DISCLAIMER

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.
# TABLE OF CONTENTS

- **Disclaimer** | 1
- **Introduction** | 3
- **Product Overview** | 4
  - Architecture | 4
  - Supported Platforms and configurations | 4
- **What’s New in Oracle Zero Downtime Migration 19.7** | 5
  - ZDMCLI Command Enhancements | 5
  - ZDMCLI Response File Enhancements | 5
  - Bug Fixes | 6
- **Requirements** | 7
  - Zero Downtime Migration Service Host Requirements | 7
  - Source Database Requirements | 7
  - Target Database Requirements | 8
  - Backup requirements | 9
  - Oracle NET Services (SQL*Net) Connectivity Requirements | 9
  - Port Requirements | 9
- **ZDM Installation & Pre-Migration Configuration** | 10
  - Oracle ZDM Service Node Configuration | 10
  - Configuring Connectivity between ZDM Service Node and Source and Target Database Servers | 15
  - Configuring Connectivity between the Source and Target Database Servers | 16
    - Option 1 - SCAN Connectivity | 16
    - Option 2 - SSH Tunnel | 17
  - Creating a Target Database | 17
  - Setting up a Transparent Data Encryption Wallet | 21
  - Creating Object Storage for Database Backup | 21
  - Generating an Authentication Token | 23
  - Preparing the Response File | 25
    - Response File Parameters | 25
- **Database Migration with ZDM** | 29
  - Performing a Database Migration pre-check process | 29
  - Performing a Database Migration | 31
- **Frequently Asked Questions & Common Issues** | 33
  - F.A.Q | 33
  - Known Issues | 35
  - Troubleshooting & Other Resources | 35
INTRODUCTION

Oracle Database is the best enterprise database, offering the highest levels of performance, availability, scalability and security to customers around the world. It also offers the most flexible deployment options – whether on-premises, in the public cloud leveraging Oracle Infrastructure (OCI), or in a Cloud at Customer platform leveraging Exadata Cloud at Customer (ExaCC), in your own data center.

Oracle Zero Downtime Migration (hereinafter ZDM) is a software solution that allows you to directly and seamlessly migrate your on-premises Oracle Databases to the Oracle Cloud – whether in OCI or ExaCC. Oracle ZDM supports a wide range of Oracle Database versions, and – as the name implies, it ensure that there is minimal to no production database impact during the migration, following the principles of Oracle Maximum Availability Architecture.

This technical brief is a step by step guide for migrating your on-premises Oracle Databases to the Oracle Cloud with Zero Downtime Migration. The scenario used for this migration comprises of a source database running on Oracle Cloud Infrastructure DBCS Virtual Machines (emulating an on-premises environment) and a target database running on Oracle Cloud Infrastructure DBCS Virtual Machines. Oracle ZDM will run on a separate node, as indicated in our documentation and will connect to both source and target to perform the migration.

All requirements related to the installation of the Oracle ZDM service node, the source database to be migrated, the target database recipient of the migration process, the backup and networking used, will be properly covered. The migration process will be dissected and done on a step by step fashion and the most frequently asked questions regarding the product and the overall migration process will be answered here.
PRODUCT OVERVIEW

Architecture

Oracle Zero Downtime Migration (ZDM) is Oracle Maximum Availability Architecture (MAA)-recommended solution to migrate Oracle Databases to the Oracle Cloud. ZDM has been designed with the goal of keeping the migration process as simple as possible and ensuring at least impact on production workloads. The source databases to be migrated can be on-premises or deployed on Oracle Public Cloud Gen 1. The target databases may be deployed as Database Cloud Service on Oracle Cloud Infrastructure (OCI) Bare Metal, Virtual Machine, Exadata Cloud Service, or Exadata Cloud at Customer. ZDM automates the entire process of migration, reducing the chance of human errors. ZDM leverages Oracle Database-integrated high availability (HA) technologies such as Oracle Data Guard and follows all MAA best practices that ensures zero to no downtime of production environments.

ZDM performs a zero downtime migration in eight simple steps, which can be scheduled and monitored as needed. The following section provides details on each of these steps.

Supported Platforms and configurations

Oracle ZDM supports the following Oracle Database versions: 11.2.0.4, 12.1.0.2, 12.2.0.1, 18c, 19c.

The source and target databases should be in the same database version. Oracle ZDM supports Oracle Databases hosted on Linux operating systems. The source database can be a single instance database migrating to a single instance or a RAC database, or it can also be a RAC One Node / RAC database, migrating to a RAC database.

Oracle ZDM supports Enterprise & Standard Edition Oracle Databases as source databases. Enterprise Edition Databases are migrated leveraging Oracle Data Guard; Standard Edition Databases are migrated in an offline manner using a backup and restore methodology. Oracle ZDM allows for the source database to be a non-CDB or a container database (CDB) with one or more Pluggable Databases (PDBs). If the source Database is a non-CDB, it will be migrated as a non-CDB. In case of a CDB with one or more PDBs, ZDM migrates it to a CDB with the same set of PDBs as in the source CDB.

ZDM supports on-premises databases to be migrated to:

1. Oracle Database Cloud Service Bare Metal
2. Oracle Database Cloud Service Virtual Machine
3. Exadata Cloud Service
4. Exadata Cloud at Customer

ZDM Supports the following backup mediums: OCI Object Storage, Oracle Zero Data Loss Recovery Appliance, NFS Storage.
WHAT'S NEW IN ORACLE ZERO DOWNTIME MIGRATION 19.7

Zero Downtime Migration Patch Release 19.7 enhances the existing functionality of ZDM while providing bug fixes and more control of the migration process. Here’s what’s new:

ZDMCLI Command Enhancements

The `ZDMCLI MIGRATE DATABASE` command has new parameters that facilitate and automates the use of wallet-based credentials, also it removed the requirement for the `-targethome` parameter since the target database home is now automatically discovered. The new wallet-based parameters lets users specify the full path for the source sys wallet file, the OSS backup user wallet file and the TDE keystore wallet file, on the Zero Downtime Migration Service host, these parameters are:

- `sourcesyswallet sys_wallet_path`
- `osswallet oss_wallet_path`
- `tdekeystorewallet tde_wallet_path`

The `ZDMCLI QUERY JOB` command expanded its functionality with new parameters, providing further information on existing migration jobs. These parameters allow for more complex queries getting information on job types, most recent job, information of jobs on evaluation mode only and limiting information to job status. Furthermore, queries can now specify sourcedb, sourceSID and targetnode.

- `-job_type`, `latest`, `-eval`, `-statusonly`
- `-sourcenode`, `-sourcedb`, `-sourcesid`, `-targetnode`

ZDMCLI Response File Enhancements

Oracle ZDM uses a response file to configure relevant parameters related to source database, migration method, target database, backup location and other required elements. The following response file parameters are new in Oracle ZDM Patch Release 19.7.

