Oracle Zero Downtime Migration 21c

Step by Step Guide – Logical Migration and In-Flight Upgrade from On-Premises to DBCS and ExaCS

July 2021 | Version 1.1
Copyright © 2022, Oracle and/or its affiliates
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  
**Zero Downtime Migration** | 4  
  Architecture | 4  
  Supported Configurations | 5  
**Zero Downtime Migration Service Host** | 5  
  Zero Downtime Migration Service Host Requirements | 5  
  ZDM Service Host Installation | 6  
  ZDM Service Host Port Requirements | 11  
**Source Database** | 12  
  Source Database Requirements | 12  
  Source Database Port Requirements | 16  
**Target Database** | 17  
  Target Database Requirements | 17  
  Target Database Port Requirements | 21  
**Connectivity** | 21  
  SSH Key Pair | 21  
  Connectivity between the ZDM Service Host and the Source and Target Database Servers | 22  
    Adding ZDM Host Public Keys to the Source Database Server | 22  
    Adding ZDM Host Public Keys to the Target Database Server | 22  
    Testing Connectivity from the ZDM Service Host to Source and Target Database Server | 23  
  Authentication Token | 23  
  OCI CLI Command Line Tool | 23  
  API Signing Public Key | 24  
  Connectivity between Source and Target Database Servers | 24  
    Option 1 – SCAN Connectivity | 24  
    Option 2 – SSH Connectivity | 25  
**Backup Location** | 26  
  Object Storage Requirements | 26  
**GoldenGate Hub** | 27  
  GoldenGate Hub Requirements | 27  
**Preparing the Response File** | 28  
  Response File Parameters used in this Guide | 28  
**Logical Migration to DBCS and ExaCS with ZDM** | 33  
  Performing a Test Database Migration on Evaluation Mode | 33  
  Performing a Database Migration | 34  
**Known Issues** | 37  
**Troubleshooting & Other Resources** | 37
INTRODUCTION

Oracle customers are moving Oracle workloads into the Oracle Cloud or onto Engineered Systems at a growingly rapid pace. However, migrating workloads has been a source of challenges for many years. In particular, migrating database workloads from one system to another or into the Cloud is easier said than done.

Based on years of experience migrating Oracle workloads, Oracle has developed Zero Downtime Migration (ZDM). ZDM is Oracle's premier solution for a simplified and automated migration experience, providing zero to negligible downtime for the production system and depending on the migration scenario. ZDM allows you to directly and seamlessly migrate your on-premises Oracle Databases to and between any Oracle-owned infrastructure, including Exadata Database Machine On-Premises, Exadata Cloud at Customer (ExaC@C), and Oracle Cloud Infrastructure. Oracle ZDM supports a wide range of Oracle Database versions and, as the name implies, ensures minimal to no production database impact during the migration.

ZDM follows Oracle Maximum Availability Architecture (MAA) principles and incorporates products such as GoldenGate and Data Guard to ensure High Availability and an online migration workflow that leverages technologies such as the Recovery Manager, Data Pump, and Database Links.

This technical brief is a step-by-step guide for migrating your on-premises Oracle Databases to the Oracle Cloud with Zero Downtime Migration's new Logical workflow. The scenario used for this migration comprises a Source Database running on Compute via Marketplace to emulate an On-Premises environment and a Target Database running on Oracle Cloud Infrastructure DBCS Virtual Machines; The base process is the same for a Target Database running on Oracle Cloud Infrastructure Exadata Cloud Service.

Oracle ZDM will run on a separate node and connect to both Source and Target to perform the migration. This guide will cover all requirements related to installing the Oracle ZDM service host, the Source Database to be migrated, the Target Database recipient of the migration process, the backup and networking used. The migration process will be dissected and done in a step-by-step fashion. This guide will answer the most frequently asked questions regarding the product and the overall migration process.

The Source Database will be an 11.2.0.4 Oracle Database, and the Target Database will be a 19c Oracle Database. This guide will cover the migration process while providing an in-flight upgrade of the Source Database.

For more information on Oracle Zero Downtime Migration, please visit ZDM's product website.

1 http://oracle.com/goto/maa
2 http://www.oracle.com/goto/zdm
ZERO DOWNTIME MIGRATION

Architecture

Oracle Zero Downtime Migration (ZDM) is the Oracle Maximum Availability Architecture (MAA)-recommended solution to migrate Oracle Databases to the Oracle Cloud. ZDM’s inherent design keeps in mind the migration process as straightforward as possible and to ensures the most negligible impact on production workloads. The Source Database to be migrated can be on-premises, deployed on Oracle Public Cloud Gen 1 or Oracle Cloud Infrastructure. The Target Database deployment can be in a Database Cloud Service on Oracle Cloud Infrastructure (OCI) Virtual Machine, Exadata Cloud Service, Exadata Cloud at Customer, or Autonomous Database. ZDM automates the entire migration process, reducing the chance of human errors. ZDM leverages Oracle Database-integrated high availability (HA) technologies such as Oracle Data Guard and GoldenGate and follows all MAA best practices that ensure no significant downtime of production environments. Oracle ZDM supports both Physical and Logical Migration workflows. This technical brief covers a step-by-step guide for the Logical Migration Workflow leveraging the Object Storage as a backup location.

Figure 1 – Step-by-Step Logical Offline Migration with Data Pump and Backup Location
Step 1, Download and Configure ZDM. Step 2, ZDM Performs Validations. Step 3, ZDM Connects to Backup Location. Step 4, ZDM Exports Via Data Pump from Source to Backup Location. Step 5, ZDM Imports Data Dump Files from Backup Location to Target. Step 6, ZDM Instantiates Target Database. Step 7, ZDM Switches Over and Finalizes the Migration Process.

Figure 2 – Step-by-Step Logical Online Migration with Data Pump, GoldenGate and Backup Location
Step 1, Download & Configure ZDM. Step 2, ZDM Starts Database Migration. Step 3, ZDM Connects to Source, Target and Backup Location. Step 4, ZDM Configures GoldenGate and Captures Source Transactions. Step 5, ZDM Exports via Data Pump from Source Location. Step 6, ZDM Imports Data Pump Files from Backup Location to Target. Step 7, ZDM Configures GoldenGate and Starts Applying changes. Step 8, ZDM Switches Over and Finalizes the Migration Process.
Supported Configurations

Oracle ZDM supports Oracle Database versions 11.2.0.4, 12.1.0.2, 12.2.0.1, 18c, 19c & 21c. ZDM’s physical migration workflow requires the Source and Target Databases to be in the same database release. Starting with ZDM 21c and introducing the Logical Migration workflow, ZDM now supports database cross-version migration, thus providing an in-flight upgrade while migrating to the Oracle Cloud.

Oracle ZDM supports Oracle Databases hosted on Linux operating systems. Oracle ZDM supports single-instance databases, RAC One Node databases, or RAC databases as sources. Oracle ZDM supports Oracle Database Enterprise & Standard Edition as Source Databases.

Oracle ZDM allows the Source Database to be a non-CDB or a container database (CDB) with one or more Pluggable Databases (PDBs). Starting with release 21c, Oracle ZDM allows for non-CDB databases to be migrated to Pluggable Databases on the fly, allowing for complete conversion and adding more versatility to the migration workflow.

ZERO DOWNTIME MIGRATION SERVICE HOST

Zero Downtime Migration Service Host Requirements

Oracle Zero Downtime Migration installation must take place on a separate host, which must fulfill the following requirements:

- Linux host running on Oracle 7 (must be this OS version).
- 100 GB of free storage space
- A zdm group and a zdmuser as part of this group, please create them as follows:
  - [root@zdm-servicenode ~]# groupadd zdm -g 1001
  - [root@zdm-servicenode ~]# useradd zdmuser -g 1001
- Following packages must be installed:
  - glibc-devel
  - expect
  - unzip
  - libaio
  - oraclelinux-developer-release-el7
- No Oracle Grid Infrastructure running on it.
- All host names and IP addresses to be used must be present as entries at /etc/hosts


---


2 Step by Step Guide – Logical Migration and In-Flight Upgrade from On-Premises to DBCS and ExaCS | Oracle Zero Downtime Migration | Version 1.1

Copyright © 2022, Oracle and/or its affiliates | Public
The ZDM software can be:

- Installed manually on-premises.
- Installed manually on OCI.

This Step-by-Step Guide will cover the manual installation of the ZDM service host, including a thorough description of all necessary instructions about the deployment and configuration.

