZDM_installed_user_home*/.ssh/id_rsa.pub file to the *opc_user_home*/.ssh/authorized_keys file, with the following dependencies:

   - If the source database is on Oracle Cloud Infrastructure Classic, then add the contents of the *ZDM_installed_user_home*/.ssh/id_rsa.pub file into the *opc_user_home*/.ssh/authorized_keys file on all of the source database servers.
     Note that the `opc` user is a standard Oracle Cloud user that is used to access the target database servers.
   - If the source database servers have root access, no action is required.
   - If the target database is on Oracle Cloud Infrastructure, Exadata Cloud at Customer, or Exadata Cloud Service, then add the contents of the *ZDM_installed_user_home*/.ssh/id_rsa.pub file into the *opc_user_home*/.ssh/authorized_keys file on *all* of the target database servers.

4. Make sure that the source and target database server names specified in the command are resolvable from the Zero Downtime Migration service host through either resolving name servers or alternate ways approved by your IT infrastructure.

   One method of resolving source and target database server names is to add the source and target database server names and IP address details to the Zero Downtime Migration service host /etc/hosts file.
   For example,

   ```
   #OCI public IP two node RAC server details
   192.0.2.1 zdmhost1
   192.0.2.2 zdmhost2
   #OCIC public IP two node RAC server details
   192.0.2.6 ocicdb1
   192.0.2.7 ocicdb2
   ```

5. Make certain that port 22 in the source and target database servers accept incoming connections from the Zero Downtime Migration service host.

6. Test the connectivity from the Zero Downtime Migration service host to all source and target database servers.

```
zdmuser> ssh -i ZDM_service_node_private_key_file_location user@source/
target_database_server_name
```

For example,

```
zdmuser> ssh -i /home/zdmuser/.ssh/zdm_service_node.ppk opc@zdmhost1
zdmuser> ssh -i /home/zdmuser/.ssh/zdm_service_node.ppk opc@ocicdb1
```

> **See Also:**
>
> Zero Downtime Migration Port Requirements

Configuring Connectivity Between the Source and Target Database Servers

You can configure connectivity between the source and target database servers using one of two options.

Option 1

The source database server specified in the ZDMCLI command `-sourcenode` parameter can connect to target database instance over target SCAN through the respecitve scan port and vice versa. The SCAN of the target should be resolvable from the source database server, and the SCAN of the source should resolve from the target server. Having connectivity from both sides, you can synchronize between the source database and target database from either side. If the source database server SCAN cannot be resolved from the target database server, then the `SKIP_FALLBACK` parameter in the response file must be set to `TRUE`, and you cannot synchronize between the target database and source database.

Option 2

If connectivity through SCAN and the SCAN port is not possible between the source and target database servers, set up an SSH tunnel from the source database server to the target database server using the procedure below. Using this option, you will not be able to synchronize between the target database and source database.

Note that this procedure amounts to setting up what may be considered a temporary channel. You can choose to set up access without using an SSH tunnel.

> **Note:**
>
> The following steps refer to Oracle Cloud Infrastructure, but are also applicable to Exadata Cloud at Customer and Exadata Cloud Service.

1. Set up an SSH tunnel on the source database servers for the root user.

a. Generate a private SSH key file without a passphrase for the `opc` user on the target Oracle Cloud Infrastructure server, using the information in Generating a Private SSH Key Without a Passphrase. If the target is an Oracle RAC database, then generate a private SSH key file without a passphrase from the first Oracle RAC server.

b. Add the contents of the Oracle Cloud Infrastructure server opc_user_home/.ssh/id_rsa.pub file into the Oracle Cloud Infrastructure server opc_user_home/.ssh/authorized_keys file.

c. Copy the target Oracle Cloud Infrastructure server private SSH key file onto the source server in the /root/.ssh/ directory. If the source is an Oracle RAC database, copy the file into all of the source servers.

   For better manageability, keep the private SSH key file name the same as the target server name, and keep the .ppk extension. For example, rptest.ppk (where rptest is the target server name).
   The file permissions should be similar to the following.

```
/root/.ssh>ls -l rptest.ppk
-rw------- 1 root root 1679 Oct 16 10:05 rptest.ppk
```

d. Put the following entries in the source server /root/.ssh/config file.

```
Host *
  ServerAliveInterval 10
  ServerAliveCountMax 2

Host OCI_server_name
  HostName OCI_server_IP_address
  IdentityFile Private_key_file_location
  User OCI_user_login
  ProxyCommand /usr/bin/nc -X connect -x proxy_name:proxy_port
%h %p
```

   Where

   - *OCI_server_name* is the Oracle Cloud Infrastructure target database server name without the domain name. For an Oracle RAC database use the first Oracle RAC server name without the domain name.
   - *OCI_server_IP_address* is the Oracle Cloud Infrastructure target database server IP address. For an Oracle RAC database use the first Oracle RAC server IP address.
   - *Private_key_file_location* is the location of the private key file.
   - *OCI_user_login* is the OS user used to access the target database servers.
   - *proxy_name* is the host name of the proxy server.
   - *proxy_port* is the port of the proxy server.

   Note that the proxy setup might not be required when you are not using a proxy server for connectivity. For example, when the source database server is on Oracle Cloud Infrastructure Classic, you can remove or comment the line starting with ProxyCommand.

For example, after specifying the relevant values, the /root/.ssh/config file should be similar to the following.

```
Host *
  ServerAliveInterval 10
  ServerAliveCountMax 2

Host rptest
  HostName 192.0.2.9
  IdentityFile /root/.ssh/rptest.ppk
  User opc
  ProxyCommand /usr/bin/nc -X connect -x www-proxy.example.com:80 %h
%p
```

The file permissions should be similar to the following.

```
/root/.ssh>ls -l config
-rw------- 1 root root 1679 Oct 16 10:05 config
```

In the above example, the Oracle Cloud Infrastructure server name is rptest, and the Oracle Cloud Infrastructure server public IP address is 192.0.2.9.

If the source is an Oracle Cloud Infrastructure Classic server, the *proxy_name* is not required, so you can remove or comment the line starting with `ProxyCommand`.

If the source is an Oracle RAC database, then copy the same /root/.ssh/config file onto all of the source Oracle RAC database servers. This file will have the Oracle Cloud Infrastructure server name, Oracle Cloud Infrastructure server public IP address, and private key file location of first Oracle Cloud Infrastructure Oracle RAC server information configured.

e. Make sure that you can SSH to first target Oracle Cloud Infrastructure server from the source server before you enable the SSH tunnel.

For an Oracle RAC database, test the connection from all of the source servers to the first target Oracle Cloud Interface server.
Using the private key:

```
[root@ocic121 ~] ssh -i /root/.ssh/rptest.ppk opc@rptest
Last login: Fri Dec  7 14:53:09 2018 from 192.0.2.11

[opc@rptest ~]$
```

f. Run the following command on the source server to enable the SSH tunnel.

```
ssh -f OCI_hostname_without_domain_name -L
ssh_tunnel_port_number:OCI_server_IP_address:OCI_server_listener_port
-N
```

Where

- *OCI_hostname_without_domain_name* is the Oracle Cloud Infrastructure target database server name without a domain name. For an Oracle RAC database use the first Oracle RAC server name without domain name.

- *ssh_tunnel_port_number* is any available ephemeral port in the range (1024-65545). Make sure that the SSH tunnel port is not used by any other process in the server before using it.
- *OCI_server_listener_port* is the target database listener port number. The listener port must be open between the source database servers and Oracle Cloud Infrastructure target servers.
- *OCI_server_IP_address* is configured based on database architecture. For a single instance database, specify the Oracle Cloud Infrastructure server IP address. For an Oracle RAC database, specify the Oracle Cloud Infrastructure scan name with the domain name. If the scan name with domain name is not resolvable or not working, then specify the IP address obtained using the `lsnrctl status` command output. For example,

```
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.0.2.9)
(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.0.2.10)
(PORT=1521)))
```

The following is an example of the command run to enable the SSH tunel.

```
[root@ocic121~]ssh -f rptest -L 9000:192.0.2.9:1521 -N
```

For an Oracle RAC database, this step must be repeated on all of the source servers.

g. Test the SSH tunnel.

Log in to source server, switch to the `oracle` user and source the database environment, and run the following command.

```
tnsping localhost:ssh_tunnel_port
```

For example,

```
[oracle@ocic121 ~] tnsping localhost:9000
```

The command output is similar to the following.

```
TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on
22-JAN-2019 05:41:57
Copyright (c) 1997, 2014, Oracle.  All rights reserved.
Used parameter files:
Used HOSTNAME adapter to resolve the alias
Attempting to contact (DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=))
(ADDRESS=(PROTOCOL=TCP)(HOST=127.0.0.1)(PORT=9000)))
OK (50 msec)
```

If tnsping does not work, then the SSH tunnel is not enabled.

For Oracle RAC, this step must be repeated on all of the source servers.

2. Test connectivity from the source to target environments.

   Add the TNS entry of the target database to the source database server $ORACLE_HOME/network/admin/tnsnames.ora file.

   ```
   [oracle@sourcedb ~] tnsping target-tns-string
   ```

3. Test connectivity from the target to the source environment.

   Add the TNS entry of the source database to the target database server $ORACLE_HOME/network/admin/tnsnames.ora file

   ```
   [oracle@targetdb ~] tnsping source-tns-string
   ```

> **Note:**
>
> Database migration to Exadata Cloud at Customer using the Zero Data Loss Recovery Appliance requires mandatory SQL*Net connectivity from the target host to the source database.

Note:

> **See Also:**
>
> Zero Downtime Migration Port Requirements

Generating a Private SSH Key Without a Passphrase

If, on the Zero Downtime Migration service host, source database server, or target database server, the authentication key pairs are not available without a passphrase for the Zero Downtime Migration software installed user, you can generate a new SSH key using the following procedure.

SSH connectivity during Zero Downtime Migration operations requires direct, non-interactive access between the Zero Downtime Migration service host and the source and target database servers, and also between the source and target database servers, without the need to enter a passphrase.

> **Note:**
>
> The following steps show examples for generating a private SSH key for the software installed user. You can also use these steps for the `opc` user.