ZDM Backup-related response file parameters:

Oracle ZDM leverages database backups as part of its overall migration workflow. New parameters have been added to the response file, providining users the ability to set intervals for monitoring and reporting the progress of backup and restore operations. The intervals are set by default to ten minutes, and can be disabled by setting them to zero.

The following lists encompasses the new parameters and the related migration job phase associated to be monitored and reported.
NEW RESPONSE FILE PARAMETER | MIGRATION JOB PHASE TO BE MONITORED AND REPORTED
--- | ---
ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL | ZDM_BACKUP_FULL_SRC
ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL | ZDM_BACKUP_INCREMENTAL_SRC
ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL | ZDM_BACKUP_DIFFERENTIAL_SRC
ZDM_CLONE_TGT_MONITORING_INTERVAL | ZDM_CLONE_TGT
ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL | ZDM_OSS_RECOVER_TGT
ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL | ZDM_OSS_RESTORE_TGT

Further enhancements to the response file have been made related to source database service retention, object store access retry, custom location for CURL, pre-authenticated URL for log files upload, custom TNS_ADMIN location and SSH connection reattempt.

- **SKIP_SRC_SERVICE_RETENTION**=TRUE/FALSE if true, any source services are migrated and re-created at the target.
- **ZDM_BACKUP_RETENTION_WINDOW** number of days for ZDM created database backups to become obsolete.
- **ZDM_OPC_RETRY_WAIT_TIME** Object Store retry wait time in seconds.
- **ZDM_OPC_RETRY_COUNT** number of times to retry the Object Store.
- **ZDM_CURL_LOCATION** custom location for CURL binary on the source.
- **ZDM_LOG_OSS_PAR_URL** custom location of a pre-authenticated URL indicating log files upload location.
- **ZDM_SRC_TNS_ADMIN** specifies TNS_ADMIN location for source databases without Grid Infrastructure.
- **SRC_SSH_RETRY_TIMEOUT** time in minutes to retry SSH connection to the source server.
- **TGT_SSH_RETRY_TIMEOUT** time in minutes to retry SSH connection to the target server.

Finally, parameters related to post-migration tasks have been added in order to automate datapatch execution. These parameters allow users to skip datapatch execution post-migration (**TGT_SKIP_DATAPATCH**), let users setup a maximum wait time for datapatch operations post-migration (**MAX_DATAPATCH_DURATION_MINS**) and let users stop all databases instance minus one on the target until datapatch completion, ZDM will then restart all stopped instances (**DATAPATCH_WITH_ONE_INSTANCE_RUNNING**).

**Bug Fixes**

Oracle ZDM Patch Release 19.7, provides a series of bug fixes addressing issues regarding different aspects of the migration workflow. A list of all fixed bugs can be found at Oracle Zero Downtime Migration’s documentation release notes section at ZDM’s product page.

1 www.oracle.com/goto/zdm
REQUIREMENTS

Zero Downtime Migration Service Host Requirements

Oracle Zero Downtime Migration binaries must be installed on a separate host which fulfills the following requirements:

- Linux host running on Oracle 7
- 100 GB of free storage space
- A `zdm` group and a `zdmuser` as part of this group, please create them as follows:
  - `groupadd zdm -g 1001`
  - `useradd zdmuser -g 1001`
- Following packages must be installed:
  - `glibc-devel`
  - `expect`
  - `unzip`
  - `libaio`
  - `oraclelinux-developer-release-el7`
- All host names and IP address to be used must be present as entries at `/etc/hosts`
- Appropriate permission for source and database access (i.e. ssh keys)

Source Database Requirements

ZDM supports Oracle Database 11g release 2 (11.2.0.4) or later versions, it is important that both the source and target are at the same release version.

- Source database must be in archive log mode
- If the source database is on 12c Release 2 and later and Transparent Data Encryption is not enabled:
  - You must configure the Transparent Data Encryption (TDE) Wallet. Guidance on how to configure the wallet can be found at “Setting Up the Transparent Data Encryption Wallet”
  - Upon setting up the TDE Wallet please ensure that:
    - `WALLET_TYPE` is set to `AUTOLOGIN` or `PASSWORD`
    - Wallet `STATUS` is `OPEN`
    - Wallet must be open on all pluggable databases if the source database is a container database. The master key must be set for all the pluggable databases and the container database.
- Source databases can be Oracle RAC databases, if this is the case, `SNAPSHOT CONTROLFILE` must be configured to point to a shared location on all cluster nodes.

2 https://docs.oracle.com/en/database/oracle/zerodowntime-migration/
• SCAN listener ports on the source database server and the target database server must allow incoming connections from each other.

• The parameter DB_NAME must be the same on both the source database and the target database.

• The parameter DB_UNIQUE_NAME must be different on both the source database and the target database, this way Oracle Data Guard can identify that the source and target are different.

• The parameter SYSPASSWORD must be the same on both the source database and the target database.

• The Database version must match on both the source database and the target database.

**Target Database Requirements**

• The target database must be created prior to the migration, this guide will explain this process on a further section.

• 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 is at a higher patch level than the source database, the customer must run datapatch after the database migration has taken place.

• For Oracle RAC Databases, it is important to setup SSH connectivity between the Oracle RAC nodes for the oracle user.

• Sizing of the target database must be taken into consideration to avoid any shortfalls during the migration. As a rule of thumb, same or larger size than the source database is recommended.

• The parameter DB_NAME must be the same on both the source database and the target database.

• The parameter DB_UNIQUE_NAME must be different on both the source database and the target database, this way Oracle Data Guard can identify that the source and target are different.

• Automatic backups must be disabled for the target database, this is done differently depending on the target cloud platform:
  - OCI DBCS BM/VM: upon creation of the target database there is an option named “Enable automatic backups”, do not select it.
  - Exadata Cloud Service: upon creation of the target database there is an option named “Enable automatic backups”, do not select it.
  - Exadata Cloud at Customer: among the database options there is a section named: “Configure Backups”, go to it, and on the option “Backup Destination” select the “Type” sub-option to None.

• The target database must have Transparent Data Encryption Enabled (TDE). Also, the wallet status must be open, and the wallet type must be either autologin or password.
Backup requirements

- Ensure both the source database server and the target database server can access the backup medium (Object Store for DBCS BM/VM and ExaCS, Recovery Appliance or NFS Storage for ExaCC).
- Sizing must be taken into consideration, so that the source database backups fit without an issue in the backup medium.

Oracle NET Services (SQL*Net) Connectivity Requirements

- Please ensure that all database ports and all SCAN ports are open on both the source database server and the target database server.

Port Requirements

- **ZDM Service Node**
  - Port 22 must be open, this port is going to be used for SSH, enabling connectivity between the service node and the source database server and the service node and the target database server.

- **Source Database Server**
  - Port 1521 must be open and not blocked by a firewall, this port will be used for Oracle NET Services connectivity between the source database server and target database server. This connectivity will enable proper Data Guard Synchronization.
  - Port 443 must be open and not blocked by a firewall, this port will be used to access the Object Store.