**ZDM Service Host Installation**

In the web console, click the navigation menu on the upper left side and select Compute > Instances. Click on “Create Instance”. For the purposes of this guide, we will choose the name “zdmhost” and Oracle Linux 7.9.

![Figure 3 – Screenshot for “Create Compute Instance” window on Oracle Cloud.](image)

Click on “Specify a custom boot volume size” and put 200 GB as boot volume size. With that, we will have the required 100 GB free space:

![Figure 4 – Screenshot for “Create Compute Instance” window, Boot Volume options, on Oracle Cloud.](image)
Fill out the remaining needed information and click Create. After just a few minutes, you will be able to log in to the compute instance.


![Oracle Zero Downtime Migration](https://www.oracle.com/database/technologies/)

**Figure 5 – Screenshot for “Oracle Zero Downtime Migration Download” window on Oracle Zero Downtime Migration’s product page.**

Accept the Oracle License Agreement [https://www.oracle.com/technetwork/licenses/sqldev-license-152021.html#licenseContent](https://www.oracle.com/technetwork/licenses/sqldev-license-152021.html#licenseContent) and download the ZDM binaries on the ZDM Service host.

Log in to the ZDM host via ssh:

1. Extend the file system to make the 200 GB available. As root user:

   ```
   [opc@zdmhost ~]$ sudo -s
   [root@zdmhost opc]# df -h
   Filesystem Size Used Avail Use% Mounted on
   /dev/sda3 39G 3.3G 36G 9% /
   [root@zdmhost opc]# /usr/libexec/oci-growfs -y
   [root@zdmhost opc]# df -h
   Filesystem Size Used Avail Use% Mounted on
   /dev/sda3 192G 3.3G 189G 2% /
   ```

2. Create a new group, user, and the needed directories. As root user:

   ```
   [root@zdmhost opc]# groupadd zdm
   [root@zdmhost opc]# useradd -g zdm zdmuser
   [root@zdmhost opc]# mkdir -p /home/zdmuser/zdminstall
   [root@zdmhost opc]# mkdir /home/zdmuser/zdmhome
   [root@zdmhost opc]# mkdir /home/zdmuser/zdmbase
   [root@zdmhost opc]# chown -R zdmuser:zdm /home/zdmuser/
   ```
3 Install the required software packages. As root user:

```bash
[root@zdmhost opc]# yum -y install \
glibc-devel \nexpect \nunzip \nlibaio 
oraclelinux-developer-release-el7
[root@zdmhost opc]# yum list installed glibc-devel expect unzip libaio oraclelinux-developer-release-el7.
```

<table>
<thead>
<tr>
<th>Installed Packages</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>expect.x86_64</td>
<td>5.45-14.el7_1</td>
</tr>
<tr>
<td>@ol7_latest-x86_64</td>
<td></td>
</tr>
<tr>
<td>libaio.x86_64</td>
<td>0.3.109-13.el7</td>
</tr>
<tr>
<td>@anaconda/7.9</td>
<td></td>
</tr>
<tr>
<td>oraclelinux-developer-release-el7.x86_64</td>
<td>1.0-6.el7</td>
</tr>
<tr>
<td>@ol7_latest</td>
<td></td>
</tr>
<tr>
<td>unzip.x86_64</td>
<td>6.0-21.el7</td>
</tr>
<tr>
<td>@anaconda/7.9</td>
<td></td>
</tr>
</tbody>
</table>

4 Download the ZDM software version 21c (zdm21.2.zip) and copy the zip file to the ZDM host into the /home/zdmuser/zdminstall/ directory. Change the owner of the zip file to zdmuser. As root user:

```bash
[root@zdmhost opc]# mv /home/opc/zdm21.2.zip /home/zdmuser/zdminstall
[root@zdmhost opc]# cd /home/zdmuser/zdminstall
[root@zdmhost zdminstall]# ll
-rw-rw-r--. 1 opc opc 768112810 May  7 02:34 zdm21.2.zip
[root@zdmhost zdminstall]# chown zdmuser:zdm /home/zdmuser/zdminstall/zdm21.2.zip
```

5 Install the ZDM software. As zdmuser:

```bash
root@zdmhost zdminstall]# su - zdmuser
[zdmmuser@zdmhost ~]# echo "ORACLE_HOME=/home/zdmuser/zdmmhome; export ORACLE_HOME" >> ~/.bashrc
[zdmmuser@zdmhost ~]# echo "ORACLE_BASE=/home/zdmuser/zdmbase; export ORACLE_BASE" >> ~/.bashrc
[zdmmuser@zdmhost ~]# echo "ZDM_BASE=$ORACLE_BASE; export ZDM_BASE" >> ~/.bashrc
[zdmmuser@zdmhost ~]# echo "ZDM_HOME=/home/zdmuser/zdmmhome; export ZDM_HOME" >> ~/.bashrc
[zdmmuser@zdmhost ~]# echo "ZDM_INSTALL_LOC=/home/zdmuser/zdminstall; export ZDM_INSTALL_LOC" >> ~/.bashrc
[zdmmuser@zdmhost ~]# cat ~/.bashrc
ORACLE_HOME=/home/zdmuser/zdmmhome; export ORACLE_HOME
ORACLE_BASE=/home/zdmuser/zdmbase; export ORACLE_BASE
ZDM_BASE=$ORACLE_BASE; export ZDM_BASE
```
ZDM_HOME=/home/zdmuser/zdmhome; export ZDM_HOME
ZDM_INSTALL_LOC=/home/zdmuser/zdminstall; export ZDM_INSTALL_LOC

[zdmuser@zdmhost ~]$ source ~/.bashrc

[zdmuser@zdmhost ~]$ cd /home/zdmuser/zdminstall/

[zdmuser@zdmhost zdminstall]$ unzip zdm21.2.zip

[zdmuser@zdmhost zdminstall]$ cd zdm21.2

-- Proceed to execute ZDM’s installation script zdmuser:

[zdmuser@zdmhost zdm21.2]$ ./zdminstall.sh setup

oraclehome=${ZDM_HOME} \
oraclebase=${ZDM_BASE} \
ziploc=./zdm_home.zip -zdm

ZDM kit home: /home/zdmuser/zdminstall/zdm21.2
/home/zdmuser/zdminstall/zdm21.2

Unzipping shiphome to ZDM home...

Unzipping shiphome...

Shiphome unzipped successfully.

##### Performing GridHome Software Only Installation #####

Installation log location: 
/home/zdmuser/zdmbase/crsdata/zdmhost/rhp/logs/runInstaller_1620355645.out

making dir /home/zdmuser/zdmbase/crsdata/zdmhost/rhp/conf

Generating Preference file

/generate /home/zdmuser/zdmbase/crsdata/zdmhost/rhp/conf/rhp.pref

Using port 8897 for MySQL

Generating Root Certificate

Cluster root certificate generated successfully.

Generating CA CERTS file

spawn /home/zdmuser/zdmhome/bin/crskeytoolctl -copycacerts -filestore
/home/zdmuser/zdmbase/crsdata/zdmhost/security

Enter JRE cacerts truststore password:

JRE cacerts copied to file [/home/zdmuser/zdmbase/crsdata/zdmhost/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
---------------------------------------
Storing to wallet
---------------------------------------
cacerts crskeytoolctl.log cwallet.sso cwallet.sso.lck
---------------------------------------
Generating random password
---------------------------------------
-rw-------. 1 zdmuser zdm 4325 May  7 02:47 /home/zdmuser/zdmbase/crsdata/zdmhost/security/cwallet.sso
-rw-------. 1 zdmuser zdm 4325 May  7 02:47 /home/zdmuser/zdmbase/crsdata/zdmhost/security/cwallet.sso
RHP_PT.ZDM21_LINUX.X64_210228.2
rhpctl working
label_date is: 210228.2
---------------------------------------
Setting up MySQL...
---------------------------------------
mysqld will log errors to
/home/zdmuser/zdmbase/crsdata/zdmhost/rhp/mysql/metadata/mysql-error.log
mysqld is running as pid 20936
---------------------------------------
Storing to wallet
---------------------------------------
cacerts crskeytoolctl.log cwallet.sso cwallet.sso.lck
---------------------------------------
Generating random password
---------------------------------------
-rw-------. 1 zdmuser zdm 4437 May  7 02:47 /home/zdmuser/zdmbase/crsdata/zdmhost/security/cwallet.sso
-rw-------. 1 zdmuser zdm 4437 May  7 02:47 /home/zdmuser/zdmbase/crsdata/zdmhost/security/cwallet.sso
spawn /home/zdmuser/zdmbase/mysql/server/bin/mysql --socket=/home/zdmuser/zdmbase/crsdata/zdmhost/rhp/mysql/metadata/mysql.sock -u root
---------------------------------------
Creating MySQL DB and user...
spawn /home/zdmuser/zdmhome/mysql/server/bin/mysql --
socket=/home/zdmuser/zdmbase/crsdata/zdmhost/rhp/mysql/metadata/mysql.sock -u root -p -e
CREATE DATABASE IF NOT EXISTS GHSUSER21;
spawn /home/zdmuser/zdmhome/mysql/server/bin/mysql --
socket=/home/zdmuser/zdmbase/crsdata/zdmhost/rhp/mysql/metadata/mysql.sock -u root -p
spawn /home/zdmuser/zdmhome/mysql/server/bin/mysql --
socket=/home/zdmuser/zdmbase/crsdata/zdmhost/rhp/mysql/metadata/mysql.sock -u root -p -e
GRANT ALTER, CREATE, CREATE TEMPORARY TABLES, CREATE VIEW, DELETE, DROP, INDEX, INSERT,
LOCK TABLES, REFERENCES, SELECT, SHOW VIEW, UPDATE ON GHSUSER21.* TO
'GHSUSER21'@'localhost';
current node is active node
spawn /home/zdmuser/zdmhome/mysql/server/bin/mysqladmin --defaults
file=/home/zdmuser/zdmbase/crsdata/zdmhost/rhp/conf/my.cnf -u root -p shutdown
ZDM service setup finished successfully...

6 Start ZDM and check the status. As zdmuser:

```
[zdmuser@zdmhost zdm21.2]$ $ZDM_HOME/bin/zdmservice start
Return code is 0
Server started successfully.
[zdmuser@zdmhost zdm21.2]$ $ZDM_HOME/bin/zdmservice status

----------------------------------------
| Service Status                      |
----------------------------------------
| Running:   true                      |
| Tranferport: |                        |
| Conn String: jdbc:mysql://localhost:8897/ |
| RMI port:  8895                      |
| HTTP port: 8896                      |
| Wallet path: /home/zdmuser/zdmbase/crsdata/zdmhost/security |
```

ZDM Service Host Port Requirements

Please find here a simplified table with the ports required for communication between the Zero Downtime Migration service host and the Source and Target Database servers.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>22</td>
<td>SSH</td>
</tr>
<tr>
<td>TCP</td>
<td>1521, 2484 or a DB SCAN listener port</td>
<td>SQL*NET</td>
</tr>
<tr>
<td>SSL</td>
<td>443</td>
<td>OCI &amp; GG REST Endpoint</td>
</tr>
</tbody>
</table>
SOURCE DATABASE

Source Database Requirements

ZDM supports Oracle Database 11g release 2 (11.2.0.4) or later versions; the Source Database must meet the following prerequisites before the Logical Migration process starts:

• The Source Database must be running in ARCHIVELOG mode. If the database is in NOARCHIVELOG, you will need to shut down the database to switch it to ARCHIVELOG mode. Please follow instructions on the Database Admin Guide “Changing the Database Archiving Mode” section 4.

• If enabling TDE on an 11g database, then ensure the TDE 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).

• The Source Database must use a server parameter file (SPFILE).

• The system time of the Zero Downtime Migration service host and Source Database server should be in sync with your Oracle Cloud Infrastructure target.

• Source Databases deployed using Oracle Grid Infrastructure and not registered using SRVCTL, must be registered before the migration.

• There are no requirements for Source and Target Databases versions to be the same for the Logical Migration workflow; this means you can migrate from a lower source level (i.e., 11.2.0.4) to a higher target level (i.e., 19c). Bear in mind that this only applies to the Logical Migration workflow; ZDM's Physical Migration still requires that both Source and Target are at the same version level.

The Logical Migration workflow has two methodologies, Offline with Data Pump and Online using a combination of Data Pump and GoldenGate. Oracle's ZDM Product Documentation section “Preparing for a Logical Migration” 5 subsection “Prepare the Source Database for Logical Migration” describes the necessary prerequisites. Please do as instructed below:

• If the Source is Oracle Database 11.2, apply the mandatory 11.2.0.4 RDBMS patches on the Source Database.

• See My Oracle Support note Oracle GoldenGate -- Oracle RDBMS Server Recommended Patches (Doc ID 1557031.1) 6
  • Database PSU 11.2.0.4.200414 includes a fix for Oracle GoldenGate performance bug 28849751 - IE PERFORMANCE DEGRADES WHEN NETWORK LATENCY BETWEEN EXTRACT AND CAPTURE IS MORE THAN 8MS
  • OGG RDBMS patch 31704157 MERGE REQUEST ON TOP OF DATABASE PSU 11.2.0.4.200414 FOR BUGS 31182000 20448066 - This patch combines mandatory fixes for Oracle GoldenGate Microservices bug 20448066 DBMS_XSTREAM_GG_APIS_SHOULD_BE_ALLOWED_FOR_SCA_PROCESSES and required OGG RDBMS patch 31182000 MERGE REQUEST ON TOP OF DATABASE PSU 11.2.0.4.200414 FOR BUGS 2990912 12668795.
  • Although MOS note 1557031.1 mentions OGG patch 31177512, it conflicts with a patch for bug 20448066. As such, OGG patch 31704157 should be used instead of OGG patch 31177512.

• Enable FORCE LOGGING to ensure that all changes are found in the redo by the Oracle GoldenGate Extract process.

6 https://support.oracle.com/cloud/faces/DocumentDisplay?id=1557031.1
• Enable database minimal supplemental logging.

• Enable initialization parameter ENABLE_GOLDENGATE_REPLICATION.

• Install the UTL_SPADV or UTL_RPADV package for Integrated Extract performance analysis. See Collecting XStream Statistics Using the UTL_RPADV Package. Note that the package changes name from UTL_SPADV to UTL_RPADV in Oracle Database 19c.

• If the source is Oracle Database 12.1.0.2 or a later release, apply mandatory RDBMS patches on the source database.

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

• Create a GoldenGate administration user, ggadmin, granting all of the required permissions.

```
-- Enable ARCHIVELOG mode:
SQL> select log_mode from v$database;
LOG_MODE
--------
ARCHIVELOG
-- Issue the following command to determine whether the database is in supplemental logging mode and in forced logging mode. If the result is YES for both queries, the database meets the Oracle GoldenGate requirement. If the result no, set it:
SQL> SELECT supplemental_log_data_min, force_logging FROM v$database;
SUPPLEMENT FOR
--------
NO NO
-- Enable database minimal supplemental logging.
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
Database altered.

-- Enable database forcelogging.
SQL > ALTER DATABASE FORCE LOGGING;
Database altered.

SQL> SELECT supplemental_log_data_min, force_logging FROM v$database;
SUPPLEMENT FOR
--------
YES YES

-- Enable initialization parameter ENABLE_GOLDENGATE_REPLICATION:
SQL> show parameter ENABLE_GOLDENGATE_REPLICATION
NAME TYPE VALUE
---------------------------------------------------------
```
enable_goldengate_replication boolean FALSE

SQL> alter system set ENABLE_GOLDENGATE_REPLICATION=TRUE scope=both;
System altered.

SQL> show parameter ENABLE_GOLDENGATE_REPLICATION
NAME TYPE VALUE
--------------------------------- ------- --------------------------
enable_goldengate_replication boolean TRUE

-- Create a GoldenGate administration user in 11g, ggadmin, granting all of the permissions listed below:
SQL> create user ggadmin identified by WElcome##1234 default tablespace users temporary tablespace temp;
User created.

SQL> grant connect, resource to ggadmin;
Grant succeeded.

SQL> grant unlimited tablespace to ggadmin;
Grant succeeded.

SQL> alter user ggadmin quota 100M on users;
User altered.

SQL> grant select any dictionary to ggadmin;
Grant succeeded.

SQL> grant create view to ggadmin;
Grant succeeded.

SQL> grant execute on dbms_lock to ggadmin;
Grant succeeded.

