

Oracle® Database

Move to Oracle Cloud Using Zero Downtime Migration



Release 19c (19.7)

F31784-01

October 2020

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Database Move to Oracle Cloud Using Zero Downtime Migration, Release 19c (19.7)

F31784-01

Copyright © 2019, 2020, Oracle and/or its affiliates.

Primary Author: Virginia Beecher

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

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

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

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

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

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

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

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

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

Contents

Preface	
<hr/>	
Audience	viii
Documentation Accessibility	viii
Related Documents	viii
Conventions	ix
1	Introduction to Zero Downtime Migration
<hr/>	
About Zero Downtime Migration	1-1
Zero Downtime Migration Capabilities	1-3
Supported Migration Paths	1-3
Supported Backup Media	1-4
Supported Configurations	1-5
Supported Platforms	1-6
Supported Database Versions for Migration	1-6
Supported Database Architectures for Migration	1-7
Zero Downtime Migration Security Provisions	1-7
Zero Downtime Migration Database Server Access	1-8
Target Placeholder Database Environment	1-9
Zero Downtime Migration Operational Phases	1-9
2	Setting Up Zero Downtime Migration Software
<hr/>	
Performing a New Zero Downtime Migration Software Installation	2-1
Prepare a Host for Zero Downtime Migration Software Installation	2-1
Install Zero Downtime Migration Software	2-2
Updating Zero Downtime Migration Software	2-4
Uninstalling Zero Downtime Migration Software	2-5
3	Preparing for Database Migration
<hr/>	
Configuring Connectivity Prerequisites	3-1

Configuring Connectivity From the Zero Downtime Migration Service Host to the Source and Target Database Servers	3-2
Configuring SUDO Access	3-3
Configuring Connectivity Between the Source and Target Database Servers	3-4
Option 1: Use SCAN	3-4
Option 2: Set up an SSH Tunnel	3-5
Generate SSH Keys Without a Passphrase	3-8
Preparing the Source and Target Databases	3-9
Source Database Prerequisites	3-10
Target Database Prerequisites	3-11
Setting Up the Transparent Data Encryption Wallet	3-13
Preparing the Response File	3-15
Response File Settings for Migration to Oracle Cloud Infrastructure	3-16
Response File Settings for Migration to Exadata Cloud Service	3-18
Response File Settings for Exadata Cloud at Customer with Zero Data Loss Recovery Appliance Backup	3-19
Response File Settings for Exadata Cloud at Customer with Object Storage Backup	3-20
Response File Settings for Exadata Cloud at Customer with NFS Backup	3-22
Response File Settings for Offline Migration (Backup and Recovery)	3-24
Preparing for Automatic Application Switchover	3-25
Customizing a Migration Job	3-26
Registering Action Plug-ins	3-26
Creating an Action Template	3-27
Updating Action Plug-ins	3-27
Associating an Action Template with a Migration Job	3-28

4 Migrating Your Database with Zero Downtime Migration

Migrate the Database	4-1
Query Migration Job Status	4-6
List Migration Job Phases	4-6
Pause and Resume a Migration Job	4-7
Rerun a Migration Job	4-8
Terminate a Running Migration Job	4-8
Post-Migration Tasks	4-8
Run Datapatch on the Target Database	4-9

5 Troubleshooting Zero Downtime Migration

Handling Migration Job Failures	5-1
---------------------------------	-----

A Zero Downtime Migration Port Requirements

B Zero Downtime Migration Encryption Requirements

C Provide Passwords Non-Interactively Using a Wallet

D Zero Downtime Migration Process Phases

E Zero Downtime Migration Response File Parameters Reference

BACKUP_PATH	E-5
DATAPATCH_WITH_ONE_INSTANCE_RUNNING	E-5
HOST	E-6
MAX_DATAPATCH_DURATION_MINS	E-6
MIGRATION_METHOD	E-7
OPC_CONTAINER	E-7
PLATFORM_TYPE	E-8
SHUTDOWN_SRC	E-8
SKIP_FALLBACK	E-8
SKIP_SRC_SERVICE_RETENTION	E-9
SRC_BASTION_HOST_IP	E-9
SRC_BASTION_IDENTITY_FILE	E-9
SRC_BASTION_PORT	E-10
SRC_BASTION_USER	E-11
SRC_CONFIG_LOCATION	E-11
SRC_HOST_IP	E-12
SRC_HTTP_PROXY_PORT	E-12
SRC_HTTP_PROXY_URL	E-12
SRC_OSS_PROXY_HOST	E-13
SRC_OSS_PROXY_PORT	E-13
SRC_RMAN_CHANNELS	E-13
SRC_SSH_RETRY_TIMEOUT	E-13
SRC_TIMEZONE	E-14
SRC_ZDLRA_WALLET_LOC	E-14
TGT_BASTION_HOST_IP	E-14
TGT_BASTION_IDENTITY_FILE	E-15
TGT_BASTION_PORT	E-16
TGT_BASTION_USER	E-16

TGT_CONFIG_LOCATION	E-17
TGT_DATAACFS	E-17
TGT_DATADG	E-18
TGT_DB_UNIQUE_NAME	E-18
TGT_HOST_IP	E-19
TGT_HTTP_PROXY_PORT	E-19
TGT_HTTP_PROXY_URL	E-20
TGT_OSS_PROXY_HOST	E-20
TGT_OSS_PROXY_PORT	E-20
TGT_RECOACFS	E-21
TGT_RECODG	E-21
TGT_REDOACFS	E-22
TGT_REDODG	E-23
TGT_RMAN_CHANNELS	E-23
TGT_SKIP_DATAPATCH	E-23
TGT_SSH_RETRY_TIMEOUT	E-24
TGT_SSH_TUNNEL_PORT	E-24
TGT_ZDLRA_WALLET_LOC	E-24
ZDLRA_CRED_ALIAS	E-25
ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL	E-25
ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL	E-26
ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL	E-27
ZDM_BACKUP_RETENTION_WINDOW	E-27
ZDM_CLONE_TGT_MONITORING_INTERVAL	E-27
ZDM_CURL_LOCATION	E-28
ZDM_LOG_OSS_PAR_URL	E-28
ZDM_OPC_RETRY_COUNT	E-28
ZDM_OPC_RETRY_WAIT_TIME	E-29
ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL	E-29
ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL	E-29
ZDM_SRC_TNS_ADMIN	E-30
ZDM_USE_EXISTING_UNDO_SIZE	E-30

F Zero Downtime Migration ZDMCLI Command Reference

abort job	F-1
add imagetype	F-2
add useraction	F-2
migrate database	F-3
modify useraction	F-6
query job	F-6

Index

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](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

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.
<i>italic</i>	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 how Zero Downtime Migration works, and its requirements and supported configurations.

- [About Zero Downtime Migration](#)
Zero Downtime Migration gives you a quick and easy way to move databases to Oracle Cloud Database services.
- [Zero Downtime Migration Capabilities](#)
The Zero Downtime Migration service has many benefits and is highly customizable.
- [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 Backup Media](#)
Part of the Zero Downtime Migration process involves creating a backup of the source database and restoring it to the target database. Zero Downtime Migration supports Oracle Cloud Infrastructure Object Storage, Zero Data Loss Recovery Appliance, or NFS storage backup media, depending on your target environment.
- [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 you configure a placeholder database target environment before beginning the migration process. 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.
- [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 databases to Oracle Cloud Database services.

With Zero Downtime Migration, you can migrate Oracle databases from on premises, Oracle Cloud Infrastructure Classic, or even from one Oracle Cloud Infrastructure

region to another. The databases can be moved to Oracle Cloud Infrastructure, Exadata Cloud at Customer, and Exadata Cloud Service without incurring any significant downtime, by leveraging technologies such as Oracle Active Data Guard and Oracle Recovery Manager (RMAN).

Zero Downtime Migration provides a robust, flexible, and resumable migration process that is also easy to roll back. You can perform and manage a database migration of an individual database or perform database migrations at a fleet level. 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.

The Zero Downtime Migration software is a service with a command line interface that you install and run on a host that you provision. 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 supports both online and offline (backup and recovery) migration.

Online Migration

Using the online migration method, Zero Downtime Migration performs the following tasks:

- Back up the source database to Oracle Cloud Infrastructure Object Storage, Zero Data Loss Recovery Appliance, or NFS
- Create a standby database in the target environment from the backup, with Data Guard configuration, Oracle Data Guard Maximum Performance protection mode, and asynchronous (ASYNC) redo transport mode
- Synchronize the source and target databases
- Switch over to the target database as the primary database, during which there is a minimal period of downtime

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 source database 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 (ship redo) from the new primary to the new standby on the source database. This configuration makes it possible to perform a switchover with minimal downtime, if you need to fall back to the original source database.

However, if there is no SQL*Net connectivity between the new primary and the new standby after the switchover, there is no data synchronization (ship redo) from the new primary to the new standby on the source database. With this configuration you cannot fall back to the original source database.

Note that Transparent Data Encryption (TDE) is enabled on Oracle databases in the Oracle Cloud by default. Zero Downtime Migration handles the encryption of your target database, even if TDE is not enabled on the source Oracle database. 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 (the source) are encrypted. Therefore, if you decide to switch back and role swap again, making the source database the primary again and the database in the Oracle Cloud

the standby, the source database will not be able to read the newly encrypted changed blocks applied by the redo logs unless TDE is enabled on the source database.

In order to avoid any post-migration issues, the recommended best practice is to perform testing and validation on the target database before Zero Downtime Migration performs the switchover phase of the migration process. There are options outside of Zero Downtime Migration for testing a snapshot standby database. When testing and validation are complete and you are ready to proceed with switchover, you can delete the snapshot standby database, instruct Zero Downtime Migration to perform the switchover, and finalize the migration process.

Offline Migration

Using the offline migration method, Zero Downtime Migration backs up the source database to Oracle Cloud Infrastructure Object Storage and instantiates a new database from this Object Storage backup to Oracle Cloud Infrastructure, Exadata Cloud at Customer, or Exadata Cloud Service. Note that the only available method for migrating Oracle Database Standard Edition is the offline migration method.

The offline migration method is similar to cloning a database. The target database has no relationship to the source, so there is no data synchronization or fallback capability. No SQL*Net connectivity is needed between the source and target database servers.

Zero Downtime Migration Capabilities

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

- **Audit capability** - All custom user actions are audited including actions performed by the migration job.
- **Work flow customization** - Work flow actions (marked by phases) can be customized with pre-user-action and post-user-action 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 termination** - You can terminate a running migration job, rather than waiting for it to complete.
- **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 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 are supported migration paths:

- **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.
- **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.
- **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.
- **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.
- **Oracle Cloud Infrastructure Database to Another Oracle Cloud Infrastructure Region**
You can migrate a database from one Oracle Cloud Infrastructure region to another Oracle Cloud Infrastructure region with Zero Downtime Migration. For example, you can move a database from the `phoenix` region to the `frankfurt` or `ashburn` region.

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

Supported Backup Media

Part of the Zero Downtime Migration process involves creating a backup of the source database and restoring it to the target database. Zero Downtime Migration supports Oracle Cloud Infrastructure Object Storage, Zero Data Loss Recovery Appliance, or NFS storage backup media, depending on your target environment.

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

Oracle Cloud Infrastructure Object Storage

Object Storage is supported as a backup medium when migrating a database to Oracle Cloud Infrastructure, Exadata Cloud Service, or Exadata Cloud at Customer.

If you back up the database to Object Storage, then the Zero Downtime Migration service initiates the source database backup and restores it to the target environment, so Object Storage must be accessible from both the source and target environments.

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.

Zero Data Loss Recovery Appliance

Zero Data Loss Recovery Appliance is supported as a backup medium when migrating a database to an Exadata Cloud at Customer target.

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 parameters, 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.

Refer to the Zero Data Loss Recovery Appliance documentation for information about creating backups.

Network File System (NFS)

NFS is supported as a backup medium when migrating a database to an Exadata Cloud at Customer target.

If you choose to back up the database to an NFS mount, then the Zero Downtime Migration service initiates the source database backup and restores it to the Exadata Cloud at Customer target environment. The NFS should be accessible from both the source and target environments.

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 the following database architecture implementations.

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 (Linux-x86-64) or later releases.

You can deploy the Zero Downtime Migration service on a standalone server on-premises or on a standalone Linux server (compute instance) in the Oracle Cloud. Oracle Linux is the supported platform for the Zero Downtime Migration service host.

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)
- Oracle Database 19c
- All subsequent Oracle Database releases

 **Note:**

Because Zero Downtime Migration leverages Oracle Data Guard, you must have the same operating system and database version on both source and target. However, note that, while Standard Edition databases can use Zero Downtime Migration, they must use the offline migration method which is based on a backup and restore methodology and does not leverage Data Guard.

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 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

 **Note:**

Zero Downtime Migration does not support migration from a non-CDB "traditional" database to a CDB in the multitenant architecture.

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 service host to the source database server and the target database server 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 `$ZDM_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 service 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 you configure a placeholder database target environment before beginning the migration process. 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.

During the migration process, Zero Downtime Migration service host restores the source database to this placeholder database target environment by dropping the placeholder database and recreating a database in the target environment with the same `db_name` as that of source database.

Any database parameters for the target database, including SGA parameters, are maintained during the migration, and the migrated database runs with this same configuration.

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. You can make any modifications to database parameters 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

Whether you are performing a new Zero Downtime Migration software installation, updating existing software to the latest release, or removing the software, read the appropriate topics carefully as there may have been changes since the last time you performed the task.

Always 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 software installation and updates.

- [Performing a New Zero Downtime Migration Software Installation](#)
If a host has not had Zero Downtime Migration software installed on it previously, verify that it complies with the requirements and perform any pre-installation tasks, then download and install the software. Once the software is installed, the host is referred to as the Zero Downtime Migration service host.
- [Updating Zero Downtime Migration Software](#)
If you already have Zero Downtime Migration software installed on a host, you can update it to the latest release. Zero Downtime Migration software updates give you the latest fixes while retaining existing job information, metadata, and log files.
- [Uninstalling Zero Downtime Migration Software](#)
Remove Zero Downtime Migration software from the Zero Downtime Migration service host.

Performing a New Zero Downtime Migration Software Installation

If a host has not had Zero Downtime Migration software installed on it previously, verify that it complies with the requirements and perform any pre-installation tasks, then download and install the software. Once the software is installed, the host is referred to as the Zero Downtime Migration service host.

- [Prepare a Host for Zero Downtime Migration Software Installation](#)
Provision a host with the following prerequisites and complete the following preinstallation tasks before installing Zero Downtime Migration software on it.
- [Install Zero Downtime Migration Software](#)
Download the Zero Downtime Migration software and install it on the Zero Downtime Migration service host.

Prepare a Host for Zero Downtime Migration Software Installation

Provision a host with the following prerequisites and complete the following preinstallation tasks before installing Zero Downtime Migration software on it.

- 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.
- Zero Downtime Migration software requires a standalone Linux host running Oracle Linux 7 or later.
- The Zero Downtime Migration service host must be able to connect to the source and the target database servers.
- 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.

Install Zero Downtime Migration Software

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

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 zdmversion.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
```

- zdminstall.sh is the installation script
- oraclehome is the Oracle Home where the Zero Downtime Migration software 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 compressed 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. \$ZDM_HOME/inventory/orainstRoot.sh
2. \$ZDM_HOME/root.sh

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

```
zdmuser> $ZDM_HOME/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> $ZDM_HOME/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> $ZDM_HOME/bin/zdmservice status
-----
Service Status
-----
Running: true
```

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

Updating Zero Downtime Migration Software

If you already have Zero Downtime Migration software installed on a host, you can update it to the latest release. Zero Downtime Migration software updates give you the latest fixes while retaining existing job information, metadata, and log files.

Before you begin the software update, review the following requirements.

- Verify that your existing Zero Downtime Migration software install location has at least 15GB free space.
- **Important:** Run the update script from outside of the currently installed Zero Downtime Migration home.

Running the script from within a Zero Downtime Migration home results in home install and uninstall failures and leaves the service in an inconsistent state.

- The path specified in `ziploc` should have read/write access for `zdmuser`.
 - All of the commands in the following procedure should be run as the existing Zero Downtime Migration software owner. For example, run as `zdmuser` in the examples that follow.
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. Change to the directory to where Zero Downtime Migration software is downloaded and unzip the software.

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

3. Run the `zdminstall.sh` script as the existing Zero Downtime Migration home owner to update the software from the software download location.