- **Target Database Server**
  - Port 1521 must be open and not blocked by a firewall, this port will be used for Oracle NET Services connectivity between the source database server and target database server. This connectivity will enable proper Data Guard Synchronization.
  - Port 443 must be open and not blocked by a firewall, this port will be used to access the Object Store.
Oracle ZDM Service Node Configuration

Accept the Oracle License Agreement https://www.oracle.com/technetwork/licenses/sqldev-license-152021.html#licenseContent and download the ZDM binaries on the ZDM Service host. Upon download, switch to the previously created zdmuser user. Unzip the binaries using the `unzip` command. Remember that the ZDM service node runs on a separate host, the best practice for the migration looks as per the figure below:

Upon extracting the binaries, create the following directories:

- **oraclehome**: directory where ZDM will be installed
- **oraclebase**: directory where all ZDM's configuration files and logs will be stored
Install the required packages, glibc-devel, expect, etc:

- [root@zdm-servicenode ~]# yum install glibc-devel*
- [root@zdm-servicenode ~]# yum install expect

Proceed to execute ZDM’s installation script zdmuser:

./zdminstall.sh setup oraclehome=zdm_oracle_home oraclebase=zdm_base_directory ziploc=zdm_loc -zdm

Bear in mind that ziploc refers to the full path zdm_home.zip is located. This file is part of the uncompressed bundle from the downloadable binaries from www.oracle.com/goto/zdm

Due to ZDM inherent relationship with Grid Infrastructure, a set of messages requiring scripts to be ran as a root user will appear upon installation completion. Please disregard this messages, ZDM’s installation process does not require any actions as root. Furthermore, as shown below, the installer will request for certain scripts to be run as root, please disregard this.

If you have already installed ZDM and want to perform an upgrade, please visit the latest product documentation, accessible from our product page www.oracle.com/goto/zdm for a step by step guide on how to upgrade ZDM.

Upon successful installation and before performing any Oracle Database migration, the ZDM service must be started and installation success must be verified. In order to so, please execute the following as zdmuser:

/zdm_oracle_home/bin/zdmservice start
/zdm_oracle_home/bin/zdmservice status

The zdmservice start command will start the zdmservice. The zdmservice status command should return the service status and show running:true. This guarantees that the zdmservice was properly installed and it is up and running. If the installation process failed or there are errors present, please refer to the Frequently Asked Questions section at the end of this guide.
[zdmuser@zdm-servicenode media]$ ./zdminstall.sh setup oraclehome=/u01/app/oracle/zdmhome oraclebase=/u01/app/oracle/zdmbase ziploc=/u01/media/zdm_home.zip -zdm

Unzipping shiphome to gridhome

Unzipping shiphome...

Shipped unzipped successfully.

##### Starting GridHome Software Only Installation #####

Launching Oracle Grid Infrastructure Setup Wizard...

**A WARNING WILL BE DISPLAYED HERE, PLEASE DISREGARD**

The response file for this session can be found at:

/u01/app/oracle/zdmhome/install/response/grid_.rsp

You can find the log of this install session at:

/tmp/GridSetupActions/gridSetupActions.log

As a root user, execute the following script(s):

1. /u01/app/oracle/zdmhome/inventory/orainstRoot.sh
2. /u01/app/oracle/zdmhome/root.sh

Execute /u01/app/oracle/zdmhome/inventory/orainstRoot.sh on the following nodes:

[zdm-servicenode]

Execute /u01/app/oracle/zdmhome/root.sh on the following nodes:

[zdm-servicenode]

Successfully Setup Software with warning(s).

Moved the install session logs to:

/u01/app/oracle/zdmhome/inventory/logs/GridSetupActions

=*=*=*

PLEASE IGNORE THE MESSAGE ABOVE TO RUN SCRIPT(S) AS ROOT USER.

ZDM INSTALL DOES NOT NEED ANY ACTIONS AS ROOT.

=*=*=*

making dir /u01/app/oracle/zdmbase/crsdata/zdm-servicenode/rhp/conf

Generating Preference file

/u01/app/oracle/zdmbase/crsdata/zdm-servicenode/rhp/conf/rhp.pref

Generating Root Certificate

Cluster root certificate generated successfully.

Generating CA CERTS file

spawn /u01/app/oracle/zdmhome/bin/crskeytoolctl -copycacerts -filestore /u01/app/oracle/zdmbase/crsdata/zdm-servicenode/security/

Enter JRE cacerts truststore password:

JRE cacerts copied to file [/u01/app/oracle/zdmbase/crsdata/zdm-servicenode/security/cacerts].

Generating nogi.enabled file

nogi.enabled file generated successfully

Generating standalone_config.properties file

Setting base folder permissions

Copying service script to bin folder in Oracle Home

ZDM setup without GI finished successfully...
No instance detected, starting zdmservice

[jwcctl debug] Environment ready to start JWC
[jwcctl debug] Return code of initialization: [0]

[jwcctl debug] ... BEGIN_DEBUG [Action= start] ...
Start JWC
[jwcctl debug] Loading configuration file: /u01/app/oracle/zdmbase/crsdata/zdm-servicenode/rhp/conf/jwc.properties
[jwcctl debug] oracle.jmx.login.credstore = CRSCRED
[jwcctl debug] oracle.jmx.login.args = DOMAIN=rhp
[jwcctl debug] oracle.rmi.url = service:jmx:rmi:///0:{1,number,#}/jndi/rmi:///0:{1,number,#}/jmxrmi
[jwcctl debug] oracle.http.url = http://(0):{1,number,#}/rhp/gridhome
[jwcctl debug] oracle.jwc.tls.clientauth = false
[jwcctl debug] oracle.jwc.container.clientfactory = RELOADABLE
[jwcctl debug] oracle.jwc.lifecycle.start.log.fileName = JWCStartEvent.log
[jwcctl debug] Get JWC PIDs
[jwcctl debug] Done Getting JWC PIDs
[jwcctl debug] ... JWC containers not found ...
[jwcctl debug] Get JWC PIDs
[jwcctl debug] Done Getting JWC PIDs
[jwcctl debug] ... JWC Container (pid=26626) ...
[jwcctl debug] ... JWC Container running (pid=26626) ...
You can verify that the ZDM service was setup successfully by trying to query an image, in order to so, please execute the following as zdmuser:

```
./zdm_oracle_home/bin/zdmcli query image
```

```
[zdmuser@zdm-servicenode]$ ./zdmcli query image
x.x.x.x.host.com: Audit ID: 2
No image has been configured
```
Configuring Connectivity between ZDM Service Node and Source and Target Database Servers

The ZDM service node must connect securely to the source database server and the target database server to ensure a swift migration process. In order to do so, ssh authentication key pairs must be available (without a passphrase) for the recently created zdmuser. In case the keys are not present or they require a passphrase, you can generate them following our “Generating a Private SSH Key Without a Passphrase” section in our documentation.

https://docs.oracle.com/en/database/oracle/zerodowntime-migration/

Upon creation of the private ssh key files, the following files will be available: id_rsa and id_rsa.pub files. It is recommended as a best practice to rename the id_rsa.pub file to <zdm_service_node_name>.ppk. Finally the public keys must be added to the source database server and the target database server (opc_user_home/.ssh/authorized_keys) files.