SQL> exec dbms_goldengate_auth.GRANT_ADMIN_PRIVILEGE('ggadmin');
PL/SQL procedure successfully completed.
-- For the purpose of this Step by Step guide, the Source Database version is 11.2, however if the Source Database version were to be 12.1 or higher and multitenant (CDB), then there is a need to also create the user c##ggadmin in CDB$ROOT as shown here. This value (username) will then be present on the response file under the parameter SOURCECONTAINERDATABASE_GGADMINUSERNAME

    SQL> create user c##ggadmin identified by WElcome##1234 default tablespace users temporary tablespace temp;
    SQL> grant connect, resource to c##ggadmin container=all;
    SQL> grant unlimited tablespace to c##ggadmin;
    SQL> grant select any dictionary to c##ggadmin container=all;
    SQL> alter user c##ggadmin quota 100M on users;
    SQL> grant create view to c##ggadmin container=all;
    SQL> grant execute on dbms_lock to c##ggadmin container=all;
    SQL> exec dbms_goldengate_auth.GRANT_ADMIN_PRIVILEGE('c##ggadmin',container=>' all');

-- Ensure the patches in the Source Database host satisfies the patch level covered on the MyOracleSupport document Oracle GoldenGate -- Oracle RDBMS Server Recommended Patches (Doc ID 1557031.1):

    oracle@source OPatch]$ ./opatch lspatches
    32248879;
    30508206;UPDATE PERL IN 11.2.0.4 DATABASE ORACLE HOME TO V5.28.2
    31335037;
    28602216;
    31983472;Database Patch Set Update : 11.2.0.4.210119 (31983472)
    29938455;OCW Patch Set Update : 11.2.0.4.191015 (29938455)
    OPatch succeeded.

-- The Target Database time zone version must be the same as the Source Database time zone version. To check the current time zone version on Source Side:

    SQL> SELECT * FROM v$timezone_file;
    FILENAME   VERSION
    ------------   -------
    timezlrg_35.dat     35

--Check the current service name on the source:
SQL> select value from v$parameter where name='service_names';
VALUE
---------------------------------
SOURCE_DB.xyz.xyz.xyz.com

-- For offline logical migrations, for optimal Data Pump performance, it is recommended that you set STREAMS_POOL_SIZE to at least 2GB
SQL> alter system set streams_pool_size = 2G;
System altered.

- SSL/TLS: in case the target source is configured with TLS and self-signed database server certificates, please add the self-signed certificate to the ZDM server home by executing the following:

  keytool -import -keystore ZDM_HOME/jdk/jre/lib/security/cacerts -trustcacerts -alias "src ca cert" -file source_db_server-certificate

- SSL/TLS: please store the wallet that contains the TLS authentication certificates in the correct location on the GoldenGate hub: /u02/deployments/deployment_name/etc

For the purposes of this guide, a 11.2.0.4 Source Database was configured. For more information, please refer to Oracle’s ZDM Product Documentation section “Preparing for a Logical Migration”, subsection “Source Database Prerequisites for Logical Migration”.

Source Database Port Requirements

Please find here a simplified table with the ports required for communication between the Source Database, the Zero Downtime Migration service host, the Target Database server, and the Oracle Cloud Object Store.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>22</td>
<td>SSH</td>
</tr>
<tr>
<td>TCP</td>
<td>1521, 2484</td>
<td>SQL*NET</td>
</tr>
<tr>
<td>SSL</td>
<td>443</td>
<td>Database Backup Store. OCI OSS.</td>
</tr>
</tbody>
</table>

TARGET DATABASE

Target Database Requirements

Zero Downtime Migration will migrate the Source Database to an Oracle Cloud Infrastructure database. This step-by-step guide covers the basics of migrating to a database on DBCS or Exadata Cloud Service.

Create a placeholder database on the Target Cloud service before starting the migration process. This placeholder Target Database must comply with the following requirements:

• **Sizing**: please ensure that the shape chosen will suffice for the Source Database sizing and any future increment in size.

• **Version**: the Target Database must be of the same version or higher than the Source Database version. Migration to a lower version database is not supported.

• **Character set**: the character set on the Source and Target Database must be the same.

• **Database time zone**: the database time zone on the Target Database must be equal to or lower than the Source Database time zone.

• **SSL/TLS**: for Target Databases configured to use SSL/TLS, store the wallet containing the TLS authentication certificates in the correct location on the GoldenGate hub:
  - /u02/deployments/deployment_name/etc

Please log in to your Oracle Cloud Account and access the Database systems tab by clicking on the Bare Metal, VM, and Exadata Menu. Please proceed to select the appropriate values regarding: Compartment, DB System name, Availability domain, shape, etc.

Log in to the Target Database via SSH and verify that the database time zone is equal to or lower than the Source Database.

```sql
-- TARGET Time zone check
SQL> SELECT * FROM v$timezone_file;
FILENAME  VERSION  CON_ID
----------  --------  -------
```

Figure 6 – Screenshot for “Create DB System” window on Oracle Cloud.
-- SRVCTL Configuration Show current settings on Target
[oracle@target ~]$ srvctl config database -d TARGET_DB -a
Database unique name: TARGET_DB_nrt1wq
Database name: TARGET_DB
Oracle home: /u01/app/oracle/product/19.0.0.0/dbhome_1
Oracle user: oracle
Spfile: +DATA/TARGET_DB/PARAMETERFILE/spfile.269.1071901813
Password file:
Domain: publicsubnet.vcnjpantechning.oraclevcn.com
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: AUTOMATIC
Server pools:
Disk Groups: RECO,DATA
Mount point paths:
Services:
 Type: SINGLE
 Database is enabled
 Database is individually enabled on nodes:
 Database is individually disabled on nodes:
 OSDBA group: dba
 OSOPER group: dbaoper
 Database instance: TARGET_DB
 Configured nodes: target
 CSS critical: no
 CPU count: 0
 Memory target: 0
 Maximum memory: 0
 Default network number for database services:
 Database is administrator managed

-- Use the crsctl query crs releaseversion command to display the version of the Oracle Clusterware software
[grid@target ~]$ crsctl query crs softwareversion -all
Oracle Clusterware version on node [target] is [19.0.0.0.0]
--Check the current service name on the target.
SQL> select value from v$parameter where name='service_names';
VALUE
--------------------------------------------------------------------------------
TARGET_DB.xyz.xyz.xyz.com

-- Ensure that the wallet STATUS is OPEN and WALLET_TYPE is AUTOLOGIN

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 keystore before migration begins.

SQL> select WRL_TYPE,WRL_PARAMETER,STATUS,CON_ID FROM v$encryption_wallet;

WRL_TYPE WRL_PARAMETER STATUS CON_ID
----------- ------------------------------------------ ------ -----
FILE /opt/oracle/dcs/commonstore/wallets/tde/TARGET_DB/ OPEN 1
FILE OPEN 2
FILE OPEN 3

SQL> show pdbs

CON_ID CON_NAME OPEN MODE RESTRICTED
-------- ------------------------ --------- --------
2 PDB$SEED READ ONLY NO
3 PDB1 READ WRITE NO

SQL> alter session set container=pdb1;
Session altered.

SQL> select owner,tablespace_name from dba_tables where owner = 'TEST';

OWNER TABLESPACE_NAME
--------- ------------------------
TEST TEST

SQL> select d.TABLESPACE_NAME, d.FILE_NAME from dba_data_files d, v$datafile v where d.FILE_ID = v.FILE# order by d.TABLESPACE_NAME, d.FILE_NAME;

TABLESPACE FILE_NAME
--------- ------------------------
SYSAUX +DATA/TARGET_DB/C1B89AAB24203930E0533100000A0129/DATAFILE/sysaux.271.1071902309
SYSTEM +DATA/TARGET_DB/C1B89AAB24203930E0533100000A0129/DATAFILE/system.276.1071902299
TEST +DATA/TARGET_DB/C1B89AAB24203930E0533100000A0129/DATAFILE/test.282.1072487049
UNDOTBS1 +DATA/TARGET_DB/C1B89AAB24203930E0533100000A0129/DATAFILE/undotbs1.272.1071902319
USERS +DATA/TARGET_DB/C1B89AAB24203930E0533100000A0129/DATAFILE/users.275.1071902287
-- Create a GoldenGate administration user, ggadmin (in the PDB in case of Multitenant):
SQL> alter session set container=pdb1;
SQL> create user ggadmin identified by WElcome##1234 default tablespace users temporary tablespace temp;
SQL> grant connect, resource to ggadmin;
SQL> grant unlimited tablespace to ggadmin;
SQL> alter user ggadmin quota 100M on users;
SQL> grant select any dictionary to ggadmin;
SQL> grant create view to ggadmin;
SQL> grant execute on dbms_lock to ggadmin;
SQL> exec dbms_goldengate_auth.GRANT_ADMIN_PRIVILEGE('ggadmin');

-- You need to connect to a CDB, Enable initialization parameter ENABLE_GOLDENGATE_REPLICATION:
SQL> show parameter ENABLE_GOLDENGATE_REPLICATION
NAME TYPE VALUE
------------------------- ----------- ------------------------------
e_enable_goldengate_replication boolean FALSE

SQL> alter system set ENABLE_GOLDENGATE_REPLICATION=TRUE scope=both;
System altered.