Run the following command as the Zero Downtime Migration software installed user on the Zero Downtime Migration service host.

```
zdmuser> ssh-keygen
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/home/opc/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/opc/.ssh/id_rsa.
Your public key has been saved in /home/opc/.ssh/id_rsa.pub.
The key fingerprint is:
c7:ed:fa:2c:5b:bb:91:4b:73:93:c1:33:3f:23:3b:30 opc@rhost1
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|                 |
|         . . .   |
|        S o . =  |
|         . E . * |
|            X.+o.|
|           .= Bo.o|
|           o+*o.  |
+-----------------+
```

This command generates the id_rsa and id_rsa.pub files in the zdmuser home, for example, /home/zdmuser/.ssh.

You can add the public key (for example, /home/zdmuser/.ssh/id_rsa.pub) to the source and target database servers using the Oracle Cloud Infrastructure Console, or you can add it manually to the authorized_keys file on those servers, as shown below.

Add the contents of the Zero Downtime Migration service host /home/zdmuser/.ssh/id_rsa.pub file to the Oracle Cloud Infrastructure server opc user /home/opc/.ssh/authorized_keys file, as shown here.

```
[opc@rptest.ssh]$ export PS1='$PWD>'
/home/opc/.ssh>ls
authorized_keys  authorized_keys.bkp  id_rsa  id_rsa.pub  known_hosts
zdmkey
/home/opc/.ssh>cat id_rsa.pub >> authorized_keys
```

You should save the private key in a separate, secure file, and use it to connect to the source and target database servers. For example, create a zdm_service_node.ppk file with permissions set to 600, and put the private key file into it on the Zero Downtime Migration service host software installed user home/.ssh to connect source and target database servers.

Setting Up the Transparent Data Encryption Wallet

For Oracle Database 12c Release 2 and later, if the source database does not have TDE enabled, then it is mandatory that you configure the TDE wallet before migration begins. Enabling TDE on Oracle Database 11g Release 2 (11.2.0.4) and Oracle Database 12c Release 1 is not required.

If Transparent Data Encryption (TDE) is not already configured as required on the source and target databases, use the following instructions to set up the (TDE) wallet. TDE should be enabled, the `WALLET` status on both source and target databases must be set to `OPEN`, and the `WALLET_TYPE` must be set to `AUTOLOGIN`.

1. Set `ENCRYPTION_WALLET_LOCATION` in $ORACLE_HOME/network/admin/sqlnet.ora file.

```
$ cat /u01/app/oracle/product/12.2.0.1/dbhome_2/network/admin/sqlnet.ora

ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=FILE)
  (METHOD_DATA=(DIRECTORY=/u01/app/oracle/product/12.2.0.1/dbhome_2/
network/admin/)))
```

2. Connect to the database and configure the keystore.

```
$ sqlplus "/as sysdba"
SQL> ADMINISTER KEY MANAGEMENT CREATE KEYSTORE '/u01/app/oracle/product/
12.2.0.1/dbhome_2/network/admin'
 identified by **********;
keystore altered.
```

For a non-CDB environment, run the following command.

```
SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY **********;
keystore altered.
```

For a CDB environment, run the following command.

```
SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY **********
container = ALL;
```

For a non-CDB environment, run the following command.

```
SQL> ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY ********** with
backup;
keystore altered.
```

For a CDB environment, run the following command.

```
SQL> ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY ********** with
backup container = ALL;
```

Then run,

```
SQL> select * FROM v$encryption_keys;
```

3. Set up autologin.

```
SQL> SELECT * FROM v$encryption_wallet;

WRL_TYPE    WRL_PARAMETER
--------------------
--------------------------------------------------------------------------
------
STATUS                         WALLET_TYPE          WALLET_OR
```

```
FULLY_BAC    CON_ID
------------------------------ -------------------- ---------
---------            ----------
FILE        /u01/app/oracle/product/12.2.0.1/dbhome_2/network/admin/
OPEN                           PASSWORD             SINGLE
NO         0

SQL> ADMINISTER KEY MANAGEMENT CREATE AUTO_LOGIN KEYSTORE FROM
KEYSTORE
 '/u01/app/oracle/product/12.2.0.1/dbhome_2/network/admin/'
IDENTIFIED BY **********;
keystore altered.
```

If you are using an Oracle RAC database, copy the files below to the same location on each cluster node, or to a shared file system.

```
/u01/app/oracle/product/12.2.0.1/dbhome_2/network/admin/ew*
/u01/app/oracle/product/12.2.0.1/dbhome_2/network/admin/cw*

SQL> SELECT * FROM v$encryption_wallet;
WRL_TYPE    WRL_PARAMETER
--------------------
--------------------------------------------------------------------------
------------
STATUS                         WALLET_TYPE          WALLET_OR
FULLY_BAC        CON_ID
------------------------------ -------------------- ---------
---------            ----------
FILE        /u01/app/oracle/product/12.2.0.1/dbhome_2/network/admin/
OPEN                           PASSWORD             SINGLE
NO                0
```

At this stage, the `PASSWORD` based wallet is enabled. To enable an `AUTOLOGIN` based wallet, complete the remaining steps in this procedure.

Close the password wallet.

```
SQL> administer key management set keystore close identified by
**********;
keystore altered.
```

Then verify that autologin is configured. Set `TDE WALLET` status to `OPEN` and `WALLET_TYPE` to `AUTOLOGIN`, otherwise the wallet configuration is not correctly set up.

```
$ sqlplus "/as sysdba"
SQL> SELECT * FROM v$encryption_wallet;
WRL_TYPE WRL_PARAMETER
--------------------
--------------------------------------------------------------------------
------------
STATUS WALLET_TYPE WALLET_OR FULLY_BAC CON_ID
------------------------------ -------------------- ---------
```

```
--------- ---------
FILE /u01/app/oracle/product/12.2.0.1/dbhome_2/network/admin/
OPEN AUTOLOGIN SINGLE NO
```

Upon migration of your Oracle Database to the Oracle Cloud, bear in mind that Oracle databases in the Oracle Cloud are TDE enabled by default. Zero Downtime Migration will take care of the encryption of your target database, even if your source Oracle Database is not TDE enabled by default. However, once the switchover phase of the migration has taken place, the redo logs that the new primary database in the Oracle Cloud sends to the new standby database on your premises will be encrypted. Therefore, if you decide to switch back and role swap again making the on-premises database the primary again and the database in the Oracle Cloud the standby, the on-premises database will not be able to read the newly encrypted changed blocks applied by the redo logs unless TDE is enabled on-premises.

In order to avoid post migration conflict, prior to performing the original switchover as part of the migration process, the recommended best practice is to perform appropriate testing and validation. There are options outside of Zero Downtime Migration for testing with a snapshot standby database, and once you are ready to proceed, delete the snapshot standby database and instruct Zero Downtime Migration to perform the switchover and finalize the migration process.

Preparing the Database for Migration

Prepare the source and target databases for the migration.

See the following topics for information about preparing the source and target databases for migration.

- Source Database Prerequisites
  Meet the prerequisites on the source database before the Zero Downtime Migration process starts.
- Target Database Prerequisites
  The following prerequisites must be met on the target database before you begin the Zero Downtime Migration process.
- Preparing for Migration to Oracle Cloud Infrastructure
  Complete the following preparation before migrating data to an Oracle Cloud Infrastructure virtual machine or bare metal target.
- Preparing for Migration to Exadata Cloud Service
  Complete the following preparation before migrating data to an Exadata Cloud Service target.
- Preparing for Migration to Exadata Cloud at Customer
  Complete the following preparation before migrating data to an Exadata Cloud at Customer target.
- Preparing for Offline Migration (Backup and Recovery)
  Complete the following preparations before migrating a database to an Oracle Cloud Infrastructure, Exadata Cloud at Customer, or Exadata Cloud Service target environment.

Source Database Prerequisites

Meet the prerequisites on the source database before the Zero Downtime Migration process starts.

1. The source database must be running in archive log mode.
2. For Oracle Database 12c Release 2 and later, if the source database does not have Transparent Data Encryption (TDE) enabled, then it is mandatory that you configure the TDE wallet before migration begins. The `WALLET_TYPE` can be `AUTOLOGIN` (preferred) or `PASSWORD` based.
3. Ensure that the wallet `STATUS` is `OPEN` and `WALLET_TYPE` is `AUTOLOGIN` (For an `AUTOLOGIN` wallet type), or `WALLET_TYPE` is `PASSWORD` (For a `PASSWORD` based wallet type). For a multitenant database, ensure that the wallet is open on all PDBs as well as the CDB, and the master key is set for all PDBs and the CDB.

   ```
   SQL> SELECT * FROM v$encryption_wallet;
   ```

4. If the source is an Oracle RAC database, and `SNAPSHOT CONTROLFILE` is not on a shared location, configure `SNAPSHOT CONTROLFILE` to point to a shared location on all Oracle RAC nodes to avoid the ORA-00245 error during backups to Oracle Object Store.

   For example, if the database is deployed on ASM storage,

   ```
   $ rman target /
   RMAN> CONFIGURE SNAPSHOT CONTROLFILE NAME TO '+DATA/
   snapcf_matrix.f';
   ```

   If the database is deployed on an ACFS file system, specify the shared ACFS location in the above command.
5. Verify that port 22 on the source and target database servers allow incoming connections from the Zero Downtime Migration service host.
6. Ensure that the scan listener ports (1521, for example) on the source database servers allow incoming connections from the target database servers and vice versa.

   Alternate SQL connectivity should be made available if a firewall blocks incoming remote connection using the SCAN listener port.
7. To preserve the source database Recovery Time Objective (RTO) and Recovery Point Objective (RPO) during the migration, the existing RMAN backup strategy should be maintained.

   During the migration a dual backup strategy will be in place; the existing backup strategy and the strategy used by Zero Downtime Migration. Avoid having two RMAN backup jobs running simultaneously (the existing one and the one initiated by Zero Downtime Migration). If archive logs were to be deleted on the source database, and these archive logs are needed by Zero Downtime Migration to instantiate the target cloud database, then these files should be restored so that Zero Downtime Migration can continue the migration process.

See Also:

Setting Up the Transparent Data Encryption Wallet
Zero Downtime Migration Port Requirements

Target Database Prerequisites

The following prerequisites must be met on the target database before you begin the Zero Downtime Migration process.

