



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 © 2021, Oracle and/or its affiliates

DISCLAIMER

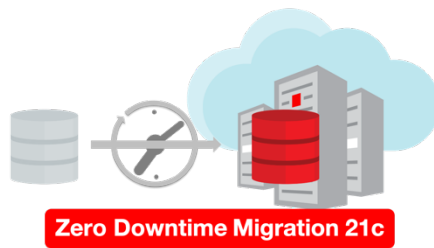
This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed 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	23
Adding ZDM Host Public Keys to the Target Database Server	23
Testing Connectivity from the ZDM Service Host to Source and Target Database Server	23
Authentication Token	23
OCI CLI Command Line Tool	24
API Signing Public Key	24
Connectivity between Source and Target Database Servers	25
Option 1 – SCAN Connectivity	25
Option 2 – SSH Connectivity	26
Backup Location	27
Object Storage Requirements	27
GoldenGate Hub	28
GoldenGate Hub Requirements	28
Preparing the Response File	33
Response File Parameters used in this Guide	33
Logical Migration to DBCS and ExaCS with ZDM	38
Performing a Test Database Migration on Evaluation Mode	38
Performing a Database Migration	39
Known Issues	42
Troubleshooting & Other Resources	42



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) principles¹ 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.²

¹ <http://oracle.com/goto/maa>

² [Http://www.oracle.com/goto/zdm](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.

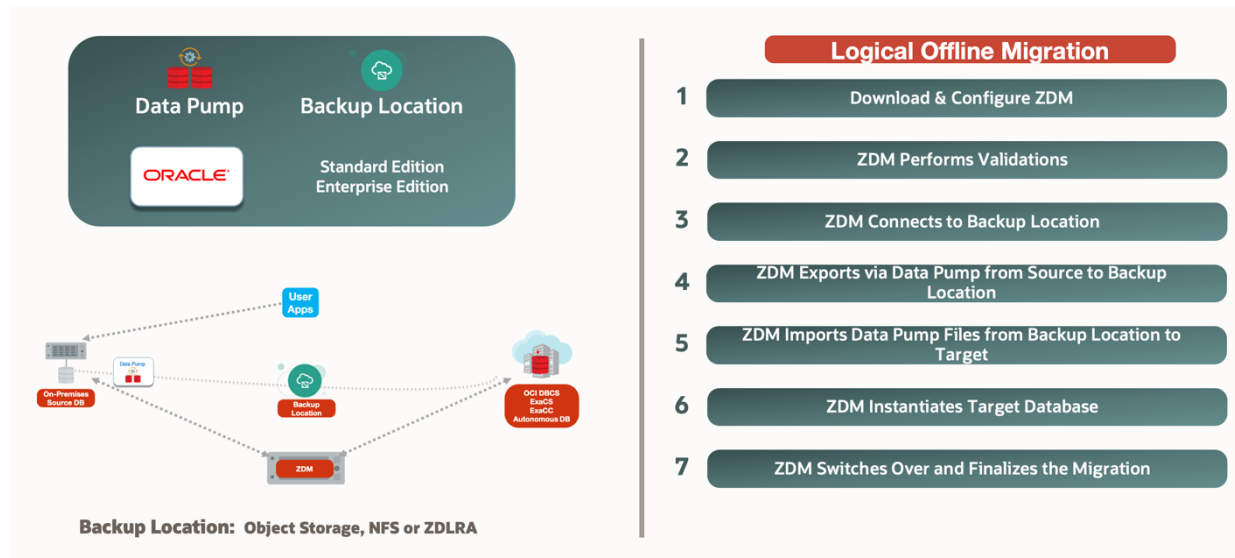


Figure 1 – Step-by-Step Logical Offline Migration with Data Pump and Backup Location