SQL> show parameter ENABLE_GOLDENGATE_REPLICATION
NAME TYPE VALUE
------------------------- ----------- ------------------------------
e_enable_goldengate_replication boolean TRUE

-- DATA PUMP Required Target preparation
SQL> alter user SYSTEM identified by Welcome##1234;
User Altered.

-- Grant DATAPUMP_IMP_FULL_DATABASE role to SYSTEM
SQL> grant DATAPUMP_IMP_FULL_DATABASE to SYSTEM;
Grant succeeded.
Target Database Port Requirements

Please find here a simplified table with the ports required for communication between the Target Database server, the Zero Downtime Migration service host, the Source Database, and Oracle Cloud Object Store.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>22</td>
<td>SSH</td>
</tr>
<tr>
<td>TCP</td>
<td>1521, 2484</td>
<td>SQL*NET</td>
</tr>
<tr>
<td>SSL</td>
<td>443</td>
<td>Database Backup Store. OCI OSS.</td>
</tr>
</tbody>
</table>

CONNECTIVITY

SSH Key Pair

ZDM connects via SSH to the Source and Target Database servers; hence a SSH key pair for the zdmuser is required. As zdmuser, execute the following:

```
[zdmuser@zdmhost ~]$ mkdir ~/.ssh
[zdmuser@zdmhost ~]$ chmod 700 ~/.ssh
[zdmuser@zdmhost ~]$ /usr/bin/ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/zdmuser/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/zdmuser/.ssh/id_rsa.
Your public key has been saved in /home/zdmuser/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:qQd25G01EtiUCUwdGTYPHeeps6f2VqJHV6cHBeUSWg zdmuser@zdmhost
The key's randomart image is:
   +---[RSA 2048]----+
   | xxyxyxyxyx    |
   | xxyxyxyxy     |
   | xyyxyxxyy     |
   | xyyxyxxyy     |
   | xyyxyxxyx     |
   | xyyxyxxyxy    |
   | xyyxyxxyxy    |
   | xyyxyxxyxy    |
   | xyyxyxxyxy    |
   | yyyyyy         |
```

```
+----[SHA256]------+

[zdmuser@zdmhost ~]$ cd ~/.ssh
[zdmuser@zdmhost .ssh]$ cat id_rsa.pub >> authorized_keys
[zdmuser@zdmhost .ssh]$ chmod 600 authorized_keys
[zdmuser@zdmhost .ssh]$ ll
```
You can find more information on ZDM Product’s documentation section Generating a Private SSH Key Without a Passphrase.

Before continuing with the migration environment setup, please add the contents of the id_rsa.pub file authorized_keys locations across all target database servers.

**Connectivity between the ZDM Service Host and the Source and Target Database Servers**

Configure /etc/hosts as a first step to ensure connectivity between the ZDM Service Host and the Source and Target Database servers. As the root user on the ZDM Service host, adding the Source Database server, Target Database server, and OGG Hub information:

```
[root@zdmhost zdminstall]# vi /etc/hosts
[root@zdmhost opc]# vi /etc/hosts
10.0.0.XX source.publicsubnet.xyz.xyz.com source  ##Source Database
10.0.0.YY target.publicsubnet.xyz.xyz.com target  ##Target Database
10.0.0.XYZ oggl9cora.publicsubnet.xyz.xyz.com oggl9cora ##OGG Hub
10.0.0.ABC zdmhost.publicsubnet.xyz.xyz.com zdmhost  ##ZDM Host
```

**Adding ZDM Host Public Keys to the Source Database Server**

Add the ZDM Host Public Keys to the Source Database Server Authorized Key files by executing the following on the Source Database server:

```
[opc@source ~]$ cd /home/opc/.ssh
[opc@source .ssh]$ echo "ssh-rsa SSHKEYSSHKEY zdmuser@zdmhost" >> authorized_keys
```

**Adding ZDM Host Public Keys to the Target Database Server**

Add the ZDM Host Public Keys to the Target Database Server Authorized Key files by executing the following on the Target Database server:

```
[opc@target ~]$ cd /home/opc/.ssh
[opc@target .ssh] $ echo "ssh-rsa SSHKEYSSHKEY zdmuser@zdmhost" >> authorized_keys
```

---

Testing Connectivity from the ZDM Service Host to Source and Target Database Server

To test the connectivity from the ZDM Service Host to the Source and Target Database Server, execute the following as the zdmuser on the ZDM Service Host:

```
[zdmuser@zdmhost .ssh]$ ssh -i zdm.ppk opc@source
[zdmuser@zdmhost .ssh]$ ssh -i zdm.ppk opc@target
```

Authentication Token

The OCI user requires an Authentication Token, which can be created from the user’s detail page, as explained below. Click on the “Auth Tokens” option and then on the “Generate Token” button.

![Screenshot for “Auth Tokens” window on Oracle Cloud.](image)

ZDM uses the Auth Token during the migration; hence, it is of the utmost importance that it is securely copied and stored.

OCI CLI Command Line Tool

During the migration, the Oracle Cloud Infrastructure command-line tool (OCI CLI) accesses OCI resources, among other tasks. To install the OCI CLI on the ZDM Service host, as the zdmuser execute as follows:

```
[zdmuser@zdmhost ~]$ mkdir ~/.ssh
[zdmuser@zdmhost .ssh]$ oci --version
2.24.3
```

```
[zdmuser@zdmhost lib]$ oci setup config
Enter a location for your config [/home/zdmuser/.oci/config]:
Enter a user OCID: ocid1.user.oc1..xyz
Enter a tenancy OCID: ocid1.tenancy.oc1..xyz
Do you want to generate a new API Signing RSA key pair? (If you decline you will be asked to supply the path to an existing key.) [Y/n]: Y
Enter a directory for your keys to be created [/home/zdmuser/.oci]:
Enter a name for your key [oci_api_key]:
```
Public key written to: /home/zdmuser/.oci/oci_api_key_public.pem
Enter a passphrase for your private key (empty for no passphrase):
Private key written to: /home/zdmuser/.oci/oci_api_key.pem
Config written to /home/zdmuser/.oci/config

API Signing Public Key

ZDM uses an API Signing Public Key to call REST APIs. Upload the key to the OCI user in the cloud console, from the ZDM Service host, as the zdmuser, by executing as follows:

```
[zdmuser@zdmhost ~]$ mkdir ~/.ssh
[zdmuser@zdmhost lib]$ cat /home/zdmuser/.oci/oci_api_key_public.pem
-----BEGIN PUBLIC KEY----
PUBLICKEYPUBLICKEY
-----END PUBLIC KEY----
```

Proceed to copy the output of this command and then go to the OCI user’s page. There, click on “API Keys” and then click on “Add API Key”. Select the option labeled as “Paste Public Key”, then paste the output from the command just copied above; once copied, click “Add”.

```
Figure 8 – Screenshot for “Add API Keys” window on Oracle Cloud.
```

The fingerprint generated and shown below the “Add API Key” button should match the fingerprint in the config file on the ZDM Service host. To verify this, in the ZDM Service host, as the zdmuser, execute as follows:

```
[zdmuser@zdmhost lib]$ cat /home/zdmuser/.oci/config | grep fingerprint
```

Connectivity between Source and Target Database Servers

It is possible to connect the Source and Target Database servers in two different ways, SQL*Net connectivity using SCAN or via SSH. The following explains how to configure both types of connectivity according to your requirements/restrictions.