```
zdmuser> ./zdminstall.sh update oraclehome=existing_zdm_oracle_home
ziploc=zdm_software_location -zdm
```

- `zdminstall.sh` is the installation and update script
- `oraclehome` is the Oracle Home value where the existing Zero Downtime Migration software is installed
- `ziploc` is the location of the compressed software file (zip) included in the Zero Downtime Migration kit

For example,

```
zdmuser>/u01/app/oracle/zdm/shiphome/update/zdminstall.sh update
oraclehome=/u01/app/zdmhome
ziploc=/u01/app/oracle/zdm/shiphome/update/zdm_home.zip -zdm
```

The update script does the following operations.

- a. Backs up the existing Zero Downtime Migration home (ZDM_HOME) and ZDM_BASE into software download location
 - b. Stops the currently running Zero Downtime Migration service
 - c. Removes the currently installed Zero Downtime Migration home
 - d. Installs the new binaries in the Zero Downtime Migration home
 - e. Restores the configuration data.
4. Start the Zero Downtime Migration service as user `zdmuser`.

```
zdmuser> $ZDM_HOME/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> $ZDM_HOME/bin/zdmservice stop
```

5. 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> $ZDM_HOME/bin/zdmservice status
-----
                Service Status
-----

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

Uninstalling Zero Downtime Migration Software

Remove Zero Downtime Migration software from the Zero Downtime Migration service host.

All commands are run as `zdmuser`.

1. Stop the Zero Downtime Migration service.

```
zdmuser> $ZDM_HOME/bin/zdmservice stop
```

2. Run the following command to uninstall the software.

```
zdmuser> $ZDM_HOME/bin/zdmservice deinstall
```


3

Preparing for Database Migration

Before starting a Zero Downtime Migration database migration you must configure connectivity between the servers, prepare the source and target databases, set parameters in the response file, and configure any required migration job customization.

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

- [Configuring Connectivity Prerequisites](#)
Connectivity must be set up between the Zero Downtime Migration service host and the source and target database servers.
- [Preparing the Source and Target Databases](#)
See the following topics for information about preparing the source and target databases for migration.
- [Preparing the Response File](#)
Set the response file parameters for the migration target and backup medium you are using in the migration process.
- [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 SUDO Access](#)
You may need to grant certain users authority to perform operations using `sudo` on the source and target database servers.
- [Configuring Connectivity Between the Source and Target Database Servers](#)
You have two options for configuring connectivity between the source and target database servers: SCAN or SSH.

- [Generate SSH Keys Without a Passphrase](#)
You can generate a new SSH key without a passphrase if on the Zero Downtime Migration service host the authentication key pairs are not available without a passphrase for the Zero Downtime Migration software installed user.

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 [Generate SSH Keys 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_host.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:

For the source database server:

- If the source database server is accessed with the root user, then no action is required.
- If the source database server is accessed through SSH, 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.

For the target database server:

- Because the target database server is on cloud only and access is through SSH, 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.

Note that the `opc` user is a standard Oracle cloud user that is used to access database servers, but you can use any user and you can use different users for the source and target database servers.

4. Make sure that the source and target database server names 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.

In the following example, the IP address entries are shown as `192.x.x.x`, but you must add your actual public IP addresses.

```
#OCI public IP two node RAC server details
192.0.2.1 ocidb1
```

```
192.0.2.2 ocidb2
#OCIC public IP two node RAC server details
192.0.2.3 ocicdb1
192.0.2.4 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_host_private_key_file_location
user@source/target_database_server_name
```

For example,

```
zdmuser> ssh -i /home/zdmuser/.ssh/zdm_service_host.ppk opc@ocidb1
zdmuser> ssh -i /home/zdmuser/.ssh/zdm_service_host.ppk opc@ocicdb1
```

 **Note:**

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 without the need to enter a passphrase.

 **See Also:**

[Zero Downtime Migration Port Requirements](#)

Configuring SUDO Access

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

For source database servers:

- If the source database server is accessed with the `root` user, then there is no need to configure Sudo operations.
- If the source database server is accessed through SSH, then configure Sudo operations to run without prompting for a password for the database installed user and the `root` user.

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

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

For target database servers:

- Because the target database server is on the cloud only, any Sudo operations are configured already. Otherwise, configure all Sudo operations to run without prompting for a password for the database installed user and the `root` user.

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

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

Note, for example, if the login user is `opc`, then you can enable Sudo operations for the `opc` user.

Configuring Connectivity Between the Source and Target Database Servers

You have two options for configuring connectivity between the source and target database servers: SCAN or SSH.

Configure connectivity using one of the following options.

- **Option 1: Use SCAN**
To use this option, the SCAN of the target should be resolvable from the source database server, and the SCAN of the source should be resolvable from the target server.
- **Option 2: Set up an SSH Tunnel**
If connectivity using 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.

Option 1: Use SCAN

To use this option, the SCAN of the target should be resolvable from the source database server, and the SCAN of the source should be resolvable from the target server.

The specified source database server in the `ZDMCLI MIGRATE DATABASE` command `-sourcenode` parameter can connect to the target database instance over target SCAN through the respective SCAN port and vice versa.

With SCAN connectivity from both sides, the source database and target databases can synchronize from either direction. 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 the target database and source database cannot synchronize after switchover.

Test Connectivity

To test connectivity from the source to the target environment, 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
```

To 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 database server to the source database server.

 **See Also:**

[Zero Downtime Migration Port Requirements](#)

Option 2: Set up an SSH Tunnel

If connectivity using 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.

The following procedure sets up an SSH tunnel on the source database servers for the root user. Note that this procedure amounts to setting up what may be considered a temporary channel. Using this connectivity option, you will not be able to synchronize between the target database and source database after switchover, and with this configuration you cannot fall back to the original source database.

 **Note:**

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

1. Generate an SSH key file without a passphrase for the `opc` user on the target Oracle Cloud Infrastructure server, using the information in [Generate SSH Keys Without a Passphrase](#). If the target is an Oracle RAC database, then generate an SSH key file without a passphrase from the first Oracle RAC server.
2. 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.
3. 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, `ocidb1.ppk` (where `ocidb1` is the target server name).

The file permissions should be similar to the following.

```
/root/.ssh>ls -l ocidb1.ppk
-rw----- 1 root root 1679 Oct 16 10:05 ocidb1.ppk
```

4. 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 on the source database server, which you copied from the target database server in step 3 above.
- *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 ocidb1
  HostName 192.0.2.1
  IdentityFile /root/.ssh/ocidb1.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 `ocidb1`, and the Oracle Cloud Infrastructure server public IP address is `192.0.2.1`.

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.

5. Make sure that you can SSH to the 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@ocidb1 ~] ssh -i /root/.ssh/ocidb1.ppk opc@ocidb1
Last login: Fri Dec 7 14:53:09 2018 from 192.0.2.3
```

```
[opc@ocidb1 ~]$
```

Note:

SSH connectivity requires direct, non-interactive access between the source and target database servers, without the need to enter a passphrase.

6. 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_por
t -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 the IP address of the target database server. 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 tunnel.

```
[root@ocidb1~]ssh -f ocidb1 -L 9000:192.0.2.9:1521 -N
```

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

7. 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@ocidb1 ~] 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.

Generate SSH Keys Without a Passphrase

You can generate a new SSH key without a passphrase if on the Zero Downtime Migration service host the authentication key pairs are not available without a passphrase for the Zero Downtime Migration software installed user.

- [Setting Up the Transparent Data Encryption Wallet](#)
For Oracle Database 12c Release 2 and later, if the source and target databases do not have Transparent Data Encryption (TDE) enabled, then it is mandatory that you configure the TDE wallet before migration begins.

Source Database Prerequisites

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

- The source database must be running in archive log mode.
- Configure the TDE wallet on Oracle Database 12c Release 2 and later. Enabling TDE on Oracle Database 11g Release 2 (11.2.0.4) and Oracle Database 12c Release 1 is optional.

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.

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;
```

- 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.

- Verify that port 22 on the source and target database servers allow incoming connections from the Zero Downtime Migration service host.
- Ensure that the scan listener ports (1521, for example) on the source database servers allow incoming connections from the target database servers and outgoing connections to the target database servers.

Alternate SQL connectivity should be made available if a firewall blocks incoming remote connection using the `SCAN` listener port.

- 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 synchronize the target cloud database, then these files should be restored so that Zero Downtime Migration can continue the migration process.

- If the source database is deployed using Oracle Grid Infrastructure and the database is not registered using SRVCTL, then you must register the database before the migration.
- The source database must use a server parameter file (SPFILE).
- The source database must have a password file in location `$ORACLE_HOME/dbs/orapwORACLE_SID`; otherwise, create it using the `ORAPWD` utility.
- If RMAN is not already configured to automatically back up the control file and SPFILE, then set `CONFIGURE CONTROLFILE AUTOBACKUP` to `ON` and revert the setting back to `OFF` after migration is complete.

```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP ON;
```

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.

- You must create a placeholder target database using Grid Infrastructure based Database Services before database migration begins.

Note:

For this release, only Grid Infrastructure-based database services are supported as targets. For example, an LVM-based instance or an instance created in compute node without Grid Infrastructure are not supported targets.

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 parameters**

- * `DB_NAME` - If the target database is Exadata Cloud Service or Exadata Cloud at Customer, then the database `DB_NAME` should be the same as the source database `DB_NAME`. If the target database is Oracle Cloud Infrastructure, then the database `DB_NAME` can be the same as or different from the source database `DB_NAME`.
- * `DB_UNIQUE_NAME`: If the target database is Oracle Cloud Infrastructure, Exadata Cloud Service, or Exadata Cloud at Customer, 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.
- **Match the source SYS password** - Specify a `SYS` password that matches that of 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**.
- 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 Jan 2020 PSU/BP and the target database is at April 2020 PSU/BP), then you must run the `datapatch` utility after database migration.

- The target database time zone version must be the same as the source database time zone version. To check the current time zone version, query the `v$timezone_file` view as shown here, and upgrade the time zone file if necessary.

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

- Verify that the TDE wallet folder exists, and ensure that the wallet `STATUS` is `OPEN` and `WALLET_TYPE` is `AUTOLOGIN` (For an auto-login wallet type), or `WALLET_TYPE` is `PASSWORD` (For a password-based wallet). 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;
```

- The target database must use a server parameter file (SPFILE).
- 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.
- 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.
- Make sure adequate storage is provisioned and available on the object store to accommodate the source database backup.

- Verify that ports 22 and 1521 on the target servers in the Oracle Cloud Infrastructure, Exadata Cloud Service, or Exadata Cloud at Customer environment are open and not blocked by a firewall.
- 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](#)

Setting Up the Transparent Data Encryption Wallet

For Oracle Database 12c Release 2 and later, if the source and target databases do not have Transparent Data Encryption (TDE) enabled, then it is mandatory that you configure the TDE wallet before migration begins.

TDE should be enabled and the `TDE WALLET` status on both source and target databases must be set to `OPEN`. The `WALLET_TYPE` can be `AUTOLOGIN`, for an auto-login wallet (preferred), or `PASSWORD`, for a password-based wallet. On a multitenant database, make sure that the wallet is open on all PDBs as well as the CDB, and that the master key is set for all PDBs and the CDB.

If TDE is not already configured as required on the source and target databases, use the following instructions to set up the TDE wallet.

For a password-based wallet, you only need to do steps 1, 2, and 4; for an auto-login wallet, complete all of the steps.

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

```
/home/oracle>cat /u01/app/oracle/product/12.2.0.1/dbhome_4/network/  
admin/sqlnet.ora
```

```
ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=FILE)  
(METHOD_DATA=(DIRECTORY=/u01/app/oracle/product/12.2.0.1/dbhome_4/  
network/admin)))
```

For an Oracle RAC instance, also set `ENCRYPTION_WALLET_LOCATION` in the second Oracle RAC node.

2. Create and configure the keystore.

- a. Connect to the database and create 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 password;
```

- b. Open the keystore.

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

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

For a CDB environment, run the following command.

```
SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY
password container = ALL;
keystore altered.
```

- c. Create and activate the master encryption key.

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

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

For a CDB environment, run the following command.

```
SQL> ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY password
with backup container = ALL;
keystore altered.
```

- d. Query V\$ENCRYPTION_KEYS to get the wallet status, wallet type, and wallet location.

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

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
```

The configuration of a password-based wallet is complete at this stage, and the wallet is enabled with status `OPEN` and `WALLET_TYPE` is shown as `PASSWORD` in the query output above.

Continue to step 3 only if you need to configure an auto-login wallet, otherwise skip to step 4.

3. For an auto-login wallet only, complete the keystore configuration.

a. Create the auto-login keystore.

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

b. Close the password-based wallet.

```
SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE CLOSE IDENTIFIED BY
password;
keystore altered.
```

c. Query `V$ENCRYPTION_WALLET` to get the wallet status, wallet type, and wallet location.

```
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
```

In the query output, verify that the TDE wallet `STATUS` is `OPEN` and `WALLET_TYPE` set to `AUTOLOGIN`, otherwise the auto-login wallet is not set up correctly.

This completes the auto-login wallet configuration.

4. Copy the wallet files to the second Oracle RAC node.

If you configured the wallet in a shared file system for Oracle RAC, or if you are enabling TDE for a single instance database, then no action is required.

If you are enabling TDE for Oracle RAC database without shared access to the wallet, copy the following files to the same location on second node.

- `/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*`

Preparing the Response File

Set the response file parameters for the migration target and backup medium you are using in the migration process.

The response file settings in the following topics show you how to configure a typical use case. To further customize your configuration you can find additional parameters described in [Zero Downtime Migration Response File Parameters Reference](#).

- [Response File Settings for Migration to Oracle Cloud Infrastructure](#)
Configure the following response file settings to migrate data to an Oracle Cloud Infrastructure virtual machine or bare metal target.
- [Response File Settings for Migration to Exadata Cloud Service](#)
Configure the following response file settings to migrate data to an Exadata Cloud Service target.
- [Response File Settings for Exadata Cloud at Customer with Zero Data Loss Recovery Appliance Backup](#)
Configure the following response file settings to migrate data to an Exadata Cloud at Customer target using Zero Data Loss Recovery Appliance as the backup medium.
- [Response File Settings for Exadata Cloud at Customer with Object Storage Backup](#)
Configure the following response file settings to migrate data to an Exadata Cloud at Customer target using Oracle Cloud Infrastructure Object Storage service as the backup medium.
- [Response File Settings for Exadata Cloud at Customer with NFS Backup](#)
Configure the following response file settings to migrate data to an Exadata Cloud at Customer target using NFS storage as the backup medium.
- [Response File Settings for Offline Migration \(Backup and Recovery\)](#)
Configure the following response file settings before migrating a database offline to an Oracle Cloud Infrastructure, Exadata Cloud at Customer, or Exadata Cloud Service target environment.

Response File Settings for Migration to Oracle Cloud Infrastructure

Configure the following response file settings to migrate data to an Oracle Cloud Infrastructure virtual machine or bare metal target.

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 as follows.

- Set `TGT_DB_UNIQUE_NAME` to the target database `DB_UNIQUE_NAME` value. To find `DB_UNIQUE_NAME` run

```
SQL> show parameter db_unique_name
```
- Set `PLATFORM_TYPE` to `VMDB`.
- Set `MIGRATION_METHOD` to `DG_OSS`, where `DG` stands for Data Guard and `OSS` stands for Object Storage service.
- If SSH tunneling is set up, set the `TGT_SSH_TUNNEL_PORT` parameter.
- Zero Downtime Migration automatically discovers the location for data, redo, and reco storage volumes from the specified target database. If you need to override the discovered values, specify the target database data files storage (ASM or ACFS) location using the appropriate set of parameters.

- ASM: TGT_DATADG, TGT_REDODG, and TGT_RECODG
- ACFS: 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.
- If the target database environment is at a higher patch level than the source database (for example, if the source database is at Jan 2020 PSU/BP and the target database is at April 2020 PSU/BP), then use the TGT_SKIP_DATAPATCH=FALSE parameter to run the datapatch utility to apply a database patch on the target database as part of the post-migration tasks. Otherwise, you need to run the datapatch utility manually after the migration.
- Set ZDM_LOG_OSS_PAR_URL to the Cloud Object Store pre-authenticated URL if you want to upload migration logs onto Cloud Object Storage. For information about getting a pre-authenticated URL see Oracle Cloud documentation at <https://docs.cloud.oracle.com/en-us/iaas/Content/Object/Tasks/usingpreauthenticatedrequests.htm#usingconsole>.
- Set *phase_name*_MONITORING_INTERVAL=*n mins* if you want Zero Downtime Migration to monitor and report the status of backup and restore operations at the configured time interval during the migration. The default interval value is 10 minutes. To disable monitoring, set these values to 0 (zero).