Edit the ZDM service node /etc/hosts file, adding the source and target database servers hostname and IP information:

```
#Source
111.1.1.1 source.node-1234 source

#OCI Target
129.1.2.3 target.node-1234-oci target
```

Please bear in mind that the alias of both source and target should be the name without the domain information. Finally, test the connectivity from the ZDM Service node to the source database server and target database server, using the ssh -i command.

```
[zduser@zdm-servicenode ~]# ssh -i zdm_service_tool.ppk opc@source
[zduser@zdm-servicenode ~]# ssh -i zdm_service_tool.ppk opc@target
```

```
[zduser@tool .ssh]$ ssh -i zdm_service_tool.ppk opc@source
The authenticity of host 'source (X.X.X.X)' can't be established.
RSA key fingerprint is ……
RSA key fingerprint is ……
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'source,X.X.X.X' (RSA) to the list of known hosts.
-- WARNING -- This system is for the use of authorized users only. Individuals using this computer system without authority or in excess of their authority are subject to having all their activities on this system monitored and recorded by system personnel. Anyone using this system expressly consents to such monitoring and is advised that if such monitoring reveals possible evidence of criminal activity system personnel may provide the evidence of such monitoring to law enforcement officials.

Last login: from inet-x-x-x-x.com
*** ALERTS::INFO BEGIN ***
ALERTS:: System Alert: [Database User accounts check Failed]
ALERTS:: Please check /var/opt/oracle/log/monitor_alerts/monitor_alerts.log
*** ALERTS::INFO END ***
[opc@source ~]$ exit
```
Configuring Connectivity between the Source and Target Database Servers

The source database server and the target database server must be able to connect via Oracle NET Services (SQL* net) in order to guarantee Data Guard operations. There are two options for connectivity between source and target:

1. The first option to ensure connectivity can be over SCAN, through the source/target SCAN port which must be resolvable from both the source database server and the target database server.

2. The second option to ensure connectivity requires setting up an SSH tunnel from the source database server to the target database server.

Option 1 - SCAN Connectivity

In order to guarantee connectivity via SCAN between the source and target database servers, both /etc/hosts files must be edited.

Source database server /etc/hosts file entry to be added:

```
#Source
111.1.1.1 source.node-1234-oci source
111.1.1.1 source-priv.node-1234-oci source-priv
111.1.1.1 source-vip.node-1234-oci source-vip
111.1.1.1 source-scan.node-1234-oci source-scan
```

Target database server /etc/hosts file entry to be added:

```
#Target
129.1.2.3 target.node-1234-oci target
129.1.2.3 target-scan.node-1234-oci target-scan
```

Target database server /etc/hosts file entry to be added:

```
#Target
129.1.2.3 target.node-1234-oci target
129.1.2.3 target-priv.node-1234-oci target-priv
129.1.2.3 target-vip.node-1234-oci target-vip
129.1.2.3 target-scan.node-1234-oci target-scan
```

```
#Source
111.1.1.1 source.node-1234 source
111.1.1.1 source-scan.node-1234 source-scan
```

```bash
[zmuser@tool .ssh]$ ssh -i zdm_service_tool.ppk opc@target
The authenticity of host 'target (x.x.x.x)' can't be established.
ECDSA key fingerprint is ...
ECDSA key fingerprint is ...
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'target, x.x.x.x' (ECDSA) to the list of known hosts.
Last login: from x.x.x.x
[opc@target ~]$ exit
```
In the case of a source database that is not Grid Infrastructure based, then, the source scan entry in the /etc/hosts target file is not required.

Bear in mind that some of these entries might already be on the /etc/hosts files by default, this is just a sample of how the files should look for ZDM to work properly.

Afterwards, connectivity can be tested via the tnsping utility, trying to reach the target database server from the source database server and viceversa via port 1521.

```
[oracle@source ~]$ tnsping target:1521
[oracle@target ~]$ tnsping source:1521
```

**Option 2 - SSH Tunnel**

In order to setup an SSH tunnel between the source and target database server, please follow the steps outlined on ZDM's product documentation section "Configuring Connectivity between the Source and Target Database Servers" subsection "Option 2".

https://docs.oracle.com/en/database/oracle/zero-downtime-migration/

**Creating a Target Database**

In order to create a placeholder target database, login to your Oracle Cloud account, and access the DB systems tab by clicking on the Bare Metal, VM and Exadata menu.

Click on the Create DB System option.
Select the appropriate values regarding: Compartment, DB System name, Availability domain and Shape.

<table>
<thead>
<tr>
<th>Compartment</th>
<th>DB System name</th>
<th>Availability domain</th>
<th>Shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD-1</td>
<td>Oracle ZDM</td>
<td>AD-2</td>
<td>VM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AD-3</td>
<td>Virtual Machine</td>
</tr>
</tbody>
</table>

Configure the DB system by selecting: Total node count and Oracle Database software edition. Choose the Storage Management Software, bear in mind that Oracle ZDM only supports migrations to Databases which use Oracle Grid Infrastructure. Logical Volume Manager is not supported.
Select **Upload SSH Keys** and Choose a license type.

Specify the network information according to your setup, this includes Hostname prefix, Host domain name and Host and domain URL, this last value is autogenerated and determined by the prefix and the host domain name.
Proceed to provide information related to the target database, this includes: **Database name**, **Database version**, and the **administrator credentials**. Please bear in mind that **Database name** value should be the same as the source database name.

Finally, select **Transaction Processing** as the database workload type and if required change your character set to the desired character set. When you are ready, click on **Create DB System**.
Setting up a Transparent Data Encryption Wallet

Oracle Cloud encrypts all of its databases by default, to do so, Oracle uses its Transparent Data Encryption (TDE) technology. This means, that when a customer migrates a database to the Oracle Cloud, even if the source database was not encrypted, the target database will be encrypted with TDE.

Oracle Zero Downtime Migration supports unencrypted databases as source databases, taking care of the target database encryption during the migration process. Since not all Oracle source databases are encrypted by default, there are several steps that must be taken, specifically if the source database is on 12c release 2 or later. In such cases, a Transparent Data Encryption wallet must be configured prior to performing the migration.

Please visit Oracle Zero Downtime Migration Documentation’s Setting Up the Transparent Data Encryption Wallet section, for a step by step guide on how to perform this task:

https://docs.oracle.com/en/database/oracle/zero-downtime-migration/

Creating Object Storage for Database Backup

Oracle Zero Downtime migration leverages the Object Storage to place the RMAN generated backups of the source database. This backups are transferred from the source database to the object storage via https connectivity. Once ZDM completes this process, it instantiates a standby database leveraging this backup files. Please follow this steps to create an Object Storage for Database Backup.