Option 1 – SCAN Connectivity

The first step to configure SCAN Connectivity is to edit the /etc/hosts files in both the Source and Target servers:
Source Server

```bash
[root@source opc]# cat /etc/hosts
10.0.0.XX  source.publicsubnet.xyz.xyz.com  source  ##Source Database
10.0.0.YY  target.publicsubnet.xyz.xyz.com  target  ##Target Database
10.0.0.XYZ oggl19cora.publicsubnet.xyz.xyz.com oggl19cora  ##OGG Hub
10.0.0.ABC zdmhost.publicsubnet.xyz.xyz.com  zdmhost  ##ZDM Host
```

Target Server

```bash
[root@target opc]# cat /etc/hosts
10.0.0.xx  target.publicsubnet.xyz.xyz.com  target
192.xx.yy.zz target-priv.publicsubnet.xyz.xyz.com target-priv
10.0.0.yy  target-vip.publicsubnet.xyz.xyz.com target-vip
10.0.0.yy  target-scan.publicsubnet.xyz.xyz.com target-scan
10.0.0.zz  source.publicsubnet.xyz.xyz.com  source
10.0.0.zz  source-scan.publicsubnet.xyz.xyz.com source-scan
```

Testing Connectivity between Source and Target via SCAN

```bash
--TESTING CONNECTIVITY FROM SOURCE TO TARGET
[oracle@source ~]$ tnsping target:1521
TNS Ping Utility for Linux: Version 11.2.0.4.0 - Production on 07-JUL-2021 00:34:01
Used HOSTNAME adapter to resolve the alias
Attempting to contact
(DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=))(ADDRESS=(PROTOCOL=TCP)(HOST=10.0.0.xy)(PORT=1521)))
OK (0 msec)
--TESTING CONNECTIVITY FROM SOURCE TO TARGET
[oracle@target ~]$ tnsping source:1521
Used HOSTNAME adapter to resolve the alias
Attempting to contact
(DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=))(ADDRESS=(PROTOCOL=tcp)(HOST=10.0.0.xx)(PORT=1521)))
OK (0 msec)
```

Option 2 – SSH Connectivity

Connectivity via SCAN might not be feasible between the Source and Target servers; if this is the case, an SSH Tunnel can be set up; to achieve this, follow the steps outlined in Oracle ZDM’s product documentation section Option 2: Set up an SSH Tunnel9.

---

BACKUP LOCATION

Object Storage Requirements

When migrating to OCI Native Databases, ZDM requires a backup location to export Data Pump dump files. The Target Database service can leverage them for instantiating the Target Database; this backup location will be an Object Storage within the customer’s tenancy. To create an Object Storage bucket, execute the following steps as the zdmuser:

```
[zmuser@zdmhost lib]$ oci os bucket create --compartment-id ocid1.compartment.oc1..xyz --name zdmbucket
{
    "data": {
        "approximate-count": null,
        "approximate-size": null,
        "auto-tiering": null,
        "compartment-id": "ocid1.compartment.oc1..xyz",
    }
    "create-by": "oci1.user.oc1..xyz",
    "defined-tags": {
        "default_tags": {
            "CreatedBy": "xyz@zdmtesting.xyzxyz",
            "CreatedOn": "2021-07-05T05:08:28.372Z"
        }
    },
    "etag": "d4162e72-249a-4e45-b163-ced36fd412f",
    "freeform-tags": {},
    "id": "ocid1.bucket.oc1.xyz.xyz",
    "is-read-only": false,
    "kms-key-id": null,
    "metadata": {},
    "name": "zdmbucket",
    "namespace": "sehubjapacprod",
    "object-events-enabled": false,
    "object-lifecycle-policy-etag": null,
    "public-access-type": "NoPublicAccess",
    "replication-enabled": false,
    "storage-tier": "Standard",
    "time-created": "2021-07-05T05:08:28.388000+00:00",
    "versioning": "Disabled"
},
    "etag": "xyz-xyz-xyz-xyz-xyz-xyz-xyz-xyz-xyz-xyz-xyz-xyz"`
Next, open the Object Storage Page from within your Oracle Cloud Account; it should show that recently created bucket as per below:

![Figure 9 – Screenshot for “Object Storage > Bucket Details” window on Oracle Cloud.](image)

**GOLDENGATE HUB**

**GoldenGate Hub Requirements**

Logical Online migration to both DBCS and ExaCS requires an Oracle GoldenGate Microservices hub. To set this up, please follow the instructions as per:

- [Oracle GoldenGate Microservices on Oracle Cloud Marketplace](#)

Deploy the Oracle GoldenGate Microservices using the specific image from the Oracle Cloud Marketplace located in:

- [Oracle GoldenGate – Database Migrations](#)

Please visit the link and from the main menu, choose “Oracle GoldenGate - Database Migrations”, and continue with the default version.

![Figure 10 – Screenshot for “Oracle GoldenGate – Database Migrations” window on Oracle Cloud.](image)

Upon Creating the GoldenGate Hub, Connect to the VM via SSH using its Public IP address to get the oggadmin user password:

```bash
-bash-4.2$ cat /home/opc/ogg-credentials.json
{"username": "oggadmin", "credential": "xyzxyzxyzxyz"}
```

Proceed to add the server's hostname and IP information into the ZDM host `/etc/hosts` file.