```
ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL=
ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL=
ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL=
ZDM_CLONE_TGT_MONITORING_INTERVAL=
ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL=
ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL=
```

- Set ZDM_BACKUP_RETENTION_WINDOW=*number of days* if you wish to retain source database backup after the migration.
- Set ZDM_SRC_TNS_ADMIN=*TNS_ADMIN value* in case of custom location.
- To access the Oracle Cloud Object Storage, set the following parameters in the response file.
 - Set HOST to the cloud storage REST endpoint URL.
 - * For Oracle Cloud Infrastructure storage the typical value format is HOST=https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/ObjectStorageNamespace
 - To find the Object Storage Namespace value, log in to the Cloud Console and select **Menu > Administration > Tenancy Detail**, and in the **Object Storage Settings** section find **Value against entry Object Storage Namespace:**
 - * For Oracle Cloud Infrastructure Classic storage the typical value format is HOST=https://acme.storage.oraclecloud.com/v1/Storage-tenancyname
 - 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.

Response File Settings for Migration to Exadata Cloud Service

Configure the following response file settings to migrate data to an Exadata Cloud Service target.

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 as follows.

- Set `TGT_DB_UNIQUE_NAME` to the target database `DB_UNIQUE_NAME` value. To find `DB_UNIQUE_NAME` run

```
SQL> show parameter db_unique_name
```
- Set `PLATFORM_TYPE` to `EXACS`.
- Set `MIGRATION_METHOD` to `DG_OSS`, where `DG` stands for Data Guard and `OSS` stands for Object Storage service.
- If SSH tunneling is set up, set the `TGT_SSH_TUNNEL_PORT` parameter.
- Zero Downtime Migration automatically discovers the location for `data`, `redo`, and `reco` storage volumes from the specified target database. If you need to override the discovered values, specify the target database data files storage (ASM or ACFS) location using the appropriate set of parameters.
 - ASM: `TGT_DATADG`, `TGT_REDODG`, and `TGT_RECODG`
 - ACFS: `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.
- If the target database environment is at a higher patch level than the source database (for example, if the source database is at Jan 2020 PSU/BP and the target database is at April 2020 PSU/BP), then use the `TGT_SKIP_DATAPATCH=FALSE` parameter to run the `datapatch` utility to apply a database patch on the target database as part of the post-migration tasks. Otherwise, you need to run the `datapatch` utility manually after the migration.
- Set `ZDM_LOG_OSS_PAR_URL` to the Cloud Object Store pre-authenticated URL if you want to upload migration logs onto Cloud Object Storage. For information about getting a pre-authenticated URL see Oracle Cloud documentation at <https://docs.cloud.oracle.com/en-us/iaas/Content/Object/Tasks/usingpreauthenticatedrequests.htm#usingconsole>.
- Set `phase_name_MONITORING_INTERVAL=n mins` if you want Zero Downtime Migration to monitor and report the status of backup and restore operations at the configured time interval during the migration. The default interval value is 10 minutes. To disable monitoring, set these values to 0 (zero).

```
ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL=  
ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL=  
ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL=
```

```
ZDM_CLONE_TGT_MONITORING_INTERVAL=
ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL=
ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL=
```

- Set `ZDM_BACKUP_RETENTION_WINDOW=number of days` if you wish to retain source database backup after the migration.
- Set `ZDM_SRC_TNS_ADMIN=TNS_ADMIN value` in case of custom location.
- To access the Oracle Cloud Object Storage, set the following parameters in the response file.
 - Set `HOST` to the cloud storage REST endpoint URL.
 - * For Oracle Cloud Infrastructure storage the typical value format is `HOST=https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/ObjectStorageNamespace`
 To find the Object Storage Namespace value, log in to the Cloud Console and select **Menu > Administration > Tenancy Detail**, and in the **Object Storage Settings** section find **Value against entry Object Storage Namespace**:
 - * For Oracle Cloud Infrastructure Classic storage the typical value format is `HOST=https://acme.storage.oraclecloud.com/v1/Storage-tenancyname`
 - 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.

Response File Settings for Exadata Cloud at Customer with Zero Data Loss Recovery Appliance Backup

Configure the following response file settings to migrate data to an Exadata Cloud at Customer target using Zero Data Loss Recovery Appliance as the backup medium.

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 as follows.

- Set `TGT_DB_UNIQUE_NAME` to the target database `DB_UNIQUE_NAME` value. To find `DB_UNIQUE_NAME` run

```
SQL> show parameter db_unique_name
```

For Cloud type Exadata Cloud at Customer Gen 1, set `TGT_DB_UNIQUE_NAME` to a different `DB_UNIQUE_NAME` not currently in use

- 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` for the wallet location, for example,
`TGT_ZDLRA_WALLET_LOC=target_database_oracle_home/dbs/zdlra.`
- Set `ZDLRA_CRED_ALIAS` for the wallet credential alias, for example,

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

- Zero Downtime Migration automatically discovers the location for data, redo, and `reco` storage volumes from the specified target database. If you need to override the discovered values, specify the target database data files storage (ASM or ACFS) location using the appropriate set of parameters.
 - **ASM:** `TGT_DATADG`, `TGT_REDOAG`, and `TGT_RECLOG`
 - **ACFS:** `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.
- If the target database environment is at a higher patch level than the source database (for example, if the source database is at Jan 2020 PSU/BP and the target database is at April 2020 PSU/BP), then use the `TGT_SKIP_DATAPATCH=FALSE` parameter to run the datapatch utility to apply a database patch on the target database as part of the post-migration tasks. Otherwise, you need to run the datapatch utility manually after the migration.
- Set `phase_name_MONITORING_INTERVAL=n mins` if you want Zero Downtime Migration to monitor and report the status of the restore operation at the configured time interval during the migration. The default interval value is 10 minutes. To disable monitoring, set the value to 0 (zero).

```
ZDM_CLONE_TGT_MONITORING_INTERVAL=
```

- Set `ZDM_SRC_TNS_ADMIN=TNS_ADMIN value` in case of custom location.

Response File Settings for Exadata Cloud at Customer with Object Storage Backup

Configure the following response file settings to migrate data to an Exadata Cloud at Customer target using Oracle Cloud Infrastructure Object Storage service as the backup medium.

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 as follows.

- Set `TGT_DB_UNIQUE_NAME` to the target database `DB_UNIQUE_NAME` value. To find `DB_UNIQUE_NAME` run

```
SQL> show parameter db_unique_name
```

For Cloud type Exadata Cloud at Customer Gen 1, set `TGT_DB_UNIQUE_NAME` to a different `DB_UNIQUE_NAME` not currently in use

- Set `PLATFORM_TYPE` to `EXACC`.
- Set `MIGRATION_METHOD` to `DG_OSS`, where `DG` stands for Data Guard and `OSS` for the Object Storage service.
- Zero Downtime Migration automatically discovers the location for `data`, `redo`, and `reco` storage volumes from the specified target database. If you need to override the discovered values, specify the target database data files storage (ASM or ACFS) location using the appropriate set of parameters.
 - ASM: `TGT_DATADG`, `TGT_REODG`, and `TGT_RECODG`
 - ACFS: `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.
- If the target database environment is at a higher patch level than the source database (for example, if the source database is at Jan 2020 PSU/BP and the target database is at April 2020 PSU/BP), then use the `TGT_SKIP_DATAPATCH=FALSE` parameter to run the datapatch utility to apply a database patch on the target database as part of the post-migration tasks. Otherwise, you need to run the datapatch utility manually after the migration.
- Set `phase_name_MONITORING_INTERVAL=n mins` if you want Zero Downtime Migration to monitor and report the status of backup and restore operations at the configured time interval during the migration. The default interval value is 10 minutes. To disable monitoring, set these values to 0 (zero).

```
ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL=  
ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL=  
ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL=  
ZDM_CLONE_TGT_MONITORING_INTERVAL=  
ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL=  
ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL=
```

- Set `ZDM_BACKUP_RETENTION_WINDOW=number of days` if you wish to retain source database backup after the migration.
- Set `ZDM_SRC_TNS_ADMIN=TNS_ADMIN value` in case of custom location.
- To access the Oracle Cloud Object Storage, set the following parameters in the response file.

The source database is backed up to the specified container and restored to Exadata Cloud at Customer using RMAN SQL*Net connectivity.

- Set `HOST` to the cloud storage REST endpoint URL.

- * For Oracle Cloud Infrastructure storage the typical value format is `HOST=https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/ObjectStorageNamespace`

To find the Object Storage Namespace value, log in to the Cloud Console and select **Menu > Administration > Tenancy Detail**, and in the **Object Storage Settings** section find **Value against entry Object Storage Namespace**:
- * For Oracle Cloud Infrastructure Classic storage the typical value format is `HOST=https://acme.storage.oraclecloud.com/v1/Storage-tenancyname`
- 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.

Response File Settings for Exadata Cloud at Customer with NFS Backup

Configure the following response file settings to migrate data to an Exadata Cloud at Customer target using NFS storage as the backup medium.

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 as follows.

- Set `TGT_DB_UNIQUE_NAME` to the target database `DB_UNIQUE_NAME` value. To find `DB_UNIQUE_NAME` run

```
SQL> show parameter db_unique_name
```

For Cloud type Exadata Cloud at Customer Gen 1, set `TGT_DB_UNIQUE_NAME` to a different `DB_UNIQUE_NAME` not currently in use

- 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 example, 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.

- Zero Downtime Migration automatically discovers the location for `data`, `redo`, and `reco` storage volumes from the specified target database. If you need to override the discovered values, specify the target database data files storage (ASM or ACFS) location using the appropriate set of parameters.
 - ASM: `TGT_DATADG`, `TGT_REDODG`, and `TGT_RECODG`
 - ACFS: `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.
- If the target database environment is at a higher patch level than the source database (for example, if the source database is at Jan 2020 PSU/BP and the target database is at April 2020 PSU/BP), then use the `TGT_SKIP_DATAPATCH=FALSE` parameter to run the datapatch utility to apply a database patch on the target database as part of the post-migration tasks. Otherwise, you need to run the datapatch utility manually after the migration.
- Set `phase_name_MONITORING_INTERVAL=n mins` if you want Zero Downtime Migration to monitor and report the status of backup and restore operations at the configured time interval during the migration. The default interval value is 10 minutes. To disable monitoring, set these values to 0 (zero).

```
ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL=
ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL=
ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL=
ZDM_CLONE_TGT_MONITORING_INTERVAL=
ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL=
ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL=
```

- Set `ZDM_BACKUP_RETENTION_WINDOW=number of days` if you wish to retain source database backup after the migration.
- Set `ZDM_SRC_TNS_ADMIN=TNS_ADMIN value` in case of custom location.

Response File Settings for Offline Migration (Backup and Recovery)

Configure the following response file settings before migrating a database offline to an Oracle Cloud Infrastructure, Exadata Cloud at Customer, or Exadata Cloud Service target environment.

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 as follows.

- Set `TGT_DB_UNIQUE_NAME` to the target database `DB_UNIQUE_NAME` value. To find `DB_UNIQUE_NAME` run

```
SQL> show parameter db_unique_name
```

- 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`.
- Where Object Storage Service is used for the backup medium, set `MIGRATION_METHOD` to `BACKUP_RESTORE_OSS`.

The Exadata Cloud at Customer platform can also use the NFS backup medium. If this is the case, set `MIGRATION_METHOD` to `BACKUP_RESTORE_NFS`, and ignore the Oracle Cloud Object Storage parameter settings.

- Zero Downtime Migration automatically discovers the location for `data`, `redo`, and `reco` storage volumes from the specified target database. If you need to override the discovered values, specify the target database data files storage (ASM or ACFS) location using the appropriate set of parameters.
 - ASM: `TGT_DATADG`, `TGT_REDODG`, and `TGT_RECODG`
 - ACFS: `TGT_DATAACFS`, `TGT_REDOACFS`, and `TGT_RECOACFS`
- If the target database environment is at a higher patch level than the source database (for example, if the source database is at Jan 2020 PSU/BP and the target database is at April 2020 PSU/BP), then use the `TGT_SKIP_DATAPATCH=FALSE` parameter to run the datapatch utility to apply a database patch on the target database as part of the post-migration tasks. Otherwise, you need to run the datapatch utility manually after the migration.
- Set `ZDM_LOG_OSS_PAR_URL` to the Cloud Object Store pre-authenticated URL if you want to upload migration logs onto Cloud Object Storage. For information about getting a pre-authenticated URL see Oracle Cloud documentation at <https://docs.cloud.oracle.com/en-us/iaas/Content/Object/Tasks/usingpreauthenticatedrequests.htm#usingconsole>.
- Set `phase_name_MONITORING_INTERVAL=n mins` if you want Zero Downtime Migration to monitor and report the status of backup and restore operations at

the configured time interval during the migration. The default interval value is 10 minutes. To disable monitoring, set these values to 0 (zero).

```
ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL=
ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL=
ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL=
ZDM_CLONE_TGT_MONITORING_INTERVAL=
ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL=
ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL=
```

- Set `ZDM_BACKUP_RETENTION_WINDOW=number of days` if you wish to retain source database backup after the migration.
- Set `ZDM_SRC_TNS_ADMIN=TNS_ADMIN value` in case of custom location.
- To access the Oracle Cloud Object Storage, set the following parameters in the response file.
 - Set `HOST` to the cloud storage REST endpoint URL.
 - * For Oracle Cloud Infrastructure storage the typical value format is `HOST=https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/ObjectStorageNamespace`
To find the Object Storage Namespace value, log in to the Cloud Console and select **Menu > Administration > Tenancy Detail**, and in the **Object Storage Settings** section find **Value against entry Object Storage Namespace**:
 - * For Oracle Cloud Infrastructure Classic storage the typical value format is `HOST=https://acme.storage.oraclecloud.com/v1/Storage-tenancyname`
 - 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.

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 Active Data Guard Best Practices](https://www.oracle.com/goto/maa) 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 user action 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, `zdmuser`) to add user actions to a database migration job. Adding user actions `zdmvaltgt` and `zdmvalsrc` with the `ADD USERACTION` command would look like the following.

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

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

In the above command, the scripts `useract.sh` and `useract1.sh`, specified in the `-actionscript` option, 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> $ZDM_HOME/bin/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> $ZDM_HOME/bin/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_zdm_service_node  
node2:node2:identity_file_path_available_on_zdm_service_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.
- [Terminate a Running Migration Job](#)
If you want to resubmit a database migration job for a specified database, you must first terminate 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 and target database servers are accessed with

a private key file, 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 `$ZDM_HOME/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. It is highly recommended that for each migration you run `migrate database` in evaluation mode first. This 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.

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

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.

```
su - zdmuser
```

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

```
zdmuser> $ZDM_HOME/bin/zdmcli migrate database -sourcedb
source_db_unique_name_value
-sourcenode source_database_server_name -srcroot
-targetnode target_database_server_name
-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> $ZDM_HOME/bin/zdmcli migrate database -sourcedb zdmsdb -
sourcenode ocidb1
-srcroot -targetnode ocidb1 -backupuser backup_user@example.com
-rsp /u01/app/zdmhome/rhp/zdm/template/zdm_template_zdmsdb.rsp -
tgtauth zdmauth
-tgtarg1 user:opc -tgtarg2 identity_file:/home/zdmuser/.ssh/
zdm_service_host.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> $ZDM_HOME/bin/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
-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> $ZDM_HOME/bin/zdmcli migrate database -sourcedb zdmsdb -
sourcenode ocidb1 -srcauth zdmauth
-srcarg1 user:opc -srcarg2 identity_file:/home/zdmuser/.ssh/
zdm_service_host.ppk
-srcarg3 sudo_location:/usr/bin/sudo -targetnode ocidb1 -backupuser
backup_user@example.com
-rsp /u01/app/zdmhome/rhp/zdm/template/zdm_template_zdmsdb.rsp -
tgtauth zdmauth -tgtarg1 user:opc
-tgtarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.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.

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.

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.

If you want to run the command without providing passwords at the command line, see [Provide Passwords Non-Interactively Using a Wallet](#).

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> $ZDM_HOME/bin/zdmcli migrate database -sourcedb
source_db_unique_name_value
-sourcenode source_database_server_name -srcroot
-targetnode target_database_server_name
-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> $ZDM_HOME/bin/zdmcli migrate database -sourcedb zdmsdb -
sourcenode ocidb1 -srcroot
-targetnode ocidb1 -backupuser backup_user@example.com -
rsp /u01/app/zdmhome/rhp/zdm/template/zdm_template_zdmsdb.rsp
-tgtauth zdmauth -tgtarg1 user:opc -tgtarg2 identity_file:/home/
zdmuser/.ssh/zdm_service_host.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> $ZDM_HOME/bin/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
-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> $ZDM_HOME/bin/zdmcli migrate database -sourcedb zdmsdb -
sourcenode ocidb1 -srcauth zdmauth
-srcarg1 user:opc -srcarg2 identity_file:/home/zdmuser/.ssh/
zdm_service_host.ppk
-srcarg3 sudo_location:/usr/bin/sudo -targetnode ocidb1 -backupuser
backup_user@example.com
-rsp /u01/app/zdmhome/rhp/zdm/template/zdm_template_zdmsdb.rsp -
tgtauth zdmauth -tgtarg1 user:opc
-tgtarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.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.

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.

If you want to run the command without providing passwords at the command line, see [Provide Passwords Non-Interactively Using a Wallet](#).