Login to your Oracle Cloud Account and click on the Object Storage menu and the Object Storage option.
Select your compartment and click on the **Create Bucket** option. It is important to bear in mind that the compartment must be in the same local region where you have created the target placeholder database.

Enter a desired Bucket Name, select the Standard for the Storage Tier and Select Encryption using Oracle Managed Keys. Finally, click on the **Create Bucket** option.
Generating an Authentication Token

Login to your Oracle Cloud Account, click on the **Identity** option and click on the **Users** option.

Select the user you want to use for Object Storage authentication, click the **Auth Tokens** option and click the **Generate Token** option.
Bear in mind that it is important to copy the token when prompted to do so, the token will not afterwards. If you do not copy it now, you will need to create a new one later.
## Preparing the Response File

### Response File Parameters

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TGT_DB_UNIQUE_NAME</td>
<td>Let’s the ZDM service node know the target database db_unique_name value</td>
</tr>
</tbody>
</table>
| MIGRATION_METHOD           | • DG_OSS → ZDM will leverage Data Guard for the migration process and will use the object store for the backups that are required for the standby database instantiation.  
                         | • DG_STORAGEPATH → ZDM will leverage Data Guard for the migration process, when selected, this option specifies that a new backup is required and that the backup location must be specified on the BACKUP_PATH parameter.  
                         | • DG_EXTBACKUP → ZDM will leverage Data Guard for the migration process, this option specifies that existing backup will be used (hence ZDM will not perform a new backup), the backup location must be specified on the BACKUP_PATH parameter.  
                         | • DG_ZDLRA → ZDM will leverage Data Guard for the migration process, this option specifies that the backup location to be used will be the Zero Data Loss Recovery Appliance. This is specific for migrations where ExaCC is the target platform.  
                         | • BACKUP_RESTORE_OSS → ZDM will perform the migration on an offline fashion, using a backup and restore methodology, this method uses the object store as a backup location. This is for source databases that are Standard Edition.  
                         | • BACKUP_RESTORE_NFS → ZDM will perform the migration on an offline fashion, using a backup and restore methodology, this method uses NFS storage as a backup location. This is for source databases that are Standard Edition. |
| PLATFORM_TYPE              | • VMDB → Specifies the target platform for the migration process, this option is valid for Virtual Machine or Bare Metal Migrations.  
                         | • EXACC → Specifies the target platform for the migration process, this option is valid for Exadata Cloud at Customer.  
<pre><code>                     | • EXACS → Specifies the target platform for the migration process, this option is valid for Exadata Cloud Service. |
</code></pre>
<p>| SRC_HTTP_PROXY_URL         | Specifies ssh proxy url for the source database server in case it is required |</p>
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRC_HTTP_PROXY_PORT</td>
<td>Specifies ssh proxy port for the source database server in case it is required.</td>
</tr>
<tr>
<td>SRC_CONFIG_LOCATION</td>
<td>Specifies the ssh configuration file location for the source database server in case it is not present in the default location: <code>&lt;user_home&gt;/ssh/config</code></td>
</tr>
<tr>
<td>SRC_TIMEZONE</td>
<td>Timezone value of the source database, only required for single instance source databases where Grid Infrastructure is not configured.</td>
</tr>
<tr>
<td>SRC_OSS_PROXY_HOST</td>
<td>Specifies proxy url for the source database server in case it is required to connect to the object store.</td>
</tr>
<tr>
<td>SRC_OSS_PROXY_PORT</td>
<td>Specifies proxy port for the source database server in case it is required to connect to the object store.</td>
</tr>
<tr>
<td>TGT_HTTP_PROXY_URL</td>
<td>Specifies ssh proxy url for the target database server in case it is required.</td>
</tr>
<tr>
<td>TGT_HTTP_PROXY_PORT</td>
<td>Specifies ssh proxy port for the target database server in case it is required.</td>
</tr>
<tr>
<td>TGT_CONFIG_LOCATION</td>
<td>Specifies the ssh configuration file location for the target database server in case it is not present in the default location: <code>&lt;user_home&gt;/ssh/config</code></td>
</tr>
<tr>
<td>TGT_SSH_TUNNEL_PORT</td>
<td>In the case of tunneling configuration being present, this parameter specifies the port where the ssh tunnels between source and target for the setup of the Oracle NET Services (SQL * Net) connectivity.</td>
</tr>
<tr>
<td>TGT_OSS_PROXY_HOST</td>
<td>Specifies proxy url for the target database server in case it is required to connect to the object store.</td>
</tr>
<tr>
<td>TGT_OSS_PROXY_PORT</td>
<td>Specifies proxy port for the source database server in case it is required to connect to the object store.</td>
</tr>
<tr>
<td>TGT_DATADG, TGT_REDODG, TGT_RECODG</td>
<td>In case the target database is using ASM for datafiles this value specifies the database data files properties.</td>
</tr>
<tr>
<td>TGT_DATAACFS, TGT_REDOACFS, TGT_RECOACFS</td>
<td>In case the target database is using ACFS for datafiles this value specifies the database data files properties.</td>
</tr>
<tr>
<td>HOST</td>
<td>Specifies the Object Storage Namespace, this value can be found under the cloud console menu &gt; administration &gt; tenancy detail. The tenancy information section show the valid url for the object storage namespace.</td>
</tr>
<tr>
<td>OPC_CONTAINER</td>
<td>Object storage bucket name.</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SRC_ZDLRA_WALLET_LOC</td>
<td>Wallet location in the source database server when the Zero Data Loss Recovery Appliance is used as backup medium.</td>
</tr>
<tr>
<td>TGT_ZDLRA_WALLET_LOC</td>
<td>Wallet location in the target database server when the Zero Data Loss Recovery Appliance is used as backup medium.</td>
</tr>
<tr>
<td>ZDLRA_CRED_ALIAS</td>
<td>Wallet credential alias when the Zero Data Loss Recovery Appliance is used as backup medium.</td>
</tr>
<tr>
<td>BACKUP_PATH</td>
<td>When the following options are specified on the MIGRATION_METHOD field -&gt; DG_STORAGEPATH</td>
</tr>
<tr>
<td>SKIP_FALLBACK</td>
<td>Specifies if redo logs from the target database are not be sent to the source database upon role swap. The default value is FALSE.</td>
</tr>
<tr>
<td>TGT_RETAIN_DB_UNIQUE_NAME</td>
<td>***</td>
</tr>
<tr>
<td>SHUTDOWN_SRC</td>
<td>Specifies if the source database is to be shutdown after the migration, if the value is set to TRUE it performs the latter, if it is set to FALSE it does not.</td>
</tr>
<tr>
<td>SRC_RMAN_CHANNELS</td>
<td>Specifies the number of Recovery Manager Channels at the source database server used to perform RMAN backups, default value is 10.</td>
</tr>
<tr>
<td>TGT_RMAN_CHANNELS</td>
<td>Specifies the number of Recovery Manager Channels at the target database server used to perform RMAN backups, default value is 10.</td>
</tr>
<tr>
<td>ZDM_BACKUP_FULL_SRC_MONITORING_INTERVAL</td>
<td>specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_BACKUP_FULL_SRC migration job phase.</td>
</tr>
<tr>
<td>ZDM_BACKUP_INCREMENTAL_SRC_MONITORING_INTERVAL</td>
<td>specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_BACKUP_INCREMENTAL_SRC migration job phase.</td>
</tr>
<tr>
<td>ZDM_BACKUP_DIFFERENTIAL_SRC_MONITORING_INTERVAL</td>
<td>specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_BACKUP_DIFFERENTIAL_SRC migration job phase.</td>
</tr>
<tr>
<td>ZDM_CLONE_TGT_MONITORING_INTERVAL</td>
<td>specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_CLONE_TGT migration job phase.</td>
</tr>
<tr>
<td>ZDM_OSS_RECOVER_TGT_MONITORING_INTERVAL</td>
<td>specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_OSS_RECOVER_TGT migration job phase.</td>
</tr>
<tr>
<td>ZDM_OSS_RESTORE_TGT_MONITORING_INTERVAL</td>
<td>specifies the time interval, in minutes, at which to monitor and report the progress of the ZDM_OSS_RESTORE_TGT migration job phase.</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SKIP_SRC_SERVICE_RETENTION</td>
<td>lets the user skip source database service retention. Default value is FALSE.</td>
</tr>
<tr>
<td>ZDM_BACKUP_RETENTION_WINDOW</td>
<td>lets the user specify the number of days after which database backups created by Zero Downtime Migration become obsolete. The default is 60 (days).</td>
</tr>
<tr>
<td>ZDM_OPC_RETRY_WAIT_TIME</td>
<td>lets the user specify the Object Store retry wait time in seconds. The default value is 529 (seconds).</td>
</tr>
<tr>
<td>ZDM_OPC_RETRY_COUNT</td>
<td>lets the user specify how many times to retry the Object Store. The default value is 0 (zero).</td>
</tr>
<tr>
<td>ZDM_CURL_LOCATION</td>
<td>lets the user specify a custom location for CURL binary on the source. The default value is /usr/bin/curl.</td>
</tr>
<tr>
<td>ZDM_LOG_OSS_PAR_URL</td>
<td>lets the user specify a pre-authenticated URL to which to upload log files. The logs also capture the current phase as well as the execution status of the phase.</td>
</tr>
<tr>
<td>ZDM_SRC_TNS_ADMIN</td>
<td>lets the user specify a custom location for TNS_ADMIN on the source database when there is no Grid Infrastructure.</td>
</tr>
<tr>
<td>SRC_SSH_RETRY_TIMEOUT</td>
<td></td>
</tr>
<tr>
<td>TGT_SSH_RETRY_TIMEOUT</td>
<td></td>
</tr>
<tr>
<td>TGT_SKIP_DATAPATCH</td>
<td>lets the user specify whether or not Zero Downtime Migration runs the datapatch utility on the target database as part of the post-migration tasks. The default value is FALSE, which allows Zero Downtime Migration to run datapatch.</td>
</tr>
<tr>
<td>MAX_DATAPATCH_DURATION_MINS</td>
<td>lets the user configure a timeout value, in minutes, after which, if the datapatch utility has failed to complete, then the operation is stopped. There is no default value. By default, Zero Downtime Migration waits until datapatch completes.</td>
</tr>
<tr>
<td>DATAPATCH_WITH_ONE_INSTANCE_RUNNING</td>
<td>when set to TRUE, lets Zero Downtime Migration stop all instances except one running on the target database server when the datapatch utility runs. When datapatch completes, all of the stopped instances are started. The default value is FALSE.</td>
</tr>
</tbody>
</table>
DATABASE MIGRATION WITH ZDM