```bash
-bash-4.2$ cat /etc/hosts
10.0.0.xyz ogg19cora.x.y.z.com ogg19cora
```
PREPARING THE RESPONSE FILE

Oracle Zero Downtime Migration leverages a response file that is fully customizable by the customer. For the logical migration methodology, a wide array of parameters allows the customer to configure the migration according to the appropriate use case. This step-by-step guide uses a specific set of response file parameters; a detailed description is present below. For more information on the complete set of response file parameters for logical migration, refer to ZDM’s Product Documentation section Zero Downtime Migration Logical Migration Response File Parameters Reference.10

Response File Parameters used in this Guide

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIGRATION_TYPE</td>
<td>• ONLINE_LOGICAL → ZDM will leverage Data Pump and Oracle GoldenGate for the migration process. <em>(Used in this guide)</em>&lt;br&gt;• OFFLINE_LOGICAL → ZDM will leverage Data Pump only for the migration process</td>
</tr>
<tr>
<td>DATA_TRANSFER_MEDIUM</td>
<td>• OSS → Object Storage Service. <em>(Used in this guide)</em>&lt;br&gt;• NFS → Network File System (for ExaC@C targets only)&lt;br&gt;• DBLINK → Direct transfer of data over database link&lt;br&gt;• COPY → secure copy (for user managed targets only)</td>
</tr>
<tr>
<td>TARGETDATABASE_OCID</td>
<td>• Specifies the connection details for the Oracle Cloud resource identifier for the target database.</td>
</tr>
<tr>
<td>TARGETDATABASE_ADMINUSERNAME</td>
<td>• Specifies the Target Database administrator’s user name.</td>
</tr>
<tr>
<td>SOURCEDATABASE_ADMINUSERNAME</td>
<td>• Specifies the Source Database administrator’s user name.</td>
</tr>
<tr>
<td>SOURCEDATABASE_CONNECTIONDETAILS_HOST</td>
<td>• Specifies the listener host name or IP address. This is only for DBCS/ExaCS/ExaC@C targets.</td>
</tr>
<tr>
<td>SOURCEDATABASE_CONNECTIONDETAILS_PORT</td>
<td>• Specifies the listener port number. This is only for DBCS/ExaCS/ExaC@C targets.</td>
</tr>
<tr>
<td>TARGETDATABASE_CONNECTIONDETAILS_SERVICENAME</td>
<td>• Specifies the fully qualified service name</td>
</tr>
<tr>
<td>TARGETDATABASE_CONNECTIONDETAILS_HOST</td>
<td>• Specifies the listener host name or IP address.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TARGETDATABASE_CONNECTIONDETAILS_SERVICENAME</td>
<td>Specifies the fully qualified service name.</td>
</tr>
<tr>
<td>OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_TENANTID</td>
<td>Specifies the OCID of the OCI tenancy.</td>
</tr>
<tr>
<td>OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_USERID</td>
<td>Specifies the OCID of the IAM user.</td>
</tr>
<tr>
<td>OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_FINGERPRINT</td>
<td>Specifies the fingerprint of the public API key.</td>
</tr>
<tr>
<td>OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_PRIVATEKEYFILE</td>
<td>Specifies the absolute path of the API private key file.</td>
</tr>
<tr>
<td>OCIAUTHENTICATIONDETAILS_REGIONID</td>
<td>Specifies the OCI region identifier.</td>
</tr>
<tr>
<td>SOURCEDATABASE_GGADMINUSERNAME</td>
<td>Specifies the GoldenGate administrator’s user name.</td>
</tr>
<tr>
<td>TARGETDATABASE_GGADMINUSERNAME</td>
<td>Specifies the GoldenGate administrator’s user name for the PDB.</td>
</tr>
<tr>
<td>GOLDENGATEHUB_ADMINUSERNAME</td>
<td>Specifies the GoldenGate Hub administrator’s user name.</td>
</tr>
<tr>
<td>GOLDENGATEHUB_URL</td>
<td>Specifies the GoldenGate Hub’s REST endpoint.</td>
</tr>
<tr>
<td>GOLDENGATEHUB_SOURCEDEPLOYMENTNAME</td>
<td>Specifies the name of the GoldenGate Microservices deployment to operate on the Source Database.</td>
</tr>
<tr>
<td>GOLDENGATEHUB_TARGETDEPLOYMENTNAME</td>
<td>Specifies the name of the GoldenGate Microservices deployment to operate on the Target Database.</td>
</tr>
<tr>
<td>GOLDENGATEHUB_COMPUTEID</td>
<td>Specifies the Oracle Cloud identifier of the VM.</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| DATAPUMPSETTINGS_JOBMODE | Specifies the Data Pump export mode:  
  - FULL  
  - SCHEMA  
  - TABLE  
  - TABLESPACE  
  - TRANSPORTABLE (not supported by ZDM) |
| INCLUDEOBJECTS-N | Specifies database objects to include for the migration; an integer must replace the N; this parameter can appear N number of times, increase the number and add new objects as required. |
| DATAPUMPSETTINGS_DATAPUMPPARAME TERS_IMPORTPARALLELISMDEGREE | Specifies the maximum number of worker processes that a Data Pump import job can use. |
| DATAPUMPSETTINGS_DATAPUMPPARAME TERS_EXPORTPARALLELISMDEGREE | Specifies the maximum number of worker processes that a Data Pump export job can use. |
| DATAPUMPSETTINGS_DATABUCKET_NAME SPACENAME | Specifies the object storage bucket namespace. |
| DATAPUMPSETTINGS_DATABUCKET_BUCKE TNAME | Specifies the object storage bucket name. |
| DATAPUMPSETTINGS_EXPORTDIRECTORYOBJECT_NAME | Specifies a directory name on the source server to store the Data Pump Export dump files. ZDM will create this object if it does not exist already. |
| DATAPUMPSETTINGS_EXPORTDIRECTORYOBJECT_PATH | Specifies a directory path on the source server to store the Data Pump Export dump files. ZDM will create this object if it does not exist already. |
| DATAPUMPSETTINGS_IMPORTDIRECTORYOBJECT_NAME | Specifies a directory path on the target server to store the Data Pump Export dump files. ZDM will create this object if it does not exist already. |
| DATAPUMPSETTINGS_CREATEAUTHTOKEN | Specifies if an OCI Authentication Token is needed to be created for the specified OCI user to import the Data Dump Files from the Object Storage into an Autonomous Database. For DBCS/ExaCS migration, this parameter is FALSE. |
| DATAPUMPSETTINGS_OMITENCRYPTIONCL AUSE | When enabled, this parameter sets TRANSFORM=OMIT_ENCRYPTION_CLAUSE, which directs Data Pump to suppress any encryption clauses associated with objects using encrypted columns. |
To execute the migration described in this step-by-step guide, the sample response file used will be as follows (the response file resides on the ZDM Service Host):

```
[zdmuser@zdmhost ~]$ cp /home/zdmuser/zdmhome/rhp/zdm/template/zdm_logical_template.rsp /home/zdmuser
[zdmuser@zdmhost ~]$ cd /home/zdmuser
[zdmuser@zdmhost ~]$ ll
```

total 76
```
drwxr-xr-x.  3 zdmuser zdm  40 Jul  7 03:10 bin
drwxr-xr-x.  3 zdmuser zdm  24 Jul  7 03:10 lib
drwxr-xr-x.  5 zdmuser zdm  51 Jul  7 02:47 zdmbase
drwxr-xr-x. 47 zdmuser zdm 4096 Jul  7 02:47 zdmhome
drwxr-xr-x.  47 zdmuser zdm 4096 Jul  7 02:37 zdminstall
-r-xr-xr-x.  1 zdmuser zdm 71731 Jul  8 01:55 zdm_logical_template.rsp
```

```
[zdmuser@zdmhost ~]$ mv zdm_logical_template.rsp logical_online.rsp
[zdmuser@zdmhost ~]$ chmod +w logical_online.rsp
[zdmuser@zdmhost ~]$ vi logical_online.rsp
```

```bash
# migration method
MIGRATION_METHOD=ONLINE_LOGICAL
DATA_TRANSFER_MEDIUM=OSS
```

```bash
# target db 19c OCID and ADMIN USER
TARGETDATABASE_OCID=ocid1.database.oc1.xyz.xyz
TARGETDATABASE_ADMINUSERNAME=SYSTEM
```

```bash
# source db
SOURCEDATABASE_ADMINUSERNAME=SYSTEM
SOURCEDATABASE_CONNECTIONDETAILS_HOST=source
SOURCEDATABASE_CONNECTIONDETAILS_PORT=1521
SOURCEDATABASE_CONNECTIONDETAILS_SERVICENAME=SOURCE_DB.xyz.xyz.oraclevcn.com
```

```bash
# target db (PDB)
TARGETDATABASE_CONNECTIONDETAILS_HOST=target
TARGETDATABASE_CONNECTIONDETAILS_PORT=1521
TARGETDATABASE_CONNECTIONDETAILS_SERVICENAME=pdb1.publicsubnet.xyz.xyz.com
```
# oci cli
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_TENANTID=ocid1.tenancy.oc1.xyz.xyz
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_USERID=ocid1.user.oc1..xyz
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_PRIVATEKEYFILE=/home/zdmuser/.oci/oci_api_key.pem
OCIAUTHENTICATIONDETAILS_REGIONID=ap-tokyo-1

## GoldenGate
SOURCEDATABASE_GGADMINUSERNAME=ggadmin
TARGETDATABASE_GGADMINUSERNAME=ggadmin
GOLDENGATEHUB_ADMINUSERNAME=oggadmin
GOLDENGATEHUB_URL=https://ogg19cora.publicsubnet.xyz.xyz.com
GOLDENGATEHUB_SOURCEDEPLOYMENTNAME=Source
GOLDENGATEHUB_TARGETDEPLOYMENTNAME=Target
GOLDENGATEHUB_COMPUTEID=ocid1.instance.oc1.xyz.xyz

# data pump
DATAPUMPSETTINGS_JOBMODE=SCHEMA
INCLUDEOBJECTS-1=owner:TEST
DATAPUMPSETTINGS_DATAPUMPPARAMETERS_IMPORTPARALLELISMDEGREE=2
DATAPUMPSETTINGS_DATAPUMPPARAMETERS_EXPORTPARALLELISMDEGREE=2
DATAPUMPSETTINGS_DATABUCKET_NAMESPACE_NAME=sehubjapacprod
DATAPUMPSETTINGS_DATABUCKET_BUCKETNAME=zdmbucket
DATAPUMPSETTINGS_EXPORTDIRECTORYOBJECT_NAME=DATA_PUMP_DIR
DATAPUMPSETTINGS_EXPORTDIRECTORYOBJECT_PATH=/u01/app/oracle/product/11.2.0.4/dbhome_1/rdbms/log/
DATAPUMPSETTINGS_IMPORTDIRECTORYOBJECT_NAME=DATA_PUMP_DIR
DATAPUMPSETTINGS_CREATEAUTHTOKEN=FALSE
#DATAPUMPSETTINGS_OMITENCRYPTIONCLAUSE=TRUE
Performing a Test Database Migration on Evaluation Mode

Oracle Zero Downtime Migration includes an evaluation mode that performs a dry run of the migration process; this is an optional step. It allows customers to ensure that the migration will run swiftly and will encounter no issues. When migrating with the evaluation flag on, ZDM evaluates all the different stages and will alert the user if there are any inconsistencies or potential issues; this way, customers can fix any problems beforehand. As a best practice, run a Test Database Migration before executing the migration itself. To this, please perform as follows:

```
[zdmuser@zdmhost logs]$ $ZDM_HOME/bin/zdmcli migrate database -sourcedb SOURCE_DB \
-sourcenode source \
-srcauth zdmauth \
-srcarg1 user:opc \
-srcarg2 identity_file:/home/zdmuser/.ssh/zdm.ppk \
-srcarg3 sudo_location:/usr/bin/sudo \
-targetnode target -rsp /home/zdmuser/logical_online.rsp \
-tgtauth zdmauth \
-tgtarg1 user:opc \
-tgtarg2 identity_file:/home/zdmuser/.ssh/zdm.ppk \
-tgtarg3 sudo_location:/usr/bin/sudo \n-eval
```

ZDM will then request the different required passwords and will generate a job id.