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> $ZDM_HOME/bin/zdmcli query job -jobid job-id
```

You can find the console output of the migration job in the file indicated (Result file path:) 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> $ZDM_HOME/bin/zdmcli migrate database -sourcedb zdmsdb -
sourcenode ocidb1 -srcauth zdmauth
-srcarg1 user:opc -srcarg2 identity_file:/home/zdmuser/.ssh/
zdm_service_host.ppk -srcarg3 sudo_location:/usr/bin/sudo
-targetnode ocidb1 -backupuser backup_user@example.com -rsp /u01/app/
zdmhome/rhp/zdm/template/zdm_template_zdmsdb.rsp
-tgtauth zdmauth -tgtarg1 user:opc -tgtarg2 identity_file:/home/
zdmuser/.ssh/zdm_service_host.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.

In the following example, if you specify `-pauseafter ZDM_SETUP_TGT`, the migration job will pause after completing the `ZDM_SETUP_TGT` phase.

```
zdmuser> $ZDM_HOME/bin/zdmcli migrate database -sourcedb zdmsdb -
sourcenode ocidb1
-srcauth zdmauth -srcarg1 user:opc
-srcarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk
-srcarg3 sudo_location:/usr/bin/sudo -targetnode ocidb1
-backupuser backup_user@example.com -rsp /u01/app/zdmhome/rhp/zdm/
template/zdm_template_zdmsdb.rsp -tgtauth zdmauth
-tgtarg1 user:opc -tgtarg2 identity_file:/home/zdmuser/.ssh/
zdm_service_host.ppk
-tgtarg3 sudo_location:/usr/bin/sudo -pauseafter ZDM_SETUP_TGT
```

Choosing a Migration Job Phase to Pause After

Choose a valid phase that is listed in the `migrate database ... -listphases` command output.

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

Pausing the migration job after `ZDM_SETUP_TGT` is recommended.

If you use `-pauseafter` at phase `ZDM_CONFIGURE_DG_SRC`, then 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.

Preserving Log Files During a Paused Migration Job

To prevent source and target database log files from getting cleaned up between pausing and resuming a migration job, log files are written to `$ORACLE_BASE/zdm/zdm_db_unique_name_zdm_job_id/zdm/log` in their respective source and target database servers.

Resuming a Migration Job

A paused job can be resumed any time by running the `ZDMCLI resume job` command, specifying the respective job ID.

```
zdmuser> $ZDM_HOME/bin/zdmcli resume job -jobid Job_ID  
[-pauseafter valid-phase]
```

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 `migrate database ... -listphases` command output.

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> $ZDM_HOME/bin/zdmcli resume job -jobid Job_ID
```

Terminate a Running Migration Job

If you want to resubmit a database migration job for a specified database, you must first terminate 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> $ZDM_HOME/bin/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. Skip this task if you set `TGT_SKIP_DATAPATCH=FALSE` in the response file and `datapatch` was run as part of the 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. Skip this task if you set `TGT_SKIP_DATAPATCH=FALSE` in the response file and datapatch was run as part of the migration job.

For example, if your source database is at Jan 2020 PSU/BP and the target is at April 2020 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 `YYYYMMDD_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 command `migrate database`. You can also find the log file location (Result file path) in the output of the ZDMCLI command `query job -jobid job-id`.

Determine which operational phase the migration job was in at the time of failure, and whether the phase belongs to the source (phase name contains `SRC`) or target (phase name contains `TGT`). Check the Zero Downtime Migration service host log at `$ZDM_BASE/crsdata/zdm_service_host/rhp/rhpserver.log.0`, and access the respective source or target server to check the log associated with the operational phase in `$ORACLE_BASE/zdm/zdm_db_unique_name_job-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_host/rhp/rhpserver.log.0`.

If a migration job fails, you can fix the cause of the failure and then re-run the job while monitoring the logs for progress.

A

Zero Downtime Migration Port Requirements

The ports required for communication between the Zero Downtime Migration service host, the source and target database servers, and Oracle Cloud Object Store Service are described in the following table.

Table A-1 Zero Downtime Migration Communication Ports

Initiator	Target	Protocol	Port	Purpose	Description
Zero Downtime Migration service host	Source and target database servers	TCP	22	SSH	Authentication -based operations to run Zero Downtime Migration operational phases. Source and target database servers should accept incoming connections from the Zero Downtime Migration service host.

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

Initiator	Target	Protocol	Port	Purpose	Description
Source database servers	Target database servers	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
Target database servers	Source database servers	TCP	1521	SQL*Net	<p>Should allow Oracle client connections to the database over Oracle's SQL*Net protocol. Allows redo log shipping if source 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 source database server then set SKIP_FALLBACK to TRUE in the response file to avoid this communication.</p> <p>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.</p>

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

Initiator	Target	Protocol	Port	Purpose	Description
Source database servers	Oracle Cloud Object Store Service	SSL	443	Database backup store. Create a backup of the source database to the specified Oracle Cloud Object store container.	If the chosen backup method uses Oracle Cloud Object Store Service as the backup medium, then access ports as documented Oracle Cloud Object Store Service applies.
Target database servers	Oracle Cloud Object Store Service	SSL	443	Database backup store. Restore backup of the source database from the specified Oracle Cloud Object store container to the target database.	If the chosen backup method uses 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

Provide Passwords Non-Interactively Using a Wallet

You can avoid entering passwords in the command line and run the `ZDMCLI MIGRATE DATABASE` command without user interaction.

Currently, whenever you submit the `$ZDM_HOME/bin/zdmcli migrate database` command, it prompts for the source database `SYS` password, Object Store user `swift` authentication token, and the source database Transparent Data Encryption (TDE) keystore password (if the wallet was configured as a `PASSWORD`-based TDE wallet). If you don't want to be required to enter the password at the command line, such as when you do automation using Rundeck, complete the following steps.

Run the following commands on the Zero Downtime Migration service host as Zero Downtime Migration software owner (for example, `zdmuser`).

1. Create an auto-login wallet for the source database `SYS` user.
 - a. Create a directory where you want to create and store the wallet.

```
zdmuser> mkdir sys_wallet_path
```

For example:

```
/u01/app/zdmhome> mkdir sysWallet
```

- b. Create a wallet.

```
zdmuser> $ZDM_HOME/bin/orapki wallet create -wallet  
sys_wallet_path  
-auto_login_only
```

For example

```
/u01/app/zdmhome> $ZDM_HOME/bin/orapki wallet create -wallet  
sysWallet  
-auto_login_only  
Oracle PKI Tool Release 19.0.0.0.0 - Production  
Version 19.4.0.0.0  
Copyright (c) 2004, 2019, Oracle and/or its affiliates. All  
rights reserved.
```

Operation is successfully completed.

c. Add a SYS user login credentials to wallet.

```
zdmuser> $ZDM_HOME/bin/mkstore -wrl sys_wallet_path
-createCredential store sysuser
```

At the prompt, enter the source database SYS password.

For example

```
/u01/app/zdmhome> $ZDM_HOME/bin/mkstore -wrl ./sysWallet
-createCredential store sysuser
Oracle Secret Store Tool Release 19.0.0.0.0 - Production
Version 19.4.0.0.0
Copyright (c) 2004, 2019, Oracle and/or its affiliates. All
rights reserved.
```

```
Your secret/Password is missing in the command line
Enter your secret/Password:
Re-enter your secret/Password:
```

d. Verify that the wallet files were created.

```
zdmuser> ls -l sys_wallet_path
```

For example

```
/u01/app/zdmhome> ls -l sysWallet/
total 4
-rw-----. 1 opc opc 581 Jun  2 08:00 cwallet.sso
-rw-----. 1 opc opc  0 Jun  2 08:00 cwallet.sso.lck
```

2. Create an auto-login wallet for the Object Store user.**a. Create a directory where you want to create and store the wallet.**

```
zdmuser> mkdir oss_wallet_path
```

For example

```
/u01/app/zdmhome> mkdir ossWallet
```

b. Create a wallet

```
zdmuser> $ZDM_HOME/bin/orapki wallet create -wallet
oss_wallet_path
-auto_login_only
```

For example

```
/u01/app/zdmhome> $ZDM_HOME/bin/orapki wallet create
-wallet ./ossWallet -auto_login_only
Oracle PKI Tool Release 19.0.0.0.0 -Production
Version 19.4.0.0.0
```

Copyright (c) 2004, 2019,
Oracle and/or its affiliates. All rights reserved.

Operation is successfully completed.

c. Add the Object Store user login credentials to the wallet.

```
zdmuser> $ZDM_HOME/bin/mkstore -wrl oss_wallet_path
-createCredential store ossuser
```

For the prompt,

- If the backup destination is Object Store (Bucket), then enter the user swift authentication token.
- If the backup destination is Storage Classic (Container), then enter your tenancy login password.

For example

```
/u01/app/zdmhome> $ZDM_HOME/bin/mkstore -wrl ./ossWallet
-createCredential store ossuser
Oracle Secret Store Tool Release 19.0.0.0.0 - Production
Version 19.4.0.0.0
Copyright (c) 2004, 2019, Oracle and/or its affiliates. All
rights reserved.
```

```
Your secret/Password is missing in the command line
Enter your secret/Password:
Re-enter your secret/Password:
```

d. Verify that the wallet files were created.

```
zdmuser> ls -l oss_wallet_path
```

For example

```
/u01/app/zdmhome> ls -l ./ossWallet
total 4
-rw-----. 1 opc opc 597 Jun  2 08:02 cwallet.sso
-rw-----. 1 opc opc  0 Jun  2 08:01 cwallet.sso.lck
```

3. Create an auto-login wallet for the source database TDE keystore.

a. Create a directory where you want to create and store the wallet.

```
zdmuser> mkdir tde_wallet_path
```

For example

```
/u01/app/zdmhome> mkdir tdeWallet
```

b. Create a wallet.

```
zdmuser> $ZDM_HOME/bin/orapki wallet create -wallet
tde_wallet_path
-auto_login_only
```

For example

```
/u01/app/zdmhome> $ZDM_HOME/bin/orapki wallet create -wallet ./
tdeWallet
-auto_login_only
Oracle PKI Tool Release 19.0.0.0.0 - Production
Version 19.4.0.0.0
Copyright (c) 2004, 2019, Oracle and/or its affiliates. All
rights reserved.
```

Operation is successfully completed.

c. Add the source database TDE keystore credentials to the wallet.

```
zdmuser> $ZDM_HOME/bin/mkstore -wrl tde_wallet_path
-createCredential store tdeuser
```

At the prompt, enter the TDE keystore password.

For example

```
/u01/app/zdmhome> $ZDM_HOME/bin/mkstore -wrl ./tdeWallet
-createCredential store tdeuser
Oracle Secret Store Tool Release 19.0.0.0.0 - Production
Version 19.4.0.0.0
Copyright (c) 2004, 2019, Oracle and/or its affiliates. All
rights reserved.
```

```
Your secret/Password is missing in the command line
Enter your secret/Password:
Re-enter your secret/Password:
```

d. Verify that the wallet files were created.

```
zdmuser> ls -l tde_wallet_path
```

For example

```
/u01/app/zdmhome> ls -l tdeWallet
total 4
-rw----- . 1 opc opc 581 Jun  2 08:06 cwallet.sso
-rw----- . 1 opc opc  0 Jun  2 08:04 cwallet.sso.lck
```

Setting Command Options to Access the Wallets

To specify wallet information in the ZDMCLI MIGRATE DATABASE command, set the `-sourcesyswallet`, `-osswallet`, and `-tdekeystorewallet` options as shown here.

```
zdmuser> $ZDM_HOME/bin/zdmcli migrate database
-sourcedb source_db_unique_name_value
-srcauth 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
-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 -sourcesyswallet sys_wallet_path
-osswallet oss_wallet_path -tdekeystorewallet tde_wallet_path
-eval
```

- `-sourcesyswallet sys_wallet_path` specifies the full path for the auto-login wallet file on the Zero Downtime Migration host containing the SYS password of the source database
- `-osswallet oss_wallet_path` specifies the full path for the auto-login wallet file on the Zero Downtime Migration host containing credentials for the Object Storage Sservice backup user
- `-tdekeystorewallet tde_wallet_path` specifies the full path for the auto-login wallet file on the Zero Downtime Migration host containing the TDE keystore password

Evaluation Mode Example

```
zdmuser> $ZDM_HOME/bin/zdmcli migrate database -sourcedb zdmfdb -
sourcenode ocicdb1
-srcauth zdmauth -srcarg1 user:opc
-srcarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk
-srcarg3 sudo_location:/usr/bin/sudo -targetnode ocicdb1
-backupuser backup_user@example.com
-rsp /u01/app/zdmhome/rhp/zdm/template/zdm_template_zdmfdb.rsp -tgtauth
zdmauth
-tgtarg1 user:opc -tgtarg2 identity_file:/home/zdmuser/.ssh/
zdm_service_host.ppk
-tgtarg3 sudo_location:/usr/bin/sudo -sourcesyswallet /u01/app/zdmhome/
sysWallet
-osswallet /u01/app/zdmhome/ossWallet -eval
```

Operation "zdmcli migrate database" scheduled with the job ID "1".

Migration Mode Example

```
zdmuser> $ZDM_HOME/bin/zdmcli migrate database -sourcedb zdmfdb -
sourcenode ocicdb1
-srcauth zdmauth -srcarg1 user:opc
-srcarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk
-srcarg3 sudo_location:/usr/bin/sudo -targetnode ocicdb1
-backupuser backup_user@example.com
```

```
-rsp /u01/app/zdmhome/rhp/zdm/template/zdm_template_zdmsdb.rsp -tgtauth  
zdmauth  
-tgtarg1 user:opc -tgtarg2 identity_file:/home/zdmuser/.ssh/  
zdm_service_host.ppk  
-tgtarg3 sudo_location:/usr/bin/sudo -sourcesyswallet /u01/app/zdmhome/  
sysWallet  
-osswallet /u01/app/zdmhome/osswallet
```

Operation "zdmcli migrate database" scheduled with the job ID "2".