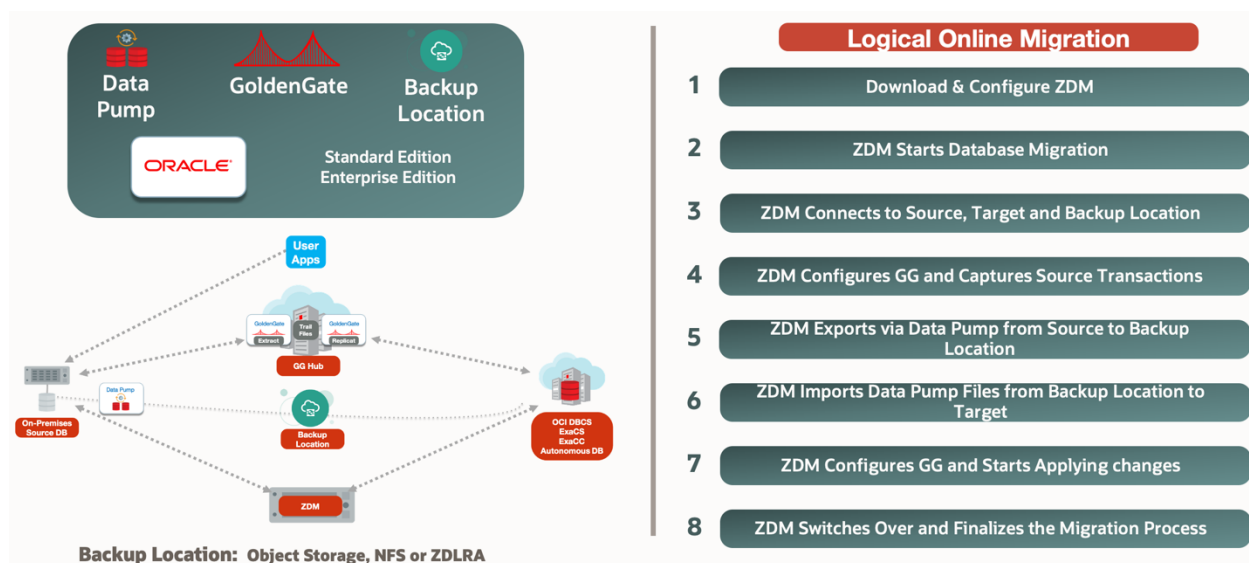


Figure 2 – Step-by-Step Logical Online Migration with Data Pump, GoldenGate and Backup Location

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`

For more information on the ZDM Service Host Requirements, please refer to Oracle ZDM's product documentation, specifically the *Setting Up Zero Downtime Migration Software*³ section.

³ <https://docs.oracle.com/en/database/oracle/zero-downtime-migration/21.1/zdmug/installing-zero-downtime-migration-software.html#GUID-A55FEDBA-236A-4006-91A5-6F28D100C5B2>

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**”. I’ll choose the name “zdmhost” and Oracle Linux 7.9.

Create Compute Instance

Create an instance to deploy and run applications, or save as a reusable Terraform stack for creating an instance with Resource Manager.

Name: zdmhost

Create in compartment: [Redacted]

Placement

The [availability domain](#) helps determine which shapes are available.

Availability domain: [Redacted]

[Show advanced options](#)

Image and shape

A [shape](#) is a template that determines the number of CPUs, amount of memory, and other resources allocated to an instance. The [image](#) is a template that determines the number of CPUs, amount of memory, and other resources allocated to an instance.

Image: Oracle Linux 7.9
Image build: 2021.04.09-0

Shape: VM.Standard.E4.Flex
Virtual Machine, 1 core OCPU, 16 GB memory, 1 Gbps network bandwidth

[Create](#) [Save as Stack](#) [Cancel](#)

Browse All Shapes

A shape is a template that determines the number of CPUs, amount of memory, and other resources allocated to a newly created instance. See [Compute Shapes](#) for more information.

Instance type: **Virtual Machine**
A virtual machine is an independent computing environment that runs on top of physical bare metal hardware.

Bare Metal Machine
A bare metal compute instance gives you dedicated physical server access for high performance.

Shape series: **AMD** Flexible OCPU count. AMD processors. **Intel** Current generation Intel processors. **Specialty and Previous Generation** Earlier generation AMD and Intel Standard GPU, and HPC shapes.