1. A placeholder target database must be created before database migration begins.

   The placeholder target database is overwritten during migration, but it retains the overall configuration.
   Pay careful attention to the following requirements:

   - **Size for the future** - When you create the database from the console, ensure that your chosen shape can accommodate the source database, plus any future sizing requirements. A good guideline is to use a shape similar to or larger in size than source database.
   - **Set name paramters** - The target database `db_name` should be the same as the source database `db_name`, and the target database `db_unique_name` parameter value must be unique to ensure that Oracle Data Guard can identify the target as a different database from the source database.
   - **Disable automatic backups** - Provision the target database from the console without enabling automatic backups.
     For Oracle Cloud Infrastructure and Exadata Cloud Service, do not select the **Enable automatic backups** option under the section **Configure database backups**.

     For Exadata Cloud at Customer, set Backup destination **Type** to `None` under the section **Configure Backups**.

2. The target database version should be the same as the source database version. The target database patch level should also be the same as (or higher than) the source database.

   If the target database environment is at a higher patch level than the source database (for example, if the source database is at Oct 2018 PSU/BP and the target database is at Jan 2019 PSU/BP), then you must run datapatch after database migration.

3. Transparent Data Encryption (TDE) should be enabled and ensure that the wallet `STATUS` is `OPEN` and `WALLET_TYPE` is `AUTOLOGIN` (for an `AUTOLOGIN` wallet type), or `WALLET_TYPE` is `PASSWORD` (for a `PASSWORD` based wallet type).

   ```
   SQL> SELECT * FROM v$encryption_wallet;
   ```

4. If the target is an Oracle RAC database, then you must set up SSH connectivity without a passphrase between the Oracle RAC servers for the oracle user.
5. Check the size of the disk groups and usage on the target database (ASM disk groups or ACFS file systems) and make sure adequate storage is provisioned and available on the target database servers.
6. Make sure adequate storage is provisioned and available on the object store to accommodate the source database backup.
7. Verify that ports 22 and 1521 on the target servers in the Oracle Cloud Infrastructure or Exadata Cloud at Customer environment are open and not blocked by a firewall.

8. Capture the output of the RMAN `SHOW ALL` command, so that you can compare RMAN settings after the migration, then reset any changed RMAN configuration settings to ensure that the backup works without any issues.

```
RMAN> show all;
```

> **See Also:**
>
> Managing User Credentials for information about generating the auth token for Object Storage backups
>
> Zero Downtime Migration Port Requirements

Preparing for Migration to Oracle Cloud Infrastructure

Complete the following preparation before migrating data to an Oracle Cloud Infrastructure virtual machine or bare metal target.

1. Prepare the response file template.

   Get the response file template, which is used to create your Zero Downtime Migration response file for the database migration procedure, from location /u01/app/zdmhome/rhp/zdm/template/zdm_template.rsp, and update the file as follows.

   - Set `TGT_DB_UNIQUE_NAME` to the target database `db_unique_name` value.
   - Set `PLATFORM_TYPE` to `VMDB`.
   - Set `MIGRATION_METHOD` to `DG_OSS`, where DG stands for Data Guard and OSS stands for for Object Storage service.
   - If an SSH proxy is required to access the source database server from the Zero Downtime Migration service host, set `SRC_HTTP_PROXY_URL` and `SRC_HTTP_PROXY_PORT`.
   - If an SSH proxy is required to access the target database server from the Zero Downtime Migration service host, set `TGT_HTTP_PROXY_URL` and `TGT_HTTP_PROXY_PORT`.
   - If SSH tunneling is set up, set the `TGT_SSH_TUNNEL_PORT` parameter.
   - Specify the target database data files storage (ASM or ACFS) properties as appropriate for (`TGT_DATADG`, `TGT_REDODG`, and `TGT_RECODG`) or (`TGT_DATAACFS`, `TGT_REDOACFS`, and `TGT_RECOACFS`).
   - Set `SKIP_FALLBACK=TRUE` if you do not want to ship redo logs from the target to the source standby either voluntarily or because there is no connectivity between the target and source.
   - Set `SHUTDOWN_SRC=TRUE`, post migration, if you wish to shut down the source database.

2. Set up Object Storage service access.

   To access the Oracle Cloud account, set the following parameters in the input file.

- Set the cloud storage REST endpoint URL value to `HOST`. The value is typically in the format https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/acme, for Oracle Cloud Infrastructure storage, or https://acme.storage.oraclecloud.com/v1/Storage-acme, for Oracle Cloud Infrastructure Classic storage.
- Set the Object Storage bucket `OPC_CONTAINER` parameter. The bucket is also referred to as a container for Oracle Cloud Infrastructure Classic storage. Make sure that the Object Storage bucket is created using the Oracle Cloud Service Console as appropriate. Make sure adequate storage is provisioned and available on the object store to accommodate the source database backup.
- If a proxy is required to access the object store from the source database server, set `SRC_OSS_PROXY_HOST` and `SRC_OSS_PROXY_PORT`.
- If a proxy is required to access the object store from the target database server, set `TGT_OSS_PROXY_HOST` and `TGT_OSS_PROXY_PORT`.

Preparing for Migration to Exadata Cloud Service

Complete the following preparation before migrating data to an Exadata Cloud Service target.

1. Prepare the response file template.

   Get the response file template, which is used to create your Zero Downtime Migration response file for the database migration procedure, from location /u01/app/zdmhome/rhp/zdm/template/zdm_template.rsp, and update the file as follows.

   - Set `TGT_DB_UNIQUE_NAME` to the target database `db_unique_name` value.
   - Set `PLATFORM_TYPE` to `EXACS`.
   - Set `MIGRATION_METHOD` to `DG_OSS`, where DG stands for Data Guard and OSS stands for for Object Storage service.
   - If an SSH proxy is required to access the source database server from the Zero Downtime Migration service host, set `SRC_HTTP_PROXY_URL` and `SRC_HTTP_PROXY_PORT`.
   - If an SSH proxy is required to access the target database server from the Zero Downtime Migration service host, set `TGT_HTTP_PROXY_URL` and `TGT_HTTP_PROXY_PORT`.
   - If SSH tunneling is set up, set the `TGT_SSH_TUNNEL_PORT` parameter.
   - Specify the target database data files storage (ASM or ACFS) properties as appropriate for (`TGT_DATADG`, `TGT_REDODG`, and `TGT_RECODG`) or (`TGT_DATAACFS`, `TGT_REDOACFS`, and `TGT_RECOACFS`).
   - Set `SKIP_FALLBACK=TRUE` if you do not want to ship redo logs from the target to the source standby, either voluntarily or because there is no connectivity between the target and the source.
   - Set `SHUTDOWN_SRC=TRUE`, post migration, if you want to shut down the source database.

2. Set up Object Storage service access.

   To access the Oracle Cloud account, set the following parameters in the input file.

   - Set the cloud storage REST endpoint URL value to `HOST`. The value is typically in the format https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/acme, for Oracle

Cloud Infrastructure storage, or https://acme.storage.oraclecloud.com/v1/Storage-acme, for Oracle Cloud Infrastructure Classic storage.

- Set the Object Storage bucket `OPC_CONTAINER` parameter. The bucket is also referred to as a container for Oracle Cloud Infrastructure Classic storage. Make sure that the Object Storage bucket is created using the Oracle Cloud Service Console as appropriate. Make sure adequate storage is provisioned and available on the object store to accommodate the source database backup.
- If a proxy is required to access the object store from the source database server, set `SRC_OSS_PROXY_HOST` and `SRC_OSS_PROXY_PORT`.
- If a proxy is required to access the object store from the target database server, set `TGT_OSS_PROXY_HOST` and `TGT_OSS_PROXY_PORT`.

Preparing for Migration to Exadata Cloud at Customer

Complete the following preparation before migrating data to an Exadata Cloud at Customer target.

1. Provision the target database.

   Configure a new placeholder database in your Exadata Cloud at Customer environment with same `db_name` as the on-premises database `db_name`.

2. Prepare the ZDMCLI input response file template.

   Get the response file template, which is used to create your Zero Downtime Migration response file for the database migration procedure, from location $ZDM_HOME/rhp/zdm/template/zdm_template.rsp, and update the file based on your backup medium as detailed in the topics that follow.

- Preparing the Template for Exadata Cloud at Customer with Zero Data Loss Recovery Appliance Backup
  When using Zero Data Loss Recovery Appliance as the backup medium for Zero Downtime Migration, set the parameters in the response file as described here.
- Preparing a Template for Exadata Cloud at Customer with Object Storage Backup
  When using Oracle Cloud Infrastructure Object Storage service as the backup medium for your Zero Downtime Migration, set the parameters in the response file as described here.
- Preparing a Template for Exadata Cloud at Customer with NFS Backup
  When using NFS storage as the backup medium for your Zero Downtime Migration, set the parameters in the response file as described here.

Preparing the Template for Exadata Cloud at Customer with Zero Data Loss Recovery Appliance Backup

When using Zero Data Loss Recovery Appliance as the backup medium for Zero Downtime Migration, set the parameters in the response file as described here.

- Set `TGT_DB_UNIQUE_NAME` to the target database `db_unique_name` value.
- Set `PLATFORM_TYPE` to `ExaCC`.
- Set `MIGRATION_METHOD` to `DG_ZDLRA`, where DG stands for Data Guard and ZDLRA for Zero Data Loss Recovery Appliance.

- Set the following Zero Data Loss Recovery Appliance parameters to use a backup residing in Zero Data Loss Recovery Appliance.
  - Set `SRC_ZDLRA_WALLET_LOC` for the wallet location, for example,

    ```
    SRC_ZDLRA_WALLET_LOC=/u02/app/oracle/product/12.1.0/dbhome_3/dbs/zdlra
    ```

  - Set `TGT_ZDLRA_WALLET_LOC`
  - Set `ZDLRA_CRED_ALIAS` for the wallet credential alias, for example,

    ```
    ZDLRA_CRED_ALIAS=zdlra_scan:listener_port/zdlra9:dedicated
    ```

- Specify the target database data files storage (ASM or ACFS) properties as appropriate. For ASM set `TGT_DATADG`, `TGT_REDODG`, and `TGT_RECODG`. For ACFS set `TGT_DATAACFS`, `TGT_REDOACFS`, and `TGT_RECOACFS`.
- Set `SKIP_FALLBACK=TRUE` if you do not want to ship redo logs from the target to the source standby, either voluntarily or because there is no connectivity between the target and the source.
- Set `SHUTDOWN_SRC=TRUE`, post migration, if you want to shut down the source database.