D

Zero Downtime Migration Process Phases

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

Example D-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> $ZDM_HOME/bin/zdmcli migrate database -sourcedb zdmsdb
-sourcenode ocicdb1 -srcauth zdmauth -srcarg1 user:opc
-srcarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk
-srcarg3 sudo_location:/usr/bin/sudo -targetnode ocidb1
-backupuser backup_user@example.com
-rsp /u01/app/zdmhome/rhp/zdm/template/zdm_template_zdmsdb.rsp
-tgtauth zdmauth -tgtarg1 user:opc
-tgtarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk
-tgtarg3 sudo_location:/usr/bin/sudo -listphases
```

Table D-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_VALIDATE_SRC	Perform validations at the source
ZDM_VALIDATE_TGT	Perform validations 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_BACKUP_FULL_SRC	Perform full backup of the source database
ZDM_BACKUP_INCREMENTAL_SRC	Perform incremental backup of the source database
ZDM_DISCOVER_SRC	Perform database discovery at the source for setting up Data Guard
ZDM_COPYFILES	Copy Oracle password file and TDE wallets from source to target

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

Phase name	Description
ZDM_SETUP_TDE_TGT	Copy TDE wallet files from the source to the target keystore location
ZDM_OSS_RESTORE_TGT	Perform full database restore
ZDM_BACKUP_DIFFERENTIAL_SRC	Perform differential backup of the source database
ZDM_OSS_RECOVER_TGT	Perform incremental restore, recovery database, and opens database with reset logs
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_POST_DATABASE_OPEN_TGT	Perform activities after database is opened, such as restore pluggable database state, DBA directories, RMAN configuration
ZDM_DATAPATCH_TGT	Runs datapatch at the target
ZDM_SHUTDOWN_SRC	Shuts down source database at the end of the migration
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

E

Zero Downtime Migration Response File Parameters Reference

The following topics describe the Zero Downtime Migration response file parameters.

- **BACKUP_PATH**
BACKUP_PATH specifies a valid path accessible at the source and target for migration backup type.
- **DATAPATCH_WITH_ONE_INSTANCE_RUNNING**
DATAPATCH_WITH_ONE_INSTANCE_RUNNING specifies whether or not to stop all instances except one running on the target database server when the datapatch utility is run. When datapatch completes all of the stopped instances are started.
- **HOST**
HOST specifies the cloud storage REST endpoint URL to access Oracle Cloud Object Storage.
- **MAX_DATAPATCH_DURATION_MINS**
MAX_DATAPATCH_DURATION_MINS specifies a timeout value, in minutes, after which if the datapatch utility has failed to complete then the operation is stopped.
- **MIGRATION_METHOD**
MIGRATION_METHOD specifies whether the migration uses Oracle Data Guard (online) or backup and restore (offline) and which media is used for the source database backup.
- **OPC_CONTAINER**
OPC_CONTAINER specifies the Object Storage bucket (called the container on Oracle Cloud Infrastructure Classic), and is set to access Oracle Cloud Object Storage.
- **PLATFORM_TYPE**
PLATFORM_TYPE specifies the target database platform.
- **SHUTDOWN_SRC**
SHUTDOWN_SRC specifies whether or not to shut down the source database after migration completes.
- **SKIP_FALLBACK**
SKIP_FALLBACK specifies whether or not to ship redo logs from the primary (target) database to the standby (source) database, either voluntarily or because there is no connectivity between the target and source database servers.
- **SKIP_SRC_SERVICE_RETENTION**
SKIP_SRC_SERVICE_RETENTION specifies whether or not to retain the source database services and run them on the target database. This parameter is only effective for the BACKUP_RESTORE_OSS and BACKUP_RESTORE_NFS migration methods.
- **SRC_BASTION_HOST_IP**
SRC_BASTION_HOST_IP specifies the bastion host IP address, if you want to connect to the source database server using a bastion host.

- **SRC_BASTION_IDENTITY_FILE**
SRC_BASTION_IDENTITY_FILE specifies the bastion user, if you want to connect to the source database server using a bastion host.
- **SRC_BASTION_PORT**
SRC_BASTION_PORT specifies the bastion host port number, if you want to connect to the source database server using a bastion host.
- **SRC_BASTION_USER**
SRC_BASTION_USER specifies the bastion user, if you want to connect to the source database server using a bastion host.
- **SRC_CONFIG_LOCATION**
SRC_CONFIG_LOCATION specifies the SSH configuration file location on the Zero Downtime Migration service host (host where Zero Downtime Migration service is running).
- **SRC_HOST_IP**
SRC_HOST_IP specifies the bastion user, if you want to connect to the source database server using a bastion host.
- **SRC_HTTP_PROXY_PORT**
SRC_HTTP_PROXY_PORT specifies the HTTPS proxy port number on the source database server if an SSH connection needs to connect using a proxy.
- **SRC_HTTP_PROXY_URL**
SRC_HTTP_PROXY_URL specifies the HTTPS proxy URL on the source database server if an SSH connection needs to connect using a proxy.
- **SRC_OSS_PROXY_HOST**
SRC_OSS_PROXY_HOST specifies the Object Storage Service proxy host on the source database server if a proxy is needed for connecting to the Object Store.
- **SRC_OSS_PROXY_PORT**
SRC_OSS_PROXY_PORT specifies the Object Storage Service proxy port number on the source database server if a proxy is needed for connecting to the Object Store.
- **SRC_RMAN_CHANNELS**
SRC_RMAN_CHANNELS specifies the number of RMAN channels to be allocated at the source database server for performing RMAN backups.
- **SRC_SSH_RETRY_TIMEOUT**
SRC_SSH_RETRY_TIMEOUT specifies a timeout value, in minutes, after which Zero Downtime Migration stops attempting SSH connections after an initial failure to connect.
- **SRC_TIMEZONE**
SRC_TIMEZONE specifies the source database server time zone, which is needed for SIDB case when there is no Grid Infrastructure configured.
- **SRC_ZDLRA_WALLET_LOC**
SRC_ZDLRA_WALLET_LOC specifies the path of the Zero Data Loss Recovery Appliance wallet on the source database server.
- **TGT_BASTION_HOST_IP**
TGT_BASTION_HOST_IP specifies the bastion host IP address, if you want to connect to the target database server using a bastion host.
- **TGT_BASTION_IDENTITY_FILE**
TGT_BASTION_IDENTITY_FILE specifies the bastion user, if you want to connect to the target database server using a bastion host.

- **TGT_BASTION_PORT**
TGT_BASTION_PORT specifies the bastion host port number, if you want to connect to the target database server using a bastion host.
- **TGT_BASTION_USER**
TGT_BASTION_USER specifies the bastion user, if you want to connect to the target database server using a bastion host.
- **TGT_CONFIG_LOCATION**
TGT_CONFIG_LOCATION specifies the SSH configuration file location on the Zero Downtime Migration service host (host where Zero Downtime Migration service is running).
- **TGT_DATAACFS**
TGT_DATAACFS specifies the location for the data files ACFS volume (*data*) on the target database. Use only if required to override the values discovered automatically by Zero Downtime Migration.
- **TGT_DATADG**
TGT_DATADG specifies the location for the data files ASM disk group (*data*) on the target database. Use only if required to override the values discovered automatically by Zero Downtime Migration.
- **TGT_DB_UNIQUE_NAME**
TGT_DB_UNIQUE_NAME is used by Zero Downtime Migration to identify the target database.
- **TGT_HOST_IP**
TGT_HOST_IP specifies the bastion user, if you want to connect to the target database server using a bastion host.
- **TGT_HTTP_PROXY_PORT**
TGT_HTTP_PROXY_PORT specifies the HTTPS proxy port if an SSH connection needs to use a proxy to connect to the target database server.
- **TGT_HTTP_PROXY_URL**
TGT_HTTP_PROXY_URL specifies the HTTPS proxy URL if an SSH connection needs to use a proxy to connect to the target database server.
- **TGT_OSS_PROXY_HOST**
TGT_OSS_PROXY_HOST specifies the Object Storage Service proxy host on the target database server if a proxy is needed for connecting to the Object Store.
- **TGT_OSS_PROXY_PORT**
TGT_OSS_PROXY_PORT specifies the Object Storage Service proxy port number on the target database server if a proxy is needed for connecting to the Object Store.
- **TGT_RECOACFS**
TGT_RECOACFS specifies the location for the fast recovery area ACFS volume (*reco*) on the target database. Use only if required to override the values discovered automatically by Zero Downtime Migration.
- **TGT_RECODG**
TGT_RECODG specifies the location for the fast recovery area ASM disk group (*reco*) on the target database. Use only if required to override the values discovered automatically by Zero Downtime Migration.
- **TGT_REDOACFS**
TGT_REDOACFS specifies the location for redo log files ACFS volume (*redo*) on the target database. Use only if required to override the values discovered automatically by Zero Downtime Migration.

- **TGT_REDODG**
TGT_REDODG specifies the location for redo log files ASM disk group (`redo`) on the target database. Use only if required to override the values discovered automatically by Zero Downtime Migration.
- **TGT_RMAN_CHANNELS**
TGT_RMAN_CHANNELS specifies the number of RMAN channels to be allocated at the target database server for performing RMAN restore.
- **TGT_SKIP_DATAPATCH**
TGT_SKIP_DATAPATCH specifies whether or not Zero Downtime Migration runs the datapatch utility on the target database as part of the post-migration tasks.
- **TGT_SSH_RETRY_TIMEOUT**
TGT_SSH_RETRY_TIMEOUT specifies the number of minutes during which retries are attempted after SSH connection failures. Retries stop when the timeout value has elapsed.
- **TGT_SSH_TUNNEL_PORT**
TGT_SSH_TUNNEL_PORT specifies the forwarding port on the source database server where the SSH tunnel to the target database server for SQL*Net connection is set up.
- **TGT_ZDLRA_WALLET_LOC**
TGT_ZDLRA_WALLET_LOC specifies the path of the Zero Data Loss Recovery Appliance wallet on the target database server.
- **ZDLRA_CRED_ALIAS**
ZDLRA_CRED_ALIAS specifies the Zero Data Loss Recovery Appliance wallet credential alias.
- **ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL**
ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_BACKUP_DIFFERENTIAL_SRC migration job phase.
- **ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL**
ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_BACKUP_FULL_SRC migration job phase.
- **ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL**
ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_BACKUP_INCREMENTAL_SRC migration job phase.
- **ZDM_BACKUP_RETENTION_WINDOW**
ZDM_BACKUP_RETENTION_WINDOW specifies the number of days after which backups created by Zero Downtime Migration become obsolete.
- **ZDM_CLONE_TGT_MONITORING_INTERVAL**
ZDM_CLONE_TGT_MONITORING_INTERVAL specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_CLONE_TGT migration job phase.
- **ZDM_CURL_LOCATION**
ZDM_CURL_LOCATION specifies a custom location for the CURL binary on the source.

- **ZDM_LOG_OSS_PAR_URL**
ZDM_LOG_OSS_PAR_URL specifies the pre-authenticated URL to use when uploading logs to Object Storage Service. The logs capture the current migration job phase and the execution status of the phase.
- **ZDM_OPC_RETRY_COUNT**
ZDM_OPC_RETRY_COUNT specifies the number of retry attempts that will be made after an initial Object Store connection failure.
- **ZDM_OPC_RETRY_WAIT_TIME**
ZDM_OPC_RETRY_WAIT_TIME specifies the number of seconds to wait after an Object Store connection failure before attempting to retry the connection.
- **ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL**
ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_OSS_RECOVER_TGT migration job phase.
- **ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL**
ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_OSS_RESTORE_TGT migration job phase.
- **ZDM_SRC_TNS_ADMIN**
ZDM_SRC_TNS_ADMIN specifies the custom location for TNS_ADMIN on the source database server when there is no Oracle Grid Infrastructure. If a Grid Infrastructure exists, then the TNS_ADMIN property must be set in the CRS resource attribute environment of the database resource.
- **ZDM_USE_EXISTING_UNDO_SIZE**
ZDM_USE_EXISTING_UNDO_SIZE specifies whether Zero Downtime Migration should use the existing undo tablespace size when creating a new undo tablespace, if required.

BACKUP_PATH

BACKUP_PATH specifies a valid path accessible at the source and target for migration backup type.

Property	Description
Syntax	BACKUP_PATH = {STORAGEPATH EXTBACKUP}
Default value	There is no default value.
Range of values	STORAGEPATH - NFS backup location EXTBACKUP - external backup location Leave this parameter value blank for other migration backup types

DATAPATCH_WITH_ONE_INSTANCE_RUNNING

DATAPATCH_WITH_ONE_INSTANCE_RUNNING specifies whether or not to stop all instances except one running on the target database server when the datapatch utility is run. When datapatch completes all of the stopped instances are started.

Property	Description
Syntax	DATAPATCH_WITH_ONE_INSTANCE_RUNNING ={TRUE FALSE}
Default value	FALSE
Range of values	TRUE - Stops all instances except one running on the target database server when running the datapatch utility. FALSE - Does not stop all instances except one running on the target database server when running the datapatch utility.

HOST

HOST specifies the cloud storage REST endpoint URL to access Oracle Cloud Object Storage.

Property	Description
Syntax	HOST = <i>rest_endpoint_url</i>
Default value	There is no default value.
Range of values	For Oracle Cloud Infrastructure storage the typical value format is <i>https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/ObjectStorageNamespace</i> For Oracle Cloud Infrastructure Classic storage the typical value format is <i>https://acme.storage.oraclecloud.com/v1/Storage-tenancy name</i>

Usage Notes

To access Oracle Cloud Object Storage, you must set both the HOST and OPC_CONTAINER parameters.

MAX_DATAPATCH_DURATION_MINS

MAX_DATAPATCH_DURATION_MINS specifies a timeout value, in minutes, after which if the datapatch utility has failed to complete then the operation is stopped.

Property	Description
Syntax	MAX_DATAPATCH_DURATION_MINS = <i>minutes</i>
Default value	There is no default value. Zero Downtime Migration waits until datapatch completes by default.

MIGRATION_METHOD

MIGRATION_METHOD specifies whether the migration uses Oracle Data Guard (online) or backup and restore (offline) and which media is used for the source database backup.

Property	Description
Syntax	MIGRATION_METHOD = {DG_OSS DG_STORAGEPATH DG_EXTBACKUP BACKUP_RESTORE_OSS BACKUP_RESTORE_NFS}
Default value	DG_OSS
Range of values	<p>DG_OSS - Oracle Data Guard migration using Object Storage Service for standby initialization</p> <p>DG_STORAGEPATH - Oracle Data Guard migration using backup location such as NFS</p> <p>DG_EXTBACKUP - Oracle Data Guard migration with existing backup in external location</p> <p>DG_ZDLRA - Oracle Data Guard migration using Zero Data Loss Recovery Appliance for standby initialization</p> <p>BACKUP_RESTORE_OSS - Offline migration using backup and restore with Object Storage Service as backup. SQL*Net connectivity is not needed between source and target</p> <p>BACKUP_RESTORE_NFS - Offline migration using backup and restore with NFS as backup. SQL*Net connectivity is not needed between source and target</p>

OPC_CONTAINER

OPC_CONTAINER specifies the Object Storage bucket (called the container on Oracle Cloud Infrastructure Classic), and is set to access Oracle Cloud Object Storage.

Property	Description
Syntax	OPC_CONTAINER = <i>bucket</i>
Default value	There is no default value.

Usage Notes

To access Oracle Cloud Object Storage, you must set both the HOST and OPC_CONTAINER parameters.

The bucket is also referred to as a container for Oracle Cloud Infrastructure Classic storage.

PLATFORM_TYPE

PLATFORM_TYPE specifies the target database platform.

Property	Description
Syntax	PLATFORM_TYPE = {VMDB EXACC EXACS}
Default value	VMDB
Range of values	VMDB - Oracle Cloud Infrastructure EXACC - Exadata Cloud at Customer EXACS - Exadata Cloud Service

SHUTDOWN_SRC

SHUTDOWN_SRC specifies whether or not to shut down the source database after migration completes.

Property	Description
Syntax	SHUTDOWN_SRC = {TRUE FALSE}
Default value	FALSE
Range of values	TRUE - Shut down the source database after migration completes. FALSE - Does not shut down the source database after migration completes.

SKIP_FALLBACK

SKIP_FALLBACK specifies whether or not to ship redo logs from the primary (target) database to the standby (source) database, either voluntarily or because there is no connectivity between the target and source database servers.

Property	Description
Syntax	SKIP_FALLBACK = {TRUE FALSE}
Default value	FALSE
Range of values	TRUE - do not ship redo logs from the primary (target) database to the standby (source) database. FALSE - ship redo logs from the primary (target) database to the standby (source) database.

SKIP_SRC_SERVICE_RETENTION

SKIP_SRC_SERVICE_RETENTION specifies whether or not to retain the source database services and run them on the target database. This parameter is only effective for the BACKUP_RESTORE_OSS and BACKUP_RESTORE_NFS migration methods.

Property	Description
Syntax	SKIP_SRC_SERVICE_RETENTION = {TRUE FALSE}
Default value	FALSE
Range of values	TRUE - Do not retain the source database services. FALSE - Retain the source database services.

SRC_BASTION_HOST_IP

SRC_BASTION_HOST_IP specifies the bastion host IP address, if you want to connect to the source database server using a bastion host.

Property	Description
Syntax	SRC_BASTION_HOST_IP = <i>IP_address</i>
Default value	There is no default value.

Usage Notes

If you want to connect to the source database server using a bastion host, values for the bastion host IP address parameter, SRC_BASTION_HOST_IP, and the source database host IP address parameter, SRC_HOST_IP, are required in the Zero Downtime Migration response file.

If you do not want to use the default value, set the following parameters for bastion host connection.

SRC_BASTION_PORT - The port number defaults to 22 if not specified.

SRC_BASTION_USER - The bastion host source user is only required if the user specified for the source `zdmauth` plug-in is different from the user of the source bastion host. The bastion user defaults to the user specified for the source `zdmauth` plug-in if the parameter is not specified.

SRC_BASTION_IDENTITY_FILE - If not specified, the value defaults to the value specified for the `identity_file` argument of the source `zdmauth` plug-in.

SRC_BASTION_IDENTITY_FILE

SRC_BASTION_IDENTITY_FILE specifies the bastion user, if you want to connect to the source database server using a bastion host.

Property	Description
Syntax	SRC_BASTION_IDENTITY_FILE = <i>identity_file</i>
Default value	If not specified, the value defaults to the value specified for the <i>identity_file</i> argument of the source <i>zdmauth</i> plug-in.

Usage Notes

If you want to connect to the source database server using a bastion host, values for the bastion host IP address parameter, SRC_BASTION_HOST_IP, and the source database server IP address parameter, SRC_HOST_IP, are required in the Zero Downtime Migration response file.

If you do not want to use the default value, set the following parameters for bastion host connection.

SRC_BASTION_PORT - The port number defaults to 22 if not specified.

SRC_BASTION_USER - The bastion host source user is only required if the user specified for the source *zdmauth* plug-in is different from the user of the source bastion host. The bastion user defaults to the user specified for the source *zdmauth* plug-in if the parameter is not specified.

SRC_BASTION_IDENTITY_FILE - If not specified, the value defaults to the value specified for the *identity_file* argument of the source *zdmauth* plug-in.

SRC_BASTION_PORT

SRC_BASTION_PORT specifies the bastion host port number, if you want to connect to the source database server using a bastion host.

Property	Description
Syntax	SRC_BASTION_PORT = <i>port_number</i>
Default value	22

Usage Notes

If you want to connect to the source database server using a bastion host, values for the bastion host IP address parameter, SRC_BASTION_HOST_IP, and the source database server IP address parameter, SRC_HOST_IP, are required in the Zero Downtime Migration response file.

If you do not want to use the default value, set the following parameters for bastion host connection.

SRC_BASTION_PORT - The port number defaults to 22 if not specified.

SRC_BASTION_USER - The bastion host source user is only required if the user specified for the source *zdmauth* plug-in is different from the user of the source bastion host. The bastion user defaults to the user specified for the source *zdmauth* plug-in if the parameter is not specified.

SRC_BASTION_IDENTITY_FILE - If not specified, the value defaults to the value specified for the `identity_file` argument of the source `zdmauth` plug-in.

SRC_BASTION_USER

SRC_BASTION_USER specifies the bastion user, if you want to connect to the source database server using a bastion host.

Property	Description
Syntax	SRC_BASTION_USER = <i>bastion_user</i>
Default value	The bastion user defaults to the user specified for the source <code>zdmauth</code> plug-in if the parameter is not specified.

Usage Notes

If you want to connect to the source database server using a bastion host, values for the bastion host IP address parameter, SRC_BASTION_HOST_IP, and the source database server IP address parameter, SRC_HOST_IP, are required in the Zero Downtime Migration response file.

If you do not want to use the default value, set the following parameters for bastion host connection.

SRC_BASTION_PORT - The port number defaults to 22 if not specified.

SRC_BASTION_USER - The bastion host source user is only required if the user specified for the source `zdmauth` plug-in is different from the user of the source bastion host. The bastion user defaults to the user specified for the source `zdmauth` plug-in if the parameter is not specified.

SRC_BASTION_IDENTITY_FILE - If not specified, the value defaults to the value specified for the `identity_file` argument of the source `zdmauth` plug-in.

SRC_CONFIG_LOCATION

SRC_CONFIG_LOCATION specifies the SSH configuration file location on the Zero Downtime Migration service host (host where Zero Downtime Migration service is running).

Property	Description
Syntax	SRC_CONFIG_LOCATION = <i>SSH_config_file_path</i>
Default value	<i>User_home/.ssh/config</i>

Usage Notes

Set SRC_CONFIG_LOCATION to the full path of the SSH configuration file location on the Zero Downtime Migration service host, for example, `/home/crsuser/.ssh/config`.

SRC_HOST_IP

SRC_HOST_IP specifies the bastion user, if you want to connect to the source database server using a bastion host.

Property	Description
Syntax	SRC_HOST_IP = <i>IP_address</i>
Default value	There is no default value.

Usage Notes

If you want to connect to the source database server using a bastion host, values for the bastion host IP address parameter, SRC_BASTION_HOST_IP, and the source database server IP address parameter, SRC_HOST_IP, are required in the Zero Downtime Migration response file.

If you do not want to use the default value, set the following parameters for bastion host connection.

SRC_BASTION_PORT - The port number defaults to 22 if not specified.

SRC_BASTION_USER - The bastion host source user is only required if the user specified for the source `zdmauth` plug-in is different from the user of the source bastion host. The bastion user defaults to the user specified for the source `zdmauth` plug-in if the parameter is not specified.

SRC_BASTION_IDENTITY_FILE - If not specified, the value defaults to the value specified for the `identity_file` argument of the source `zdmauth` plug-in.

SRC_HTTP_PROXY_PORT

SRC_HTTP_PROXY_PORT specifies the HTTPS proxy port number on the source database server if an SSH connection needs to connect using a proxy.

Property	Description
Syntax	SRC_HTTP_PROXY_PORT = <i>https_proxy_port_number</i>
Default value	There is no default value.

SRC_HTTP_PROXY_URL

SRC_HTTP_PROXY_URL specifies the HTTPS proxy URL on the source database server if an SSH connection needs to connect using a proxy.

Property	Description
Syntax	SRC_HTTP_PROXY_URL = <i>https_proxy_url</i>
Default value	There is no default value.

SRC_OSS_PROXY_HOST

`SRC_OSS_PROXY_HOST` specifies the Object Storage Service proxy host on the source database server if a proxy is needed for connecting to the Object Store.

Property	Description
Syntax	<code>SRC_OSS_PROXY_HOST = oss_proxy_host</code>
Default value	There is no default value.

Usage Notes

Set both the `SRC_OSS_PROXY_HOST` and `SRC_OSS_PROXY_PORT` parameters if a proxy is needed for connecting to the Object Store.

SRC_OSS_PROXY_PORT

`SRC_OSS_PROXY_PORT` specifies the Object Storage Service proxy port number on the source database server if a proxy is needed for connecting to the Object Store.

Property	Description
Syntax	<code>SRC_OSS_PROXY_PORT = oss_proxy_port_number</code>
Default value	There is no default value.

Usage Notes

Set both the `SRC_OSS_PROXY_HOST` and `SRC_OSS_PROXY_PORT` parameters if a proxy is needed for connecting to the Object Store.

SRC_RMAN_CHANNELS

`SRC_RMAN_CHANNELS` specifies the number of RMAN channels to be allocated at the source database server for performing RMAN backups.

Property	Description
Syntax	<code>SRC_RMAN_CHANNELS = number_of_channels</code>
Default value	10

SRC_SSH_RETRY_TIMEOUT

`SRC_SSH_RETRY_TIMEOUT` specifies a timeout value, in minutes, after which Zero Downtime Migration stops attempting SSH connections after an initial failure to connect.

Property	Description
Syntax	SRC_SSH_RETRY_TIMEOUT = <i>number_of_minutes</i>
Default value	There is no default value.

SRC_TIMEZONE

SRC_TIMEZONE specifies the source database server time zone, which is needed for SIDB case when there is no Grid Infrastructure configured.

Property	Description
Syntax	SRC_TIMEZONE = <i>source_db_time_zone</i>
Default value	There is no default value.

SRC_ZDLRA_WALLET_LOC

SRC_ZDLRA_WALLET_LOC specifies the path of the Zero Data Loss Recovery Appliance wallet on the source database server.

Property	Description