Performing a Database Migration pre-check process

To perform a database migration pre-check process, the zdmcli migrate database command with the -eval flag must be executed. For example, this is how the command would look like:

```
./zdmcli migrate database -sourcedb db1_phx -sourcenode sourcezdm -srcauth zdmauth -srcarg1 user:opc -srcarg2:/home/zdmuser/.ssh/zdm-servicenode.ppk -srcarg3 sudo_location:/usr/bin/sudo -targetnode targetzdm -targethome /u01/app/oracle/product/18.0.1 -backupuser bkpuser -rsp /u01/response/zdm_template.rsp -tgtauth zdmauth -tgtaarg1 user:opc -tgtaarg2 identity_file:/home/zdmuser/.ssh/zdm-spk -tgtaarg3 sudo_location:/usr/bin/sudo -ignore ALL -eval
```

<table>
<thead>
<tr>
<th>FLAG</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>-sourcedb or -sourcesid</td>
<td>Source database name, in this case the database name is db1_phx. If the source database is a single instance database deployed on a non Grid Infrastructure environment, the correct flag should be -sourcesid alongside the database name.</td>
</tr>
<tr>
<td>-sourcenode</td>
<td>Source database server host name, in this case it is sourcezdm.</td>
</tr>
<tr>
<td>-srcauth</td>
<td>Authentication plug-in name required to access the source database server. This let’s ZDM know that a number n of plugin arguments are to come, specifying the specific authentications to be followed.</td>
</tr>
<tr>
<td>-srcarg1</td>
<td>Plugin argument, in this case it specifies the opc user.</td>
</tr>
<tr>
<td>-srcarg2</td>
<td>Plugin argument, in this case it specifies the location of the public keys: /home/zdmuser/.ssh/zdm-servicenode.ppk.</td>
</tr>
<tr>
<td>-srcarg3</td>
<td>Plugin argument, in this case it specifies the sudo location: /usr/bin/sudo.</td>
</tr>
<tr>
<td>-backupuser</td>
<td>Username with database backup/restore privileges, when the migration starts, ZDM will request for the password for this user, please provide the authentication token for the object storage. In this case the user is bkpuser.</td>
</tr>
<tr>
<td>-rsp</td>
<td>Response file location, in this case the location is: /u01/response/zdm_template.rsp.</td>
</tr>
<tr>
<td>-tgtauth</td>
<td>Authentication plug-in required to access the target database server. This let’s ZDM know that a number n of plugin arguments are to come, specifying the specific authentications to be followed.</td>
</tr>
<tr>
<td>-tgtaarg1</td>
<td>Plugin argument, in this case it specifies the opc user.</td>
</tr>
</tbody>
</table>
-tgtarg2

Plugin argument, in this case it specifies the location of the public keys:
/home/zdmuser/.ssh/zdm-servicenode.ppk

-tgtarg3

Plugin argument, in this case it specifies the sudo location:
/usr/bin/sudo

-ignore

Ignore checks (ALL | WARNING | PATCH_CHECK)

-eval

Perform a dry run or evaluation of the migration without actually migrating the database. This allows to be sure that all migration phases will run smoothly. If there is any corrections to be made prior to migration, ZDM will alert the user.