Preparing a Template for Exadata Cloud at Customer with Object Storage Backup

When using Oracle Cloud Infrastructure Object Storage service as the backup medium for your Zero Downtime Migration, set the parameters in the response file as described here.

- Set `TGT_DB_UNIQUE_NAME` to the target database `db_unique_name` value.
- Set `PLATFORM_TYPE` to `ExaCC`.
- Set `MIGRATION_METHOD` to `DG_OSS`, where DG stands for Data Guard and OSS for the Object Storage service.
- Specify the Oracle Cloud Infrastructure Object Storage service access and container details.
  The source database is backed up to the specified container and restored to Exadata Cloud at Customer using RMAN SQL*Net connectivity.
- Specify the target database data files storage (ASM or ACFS) properties as appropriate. For ASM set `TGT_DATADG`, `TGT_REDODG`, and `TGT_RECODG`. For ACFS set `TGT_DATAACFS`, `TGT_REDOACFS`, and `TGT_RECOACFS`.
- Set `SKIP_FALLBACK=TRUE` if you do not want to ship redo logs from the target to the source standby, either voluntarily or because there is no connectivity between the target and the source.
- Set `SHUTDOWN_SRC=TRUE`, post migration, if you want to shut down the source database.

Preparing a Template for Exadata Cloud at Customer with NFS Backup

When using NFS storage as the backup medium for your Zero Downtime Migration, set the parameters in the response file as described here.

- Set `TGT_DB_UNIQUE_NAME` to the target database `db_unique_name` value.
- Set `PLATFORM_TYPE` to `ExaCC`.

- Set MIGRATION_METHOD to DG_SHAREDPATH or DG_EXTBACKUP, where DG stands for Data Guard.
  Use DG_STORAGEPATH when a new backup needs to be taken and placed on an external storage mount (for eaxmple, an NFS mount point).

  Use DG_EXTBACKUP when using an existing backup, already placed on an external shared mount (for example, NFS storage).

  Note that if MIGRATION_METHOD is set to DG_EXTBACKUP then Zero Downtime Migration does not perform a new backup.
- Set BACKUP_PATH to specify the actual NFS path which is made accessible from both the source and target database servers, for example, an NFS mount point. The NFS mount path should be same for both source and target database servers. This path does not need to be mounted on the Zero Downtime Migration service host.
  Note the following considerations:
  - The source database is backed up to the specified path and restored to Exadata Cloud at Customer using RMAN SQL*Net connectivity.
  - The path set in BACKUP_PATH should have 'rwx' permissions for the source database user, and at least read permissions for the target database user.
  - In the path specified by BACKUP_PATH, the Zero Downtime Migration backup procedure will create a directory, $BACKUP_PATH/*dbname*, and place the backup pieces in this directory.
- If you use DG_EXTBACKUP as the MIGRATION_METHOD, then you should create a standby control file backup in the specified path and provide read permissions to the backup pieces for the target database user. For example,

  ```
  RMAN> BACKUP CURRENT CONTROLFILE FOR STANDBY FORMAT '< BACKUP_PATH
  >/lower_case_dbname/standby_ctl_%U';
  ```

  Where standby_ctl_%U is a system-generated unique file name.
- Specify the target database data files storage (ASM or ACFS) properties as appropriate. For ASM set TGT_DATADG, TGT_REDODG, and TGT_RECODG. For ACFS set TGT_DATAACFS, TGT_REDOACFS, and TGT_RECOACFS.
- Set SKIP_FALLBACK=TRUE if you do not want to ship redo logs from the target to the source standby, either voluntarily or because there is no connectivity between the target and the source.
- Set SHUTDOWN_SRC=TRUE, post migration, if you want to shut down the source database.

Preparing for Offline Migration (Backup and Recovery)

Complete the following preparations before migrating a database to an Oracle Cloud Infrastructure, Exadata Cloud at Customer, or Exadata Cloud Service target environment.

1. Prepare the response file template.

   Get the response file template, which is used to create your Zero Downtime Migration response file for the database migration procedure, from

location /u01/app/zdmhome/rhp/zdm/template/zdm_template.rsp, and update the file as follows.

- Set `TGT_DB_UNIQUE_NAME` to the target database `db_unique_name` value.
- Set `PLATFORM_TYPE` to the appropriate value, depending on your target environment.
  - For Oracle Cloud Infrastructure, set `PLATFORM_TYPE=VMDB`.
  - For Exadata Cloud at Customer, set `PLATFORM_TYPE=EXACC`.
  - For Exadata Cloud Service, set `PLATFORM_TYPE=EXACS`.
- Set `MIGRATION_METHOD` to `BACKUP_RESTORE_OSS`, where OSS stands for Object Storage service.
- Specify the target database data files storage (ASM or ACFS) properties as appropriate. For ASM, set `TGT_DATADG`, `TGT_REDODG`, and `TGT_RECODG`. For ACFS set `TGT_DATAACFS`, `TGT_REDOACFS`, and `TGT_RECOACFS`.
- If an SSH proxy is required to access the source database server from the Zero Downtime Migration service host, set `SRC_HTTP_PROXY_URL` and `SRC_HTTP_PROXY_PORT`.
- If an SSH proxy is required to access the target database server from the Zero Downtime Migration service host, set `TGT_HTTP_PROXY_URL` and `TGT_HTTP_PROXY_PORT`.

2. Set up Object Storage service access.

   To access the Oracle Cloud account, set the following parameters in the input file.

   - Set the cloud storage REST endpoint URL value to `HOST`. The value is typically in the format https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/acme, for Oracle Cloud Infrastructure storage, or https://acme.storage.oraclecloud.com/v1/Storage-acme, for Oracle Cloud Infrastructure Classic storage.
   - Set the Object Storage bucket `OPC_CONTAINER` parameter. The bucket is also referred to as a container for Oracle Cloud Infrastructure Classic storage. Make sure that the Object Storage bucket is created using the Oracle Cloud Service Console as appropriate. Make sure adequate storage is provisioned and available on the object store to accommodate the source database backup.
   - If a proxy is required to access the object store from the source database server, set `SRC_OSS_PROXY_HOST` and `SRC_OSS_PROXY_PORT`.
   - If a proxy is required to access the object store from the target database server, set `TGT_OSS_PROXY_HOST` and `TGT_OSS_PROXY_PORT`.

Preparing for Automatic Application Switchover

To minimize or eliminate service interruptions on the application after you complete the database migration and switchover, prepare your application to automatically switch over connections from the source database to the target database.

In the following example connect string, the application connects to the source database, and when it is not available the connection is switched over to the target database.

```
(DESCRIPTION=
    (FAILOVER=on)(LOAD_BALANCE=on)(CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
    (ADDRESS_LIST=
```

```
        (ADDRESS=(PROTOCOL=TCP)(HOST=source_database_scan)(PORT=1521))
        (ADDRESS=(PROTOCOL=TCP)(HOST=target_database_scan)(PORT=1521)))
    (CONNECT_DATA=(SERVICE_NAME=zdm_prod_svc)))
```

On the source database, create the service, named zdm_prod_svc in the examples.

```
srvctl add service -db clever -service zdm_prod_svc -role PRIMARY
 -notification TRUE -session_state dynamic -failovertype transaction
 -failovermethod basic -commit_outcome TRUE -failoverretry 30 -
failoverdelay 10
 -replay_init_time 900 -clbgoal SHORT -rlbgoal SERVICE_TIME -preferred
clever1,clever2
 -retention 3600 -verbose
```

> **See Also:**
>
> Oracle MAA white papers about client failover best practices on the Oracle Data Guard page at https://www.oracle.com/goto/maa
> High Availability in *Oracle Database Development Guide*

Customizing a Migration Job

You can customize the Zero Downtime Migration workflow by registering action scripts or plug-ins as pre-actions or post-actions to be performed as part of the operational phases involved in your migration job.

The following topics describe how to customize a migration job.

- Registering Action Plug-ins
  Custom plug-ins must be registered to the Zero Downtime Migration service host to be plugged in as customizations for a particular operational phase.
- Creating an Action Template
  After the useraction plug-ins are registered, you create an action template that combines a set of action plug-ins which can be associated with a migration job.
- Updating Action Plug-ins
  You can update action plug-ins registered with the Zero Downtime Migration service host.
- Associating an Action Template with a Migration Job
  When you run a migration job you can specify the image type that specifies the plug-ins to be run as part of your migration job.

Registering Action Plug-ins

Custom plug-ins must be registered to the Zero Downtime Migration service host to be plugged in as customizations for a particular operational phase.

Determine the operational phase the given plug-in has to be associated with, and run the ZDMCLI command `add useraction`, specifying `-optype MIGRATE_DATABASE` and

the respective phase of the operation, whether the plug-in is run `-pre` or `-post` relative to that phase, and any on-error requirements. You can register custom plug-ins for operational phases after ZDM_SETUP_TGT in the migration job workflow.

What happens at runtime if the useraction encounters an error can be specified with the `-onerror` option, which you can set to either `ABORT`, to end the process, or `CONTINUE`, to continue the migration job even if the custom plug-in exits with an error. See the example command usage below.

Use the Zero Downtime Migration software installed user (for example, zmduser) to add useractions to a database migration job. Adding useractions zdmvaltgt and zdmvalsrc with the `add useraction` command would look like the following.

```
zdmuser> ./zdmcli add useraction -useraction zdmvaltgt -optype
MIGRATE_DATABASE
-phase ZDM_VALIDATE_TGT -pre -onerror ABORT -actionscript /home/useract.sh

zdmuser> ./zdmcli add useraction -useraction zdmvalsrc -optype
MIGRATE_DATABASE
-phase ZDM_VALIDATE_SRC -pre -onerror CONTINUE -actionscript /home/
useract1.sh
```

In the above command, the scripts /home/useract.sh and /home/useract1.sh are copied to the Zero Downtime Migration service host repository, and they are run if they are associated with any migration job run using an action template.

Creating an Action Template

After the useraction plug-ins are registered, you create an action template that combines a set of action plug-ins which can be associated with a migration job.

An action template is created using the ZDMCLI command `add imagetype`, where the image type, `imagetype`, is a bundle of all of the useractions required for a specific type of database migration. Create an image type that associates all of the useraction plug-ins needed for the migration of the database. Once created, the image type can be reused for all migration operations for which the same set of plug-ins are needed.

The base type for the image type created here must be `CUSTOM_PLUGIN`, as shown in the example below.