Syntax	SRC_ZDLRA_WALLET_LOC = <i>source_zdlra_wallet_location</i> The expected format for the location is /u02/app/oracle/product/12.1.0/dbhome_3/dbs/zdlra
Default value	There is no default value.

Usage Notes

When using Zero Data Loss Recovery Appliance as the migration backup medium, you must set the following parameters.

SRC_ZDLRA_WALLET_LOC

TGT_ZDLRA_WALLET_LOC

ZDLRA_CRED_ALIAS

TGT_BASTION_HOST_IP

TGT_BASTION_HOST_IP specifies the bastion host IP address, if you want to connect to the target database server using a bastion host.

Property	Description
Syntax	TGT_BASTION_HOST_IP = <i>bastion_ip_address</i>
Default value	There is no default value.

Usage Notes

If you want to connect to the target database server using a bastion host, you are required to configure values for the bastion host IP address parameter, `TGT_BASTION_HOST_IP`, and the target database server IP address parameter, `TGT_HOST_IP`, in the Zero Downtime Migration response file.

If you do not want to use the default values for the remaining bastion connection parameters, set the following parameters to configure the bastion host connection.

`TGT_BASTION_PORT` - The port number defaults to 22 if not specified.

`TGT_BASTION_USER` - The bastion host target user is only required if the user specified for the target `zdmauth` plug-in is different from the user of the target bastion host. The bastion user defaults to the user specified for the target `zdmauth` plug-in if the parameter is not specified.

`TGT_BASTION_IDENTITY_FILE` - If not specified, the value defaults to the value specified for the `identity_file` argument of the target `zdmauth` plug-in.

TGT_BASTION_IDENTITY_FILE

`TGT_BASTION_IDENTITY_FILE` specifies the bastion user, if you want to connect to the target database server using a bastion host.

Property	Description
Syntax	<code>TGT_BASTION_IDENTITY_FILE = <i>identity_file</i></code>
Default value	If not specified, the value defaults to the value specified for the <code>identity_file</code> argument of the target <code>zdmauth</code> plug-in.

Usage Notes

If you want to connect to the target database server using a bastion host, you are required to configure values for the bastion host IP address parameter, `TGT_BASTION_HOST_IP`, and the target database server IP address parameter, `TGT_HOST_IP`, in the Zero Downtime Migration response file.

If you do not want to use the default values for the remaining bastion connection parameters, set the following parameters to configure the bastion host connection.

`TGT_BASTION_PORT` - The port number defaults to 22 if not specified.

`TGT_BASTION_USER` - The bastion host target user is only required if the user specified for the target `zdmauth` plug-in is different from the user of the target bastion host. The bastion user defaults to the user specified for the target `zdmauth` plug-in if the parameter is not specified.

`TGT_BASTION_IDENTITY_FILE` - If not specified, the value defaults to the value specified for the `identity_file` argument of the target `zdmauth` plug-in.

TGT_BASTION_PORT

TGT_BASTION_PORT specifies the bastion host port number, if you want to connect to the target database server using a bastion host.

Property	Description
Syntax	TGT_BASTION_PORT = <i>port_number</i>
Default value	22

Usage Notes

If you want to connect to the target database server using a bastion host, you are required to configure values for the bastion host IP address parameter, TGT_BASTION_HOST_IP, and the target database server IP address parameter, TGT_HOST_IP, in the Zero Downtime Migration response file.

If you do not want to use the default values for the remaining bastion connection parameters, set the following parameters to configure the bastion host connection.

TGT_BASTION_PORT - The port number defaults to 22 if not specified.

TGT_BASTION_USER - The bastion host target user is only required if the user specified for the target `zdmauth` plug-in is different from the user of the target bastion host. The bastion user defaults to the user specified for the target `zdmauth` plug-in if the parameter is not specified.

TGT_BASTION_IDENTITY_FILE - If not specified, the value defaults to the value specified for the `identity_file` argument of the target `zdmauth` plug-in.

TGT_BASTION_USER

TGT_BASTION_USER specifies the bastion user, if you want to connect to the target database server using a bastion host.

Property	Description
Syntax	TGT_BASTION_USER = <i>bastion_user</i>
Default value	The bastion user defaults to the user specified for the target <code>zdmauth</code> plug-in if the parameter is not specified.

Usage Notes

If you want to connect to the target database server using a bastion host, you are required to configure values for the bastion host IP address parameter, TGT_BASTION_HOST_IP, and the target database server IP address parameter, TGT_HOST_IP, in the Zero Downtime Migration response file.

If you do not want to use the default values for the remaining bastion connection parameters, set the following parameters to configure the bastion host connection.

TGT_BASTION_PORT - The port number defaults to 22 if not specified.

TGT_BASTION_USER - The bastion host target user is only required if the user specified for the target `zdmauth` plug-in is different from the user of the target bastion host. The bastion user defaults to the user specified for the target `zdmauth` plug-in if the parameter is not specified.

TGT_BASTION_IDENTITY_FILE - If not specified, the value defaults to the value specified for the `identity_file` argument of the target `zdmauth` plug-in.

TGT_CONFIG_LOCATION

TGT_CONFIG_LOCATION specifies the SSH configuration file location on the Zero Downtime Migration service host (host where Zero Downtime Migration service is running).

Property	Description
Syntax	TGT_CONFIG_LOCATION = <i>SSH_config_file_path</i>
Default value	<i>User_home/.ssh/config</i>

Usage Notes

Set TGT_CONFIG_LOCATION to the full path of the SSH configuration file location on the Zero Downtime Migration service host, for example, `/home/crsuser/.ssh/config`.

TGT_DATAACFS

TGT_DATAACFS specifies the location for the data files ACFS volume (`data`) on the target database. Use only if required to override the values discovered automatically by Zero Downtime Migration.

Property	Description
Syntax	TGT_DATAACFS = <i>data_location</i>
Default value	There is no default value.

Usage Notes

Zero Downtime Migration discovers the location for ASM and ACFS data, `redo`, and `reco` storage volumes from the specified target database, making these target database storage properties optional.

If you need to override the values automatically discovered by Zero Downtime Migration, then you can set the following parameters. For example, `TGT_DATADG=+DATAAC3`

For ASM use these parameters

TGT_DATADG

TGT_REDODG

TGT_RECODG

For ACFS use these parameters

TGT_DATAACFS

TGT_REDOACFS

TGT_RECOACFS

TGT_DATADG

TGT_DATADG specifies the location for the data files ASM disk group (*data*) on the target database. Use only if required to override the values discovered automatically by Zero Downtime Migration.

Property	Description
Syntax	TGT_DATADG = <i>data_location</i>
Default value	There is no default value.

Usage Notes

Zero Downtime Migration discovers the location for ASM and ACFS *data*, *redo*, and *reco* storage volumes from the specified target database, making these target database storage properties optional.

If you need to override the values automatically discovered by Zero Downtime Migration, then you can set the following parameters. For example, TGT_DATADG=+DATA3

For ASM use these parameters

TGT_DATADG

TGT_REDO3

TGT_RECO3

For ACFS use these parameters

TGT_DATAACFS

TGT_REDOACFS

TGT_RECOACFS

TGT_DB_UNIQUE_NAME

TGT_DB_UNIQUE_NAME is used by Zero Downtime Migration to identify the target database.

Property	Description
Syntax	TGT_DB_UNIQUE_NAME = <i>value of target database DB_UNIQUE_NAME</i>
Default value	

Usage Notes

Set `TGT_DB_UNIQUE_NAME` to the target database `DB_UNIQUE_NAME` value.

If the target database is Oracle Cloud Infrastructure, Exadata Cloud Service, or Exadata Cloud at Customer, 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.

TGT_HOST_IP

`TGT_HOST_IP` specifies the bastion user, if you want to connect to the target database server using a bastion host.

Property	Description
Syntax	<code>TGT_HOST_IP = IP_address</code>
Default value	There is no default value.

Usage Notes

If you want to connect to the target database server using a bastion host, you are required to configure values for the bastion host IP address parameter, `TGT_BASTION_HOST_IP`, and the target database server IP address parameter, `TGT_HOST_IP`, in the Zero Downtime Migration response file.

If you do not want to use the default values for the remaining bastion connection parameters, set the following parameters to configure the bastion host connection.

`TGT_BASTION_PORT` - The port number defaults to 22 if not specified.

`TGT_BASTION_USER` - The bastion host target user is only required if the user specified for the target `zdmauth` plug-in is different from the user of the target bastion host. The bastion user defaults to the user specified for the target `zdmauth` plug-in if the parameter is not specified.

`TGT_BASTION_IDENTITY_FILE` - If not specified, the value defaults to the value specified for the `identity_file` argument of the target `zdmauth` plug-in.

TGT_HTTP_PROXY_PORT

`TGT_HTTP_PROXY_PORT` specifies the HTTPS proxy port if an SSH connection needs to use a proxy to connect to the target database server.

Property	Description
Syntax	<code>TGT_HTTP_PROXY_PORT = https_proxy_port_number</code>
Default value	There is no default value.

Usage Notes

Set both the `TGT_HTTP_PROXY_URL` and `TGT_HTTP_PROXY_PORT` parameters if the SSH connection needs to use an HTTPS proxy to connect to the target database server.

TGT_HTTP_PROXY_URL

TGT_HTTP_PROXY_URL specifies the HTTPS proxy URL if an SSH connection needs to use a proxy to connect to the target database server.

Property	Description
Syntax	TGT_HTTP_PROXY_URL = <i>https_proxy_url</i>
Default value	There is no default value.

Usage Notes

Set both the TGT_HTTP_PROXY_URL and TGT_HTTP_PROXY_PORT parameters if the SSH connection needs to use an HTTPS proxy to connect to the target database server.

TGT_OSS_PROXY_HOST

TGT_OSS_PROXY_HOST specifies the Object Storage Service proxy host on the target database server if a proxy is needed for connecting to the Object Store.

Property	Description
Syntax	TGT_OSS_PROXY_HOST = <i>oss_proxy_host</i>
Default value	There is no default value.

Usage Notes

Set both the TGT_OSS_PROXY_HOST and TGT_OSS_PROXY_PORT parameters if a proxy is needed for connecting to the Object Store.

TGT_OSS_PROXY_PORT

TGT_OSS_PROXY_PORT specifies the Object Storage Service proxy port number on the target database server if a proxy is needed for connecting to the Object Store.

Property	Description
Syntax	TGT_OSS_PROXY_PORT = <i>oss_proxy_port_number</i>
Default value	There is no default value.

Usage Notes

Set both the TGT_OSS_PROXY_HOST and TGT_OSS_PROXY_PORT parameters if a proxy is needed for connecting to the Object Store.

TGT_RECOACFS

TGT_RECOACFS specifies the location for the fast recovery area ACFS volume (*reco*) on the target database. Use only if required to override the values discovered automatically by Zero Downtime Migration.

Property	Description
Syntax	TGT_RECOACFS = <i>reco_location</i>
Default value	There is no default value.

Usage Notes

Zero Downtime Migration discovers the location for ASM and ACFS data, redo, and *reco* storage volumes from the specified target database, making these target database storage properties optional.

If you need to override the values automatically discovered by Zero Downtime Migration, then you can set the following parameters. For example, TGT_DATADG=+DATA3

For ASM use these parameters

TGT_DATADG

TGT_REDODG

TGT_RECODG

For ACFS use these parameters

TGT_DATAACFS

TGT_REDOACFS

TGT_RECOACFS

TGT_RECODG

TGT_RECODG specifies the location for the fast recovery area ASM disk group (*reco*) on the target database. Use only if required to override the values discovered automatically by Zero Downtime Migration.

Property	Description
Syntax	TGT_RECODG = <i>reco_location</i>
Default value	There is no default value.

Usage Notes

Zero Downtime Migration discovers the location for ASM and ACFS data, redo, and *reco* storage volumes from the specified target database, making these target database storage properties optional.

If you need to override the values automatically discovered by Zero Downtime Migration, then you can set the following parameters. For example, `TGT_DATADG=+DATA3`

For ASM use these parameters

`TGT_DATADG`

`TGT_REDODG`

`TGT_RECODG`

For ACFS use these parameters

`TGT_DATAACFS`

`TGT_REDOACFS`

`TGT_RECOACFS`

TGT_REDOACFS

`TGT_REDOACFS` specifies the location for redo log files ACFS volume (`redo`) on the target database. Use only if required to override the values discovered automatically by Zero Downtime Migration.

Property	Description
Syntax	<code>TGT_REDOACFS = redo_location</code>
Default value	There is no default value.

Usage Notes

Zero Downtime Migration discovers the location for ASM and ACFS `data`, `redo`, and `reco` storage volumes from the specified target database, making these target database storage properties optional.

If you need to override the values automatically discovered by Zero Downtime Migration, then you can set the following parameters. For example, `TGT_DATADG=+DATA3`

For ASM use these parameters

`TGT_DATADG`

`TGT_REDODG`

`TGT_RECODG`

For ACFS use these parameters

`TGT_DATAACFS`

`TGT_REDOACFS`

`TGT_RECOACFS`

TGT_REDODG

TGT_REDODG specifies the location for redo log files ASM disk group (*redo*) on the target database. Use only if required to override the values discovered automatically by Zero Downtime Migration.

Property	Description
Syntax	TGT_REDODG = <i>redo_location</i>
Default value	There is no default value.

Usage Notes

Zero Downtime Migration discovers the location for ASM and ACFS *data*, *redo*, and *reco* storage volumes from the specified target database, making these target database storage properties optional.

If you need to override the values automatically discovered by Zero Downtime Migration, then you can set the following parameters. For example, TGT_DATADG=+DATA3

For ASM use these parameters

TGT_DATADG

TGT_REDODG

TGT_RECODG

For ACFS use these parameters

TGT_DATAACFS

TGT_REDOACFS

TGT_RECOACFS

TGT_RMAN_CHANNELS

TGT_RMAN_CHANNELS specifies the number of RMAN channels to be allocated at the target database server for performing RMAN restore.

Property	Description
Syntax	TGT_RMAN_CHANNELS = <i>number_of_channels</i>
Default value	10

TGT_SKIP_DATAPATCH

TGT_SKIP_DATAPATCH specifies whether or not Zero Downtime Migration runs the datapatch utility on the target database as part of the post-migration tasks.

Property	Description
Syntax	TGT_SKIP_DATAPATCH = {TRUE FALSE}
Default value	FALSE
Range of values	TRUE - do not allow Zero Downtime Migration to run datapatch FALSE - allow Zero Downtime Migration to run datapatch

Usage Notes

If the target database environment is at a higher patch level than the source database (for example, if the source database is at Jan 2020 PSU/BP and the target database is at April 2020 PSU/BP), then set the `TGT_SKIP_DATAPATCH` parameter to `FALSE` to allow Zero Downtime Migration to run the datapatch utility on the target database as part of the post-migration tasks.

Otherwise, set the parameter to `TRUE`, and if the target database environment is at a higher patch level than the source database, you will need to run the datapatch utility manually after the migration.

TGT_SSH_RETRY_TIMEOUT

`TGT_SSH_RETRY_TIMEOUT` specifies the number of minutes during which retries are attempted after SSH connection failures. Retries stop when the timeout value has elapsed.

Property	Description
Syntax	TGT_SSH_RETRY_TIMEOUT = <i>number_of_minutes</i>
Default value	There is no default value.

TGT_SSH_TUNNEL_PORT

`TGT_SSH_TUNNEL_PORT` specifies the forwarding port on the source database server where the SSH tunnel to the target database server for SQL*Net connection is set up.

Property	Description
Syntax	TGT_SSH_TUNNEL_PORT = <i>ssh_tunnel_port_number</i>
Default value	There is no default value.

TGT_ZDLRA_WALLET_LOC

`TGT_ZDLRA_WALLET_LOC` specifies the path of the Zero Data Loss Recovery Appliance wallet on the target database server.

Property	Description
Syntax	TGT_ZDLRA_WALLET_LOC = <i>target_zdlra_wallet_location</i> The expected format for the location is /u02/app/oracle/product/12.1.0/dbhome_3/dbs/zdlra
Default value	There is no default value.

Usage Notes

When using Zero Data Loss Recovery Appliance as the migration backup medium, you must set the following parameters.

SRC_ZDLRA_WALLET_LOC

TGT_ZDLRA_WALLET_LOC

ZDLRA_CRED_ALIAS

ZDLRA_CRED_ALIAS

ZDLRA_CRED_ALIAS specifies the Zero Data Loss Recovery Appliance wallet credential alias.

Property	Description
Syntax	ZDLRA_CRED_ALIAS = <i>zdlra_wallet_alias</i> The expected format for the alias is <i>zdlra scan:listener port/zdlra9:dedicated</i>
Default value	There is no default value.

Usage Notes

When using Zero Data Loss Recovery Appliance as the migration backup medium, you must set the following parameters.

SRC_ZDLRA_WALLET_LOC

TGT_ZDLRA_WALLET_LOC

ZDLRA_CRED_ALIAS

ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL

ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_BACKUP_DIFFERENTIAL_SRC migration job phase.

Property	Description
Syntax	ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL = <i>minutes</i>
Default value	10

Usage Notes

The migration job phase monitoring interval parameters, listed below, monitor and report the backup and restore operations progress at the set time interval, specified in minutes. Note that the migration job phase for which the monitoring interval applies is prefixed to `_MONITORING_INTERVAL` in each parameter listed above.

- ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL
- ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL
- ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL
- ZDM_CLONE_TGT_MONITORING_INTERVAL
- ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL
- ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL

To disable a monitoring interval parameter, set it to 0 (zero).

ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL

ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_BACKUP_FULL_SRC migration job phase.

Property	Description
Syntax	ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL = <i>minutes</i>
Default value	10

Usage Notes

The migration job phase monitoring interval parameters, listed below, monitor and report the backup and restore operations progress at the set time interval, specified in minutes. Note that the migration job phase for which the monitoring interval applies is prefixed to `_MONITORING_INTERVAL` in each parameter listed above.

- ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL
- ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL
- ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL
- ZDM_CLONE_TGT_MONITORING_INTERVAL
- ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL
- ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL

To disable a monitoring interval parameter, set it to 0 (zero).

ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL

`ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL` specifies the time interval, in minutes, at which to monitor and report the progress of the `ZDM_BACKUP_INCREMENTAL_SRC` migration job phase.

Property	Description
Syntax	<code>ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL = minutes</code>
Default value	10

Usage Notes

The migration job phase monitoring interval parameters, listed below, monitor and report the backup and restore operations progress at the set time interval, specified in minutes. Note that the migration job phase for which the monitoring interval applies is prefixed to `_MONITORING_INTERVAL` in each parameter listed above.

- `ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL`
- `ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL`
- `ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL`
- `ZDM_CLONE_TGT_MONITORING_INTERVAL`
- `ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL`
- `ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL`

To disable a monitoring interval parameter, set it to 0 (zero).

ZDM_BACKUP_RETENTION_WINDOW

`ZDM_BACKUP_RETENTION_WINDOW` specifies the number of days after which backups created by Zero Downtime Migration become obsolete.

Property	Description
Syntax	<code>ZDM_BACKUP_RETENTION_WINDOW = days</code>
Default value	60

ZDM_CLONE_TGT_MONITORING_INTERVAL

`ZDM_CLONE_TGT_MONITORING_INTERVAL` specifies the time interval, in minutes, at which to monitor and report the progress of the `ZDM_CLONE_TGT` migration job phase.

Property	Description
Syntax	<code>ZDM_CLONE_TGT_MONITORING_INTERVAL = minutes</code>

Property	Description
Default value	10

Usage Notes

The migration job phase monitoring interval parameters, listed below, monitor and report the backup and restore operations progress at the set time interval, specified in minutes. Note that the migration job phase for which the monitoring interval applies is prefixed to `_MONITORING_INTERVAL` in each parameter listed above.

- ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL
- ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL
- ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL
- ZDM_CLONE_TGT_MONITORING_INTERVAL
- ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL
- ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL

To disable a monitoring interval parameter, set it to 0 (zero).

ZDM_CURL_LOCATION

`ZDM_CURL_LOCATION` specifies a custom location for the CURL binary on the source.

Property	Description
Syntax	<code>ZDM_CURL_LOCATION = curl_location</code>
Default value	<code>/usr/bin/curl</code>

ZDM_LOG_OSS_PAR_URL

`ZDM_LOG_OSS_PAR_URL` specifies the pre-authenticated URL to use when uploading logs to Object Storage Service. The logs capture the current migration job phase and the execution status of the phase.

Property	Description
Syntax	<code>ZDM_LOG_OSS_PAR_URL = url</code>
Default value	There is no default value. By default this parameter is disabled.

ZDM_OPC_RETRY_COUNT

`ZDM_OPC_RETRY_COUNT` specifies the number of retry attempts that will be made after an initial Object Store connection failure.

Property	Description
Syntax	<code>ZDM_OPC_RETRY_COUNT = number</code>

Property	Description
Default value	0 (zero) The default behavior is to attempt no retries.

ZDM_OPC_RETRY_WAIT_TIME

ZDM_OPC_RETRY_WAIT_TIME specifies the number of seconds to wait after an Object Store connection failure before attempting to retry the connection.

Property	Description
Syntax	ZDM_OPC_RETRY_WAIT_TIME = <i>seconds</i>
Default value	529 (seconds)

ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL

ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_OSS_RECOVER_TGT migration job phase.

Property	Description
Syntax	ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL = <i>minutes</i>
Default value	10

Usage Notes

The migration job phase monitoring interval parameters, listed below, monitor and report the backup and restore operations progress at the set time interval, specified in minutes. Note that the migration job phase for which the monitoring interval applies is prefixed to `_MONITORING_INTERVAL` in each parameter listed above.

- ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL
- ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL
- ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL
- ZDM_CLONE_TGT_MONITORING_INTERVAL
- ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL
- ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL

To disable a monitoring interval parameter, set it to 0 (zero).

ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL

ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_OSS_RESTORE_TGT migration job phase.

Property	Description
Syntax	ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL = <i>minutes</i>
Default value	10

Usage Notes

The migration job phase monitoring interval parameters, listed below, monitor and report the backup and restore operations progress at the set time interval, specified in minutes. Note that the migration job phase for which the monitoring interval applies is prefixed to `_MONITORING_INTERVAL` in each parameter listed above.

- ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL
- ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL
- ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL
- ZDM_CLONE_TGT_MONITORING_INTERVAL
- ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL
- ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL

To disable a monitoring interval parameter, set it to 0 (zero).

ZDM_SRC_TNS_ADMIN

ZDM_SRC_TNS_ADMIN specifies the custom location for TNS_ADMIN on the source database server when there is no Oracle Grid Infrastructure. If a Grid Infrastructure exists, then the TNS_ADMIN property must be set in the CRS resource attribute environment of the database resource.

Property	Description
Syntax	ZDM_SRC_TNS_ADMIN = <i>tns_admin_location</i>
Default value	There is no default value.

ZDM_USE_EXISTING_UNDO_SIZE

ZDM_USE_EXISTING_UNDO_SIZE specifies whether Zero Downtime Migration should use the existing undo tablespace size when creating a new undo tablespace, if required.

Property	Description
Syntax	ZDM_USE_EXISTING_UNDO_SIZE = {TRUE FALSE}
Default value	TRUE By default, Zero Downtime Migration uses the largest size of the existing undo tablespaces.

Property	Description
Range of values	TRUE - Use the existing undo tablespace size. FALSE - Do not use the existing undo tablespace size.

F

Zero Downtime Migration ZDMCLI Command Reference

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

- [abort job](#)
Terminates the specified job, if running.
- [add imagetype](#)
Configures a new image type of the specified name and its associated user actions.
- [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 user action with the specified name.
- [query job](#)
Gets the current status of scheduled migration jobs.
- [resume job](#)
Resumes a specified job that was paused.

abort job

Terminates the specified job, if running.

Syntax