Shape Name	OCPU	Memory (GB)	Network Bandwidth (Gbps)	Max. Total VNICs
<input checked="" type="checkbox"/> VM.Standard2.1	1	15	1	1
Local Disk: Block Storage Only				
<input type="checkbox"/> VM.Standard2.2	2	30	2	2
<input type="checkbox"/> VM.Standard2.4	4	60	4.1	4.1
<input type="checkbox"/> VM.Standard2.8	8	120	8.2	8.2
<input type="checkbox"/> VM.Standard2.16	16	240	16.4	16.4
<input type="checkbox"/> VM.Standard2.24	24	320	24.6	24.6

1 Selected

Don't see the shape you want? [View your service limits and request an increase](#)

[Select Shape](#) [Cancel](#)

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:

Boot volume

Your [boot volume](#) is a detachable device that contains the image used to boot your compute instance.

☒ **Specify a custom boot volume size**
[Volume performance](#) varies with volume size. Default boot volume size: 40.6 GB

Boot volume size (GB)
200

Integer between 50 GB and 32,768 GB (32 TB). Must be larger than the default boot volume size for the selected image.

☐ **Use in-transit encryption**
[Encrypts data](#) in transit between the instance, the boot volume, and the block volumes.

☐ **Encrypt this volume with a key that you manage**
By default, Oracle manages the keys that encrypt this volume, but you can choose a key from a vault that you have access to if you want greater control over the key's lifecycle and how it's used. [Learn more about managing your own encryption keys](#)

[Show advanced options](#)

[Create](#) [Save as Stack](#) [Cancel](#)

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.

Visit www.oracle.com/database/technologies/rac/zdm-downloads.html



Name	Download
Zero Downtime Migration Download	 Download

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.

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:

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

Installed Packages
expect.x86_64                                     5.45-14.el7_1
@ol7_latest-x86_64
libaio.x86_64                                     0.3.109-
13.el7                                           @anaconda/7.9
oraclelinux-developer-release-el7.x86_64         1.0-6.el7
@ol7_latest
unzip.x86_64                                     6.0-21.el7
@anaconda/7.9
```

4 Download the ZDM software version 21c (zdm21.1.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:

```
[root@zdmhost opc]# mv /home/opc/zdm21.1.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 zdm21.1.zip
[root@zdmhost zdminstall]# chown zdmuser:zdm /home/zdmuser/zdminstall/zdm21.1.zip
```

5 Install the ZDM software. As zdmuser:

```
[root@zdmhost zdminstall]# su - zdmuser
[zdmuser@zdmhost ~]$ echo "ORACLE_HOME=/home/zdmuser/zdmhome; export ORACLE_HOME" >>
 ~/.bashrc
[zdmuser@zdmhost ~]$ echo "ORACLE_BASE=/home/zdmuser/zdmbase; export ORACLE_BASE" >>
 ~/.bashrc
[zdmuser@zdmhost ~]$ echo "ZDM_BASE=\$ORACLE_BASE; export ZDM_BASE" >> ~/.bashrc
[zdmuser@zdmhost ~]$ echo "ZDM_HOME=/home/zdmuser/zdmhome; export ZDM_HOME" >> ~/.bashrc
[zdmuser@zdmhost ~]$ echo "ZDM_INSTALL_LOC=/home/zdmuser/zdminstall; export
ZDM_INSTALL_LOC" >> ~/.bashrc
[zdmuser@zdmhost ~]$ cat ~/.bashrc
ORACLE_HOME=/home/zdmuser/zdmhome; 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 zdm21.1.zip
[zdmuser@zdmhost zdminstall]$ cd zdm21.1
-- Proceed to execute ZDM's installation script zdmuser:
[zdmuser@zdmhost zdm21.1]$ ./zdminstall.sh setup \
oraclehome=$ZDM_HOME \
oraclebase=$ZDM_BASE \
ziploc=./zdm_home.zip -zdm
-----
ZDM kit home: /home/zdmuser/zdminstall/zdm21.1
/home/zdmuser/zdminstall/zdm21.1
-----
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
-----
/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 sucessfully
-----

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/zdmhome/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/zdmhome/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/zdmhome/crsdata/zdmhost/rhp/mysql/metadata/mysql.sock -u root -p

spawn /home/zdmuser/zdmhome/mysql/server/bin/mysql --
socket=/home/zdmuser/zdmhome/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/zdmhome/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 zdm21.1]$ $ZDM_HOME/bin/zdmshservice start

Return code is 0

Server started successfully.