For example, you can create an image type `ACTION_ZDM` that bundles both of the useractions created in the previous example, zdmvalsrc and zdmvaltgt.

```
zdmuser>./zdmcli add imagetype -imagetype ACTION_ZDM -basetype
CUSTOM_PLUGIN -useractions zdmvalsrc,zdmvaltgt
```

Updating Action Plug-ins

You can update action plug-ins registered with the Zero Downtime Migration service host.

The following example shows you how to modify the useraction zdmvalsrc to be a `-post` action, instead of a `-pre` action.

```
zdmuser>./zdmcli modify useraction -useraction zdmvalsrc -phase
ZDM_VALIDATE_SRC
 -optype MIGRATE_DATABASE -post
```

This change is propagated to all of the associated action templates, so you do not need to update the action templates.

Associating an Action Template with a Migration Job

When you run a migration job you can specify the image type that specifies the plug-ins to be run as part of your migration job.

As an example, run the migration command specifying the action template ACTION_ZDM created in previous examples, `-imagetype ACTION_ZDM`, including the image type results in running the useract.sh and useract1.sh scripts as part of the migration job workflow.

By default, the action plug-ins are run for the specified operational phase on all nodes of the cluster. If the access credential specified in the migration command option `-tgtarg2` is unique for a specified target node, then an additional auth argument should be included to specify the auth credentials required to access the other cluster nodes. For example, specify `-tgtarg2 nataddrfile:`*`auth_file_with_node_and_identity_file_mapping`*.

A typical nataddrfile for a 2 node cluster with node1 and node2 is shown here.

```
node1:node1:identity_file_path_available_on_zdmservice_node
node2:node2:identity_file_path_available_on_zdmservice_node
```

4

Migrating Your Database with Zero Downtime Migration

Evaluate the database migration job, run the job, and perform other operations during and after a database migration.

See the Zero Downtime Migration Release Notes for the latest information about known issues, My Oracle Support notes, and runbooks.

- Migrate the Database
  Perform the database migration with Zero Downtime Migration using the following procedure.
- Query Migration Job Status
  You can query the migration job status while the job is running.
- List Migration Job Phases
  You can list the operation phases involved in the migration job.
- Pause and Resume a Migration Job
  You can pause a migration job at any point after the ZDM_SETUP_TGT phase, and resume the job at any time.
- Rerun a Migration Job
  If there are any unexpected errors in the migration workflow, you can correct them and rerun the migration job.
- Terminiate a Running Migration Job
  If you want to resubmit a database migration job for a specified database, you must first terminiate the running migration job.
- Post-Migration Tasks
  The following topics describe tasks that you do after you complete the database migration job.

Migrate the Database

Perform the database migration with Zero Downtime Migration using the following procedure.

Ensure that you have met all of the prerequisites and completed the required preparations described in Preparing for Database Migration before you begin the migration procedures in this topic.

1. Obtain the necessary access credentials required.

   If Oracle Cloud Infrastructure Object Storage is used as the backup medium, obtain the Object Storage access credential. The user ID for the Oracle Cloud Infrastructure Console user and an auth token for Object Storage is required. If you are not using an existing auth token, a new auth token can be generated using the Oracle Cloud Infrastructure Console.
   If the source database server is accessed with the root user, then you need the root user password. If the source database server is accessed with the private key file, for

example, the Oracle Cloud Infrastructure Classic server, then you need the private key file. The SYS password for the source database environment is also required.

If Zero Data Loss Recovery Appliance is used as the backup medium, get the Zero Data Loss Recovery Appliance virtual private catalog (VPC) user credentials.

2. Prepare the Zero Downtime Migration response file.

   The database migration is driven by a response file that captures the essential parameters for accomplishing the task. Use the sample /u01/app/zdmhome/rhp/zdm/template/zdm_template.rsp file for example entries needed to set up the response file for your particular source, target, and backup environments.

3. Evaluate the database migration process.

   Before submitting the database migration job for the production database, perform a test migration to determine how the process may fare with your configuration and settings. To run an evaluation of the migration process, run the ZDMCLI command `migrate database` with the `-eval` option, as shown in the following example.
   Log in to the Zero Downtime Migration service host and switch to the `zdmuser` installed user, then change to the Zero Downtime Migration home bin directory.

   ```
   su - zdmuser
   zdmuser> cd /u01/app/zdmhome/bin
   ```

   For example,

   ```
   zdmuser> cd /u01/app/oracle/zdm/grid/bin
   ```

   If connectivity to the source database server is done through root credentials then the command would be the following:

   ```
   zdmuser> ./zdmcli migrate database -sourcedb
   source_db_unique_name_value
   -sourcenode source_database_server_name -srcroot
   -targetnode target_database_server_name
   -targethome target_database_ORACLE_HOME
   -backupuser Object_store_login_user_name
   -rsp response_file_location
   -tgtauth zdmauth
   -tgtarg1 user:target_database_server_login_user_name
   -tgtarg2 identity_file:ZDM_installed_user_private_key_file_location
   -tgtarg3 sudo_location:/usr/bin/sudo -eval
   ```

   For the prompts, specify the source database SYS password and the source database server root user password. If the backup destination is Object Store (Bucket), then specify user swift authentication token. If the backup destination is Storage Classic (Container) then specify your tenancy login password.

   For example,

   ```
   zdmuser> ./zdmcli migrate database -sourcedb zdmsdb -sourcenode
   srcnode
   -srcroot -targetnode rptest -targethome /u01/app/oracle/product/
   ```

```
12.1.0.2/dbhome_1
-backupuser backup_user@example.com -rsp /scratch/zdm/
zdm_template_zdmsdb.rsp
-tgtauth zdmauth -tgtarg1 user:opc -tgtarg2
identity_file:/home/oracle/.ssh/zdm_service_node.ppk -tgtarg3
sudo_location:/usr/bin/sudo -eval

Enter source database zdmsdb SYS password:
Enter source user "root" password:
Enter user "backup_user@example.com" password:
```

If connectivity to the source database server is through SSH key, then the command would be:

```
zdmuser> ./zdmcli migrate database -sourcedb source_db_unique_name_value
-sourcenode source_database_server_name -srcauth zdmauth
-srcarg1 user:source_database_server_login_user_name
-srcarg2 identity_file:ZDM_installed_user_private_key_file_location
-srcarg3 sudo_location:/usr/bin/sudo -targetnode
target_database_server_name
-targethome target_database_ORACLE_HOME_value
-backupuser Object_store_login_user_name -rsp response_file_location
-tgtauth zdmauth -tgtarg1 user:target_database_server_login_user_name
-tgtarg2 identity_file:ZDM_installed_user_private_key_file_location
-tgtarg3 sudo_location:/usr/bin/sudo -eval
```

For the prompts, specify the source database SYS password. If the backup destination is Object Store (Bucket), then specify user swift authentication token. If the backup destination is Storage Classic (Container), then specify your tenancy login password.

```
zdmuser> ./zdmcli migrate database -sourcedb zdmsdb -sourcenode ocic31 -
srcauth zdmauth
-srcarg1 user:opc -srcarg2 identity_file:/home/oracle/.ssh/
zdm_service_node.ppk
-srcarg3 sudo_location:/usr/bin/sudo -targetnode rptest -targethome
/u01/app/oracle/product/12.2.0.1/dbhome_1 -backupuser
backup_user@example.com
-rsp /u01/zdm_template_zdmsdb.rsp -tgtauth zdmauth -tgtarg1 user:opc -
tgtarg2
identity_file:/home/oracle/.ssh/zdm_service_node.ppk  -tgtarg3
sudo_location:/usr/bin/sudo -eval

Enter source database zdmsdb SYS password:
Enter user "backup_user@example.com" password:
```

Note that if a source single instance database is deployed without a Grid Infrastructure home, then in the above command use `-sourcesid` in place of `-sourcedb`.

Also, if a source database is configured for a `PASSWORD` based wallet, then add the `-tdekeystorepasswd` option to the command above, and for the prompt, specify the source database TDE keystore password value.

It is highly recommended that for each migration you run `migrate database` in evaluation mode first. The evaluation allows you to correct any potential problems in the setup and

configuration before performing the actual migration on a production database. In evaluation mode, the migration process runs without effecting the changes. It is safe to run the command with the `-eval` option as many times as needed before running the actual migration job.

4. Determine if the migration process needs to be paused and resumed before you start the database migration. Once the migration job is initiated the job system runs the job as configured.

   If the migration job needs to pause and resume at a particular point, then see the topics List Migration Job Phases and Pause and Resume Migration Job (cross references below) for more details.

5. Start the database migration process.

   The database migration job is submitted from the Zero Downtime Migration service host by the `zdmuser` user using the ZDMCLI command `migrate database`.
   If connectivity to the source database server is through root credentials, then the command would be:

```
zdmuser> ./zdmcli migrate database -sourcedb
source_db_unique_name_value
-sourcenode source_database_server_name -srcroot
-targetnode target_database_server_name
-targethome target_database_ORACLE_HOME_value
-backupuser Object_store_login_user_name
-rsp response_file_location -tgtauth zdmauth
-tgtarg1 user:target_database_server_login_user_name
-tgtarg2
identity_file:ZDM_installed_user_private_key_file_location
-tgtarg3 sudo_location:/usr/bin/sudo
```

   For the prompts, specify the source database SYS password and source database server root user password. If the backup destination is Object Store (Bucket), then specify user swift authentication token. If the backup destination is Storage Classic (Container), then specify your tenancy login password.

   For example:

```
zdmuser> ./zdmcli migrate database -sourcedb zdmsdb -sourcenode
srcnode -srcroot
-targetnode rptest -targethome /u01/app/oracle/product/12.1.0.2/
dbhome_1
-backupuser backup_user@example.com -rsp /scratch/zdm/
zdm_template_zdmsdb.rsp
-tgtauth zdmauth -tgtarg1 user:opc
-tgtarg2 identity_file:/home/oracle/.ssh/zdm_service_node.ppk
-tgtarg3 sudo_location:/usr/bin/sudo

Enter source database zdmsdb SYS password:
Enter source user "root" password:
Enter user "backup_user@example.com" password:
```

If connectivity to the source database server is through SSH key, then the command would be:

```
zdmuser> ./zdmcli migrate database -sourcedb source_db_unique_name_value
-sourcenode source_database_server_name -srcauth zdmauth
-srcarg1 user:source_database_server_login_user_name
-srcarg2 identity_file:ZDM_installed_user_private_key_file_location
-srcarg3 sudo_location:/usr/bin/sudo -targetnode
target_database_server_name
-targethome target_database_ORACLE_HOME_value
-backupuser Object_store_login_user_name -rsp response_file_location
-tgtauth zdmauth -tgtarg1 user:target_database_server_login_user_name
-tgtarg2 identity_file:ZDM_installed_user_private_key_file_location
-tgtarg3 sudo_location:/usr/bin/sudo
```

For the prompts, specify the source database SYS password. If the backup destination is Object Store (Bucket), then specify user swift authentication token. If the backup destination is Storage Classic (Container), then specify your tenancy login password.