Upon execution of the command, and password prompt, a job id will be generated:

```
./zdmcli migrate database -sourcedb db1_phx -sourcezdm -sourcenode sourcezdm -srauth zdmauth
-srcarg1 user:opc -srcarg2:/home/zdmuser/.ssh/zdm-servicenode.ppk
-srcarg3 sudo_location:/usr/bin/sudo -targetzdm -targetnode targetzdm
-targethome /u01/app/oracle/product/18.01
-backupuser bkpuser -rspsp /u01/response/zdm_template.rsp -tgtauth zdmauth -tgtarg1 user:opc
-tgtarg2 identity_file:/home/zdmuser/.ssh/id_rsa -tgtarg3 sudo_location:/usr/bin/sudo
-ignore ALL -eval
```

Enter source database db1_phx1fn SYS password:
Enter user "xxxxx" password:
Operation "zdmcli migrate database" scheduled with the job ID "4".

The second password that is requested, as per the backup user, refers to the Authentication token for the object store.
You can query the command progress with the following command:
```
./zdmcli query job -jobid 4
```
Performing a Database Migration

To perform a database migration process, the zdmcli migrate database command must be executed, now without the -eval flag. For example, this is how the command would look like:

```
./zdmcli migrate database -sourcedb db1_phx -sourcenode sourcezdm -srcauth zdmauth -srcarg1 user:opc -srcarg2 /zdmuser/.ssh/zdm-servicecnode.ppk -srcarg3 sudo_location:/usr/bin/sudo -targetnode targetzdm -targethome /u01/app/oracle/product/18.0.0user xxxxxx -rsp /u01/response/zdm_template.rsp -tgtauth zdmauth -tgtrarg1 user:opc -tgtrarg2 identity_file:/home/zdmuser/.ssh/zdm-servicenode.ppk -tgtrarg3 sudo_location:/usr/bin/sudo -ignore ALL -eval
```

Scheduled job execution start time: 2020-X. Equivalent local time: 2020-X

Current status: SUCCEEDED

Result file path: "/u01/app/oracle/zdmbase/chkbase/scheduled/job-4-2020-05.log"

Job execution start time: 2020-05-06 20:11:49
Job execution end time: 2020-05-06 20:15:15
Job execution elapsed time: 3 minutes 25 seconds

ZDM_GET_SRC_INFO .......... COMPLETED
ZDM_GET_TGT_INFO .......... COMPLETED
ZDM_SETUP_SRC ............. COMPLETED
ZDM_SETUP_TGT ............. COMPLETED
ZDM_GEN_RMAN_PASSWD ....... COMPLETED
ZDM_PREUSERACTIONS ........ COMPLETED
ZDM_PREUSERACTIONS_TGT .... COMPLETED
ZDM_VALIDATE_SRC .......... COMPLETED
ZDM_VALIDATE_TGT .......... COMPLETED

The second password that is requested, as per the backup user, refers to the Authentication token for the object store.

Upon execution of the command, and password prompt, a job id will be generated, you can query the command progress with the following command:

```
./zdmcli query job -jobid 5
```

Enter source database sourcedb db1_SYS password:
Enter user "xxxxxxx" password:
Operation "zdmcli migrate database" scheduled with the job ID "5".

Upon completion of the migration, this is how the output would look like:

```
zdmuser@zdm-servicenode bin]$ ./zdmcli query job -jobid 5
zdm-servicenode.x.x.x: Audit ID: 52
Job ID: 5
User: zdmuser
Client: zdm-servicenode
Scheduled job command: "zdmcli migrate database -sourcedb db1_phx1fn -sourcenode sourcezdm -srcauth zdmauth -srcauth2 opc -srcauth3 zdm -sourcedbname db1 -sourcedbhome /u01/app/oracle/product/18.0.0.0/dbhome_1 -srcauth3 opc -srcauth2 identity_file:/home/zdmuser/.ssh/zdm-servicenode.ppk -srcauth1 user:opc -srcauth2 identity_file:/home/zdmuser/.ssh/zdm-servicenode.ppk -srcauth1 user:opc -srcarg2 identity_file:/home/zdmuser/.ssh/zdm-servicenode.ppk -srcarg3 sudo_location:/usr/bin/sudo -targetnode targetzdm -targethome /u01/app/oracle/product/18.0.0.0/dbhome_1 -targetauth zdmauth -targetarg1 user:opc -targetarg2 identity_file:/home/zdmuser/.ssh/zdm-servicenode.ppk -targetarg3 sudo_location:/usr/bin/sudo -ignore ALL"

Current status: SUCCEEDED
Result file path: "/u01/app/oracle/zdmbase/chkbase/scheduled/job-5-2020-05.log"
Job execution start time: 2020-05-06 20:24:19
Job execution end time: 2020-05-06 20:45:15
Job execution elapsed time: 20 minutes 55 seconds

ZDM_GET_SRC_INFO .............. COMPLETED
ZDM_GET_TGT_INFO .............. COMPLETED
ZDM_SETUP_SRC ................. COMPLETED
ZDM_SETUP_TGT ................. COMPLETED
ZDM_GEN_RMAN_PASSWD ........... COMPLETED
ZDM_PREUSERACTIONS ............ COMPLETED
ZDM_PREUSERACTIONS_TGT ........ COMPLETED
ZDM_VALIDATE_SRC .............. COMPLETED
ZDM_VALIDATE_TGT .............. COMPLETED
ZDM_OBC_INST_SRC .............. COMPLETED
ZDM_OBC_INST_TGT .............. COMPLETED
ZDM_BACKUP_FULL_SRC .......... COMPLETED
ZDM_BACKUP_INCREMENTAL_SRC .... COMPLETED
ZDM_DISCOVER_SRC .............. COMPLETED
ZDM_COPYFILES ................. COMPLETED
ZDM_PREPARE_TGT ............... COMPLETED
ZDM_SETUP_TDE_TGT ............. COMPLETED
ZDM_CLONE_TGT ................. COMPLETED
ZDM_FINALIZE_TGT .............. COMPLETED
ZDM_CONFIGURE_DG_SRC .......... COMPLETED
ZDM_SWITCHOVER_SRC ............ COMPLETED
ZDM_SWITCHOVER_TGT ............ COMPLETED
ZDM_NONCDBTODPDB_PRECHECK ...... COMPLETED
ZDM_NONCDBTODPDB_CONVERSION .... COMPLETED
ZDM_POSTUSERACTIONS ........... COMPLETED
ZDM_POSTUSERACTIONS_TGT ...... COMPLETED
ZDM_CLEANUP_SRC ............... COMPLETED
ZDM_CLEANUP_TGT ............... COMPLETED
```
# FREQUENTLY ASKED QUESTIONS & COMMON ISSUES