```
$ZDM_HOME/bin/zdmcli abort job  
-jobid job_id
```

Options

Table F-1 ZDMCLI abort job Options

Option	Description
-jobid <i>job_id</i>	Unique job ID value (integer) for the scheduled job. The job ID is assigned when the job is scheduled.

add imagetype

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

Syntax

```
$ZDM_HOME/bin/zdmcli add imagetype
  -imagetype image_type
  -basetype CUSTOM_PLUGIN
  [-useractions user_action_list]
```

Options

Table F-2 ZDMCLI add imagetype Options

Option	Description
-imagetype <i>image_type</i>	Name of the image type to be created
-basetype CUSTOM_PLUGIN	The base image type for which the image type is created. Note that CUSTOM_PLUGIN is the only valid value for this mandatory argument.
-useractions <i>user_action_list</i>	Comma-separated list of user action names

add useraction

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

Syntax

```
$ZDM_HOME/bin/zdmcli add useraction
  -useraction user_action_name
  -actionscript script_name
  [-actionfile file_name]
  {-pre | -post}
  -optype MIGRATE_DATABASE
  [-phase operation_phase]
  [-onerror {ABORT | CONTINUE}]
  [-runscope
    {ONENODE |
     ALLNODES |
     AUTO}]
```

Options

Table F-3 ZDMCLI add useraction Options

Option	Description
-useraction <i>user_action_name</i>	Name of the user action
-actionscrip <i>script_name</i>	Script file to be run
-actionfile <i>file_name</i>	File associated with and needed by the user action
-pre	Runs the user action before the operation
-post	Runs the user action after the operation
-optype MIGRATE_DATABASE	Defines the operation for which the user action is configured as MIGRATE_DATABASE.
-phase <i>phase_of_operation</i>	Migration operation phase for which the user action is configured
-onerror {ABORT CONTINUE}	The response if the user action encounters an error during execution
-runscope {ONENODE ALLNODES AUTO}	The servers on which the user action is run. Specify <i>AUTO</i> to choose the run scope based on the other command options.

migrate database

Performs a migration of a database to the Oracle Cloud.

Syntax

```
$ZDM_HOME/bin/zdmcli migrate database
  {-sourcedb db_name |
   -sourcesid source_oracle_sid}
  -sourcenode source_host_name
  -targetnode target_host_name
  -rsp zdm_template_path
  -backupuser user_name
  [-targethome target_home]
  [-eval]
  [-imagetype]
  [-tdekeystorepasswd]
  [-tdemasterkey]
  [-sourcesyswallet sys_wallet_path]
  [-osswallet oss_wallet_path]
  [-tdekeystorewallet tde_wallet_path]
  [-useractiondata user_action_data]
  [{-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 F-4 ZDMCLI migrate database Options

Option	Description
-sourcedb <i>db_name</i>	Name of the source database you want to migrate
-sourcenode <i>source_host_name</i>	Host on which the source database is running
-targetnode <i>target_host_name</i>	Target server to which the source database is migrated
-targethome <i>target_home</i>	Location of the target database ORACLE_HOME
-imagetype <i>image_type</i>	Name of the user action imagetype
-useractiondata <i>user_action_data</i>	Value to be passed to <i>useractiondata</i> parameter of the user action script
-rsp <i>zdm_template_path</i>	Location of the Zero Downtime Migration response file
-sourcesid <i>source_oracle_sid</i>	ORACLE_SID of the source single instance database without Grid Infrastructure
-eval	Evaluate the migration job without actually running the migrate database command against the source and target
-backupuser <i>user_name</i>	Name of the user allowed to back up or restore the database
-srcroot	Directs Zero Downtime Migration to use root credentials to access the source database server
-srccred <i>cred_name</i>	Credential name with which to associate the user name and password credentials to access the source database server
-srcuser <i>user_name</i>	Name of the privileged user performing operations on the source database server
-srcsudouser <i>user_name</i>	Perform super user operations as sudo user name on the source database server
-srcsudopath <i>sudo_binary_path</i>	Location of sudo binary on the source database server

Table F-4 (Cont.) ZDMCLI migrate database Options

Option	Description
<code>-srcauth plug-in_name [plug-in_args]</code>	Use an authentication plug-in to access the source database server
<code>-tgtroot</code>	Use <code>root</code> credentials to access the target database server
<code>-tgtcred cred_name</code>	Credential name with which to associate the user name and password credentials to access the target database server
<code>-tgtuser user_name</code>	Name of the user performing operations on the target database server
<code>-tgtsudouser user_name</code>	Perform super user operations as <code>sudo</code> user name on the target database server
<code>-tgtsudopath sudo_binary_path</code>	Location of <code>sudo</code> binary on the target database server
<code>-tgtauth plugin_name [plugin_args]</code>	Use an authentication plug-in to access the target database server
<code>-tdekeystorepasswd</code>	Transparent Data Encryption (TDE) keystore password, required for password-based keystore or wallet
<code>-tdemasterkey</code>	Transparent Data Encryption (TDE) master encryption key
<code>-schedule timer_value</code>	Scheduled time to execute the operation, in ISO-8601 format. For example: 2016-12-21T19:13:17+05
<code>-pauseafter phase</code>	Pause the job after running the specified phase
<code>-ignoremissingpatches</code>	Proceed with the migration 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
<code>-ignore {ALL WARNING PATCH_CHECK}</code>	Ignore all checks or specific type of checks
<code>-listphases</code>	List the phases for this operation
<code>-sourcesyswallet sys_wallet_path</code>	Full path for the auto-login wallet file on the Zero Downtime Migration host containing the <code>SYS</code> password of the source database
<code>-osswallet oss_wallet_path</code>	Full path for the auto-login wallet file on the Zero Downtime Migration host containing the credential for the Object Storage Service (OSS) backup user
<code>-tdekeystorewallet tde_wallet_path</code>	Full path for the auto-login wallet file on the Zero Downtime Migration host containing the Transparent Data Encryption (TDE) keystore password

modify useraction

Modifies the configuration of the user action with the specified name.