For example,

```
zdmuser> ./zdmcli migrate database -sourcedb zdmsdb -sourcenode ocic31 -
srcauth zdmauth
-srcarg1 user:opc -srcarg2 identity_file:/home/oracle/.ssh/
zdm_service_node.ppk
-srcarg3 sudo_location:/usr/bin/sudo -targetnode rptest
-targethome /u01/app/oracle/product/12.2.0.1/dbhome_1
-backupuser backup_user@example.com -rsp /u01/zdm_template_zdmsdb.rsp
-tgtauth zdmauth -tgtarg1 user:opc
-tgtarg2 identity_file:/home/oracle/.ssh/zdm_service_node.ppk
-tgtarg3 sudo_location:/usr/bin/sudo

Enter source database zdmsdb SYS password:
Enter user "backup_user@example.com" password:
```

If a source single instance is deployed without a Grid Infrastructure home, then in the command above use `-sourcesid` in place of `-sourcedb`.

If the source database is configured for a `PASSWORD` based wallet, then add the `-tdekeystorepasswd` option to the command above, and for the prompt, specify the source database TDE keystore password value.

Note that the `-backupuser` argument takes the Object Storage access user or Zero Data Loss Recovery Appliance VPC user and is skipped if NFS is the backup medium. For NFS, the source database user should have ‘rwx’ access to the NFS path provided. If the backup destination is Object Store (Bucket), then use the user swift authentication token. If the backup destination is Storage Classic (Container), then use your tenancy login password.

The migration command checks for patch compatibility between the source and target home patch level, and expects the target home patch level to be equal to or higher than the source. If the target home patch level is not as expected, then the migration job is stopped and missing patches are reported. You can either patch the target home with the necessary patches or you can force continue the migration by appending the `-ignore PATCH_CHECK` or `-ignore ALL` option to the migration command.

The command result output indicates the job ID for the migration job, which you can use to query the status of the job.

> **See Also:**
>
> List Migration Job Phases and Pause and Resume a Migration Job

Query Migration Job Status

You can query the migration job status while the job is running.

Query the status of a database migration job using the ZDMCLI `query job` command, specifying the job ID. The job ID is shown in the command output when the database migration job is submitted.

```
zdmuser> ./zdmcli query job -jobid job-id
```

You can find the console output of the migration job in the file indicated in the `query job` command output. You can see migration progress messages in the specified file.

List Migration Job Phases

You can list the operation phases involved in the migration job.

To list the operation phases involved in the migration job, add the `-listphases` option in the ZDMCLI `migrate` command. This option will list the phases involved in the operation.

For example,

```
zdmuser> ./zdmcli migrate database -sourcedb zdmsdb -sourcenode ocic31
-srcauth zdmauth
-srcarg1 user:opc -srcarg2 identity_file:/home/oracle/.ssh/
zdm_service_node.ppk -srcarg3 sudo_location:/usr/bin/sudo -targetnode
rptest
-targethome /u01/app/oracle/product/12.2.0.1/dbhome_1
-backupuser backup_user@example.com -rsp /u01/zdm_template_zdmsdb.rsp
-tgtauth zdmauth -tgtarg1 user:opc
-tgtarg2 identity_file:/home/oracle/.ssh/zdm_service_node.ppk
-tgtarg3 sudo_location:/usr/bin/sudo -listphases
```

Pause and Resume a Migration Job

You can pause a migration job at any point after the ZDM_SETUP_TGT phase, and resume the job at any time.

To pause a migration job, specify the `-pauseafter` option in the ZDMCLI `migrate` command with a valid phase to be paused after. Choose a valid phase that is listed in the `-listphases` command output.

Pausing phases after ZDM_SETUP_TGT is recommended.

For example, if you specify `-pauseafter ZDM_SWITCHOVER_SRC`, the migration job will pause after completing the ZDM_SWITCHOVER_SRC phase.

```
zdmuser> ./zdmcli migrate database -sourcedb zdmsdb -sourcenode ocic31
-srcauth zdmauth -srcarg1 user:opc
-srcarg2 identity_file:/home/oracle/.ssh/zdm_service_node.ppk
-srcarg3 sudo_location:/usr/bin/sudo -targetnode rptest
-targethome /u01/app/oracle/product/12.2.0.1/dbhome_1
-backupuser backup_user@example.com -rsp /u01/zdm_template_zdmsdb.rsp
-tgtauth zdmauth -tgtarg1 user:opc
-tgtarg2 identity_file:/home/oracle/.ssh/zdm_service_node.ppk
-tgtarg3 sudo_location:/usr/bin/sudo -pauseafter ZDM_SWITCHOVER_SRC
```

A paused job can be resumed anytime by running the ZDMCLI `resume job` command, specifying the respective job ID. To schedule another pause, specify the `-pauseafter` option in the `resume` command with a valid phase to be paused after. Choose a valid phase later than phase currently paused at, that is listed in the `-listphases` command output.

```
zdmuser> ./zdmcli resume job -jobid Job_ID [-pauseafter valid-phase]
```

Note that `-pauseafter` allows only one phase to be specified.

Also, note that if you use `-pauseafter` in a migration job at phase `ZDM_CONFIGURE_DG_SRC`, at the end of the execution of the phase, a standby is created at the target database and synchronization occurs between source and target databases.

Rerun a Migration Job

If there are any unexpected errors in the migration workflow, you can correct them and rerun the migration job.

The errors are recorded in the job output, which can be queried using the ZDMCLI `query job` command. Upon resolving the error, the failed job can be continued from the point of failure.

Rerun the migration job by running the ZDMCLI `resume job` command, specifying the job ID of the job to be rerun, as shown here.

```
zdmuser> ./zdmcli resume job -jobid Job_ID
```

Terminiate a Running Migration Job

If you want to resubmit a database migration job for a specified database, you must first terminiate the running migration job.

Zero Downtime Migration blocks attempts to rerun the `MIGRATE DATABASE` command for a specified database if that database is already part of an ongoing migration job.

If you want to resubmit a database migration job for a specified database, you must first terminate the running migration job in either `EXECUTING` or `PAUSED` state using the `ZDMCLI ABORT JOB` command.

```
zdmuser> ./zdmcli abort job -jobid job-id
```

Post-Migration Tasks

The following topics describe tasks that you do after you complete the database migration job.

- Run Datapatch on the Target Database
  If the target database environment is at a higher patch level than the source database, you must run the datapatch utility on the target database.

Run Datapatch on the Target Database

If the target database environment is at a higher patch level than the source database, you must run the datapatch utility on the target database.

For example, if your source database is at Oct 2018 PSU/BP and the target is at Jan 2019 PSU/BP), you must run the datapatch utility. Before running datapatch on the target, ensure you apply the target patch level to the binaries at the source (standby) database.

1. If you are running a multitenant architecture, open the PDBs.

   ```
   SQL> alter pluggable database all open;
   ```

   It is recommended that you run datapatch on all of the PDBs; however, if you only want to open a subset of the PDBs in the CDB, you can use the following command instead. Datapatch only runs on the CDB and opened PDBs.

   ```
   SQL> alter pluggable database PDB_NAME open
   ```

   To run datapatch on a PDB later (previously skipped or newly plugged in), open the database using the `alter pluggable database` command and rerun the datapatch utility.

2. Go to the OPatch directory in ORACLE_HOME and run the datapatch utility.

   ```
   % cd $ORACLE_HOME/OPatch
   % ./datapatch -verbose
   ```

   The datapatch utility runs the necessary apply scripts to load the modified SQL files into the database. An entry is added to the `dba_registry_sqlpatch` view indicating the patch application.

3. Check for errors.

   Error logs are located in the $ORACLE_BASE/cfgtoollogs/sqlpatch/*patch#*/*unique patch ID* directory in the following format:
   *patch#*_apply_*database_SID_CDB_name_timestamp*.log

where *database_SID* is the database SID, *CDB_name* is the name of the multitenant container database, and *timestamp* is in the format YYYYMMMDD_HH_MM_SS.

5

Troubleshooting Zero Downtime Migration

This section describes how to handle migration job failures.

For more information about troubleshooting Zero Downtime Migration and known issues in the current release, see the Zero Downtime Migration Release Notes.

- Handling Migration Job Failures
  If your migration job fails, the following solutions can help you discover the issue.

Handling Migration Job Failures

If your migration job fails, the following solutions can help you discover the issue.

If your migration job encounters an error, refer to the migration job output logs, Zero Downtime Migration service logs, and server-specific operational phase logs present at the respective source or target database servers.

If the migration job encounters an exception (that is, fails) then the logs can provide some indication of the nature of the fault. The logs for the migration procedures executed in the source and target environments are stored on the servers in the respective source and target environments. The Zero Downtime Migration command output location is provided to you when the migration job is run with the ZDMCLI `migrate database` command. You can also find the log file location in the output of the ZDMCLI `query job -jobid` *`job-id`* command.

Determine which operational phase the migration job was in at time of failure, and whether the phase belongs to the source or target. Check the Zero Downtime Migration service host log at $ZDM_BASE/crsdata/<zdm_service_node>/rhp/rhpserver.log.0 and access the respective source or target server to check the log associated with the operational phase in /tmp/zdm-*unique id*/zdm/log.

If the Zero Downtime Migration service does not start, then check the Zero Downtime Migration service logs for process startup errors to determine the cause of the error reported. The Zero Downtime Migration service log can be found at $ZDM_BASE/crsdata/*zdm_service_node*/rhp/rhpserver.log.0.

If a migration job fails, you can fix the cause of failure and re-run the job. Note that when you re-run a migration job, a new job ID is assigned to the job. Make sure you check the new job and logs for monitoring further progress.