## F.A.Q

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>ANSWER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. What authentication method should be used for object storage authentication?</td>
<td>An authentication token can be created from the OCI Console. You can Navigate to the main Menu &gt; Identity &gt; Users. Then, you must select the user you want to use for the object storage authentication, click on the Auth Token option and click on generate token. Please copy the token upon creation, since it will not be accessible afterwards and if you want to access it you will need to generate a new one.</td>
</tr>
<tr>
<td>2. What method of migration should be used for SE (standard edition) databases migration?</td>
<td>ZDM allows for Standard Edition as source databases. The migration method recommended is BACKUP_RESTORE_OSS. This method uses the object store for the backup files.</td>
</tr>
<tr>
<td>3. Can ZDM be used to migrate database to on-premises physical servers?</td>
<td>Future versions of ZDM will allow for on-premises migration, providing that the target platform is an engineered system like an Exadata Database Machine.</td>
</tr>
<tr>
<td>4. Can we use ZDM to migrate databases that are hosted on different Linux versions than that target OS version?</td>
<td>Yes, ZDM allows for cross OS version (Linux only) migration.</td>
</tr>
<tr>
<td>5. Can source and target database be on different versions?</td>
<td>No, Oracle ZDM allows only for migration between databases of the same versions. The target database can be at a higher patch level, but then datapatch must be run post migration on the target database.</td>
</tr>
<tr>
<td>6. Does ZDM perform non-cdb to cdb conversion while migration to the Oracle Cloud?</td>
<td>No, Oracle ZDM does not allow for this type of migration. Source databases must be non-cdb for non-cdb targets in the Oracle Cloud or source databases must be cdb for cdb targets in the Oracle Cloud. Future releases of Oracle ZDM will include non-cdb to cdb conversion.</td>
</tr>
<tr>
<td>7. Does ZDM support cross platform migration?</td>
<td>No, Oracle ZDM only allows for source databases that are hosted on Linux platforms. Future releases of ZDM will include cross platform migration, while leveraging Oracle products and tools like GoldenGate and DataPump. For more information on available tools for cross platform migration please visit <a href="http://www.oracle.com/goto/move">www.oracle.com/goto/move</a></td>
</tr>
<tr>
<td>8. Can ZDM used to migrate databases hosted in one OCI region to another OCI region?</td>
<td>Yes, Oracle ZDM allows for cross OCI region migration.</td>
</tr>
<tr>
<td>9. Can ZDM used to migrate from OCI-C Database to OCI?</td>
<td>Yes, one of Oracle ZDM’s supported sources is databases hosted in OCI-Classic.</td>
</tr>
<tr>
<td>Question</td>
<td>Answer</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>10. Will ZDM migrate Apex application reports?</td>
<td>No, ZDM does not perform this task.</td>
</tr>
<tr>
<td>11. Can ZDM be instructed to not perform the switchover, basically just setting up DG?</td>
<td>Yes, Oracle ZDM allows to pause the migration process on any given stage, and hence, the migration process can be paused before the role swap and switchover phase. Upon executing the zdm migrate database command, the <code>--pauseafter</code> flag must be entered with the desired stage for pausing, in this case, <code>ZDM_CONFIGURE_DG_SRC</code></td>
</tr>
<tr>
<td>12. What are the supported OS versions for ZDM binaries?</td>
<td>Oracle ZDM runs on Oracle Linux 7</td>
</tr>
</tbody>
</table>
| 13. Do you provide interim patches for the ZDM, in-case if we are hitting a bug? | Oracle ZDM will release patches, addressing issues when required, if you are having an issue please file a service request and contact product management.  
In order to expedite resolution time, please follow the instructions in My Oracle Support Document: [SRDC – Data Collection for Database Migration Using Zero Downtime Migration (ZDM) (DOC ID 2595205.1)](https://support.oracle.com/epmos/faces/DocContentDisplay?id=2595205.1) |
| 14. Can ZDM be installed on the source or target server?               | Yes, Oracle ZDM can be installed on any location, however, as a best practices is better to have ZDM run on a separate node. Further mode, Oracle GI must not be present on the node where the Oracle ZDM service node is to be deployed. |
| 15. What are the OS packages required for ZDM install?                  | The following packages are required:  
- `glibc-devel`  
- `expect`  
- `unzip`  
- `libaio`  
- `oraclelinux-developer-release-el7` |
| 16. Is it required to have the same databases patches between source and target databases? | This is recommended as a best practice, however, the target database can be at a higher patch level, but then datapatch must be run post migration on the target database. |
| 17. What is the supported key format for ZDM to authenticate with source and target? | The supported format is RSA |
| 18. What is storage types are supported according to the target cloud platform? | • Oracle Database Cloud Service Bare Metal  
  1. OCI Object Storage  
• Oracle Database Cloud Service Virtual Machine  
  1. OCI Object Storage  
• Exadata Cloud Service  
  1. Oracle Zero Data Loss Recovery Appliance  
  2. NFS Storage  
• Exadata Cloud at Customer  
  1. Oracle Zero Data Loss Recovery Appliance |
19. Do we need a license for ZDM software?

Oracle ZDM is free of charge and follows the following licensing agreement: https://www.oracle.com/downloads/licenses/sqldev-license.html#licenseContent

20. What are the supported target type's for migration?

- Oracle Database Cloud Service Bare Metal
- Oracle Database Cloud Service Virtual Machine
- Exadata Cloud Service
- Exadata Cloud at Customer

Known Issues

All common issues are documented and updated periodically in Oracle Zero Downtime Migration’s documentation, specifically on the product release note, Known Issues section:

https://docs.oracle.com/en/database/oracle/zero-downtime-migration/

Troubleshooting & Other Resources

For Oracle ZDM log review:

- ZDM Server Logs:
  - Check - $ZDM_BASE/crsdata/<zdm_service_node>/rhp/rhpserver.log.0
- Check source node logs
  - <oracle_base>/zdm/zdm_<src_db_name>_<job_id>/zdm/log
- Check target node logs
  - <oracle_base>/zdm/zdm_<tgt_db_name>_<job_id>/zdm/log

For all Oracle Support Service Requests related to Zero Downtime Migration, please be sure to follow the instructions in My Oracle Support Document:

- SRDC – Data Collection for Database Migration Using Zero Downtime Migration (ZDM) (DOC ID 2595205.1)
- https://support.oracle.com/epmos/faces/DocContentDisplay?id=2595205.1

Please review our Maximum Availability Architecture document for best practices related to Zero Downtime Migration.

- MAA Practices for Cloud Migration Using ZDM (Doc ID 2562063.1)
- https://support.oracle.com/epmos/faces/DocContentDisplay?id=2562063.1