Syntax

```
$ZDM_HOME/bin/zdmcli modify useraction
-useraction user_action_name
[-actionscript script_name]
[-actionfile file_name]
[-pre | -post]
[-optype MIGRATE_DATABASE]
[-phase phase]
[-onerror {ABORT | CONTINUE}]
[-runscope
  {ONENODE |
  ALLNODES |
  AUTO}]
```

Options

Table F-5 ZDMCLI modify useraction Options

Option	Description
-useraction <i>user_action_name</i>	Name of the user action
-actionscript <i>script_name</i>	Script file to be run
-actionfile <i>file_name</i>	Accompanying file needed by the user action
-pre	Runs the user action before the operation
-post	Runs the user action after the operation
-optype MIGRATE_DATABASE	Defines the operation for which the user action is configured as MIGRATE_DATABASE
-onerror {ABORT CONTINUE}	Defines whether to stop or continue running if an error occurs while the user action is running
-runscope {ONENODE ALLNODES AUTO}	The servers where the user action will be run. Specify <i>AUTO</i> to choose the run scope based on the other command options.

query job

Gets the current status of scheduled migration jobs.

Syntax

```
$ZDM_HOME/bin/zdmcli query job
[-jobid job_id
  [-jobtype]]
[-sourcename source_host_name
  [-sourcedb db_name |
```

```

    -sourcesid sid]]
[-targetnode target_host_name]
[-latest]
[-eval |
-migrate]
[-status
  {SCHEDULED |
  EXECUTING |
  UNKNOWN |
  TERMINATED |
  FAILED |
  SUCCEEDED |
  PAUSED |
  ABORTED}]
[-dbname database_name]
[-since timer_value]
[-upto timer_value]
[-brief]
[-statusonly]

```

Options

Table F-6 ZDMCLI query job Options

Option	Description
-jobid <i>job_id</i>	Unique job ID value (integer) for the scheduled migration job The job ID is assigned when the migration job is scheduled.
-job_type	Returns the type of the scheduled job
-sourcenode <i>source_host_name</i>	Server on which the source database is running
-sourcedb <i>db_name</i>	Name of the source database to be migrated
-sourcesid <i>sid</i>	The ORACLE_SID of the source single instance database without Grid Infrastructure
-targetnode <i>target_host_name</i>	Target server to which the database is migrated
-latest	Returns the most recent job that matches the given criteria
-eval	Returns evaluation jobs only
-migrate	Returns migration jobs only
-status {SCHEDULED EXECUTING UNKNOWN TERMINATED FAILED SUCCEEDED PAUSED ABORTED}	Returns jobs that match the specified job status
-dbname <i>unique_db_name</i>	Specifies the database DB_UNIQUE_NAME value
-since <i>timer_value</i>	Date from which to get the jobs, in ISO-8601 format. For example: 2016-12-21T19:13:17+05

Table F-6 (Cont.) ZDMCLI query job Options

Option	Description
-upto <i>timer_value</i>	Upper limit time to which to get the jobs, in ISO-8601 format. For example: 2016-12-21T19:13:17+05
-brief	Returns job details summary only
-statusonly	Returns only the job status and current phase name

resume job

Resumes a specified job that was paused.

Syntax

```
$ZDM_HOME/bin/zdmcli resume job
  -jobid job_id
  [-pauseafter pause_phase]
```

Options

Table F-7 ZDMCLI resume job Options

Option	Description
-jobid <i>job_id</i>	Unique job ID value (integer) for the scheduled job The job ID is assigned when the migration job is scheduled.
-pauseafter <i>pause_phase</i>	Pauses the migration job after running the specified phase

Index

A

abort job command, [F-1](#), [F-8](#)
about Zero Downtime Migration, [1-1](#)
action template, [3-27](#), [3-28](#)
add imagetype command, [F-2](#)
add useraction command, [F-2](#)
architectures, [1-7](#)
architectures supported, [1-5](#)

B

backup and restore, [1-4](#)
BACKUP_PATH, [E-5](#)
bare metal target, [3-16](#), [3-18](#)
benefits, [1-3](#)

C

connectivity, [3-2](#)
 between source and target, [3-4](#)
connectivity prerequisites, [3-1](#), [F-1](#)

D

database architectures, [1-7](#)
database migration, [4-1](#)
database preparation, target, [1-9](#)
database source prerequisites, [3-10](#)
database target prerequisites, [3-11](#)
database version, [1-6](#)
database versions, [1-5](#)
datapatch utility, [4-9](#)
DATAPATCH_WITH_ONE_INSTANCE_RUNNIN
 G, [E-5](#)
deinstall, [2-5](#)

E

encryption requirements, [B-1](#)

F

failure, handling, [5-1](#)

H

HOST, [E-6](#)

I

installation prerequisites, [2-1](#)
installing, [2-1](#)
installing software, [2-2](#)
introduction, [1-1](#)

K

key pair without passphrase, [3-8](#)

L

list phases in a migration job, [4-6](#)

M

MAX_DATAPATCH_DURATION_MINS, [E-6](#)
migrate database command, [F-3](#)
migration, [4-1](#)
migration job
 list phases, [4-6](#)
 pause and resume, [4-7](#)
 rerun, [4-8](#)
 status, [4-6](#)
migration job failure, handling, [5-1](#)
migration paths, [1-3](#)
migration phases, [D-1](#)
migration preparation, [3-1](#)
migration prerequisites, [3-1](#), [F-1](#)
MIGRATION_METHOD, [E-7](#)
modify useraction command, [F-6](#)

N

NFS storage response file, [3-22](#)

O

Object Storage, [1-4](#), [3-20](#)

offline migration, [3-24](#)
 OPC_CONTAINER, [E-7](#)
 operational phases, [1-9](#), [D-1](#)

P

patch level, [4-9](#)
 paths supported, [1-3](#)
 pause a migration job, [4-7](#)
 phases, [1-9](#)
 phases of migration job, [D-1](#)
 placeholder database, [1-9](#)
 PLATFORM_TYPE, [E-8](#)
 platforms supported, [1-5](#)
 service host, [1-6](#)
 source and target database servers, [1-6](#)
 plug-in registration, [3-26](#)
 plug-in, updating, [3-27](#)
 plug-ins, [1-9](#)
 port requirements, [A-1](#)
 post-migration tasks, [4-9](#)
 prepare for migration, [3-1](#)
 prerequisites, [2-1](#)
 prerequisites for connectivity, [3-1](#), [F-1](#)
 prerequisites, source database, [3-10](#)
 prerequisites, target, [3-11](#)

Q

query job command, [F-6](#)

R

register plug-in, [3-26](#)
 requirements, encryption, [B-1](#)
 requirements, ports, [A-1](#)
 rerun a migration job, [4-8](#)
 response file
 NFS storage, [3-22](#)
 ZDLRA, [3-19](#)
 response file parameters
 BACKUP_PATH, [E-5](#)
 DATAPATCH_WITH_ONE_INSTANCE_RUNNING, [E-5](#)
 HOST, [E-6](#)
 MAX_DATAPATCH_DURATION_MINS, [E-6](#)
 MIGRATION_METHOD, [E-7](#)
 OPC_CONTAINER, [E-7](#)
 PLATFORM_TYPE, [E-8](#)
 SHUTDOWN_SRC, [E-8](#)
 SKIP_FALLBACK, [E-8](#)
 SKIP_SRC_SERVICE_RETENTION, [E-9](#)
 SRC_BASTION_HOST_IP, [E-9](#)
 SRC_BASTION_IDENTITY_FILE, [E-9](#)
 SRC_BASTION_PORT, [E-10](#)

response file parameters (*continued*)

 SRC_BASTION_USER, [E-11](#)
 SRC_CONFIG_LOCATION, [E-11](#)
 SRC_HOST_IP, [E-12](#)
 SRC_HTTP_PROXY_PORT, [E-12](#)
 SRC_HTTP_PROXY_URL, [E-12](#)
 SRC_OSS_PROXY_HOST, [E-13](#)
 SRC_OSS_PROXY_PORT, [E-13](#)
 SRC_RMAN_CHANNELS, [E-13](#)
 SRC_SSH_RETRY_TIMEOUT, [E-13](#)
 SRC_TIMEZONE, [E-14](#)
 SRC_ZDLRA_WALLET_LOC, [E-14](#)
 TGT_BASTION_HOST_IP, [E-14](#)
 TGT_BASTION_IDENTITY_FILE, [E-15](#)
 TGT_BASTION_PORT, [E-16](#)
 TGT_BASTION_USER, [E-16](#)
 TGT_CONFIG_LOCATION, [E-17](#)
 TGT_DATAACFS, [E-17](#)
 TGT_DATADG, [E-18](#)
 TGT_DB_UNIQUE_NAME, [E-18](#)
 TGT_HOST_IP, [E-19](#)
 TGT_HTTP_PROXY_PORT, [E-19](#)
 TGT_HTTP_PROXY_URL, [E-20](#)
 TGT_OSS_PROXY_HOST, [E-20](#)
 TGT_OSS_PROXY_PORT, [E-20](#)
 TGT_RECOACFS, [E-21](#)
 TGT_RECODG, [E-21](#)
 TGT_REDOACFS, [E-22](#)
 TGT_REDODG, [E-23](#)
 TGT_RMAN_CHANNELS, [E-23](#)
 TGT_SKIP_DATAPATCH, [E-23](#)
 TGT_SSH_RETRY_TIMEOUT, [E-24](#)
 TGT_SSH_TUNNEL_PORT, [E-24](#)
 TGT_ZDLRA_WALLET_LOC, [E-24](#)
 ZDLRA_CRED_ALIAS, [E-25](#)
 ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL, [E-25](#)
 ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL, [E-26](#)
 ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL, [E-27](#)
 ZDM_BACKUP_RETENTION_WINDOW, [E-27](#)
 ZDM_CLONE_TGT_MONITORING_INTERVAL, [E-27](#)
 ZDM_CURL_LOCATION, [E-28](#)
 ZDM_LOG_OSS_PAR_URL, [E-28](#)
 ZDM_OPC_RETRY_COUNT, [E-28](#)
 ZDM_OPC_RETRY_WAIT_TIME, [E-29](#)
 ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL, [E-29](#)
 ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL, [E-29](#)
 ZDM_SRC_TNS_ADMIN, [E-30](#)
 ZDM_USE_EXISTING_UNDO_SIZE, [E-30](#)
 restore from backup, [1-4](#)
 resume a migration job, [4-7](#)
 root user access, [1-8](#)

S

security, [1-7](#)
 SHUTDOWN_SRC, [E-8](#)
 SKIP_FALLBACK, [E-8](#)
 SKIP_SRC_SERVICE_RETENTION, [E-9](#)
 software installation, [2-1](#)
 source database prerequisites, [3-10](#)
 SRC_BASTION_HOST_IP, [E-9](#)
 SRC_BASTION_IDENTITY_FILE, [E-9](#)
 SRC_BASTION_PORT, [E-10](#)
 SRC_BASTION_USER, [E-11](#)
 SRC_CONFIG_LOCATION, [E-11](#)
 SRC_HOST_IP, [E-12](#)
 SRC_HTTP_PROXY_PORT, [E-12](#)
 SRC_HTTP_PROXY_URL, [E-12](#)
 SRC_OSS_PROXY_HOST, [E-13](#)
 SRC_OSS_PROXY_PORT, [E-13](#)
 SRC_RMAN_CHANNELS, [E-13](#)
 SRC_SSH_RETRY_TIMEOUT, [E-13](#)
 SRC_TIMEZONE, [E-14](#)
 SRC_ZDLRA_WALLET_LOC, [E-14](#)
 SSH key without passphrase, [3-8](#)
 SSH key-based access, [1-8](#)
 status of migration job, [4-6](#)
 supported architectures, [1-7](#)
 supported configurations, [1-5](#)
 supported database versions, [1-6](#)
 supported migration paths, [1-3](#)
 supported platforms
 service host, [1-6](#)
 source and target database servers, [1-6](#)

T

target preparation, [1-9](#)
 target prerequisites, [3-11](#)
 target, bare metal, [3-16](#), [3-18](#)
 target, virtual machine, [3-16](#), [3-18](#)
 TGT_BASTION_HOST_IP, [E-14](#)
 TGT_BASTION_IDENTITY_FILE, [E-15](#)
 TGT_BASTION_PORT, [E-16](#)
 TGT_BASTION_USER, [E-16](#)
 TGT_CONFIG_LOCATION, [E-17](#)
 TGT_DATAACFS, [E-17](#)
 TGT_DATADG, [E-18](#)
 TGT_DB_UNIQUE_NAME, [E-18](#)
 TGT_HOST_IP, [E-19](#)
 TGT_HTTP_PROXY_PORT, [E-19](#)
 TGT_HTTP_PROXY_URL, [E-20](#)
 TGT_OSS_PROXY_HOST, [E-20](#)
 TGT_OSS_PROXY_PORT, [E-20](#)
 TGT_RECOACFS, [E-21](#)
 TGT_RECODG, [E-21](#)
 TGT_REDOACFS, [E-22](#)

TGT_REDODG, [E-23](#)
 TGT_RMAN_CHANNELS, [E-23](#)
 TGT_SKIP_DATAPATCH, [E-23](#)
 TGT_SSH_RETRY_TIMEOUT, [E-24](#)
 TGT_SSH_TUNNEL_PORT, [E-24](#)
 TGT_ZDLRA_WALLET_LOC, [E-24](#)
 Transparent Data Encryption
 wallet setup, [3-13](#)

U

uninstall, [2-5](#)
 update, [2-4](#)

V

versions supported, [1-5](#), [1-6](#)
 virtual machine target, [3-16](#), [3-18](#)

Z

ZDLRA response file, [3-19](#)
 ZDLRA_CRED_ALIAS, [E-25](#)
 ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL, [E-25](#)
 ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL, [E-26](#)
 ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL, [E-27](#)
 ZDM_BACKUP_RETENTION_WINDOW, [E-27](#)
 ZDM_BASE, [1-7](#)
 ZDM_CLONE_TGT_MONITORING_INTERVAL, [E-27](#)
 ZDM_CURL_LOCATION, [E-28](#)
 ZDM_HOME, [1-7](#)
 ZDM_LOG_OSS_PAR_URL, [E-28](#)
 ZDM_OPC_RETRY_COUNT, [E-28](#)
 ZDM_OPC_RETRY_WAIT_TIME, [E-29](#)
 ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL, [E-29](#)
 ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL, [E-29](#)
 ZDM_SRC_TNS_ADMIN, [E-30](#)
 ZDM_USE_EXISTING_UNDO_SIZE, [E-30](#)
 zdmcli
 abort job, [F-1](#), [F-8](#)
 add imagetype, [F-2](#)
 add useraction, [F-2](#)
 migrate database, [F-3](#)
 modify useraction, [F-6](#)
 query job, [F-6](#)
 Zero Downtime Migration
 about, [1-1](#)