A

Zero Downtime Migration Port Requirements

Zero Downtime Migration requires a means to access on-premises and Oracle Cloud hosts, and if SSH is used for host access, then port 22 access is required.

Table A-1 Zero Downtime Migration Communication Ports

Initiator	Target	Protocol	Port	Purpose	Description
Zero Downtime Migration Service Host	On-premises database hosts and Oracle Cloud hosts	TCP	22	SSH	Authentication-based operations to run Zero Downtime Migration operational phases Source and Target servers should accept incoming connections from the Zero Downtime Migration service host.
On-premises database host	Oracle Cloud hosts	TCP	1521	SQL*Net	Should allow Oracle client connections to the database over Oracle's SQL*Net protocol Perform database queries, Data Guard sync, and configuration **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.

Table A-1 (Cont.) Zero Downtime Migration Communication Ports

Initiator	Target	Protocol	Port	Purpose	Description
Oracle Cloud hosts	On-premises database host	TCP	1521	SQL*Net	Should allow Oracle client connections to the database over Oracle's SQL*Net protocol Allows redo log shipping if on-premises database needs to be in sync with the new primary on Oracle Cloud after switchover. If there is no communication possible from Oracle Cloud to on-premises host then set `SKIP_FALLBACK` to `TRUE` in the response file to avoid this communication. **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.
On-premises database host	Oracle Cloud Object Store Service	SSL	443	Database backup store. Create a backup of the on-premises database to the specified Oracle Cloud Object store container.	If the chosen backup method involves Oracle Cloud Object Store Service as the backup medium, then access ports as documented Oracle Cloud Object Store Service applies.

Table A-1 (Cont.) Zero Downtime Migration Communication Ports

Initiator	Target	Protocol	Port	Purpose	Description
Oracle Cloud database hosts	Oracle Cloud Object Store Service	SSL	443	Database backup store. Restore backup of the on-premises database from the specified Oracle Cloud Object store container to the Oracle Cloud hosts.	If the chosen backup method involves Oracle Cloud Object Store Service as the backup medium, then access ports as documented Oracle Cloud Object Store Service applies.

B

Zero Downtime Migration Encryption Requirements

Zero Downtime Migration does not always require encryption at the source (although, all Cloud databases are encrypted by default). The following tables list specific cases when encryption is not required.

Table B-1 On-Premises Unencrypted Primary and Cloud Encrypted Standby

Operation	On-Premises Primary 11g R2	Cloud Standby 11g R2	On-Premises Primary 12c R1	Cloud Standby 12c R1	On-Premises Primary 12c R2	Cloud Standby 12c R2 and later	Notes
Data Guard initial setup for on-premises primary and cloud standby	Unencrypted	Encrypted	Unencrypted	Encrypted	Unencrypted	Encrypted	In these cases the standby database is manually encrypted after instantiation
New tablespace creation on-premises primary	Unencrypted	Unencrypted	Unencrypted	Unencrypted	Unencrypted	Unencrypted	Requires manual TDE conversion for standby database
Redo generated in on-premises primary	Unencrypted	Unencrypted	Unencrypted	Unencrypted	Unencrypted	Unencrypted	
Archived logs	Unencrypted	Unencrypted	Unencrypted	Unencrypted	Unencrypted	Unencrypted	
New and changed blocks	Unencrypted	Encrypted	Unencrypted	Encrypted	Unencrypted	Encrypted	Redo shipped from the on-premises primary to the cloud is not encrypted
Recovery in the cloud standby	N/A	Encrypted	N/A	Encrypted	N/A	Encrypted	Redo shipped from the on-premises primary to the cloud is not encrypted

Table B-2 Cloud Encrypted Primary and On-Premises Unencrypted Standby

Operation	Cloud Primary 11g R2	On-Premises Standby 11g R2	Cloud Primary 12c R1	On-Premises Standby 12c R1	Cloud Primary 12c R2	On-Premises Standby 12c R2 and later	Notes
New tablespace creation in cloud primary	Encrypted	Encrypted	Encrypted	Encrypted	Encrypted	Encrypted	ASO required for on-premises to decrypt
Redo generated in cloud primary	Encrypted	Encrypted	Encrypted	Encrypted	Encrypted	Encrypted	ASO required for on-premises to decrypt
Archived logs	Encrypted	Encrypted	Encrypted	Encrypted	Encrypted	Encrypted	ASO required for on-premises to decrypt
New and changed blocks for existing unencrypted tablespace on standby	Encrypted	Encrypted*	Encrypted	Encrypted*	Encrypted	Unencrypted	ASO is required on-premises to decrypt and encrypt * For 11g R2 and 12c R1 redo apply will encrypt only if redo is encrypted
Recovery in the on-premises standby	N/A	Encrypted	N/A	Encrypted	N/A	Unencrypted data depends on whether the datafile is encrypted	ASO required for on-premises database

C

Zero Downtime Migration Process Phases

The migration job process in Zero Downtime Migration runs in operational phases as a workflow.

Example C-1 Listing Zero Downtime Migration Process Phases

Run the ZDMCLI `migrate database` command with the `listphases` option to list the operational phases for your migration job, as shown here.

```
zdmuser> cd /u01/app/oracle/zdm/grid/bin
zdmuser> ./zdmcli migrate database -sourcedb zdmsdb -sourcenode ocic31 -
srcauth zdmauth
 -srcarg1 user:opc -srcarg2 identity_file:/home/oracle/.ssh/
zdm_service_node.ppk -srcarg3 sudo_location:/usr/bin/sudo
 -targetnode rptest
-targethome /u01/app/oracle/product/12.2.0.1/dbhome_1
-backupuser backup_user@example.com -rsp /u01/zdm_template_zdmsdb.rsp
-tgtauth zdmauth -tgtarg1 user:opc
-tgtarg2 identity_file:/home/oracle/.ssh/zdm_service_node.ppk
-tgtarg3 sudo_location:/usr/bin/sudo -listphases
```

Table C-1 Zero Downtime Migration Process Phase Descriptions

Phase name	Description
ZDM_GET_SRC_INFO	Get information about the source database
ZDM_GET_TGT_INFO	Get information about the target database
ZDM_SETUP_SRC	Set up Zero Downtime Migration helper modules on the source server
ZDM_SETUP_TGT	Set up Zero Downtime Migration helper modules on the target server
ZDM_PREUSERACTIONS	Run migration pre-useractions, if any, at the source
ZDM_PREUSERACTIONS_TGT	Run migration pre-useractions, if any, at the target
ZDM_OBC_INST_SRC	Install Oracle Database Cloud Backup Module at the source
ZDM_OBC_INST_TGT	Install Oracle Database Cloud Backup Module at the target
ZDM_GEN_RMAN_PASSWD	Generate random password for encrypting RMAN backup
ZDM_BACKUP_FULL_SRC	Perform full backup of the source database
ZDM_BACKUP_INCREMENTAL_SRC	Perform incremental backup of the source database
ZDM_VALIDATE_SRC	Perform validations at the source
ZDM_VALIDATE_TGT	Perform validations at the target
ZDM_DISCOVER_SRC	Perform database discovery at the source for setting up Data Guard

Table C-1 (Cont.) Zero Downtime Migration Process Phase Descriptions

Phase name	Description
ZDM_COPYFILES	Copy Oracle password file and TDE wallets from source to target
ZDM_OSS_STANDBY_SETUP_TDE_TGT	Copy TDE wallet files from the source to the target keystore location
ZDM_PREPARE_TGT	Prepare target for Data Guard standby creation
ZDM_CLONE_TGT	Create Data Guard standby from the Cloud backup
ZDM_FINALIZE_TGT	Finalize Data Guard standby preparation of the target
ZDM_CONFIGURE_DG_SRC	Register the Cloud standby with the source
ZDM_SWITCHOVER_SRC	Initiate switchover actions at the source
ZDM_SWITCHOVER_TGT	Complete switchover actions at the target
ZDM_POSTUSERACTIONS	Perform any post-migration useractions at the source
ZDM_POSTUSERACTIONS_TGT	Perform any post-migration useractions at the target
ZDM_CLEANUP_SRC	Perform clean up at the source
ZDM_CLEANUP_TGT	Perform clean up at the target

D

Zero Downtime Migration ZDMCLI Command Reference

The following topics describe the Zero Downtime Migration ZDMCLI command usage and options.

- add imagetype
  Configures a new image type of the specified name and its associated useractions.
- add useraction
  Configures a new user action of the specified name with its associated script and action file.
- migrate database
  Performs a migration of a database to the Oracle Cloud.
- modify useraction
  Modifies the configuration of the useraction with the specified name.
- query job
  Gets the current status of the scheduled job with the specified job ID.

add imagetype

Configures a new image type of the specified name and its associated useractions.

Syntax

```
zdmcli add imagetype -imagetype image_type -basetype
 {SOFTWARE |
         ORACLEGISOFTWARE |
         ORACLEDBSOFTWARE |
         ORACLEGGSOFTWARE |
         LINUXOS |
         CUSTOM_PLUGIN}
 [-useractions user_action_list]
```

Options

Table D-1 ZDMCLI add imagetype Options

Option	Description
-imagetype *image_type*	Name of the imagetype to be created
-basetype {SOFTWARE | ORACLEGISOFTWARE | ORACLEDBSOFTWARE | ORACLEGGSOFTWARE | LINUXOS | CUSTOM_PLUGIN}	The base image type for which the image type is created

Table D-1 (Cont.) ZDMCLI add imagetype Options

Option	Description
-useractions *user_action_list*	Comma-separated list of useraction names

> **Note:**
>
> The `-basetype CUSTOM_PLUGIN` value is not shown in the help output for this command, but it is a valid value.

add useraction

Configures a new user action of the specified name with its associated script and action file.