```
zdmhost.publicsubnet.xyz.xyz.com: Audit ID: 264
Enter source database administrative user "SYSTEM" password: WElcome##1234
Enter source database administrative user "ggadmin" password: WElcome##1234
Enter target database administrative user "SYSTEM" password: WElcome##1234
Enter target database administrative user "ggadmin" password: WElcome##1234
Enter Oracle GoldenGate hub administrative user "oggadmin" password: xyzxyzxyz
Enter Authentication Token for OCI user "ocid1.user.oc1..xyz": xyzxyxxyz
Enter Data Pump encryption password: WElcome##1234
Operation "zdmcli migrate database" scheduled with the job ID "30".
```

The generated job id can be queried for progress using the `zdmcli query job -jobid job_id` command.

```
[zdmuser@zdmhost logs]$ $ZDM_HOME/bin/zdmcli query job -jobid 30
zdmhost.publicsubnet.xyz.xyz.com: Audit ID: 270
Job ID: 30
User: zdmuser
Client: zdmhost
Job Type: "EVAL"
Scheduled job command: "zdmcli migrate database -sourcedb SOURCE_DB -sourcenode source -srcauth zdmauth -srcarg1 user:opc -srcarg2 identity_file:/home/zdmuser/.ssh/zdm.ppk -srcarg3 sudo_location:/usr/bin/sudo -targetnode target -rsp /home/zdmuser/logical_online.rsp -tgtauth zdmauth -tgtarg1 user:opc -tgtarg2 identity_file:/home/zdmuser/.ssh/zdm.ppk -tgtarg3 sudo_location:/usr/bin/sudo -eval"
Scheduled job execution start time: 2021-07-08T06:31:44Z. Equivalent local time: 2021-07-08 06:31:44
```
Performing a Database Migration

To perform the database migration once the migration command with the evaluation flag completed successfully and without errors and warnings, execute the same command without the -eval option:

```
[zdmuser@zdmhost logs]$ $ZDM_HOME/bin/zdmcli migrate database -sourcedb SOURCE_DB \
  -sourcenode source \n  -srauth zdmauth \n  -srcarg1 user:opc \n  -srcarg2 identity_file:/home/zdmuser/.ssh/zdm.ppk \n  -srcarg3 sudo_location:/usr/bin/sudo \n  -targetnode target -rsp /home/zdmuser/logical_online.rsp \n  -tgtauth zdmauth \n  -tgtarg1 user:opc \n  -tgtarg2 identity_file:/home/zdmuser/.ssh/zdm.ppk \n  -tgtarg3 sudo_location:/usr/bin/sudo \n
zdmhost.publicsubnet.xyz.xyz.com: Audit ID: 271
Enter source database administrative user "SYSTEM" password: WElcome##1234
Enter source database administrative user "ggadmin" password: WElcome##1234
Enter target database administrative user "SYSTEM" password: WElcome##1234
Enter target database administrative user "ggadmin" password: WElcome##1234
Enter Oracle GoldenGate hub administrative user "oggadmin" password: xyzxyz
Enter Authentication Token for OCI user "ocid1.user.oc1..xyz": Du>:xyzxyz
Enter Data Pump encryption password: WElcome##1234
Operation "zdmcli migrate database" scheduled with the job ID "31".
```
Proceed to periodically query the migration job with the provided migration job id until completed:

```
[zdmsuser@zdmhost ~]$ $ZDM_HOME/bin/zdmcli query job -jobid 34
zdmsuser.publicsubnet.xyz.xyz.com: Audit ID: 307
Job ID: 34
User: zdmuser
Client: zdmhost
Job Type: "MIGRATE"
Scheduled job command: "zdmcli migrate database -sourcedb SOURCE_DB -sourcenode source -srcauth zdmauth -srcarg1 user:opc -srcarg2 identity_file:/home/zdmuser/.ssh/zdm.ppk -srcarg3 sudo_location:/usr/bin/sudo -targetnode target -rsp /home/zdmuser/logical_online.rsp -tgtauth zdmauth -tgtauth user:opc -tgtaug2 identity_file:/home/zdmuser/.ssh/zdm.ppk -tgtaug3 sudo_location:/usr/bin/sudo"
Scheduled job execution start time: 2021-07-08T01:13:52Z. Equivalent local time: 2021-07-08 01:13:52
Current status: SUCCEEDED
Result file path: "/home/zdmuser/zdmbase/chkbase/scheduled/job-34-2021-07-08-01:14:18.log"

Job execution start time: 2021-07-08 01:14:18
Job execution end time: 2021-07-08 01:24:17
Job execution elapsed time: 9 minutes 59 seconds
ZDM_VALIDATE_TGT ...................... COMPLETED
ZDM_VALIDATE_SR ...................... COMPLETED
ZDM_SETUP_SRC ......................... COMPLETED
ZDM_PRE_MIGRATION_ADVISOR .......... COMPLETED
ZDM_VALIDATE_GG_HUB ................. COMPLETED
ZDM_VALIDATE_DATAPUMP_SETTINGS_SRC .... COMPLETED
ZDM_VALIDATE_DATAPUMP_SETTINGS_TGT .... COMPLETED
ZDM_PREPARE_GG_HUB .................... COMPLETED
ZDM_ADD_HEARTBEAT_SRC ................. COMPLETED
ZDM_ADD_SCHEMA_TRANDATA_SRC .......... COMPLETED
ZDM_CREATE_GG_EXTRACT_SRC .......... COMPLETED
ZDM_PREPARE_DATAPUMP_SRC .......... COMPLETED
ZDM_PREPARE_DATAPUMP_TGT .......... COMPLETED
ZDM_DATAPUMP_EXPORT_SRC .......... COMPLETED
ZDM_UPLOAD_DUMPS_SRC .......... COMPLETED
ZDM_DATAPUMP_IMPORT_TGT .......... COMPLETED
ZDM_POST_DATAPUMP_SRC .......... COMPLETED
ZDM_POST_DATAPUMP_TGT .......... COMPLETED
ZDM_ADD_HEARTBEAT_TGT .......... COMPLETED
ZDM_ADD_CHECKPOINT_TGT .......... COMPLETED
ZDM_CREATE_GG_REPLICAT_TGT .......... COMPLETED
ZDM_MONITOR_GG_LAG .......... COMPLETED
```
<table>
<thead>
<tr>
<th>Procedure</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZDM_SWITCHOVER_APP</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>ZDM_RM_GG_EXTRACT_SRC</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>ZDM_RM_GG_REPLICAT_TGT</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>ZDM_DELETE_SCHEMA_TRANDATA_SRC</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>ZDM_RM_HEARTBEAT_SRC</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>ZDM_RM_CHECKPOINT_TGT</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>ZDM_RM_HEARTBEAT_TGT</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>ZDM_CLEAN_GG_HUB</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>ZDM_POST_ACTIONS</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>ZDM_CLEANUP_SRC</td>
<td>COMPLETED</td>
</tr>
</tbody>
</table>
KNOWN ISSUES

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


TROUBLESHOOTING & OTHER RESOURCES

For Oracle ZDM log review:

- ZDM Server host logs:
  - Check - $ZDM_BASE/crsdata/zdmserver.log.0
- ZDM Server host logs:
  - Check - $ZDM_BASE/crsdata/zdmserver.log.0
- ZDM source node Data Pump logs:
  - DATAPUMPSETTINGS_EXPORTDIRECTORYOBJECT_PATH
- ZDM target node logs:
  - DATAPUMPSETTINGS_IMPORTDIRECTORYOBJECT_NAME
- Import Log:
  - OSS Bucket
- OGG hub logs:
  - /u02/deployments/<ogg_deployment_name>/var/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](https://support.oracle.com/epmos/faces/DocContentDisplay?id=2595205.1)