Syntax

```
zdmcli add useraction -useraction user_action_name -actionscript
script_name
 [-actionfile file_name]
 {-pre |
  -post} -optype
 {IMPORT_IMAGE |
  ADD_WORKINGCOPY |
  DELETE_WORKINGCOPY |
  ADD_DATABASE |
  DELETE_DATABASE |
  ADD_PDB_DATABASE |
  DELETE_PDB_DATABASE |
  MOVE_DATABASE |
  MOVE_GIHOME |
  UPGRADE_DATABASE |
  UPGRADE_GIHOME |
  ADDNODE_GIHOME |
  DELETENODE_GIHOME |
  ADDNODE_DATABASE |
  DELETENODE_DATABASE |
  ADDNODE_WORKINGCOPY |
  ZDTUPGRADE_DATABASE |
  ZDTUPGRADE_DATABASE_SNAPDB |
  ZDTUPGRADE_DATABASE_DBUA |
  ZDTUPGRADE_DATABASE_SWITCHBACK |
  MIGRATE_DATABASE }
 [-phase <operation phase>]
 [-onerror
       {ABORT |
        CONTINUE}]
 [-runscope
       {ONENODE |
        ALLNODES |
```

```
        AUTO |
        FIRSTNODEONRHPS |
        LASTNODEONRHPS |
        ALLNODESONRHPS}]
```

Options

Table D-2 ZDMCLI add useraction Options

Option	Description
-useraction *user_action_name*	Name of the useraction
-actionscript *script_name*	Script file to be run
-actionfile *file_name*	File associated with and needed by the useraction
-pre	Runs the useraction before the operation
-post	Runs the useraction after the operation
-optype {IMPORT_IMAGE | ADD_WORKINGCOPY | DELETE_WORKINGCOPY | ADD_DATABASE | DELETE_DATABASE | ADD_PDB_DATABASE | DELETE_PDB_DATABASE | MOVE_DATABASE | MOVE_GIHOME | UPGRADE_DATABASE | UPGRADE_GIHOME | ADDNODE_GIHOME | DELETENODE_GIHOME | ADDNODE_DATABASE | DELETENODE_DATABASE | ADDNODE_WORKINGCOPY | ZDTUPGRADE_DATABASE | ZDTUPGRADE_DATABASE_SNAPDB | ZDTUPGRADE_DATABASE_DBUA | ZDTUPGRADE_DATABASE_SWITCHBACK | MIGRATE_DATABASE}	Operation for which the useraction is configured
-phase *phase of operation*	Operation phase for which the useraction is configured
-onerror {ABORT | CONTINUE}	The response if the useraction encounters an error during execution
-runscope {ONENODE | ALLNODES | AUTO | FIRSTNODEONRHPS | LASTNODEONRHPS | ALLNODESONRHPS}	The servers on which the useraction is run. Specify *AUTO* to choose the run scope based on the other command options. Specify *FIRSTNODEONRHPS*, *LASTNODEONRHPS*, or *ALLNODESONRHPS* to run the useraction script on RHPS when the operation is performed on the first server, last server, or all servers.

migrate database

Performs a migration of a database to the Oracle Cloud.

Syntax

```
zdmcli migrate database
 {-sourcedb db_name |
  -sourcesid source_oracle_sid} -sourcenode node_name -targetnode node_name -
targethome target_home -rsp zdm_template
```

```
  [-eval]
  [-imagetype]
  [-tdekeystorepasswd]
  [-tdemasterkey]
  [-useractiondata user_action_data] -backupuser user_name
  [{-srcroot |
         -srccred cred_name |
         -srcuser user_name |
             {-srcsudouser sudo_user_name -srcsudopath
sudo_binary_path} |
             {-srcauth plugin_name
             [-srcarg1 name1:value1
             [-srcarg2 name2:value2...]]}}]
  {-tgtroot |
   -tgtcred cred_name |
   -tgtuser user_name |
       {-tgtsudouser sudo_user_name -tgtsudopath sudo_binary_path} |
       {-tgtauth plugin_name
           [-tgtarg1 name1:value1
           [-tgtarg2 name2:value2...]]}}
  [-schedule
       { timer_value |
         NOW }]
  [-pauseafter phase]
  [-listphases]
  [-ignoremissingpatches patch_name
       [,patch_name...]]
  [-ignore
       {ALL|
        WARNING|
        PATCH_CHECK}]
```

Options

Table D-3 ZDMCLI migrate database Options

Option	Description
-sourcedb *db_name*	Name of the source database you want to migrate
-sourcenode *node_name*	Host on which the source database is running
-targetnode *node_name*	Target server to which the source database is migrated
-targethome *target_home*	Location of the target database ORACLE_HOME
-imagetype *image_type*	Name of the useraction imagetype
-useractiondata *user_action_data*	Value to be passed to *useractiondata* parameter of the useraction script
-rsp *zdm_template*	Zero Downtime Migration template
-sourcesid *source_oracle_sid*	ORACLE_SID of the source single instance database without Grid Infrastructure
-eval	Evaluate without executing the command

Table D-3 (Cont.) ZDMCLI migrate database Options

Option	Description
-backupuser *user_name*	Name of the user allowed to backup or restore the database
-srcroot	Directs Zero Downtime Migration to use root credentials to access the source database server
-srccred *cred_name*	Credential name with which to associate the username and password credentials to access the source database server
-srcuser *user_name*	Name of the privileged user performing operations on the source database server
-srcsudouser *username*	Perform super user operations as sudo user name on the source database server
-srcsudopath *sudo_binary_path*	Location of sudo binary on the source database server
-srcauth *plug-in_name* [*plug-in_args*]	Use an authentication plug-in to access the source database server
-tgtroot	Use root credentials to access the target database server
-tgtcred *cred_name*	Credential name with which to associate the username and password credentials to access the target database server
-tgtuser *user_name*	Name of the user performing operations on the target database server
-tgtsudouser *username*	Perform super user operations as sudo user name on the target database server
-tgtsudopath *sudo_binary_path*	Location of sudo binary on the target database server
-tgtauth *plugin_name* [*plugin_args*]	Use an authentication plug-in to access the target database server
-tdekeystorepasswd	TDE keystore password, required for password based keystore/wallet
-tdemasterkey	TDE master encryption key
-schedule *timer_value*	Scheduled time to execute the operation, in ISO-8601 format. For example: 2016-12-21T19:13:17+05
-pauseafter *phase*	Pause the job after running the specified phase
-ignoremissingpatches	Proceed with the move/upgrade even though the specified patches, which are present in the source path or working copy, might be missing from the destination path or working copy
-ignore {ALL|WARNING|PATCH_CHECK}	Ignore all checks or specific type of checks
-listphases	List the phases for this operation

modify useraction

Modifies the configuration of the useraction with the specified name.

Syntax

```
zdmcli modify useraction -useraction user_action_name
 [-actionscript script_name]
 [-actionfile file_name]
 [-pre |
  -post]
 [-optype
        {IMPORT_IMAGE |
         ADD_WORKINGCOPY |
         DELETE_WORKINGCOPY |
         ADD_DATABASE |
         DELETE_DATABASE |
         ADD_PDB_DATABASE |
         DELETE_PDB_DATABASE |
         MOVE_DATABASE |
         MOVE_GIHOME |
         UPGRADE_DATABASE |
         UPGRADE_GIHOME |
         ADDNODE_GIHOME |
         DELETENODE_GIHOME |
         ADDNODE_DATABASE |
         DELETENODE_DATABASE |
         ADDNODE_WORKINGCOPY |
         ZDTUPGRADE_DATABASE |
         ZDTUPGRADE_DATABASE_SNAPDB |
         ZDTUPGRADE_DATABASE_DBUA |
         ZDTUPGRADE_DATABASE_SWITCHBACK |
         MIGRATE_DATABASE } ]
 [-phase phase]
 [-onerror
      {ABORT |
       CONTINUE}]
 [-runscope
      {ONENODE |
       ALLNODES |
       AUTO |
       FIRSTNODEONRHPS |
       LASTNODEONRHPS |
       ALLNODESONRHPS]
```

Options

Table D-4 ZDMCLI modify useraction Options

Option	Description
-useraction *user_action_name*	Name of the useraction
-actionscript *script_name*	Script file to be run
-actionfile *file_name*	Accompanying file needed by the useraction
-pre	Runs the useraction before the operation
-post	Runs the useraction after the operation

Table D-4 (Cont.) ZDMCLI modify useraction Options

Option	Description
-optype {IMPORT_IMAGE | ADD_WORKINGCOPY | DELETE_WORKINGCOPY | ADD_DATABASE | DELETE_DATABASE | ADD_PDB_DATABASE | DELETE_PDB_DATABASE | MOVE_DATABASE | MOVE_GIHOME | UPGRADE_DATABASE | UPGRADE_GIHOME | ADDNODE_GIHOME | DELETENODE_GIHOME | ADDNODE_DATABASE | DELETENODE_DATABASE | ADDNODE_WORKINGCOPY | ZDTUPGRADE_DATABASE | ZDTUPGRADE_DATABASE_SNAPDB | ZDTUPGRADE_DATABASE_DBUA | ZDTUPGRADE_DATABASE_SWITCHBACK | MIGRATE_DATABASE}	Operation for which the useraction is configured
-onerror {ABORT | CONTINUE}	The outcome if the useraction encounters an error during execution
-runscope {ONENODE | ALLNODES | AUTO | FIRSTNODEONRHPS | LASTNODEONRHPS | ALLNODESONRHPS}	The servers where the useraction will be run. Specify *AUTO* to choose the run scope based on the other command options. Specify *FIRSTNODEONRHPS*, *LASTNODEONRHPS* or *ALLNODESONRHPS* to run the useraction script on RHPS when the operaton is performed on the first server, last server, or all servers.

query job

Gets the current status of the scheduled job with the specified job ID.

Syntax

```
zdmcli query job
 [-jobid <job_id>]
 [-status
     {SCHEDULED |
      EXECUTING |
      UNKNOWN |
      TERMINATED |
      FAILED |
      SUCCEEDED |
      PAUSED |
      ABORTED}]
 [-dbname <database_name>]
 [-since <timer_value>]
 [-upto <timer_value>]
 [-brief]
```

Options

Table D-5 ZDMCLI query job Options

Option	Description
-jobid *jobid*	Unique integer job ID value for the scheduled job, which is obtained while scheduling the job
-status {SCHEDULED | EXECUTING | UNKNOWN | TERMINATED | FAILED | SUCCEEDED | PAUSED | ABORTED}	Job status for which to get the jobs
-dbname *unique_db_name*	Name of the database (DB_UNIQUE_NAME)
-since *timer_value*	Date from which to get the jobs, in ISO-8601 format. For example: 2016-12-21T19:13:17+05
-upto *timer_value*	Upper limit time to which to get the jobs, in ISO-8601 format. For example: 2016-12-21T19:13:17+05
-brief	Job details summary only

Index

Q

R

S

T

U

V

Z