[zdmuser@zdmhost zdm21.1]$ $ZDM_HOME/bin/zdmshservice status

-----

Service Status

-----

Running:           true
Transferport:
Conn String:       jdbc:mysql://localhost:8897/
RMI port:          8895
HTTP port:         8896
Wallet path:       /home/zdmuser/zdmhome/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.

Protocol	Port	Purpose
TCP	22	SSH
TCP	1521, 2484 or a DB SCAN listener port	SQL*NET
SSL	443	OCI & GG REST Endpoint

You can find more information on the Oracle Zero Downtime Migration documentation section “Zero Downtime Migration Port Requirements.”⁴

⁴ <https://docs.oracle.com/en/database/oracle/zero-downtime-migration/21.1/zdmug/zero-downtime-migration-port-requirements.html#GUID-E6F8EF13-03A5-43DD-8F98-182632C83CB3>

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 ARCHIVE LOG mode. Please follow instructions on the Database Admin Guide “*Changing the Database Archiving Mode*” section ⁵.
- 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*” ⁶ 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)⁷
 - 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.

⁵ <https://docs.oracle.com/en/database/oracle/oracle-database/21/admin/managing-archived-redo-log-files.html#GUID-C12FA833-4717-430A-8919-5AFA747087B9>

⁶ <https://docs.oracle.com/en/database/oracle/zero-downtime-migration/21.1/zdmug/preparing-for-database-migration.html#GUID-004B4498-49B9-45A8-BD1B-5449857139A0>

⁷ <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.
- 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;
SUPPLEME 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;
SUPPLEME 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;

```

```

SQL> grant unlimited tablespace to c##ggadmin;
SQL> alter user c##ggadmin quota 100M on users;
SQL> grant select any dictionary to c##ggadmin;
SQL> grant create view to c##ggadmin;
SQL> grant execute on dbms_lock to c##ggadmin;
SQL> exec dbms_goldengate_auth.GRANT_ADMIN_PRIVILEGE('c##ggadmin',container=>' all');

-- Ensure that the wallet STATUS is OPEN and WALLET_TYPE is AUTOLOGIN
Enabling TDE on 11.2.0.4 is optional.
SQL> SELECT * FROM v$encryption_wallet;

```

WRL_TYPE	WRL_PARAMETER	STATUS
file	/opt/oracle/dcs/commonstore/wallets/tde/\$ORACLE_UNQNAME	OPEN

```

-- 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.

```



```
-- DATAPUMP Configuration
SQL> alter user SYSTEM identified by WELCOME##1234;
User altered.
--Grant export operations role to SYSTEM
SQL> grant DATAPUMP_EXP_FULL_DATABASE to system;
Grant succeeded.
-- Check the current Location of DATA_PUMP_DIR in dba_directories.
SQL> select directory_path from dba_directories where directory_name = 'DATA_PUMP_DIR';
DIRECTORY_PATH
-----
/u01/app/oracle/product/11.2.0.4/dbhome_1/rdbms/log/
```

- 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 *"Prepare the Source Database for Logical Migration"*⁸.

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.

Protocol	Port	Purpose
TCP	22	SSH
TCP	1521, 2484 or a DB SCAN listener port	SQL*NET
SSL	443	Database Backup Store. OCI OSS.

You can find more information on the Oracle Zero Downtime Migration documentation section *"Zero Downtime Migration Port Requirements"*⁹.

⁸ <https://docs.oracle.com/en/database/oracle/zero-downtime-migration/21.1/zdmug/preparing-for-database-migration.html#GUID-004B4498-49B9-45A8-BD1B-5449857139A0>

⁹ <https://docs.oracle.com/en/database/oracle/zero-downtime-migration/21.1/zdmug/zero-downtime-migration-port-requirements.html#GUID-E6F8EF13-03A5-43DD-8F98-182632C83CB3>

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.

Create DB System

DB System Information Database Information

Provide information for the initial database

Database name DB0507

Database image Oracle Database 19c Change Database Image

PDB name Optional pdb1

Create administrator credentials

Username Read-Only sys

Password

Confirm password

☒ Use the administrator password for the TDE wallet

Select workload type

Transaction Processing Configure the database for a transactional workload, with bias towards high volumes of random data access.

Data Warehouse Configure the database for a decision support or data warehouse workload, with bias towards large data scanning operations.

Configure database backups

☐ Enable automatic backups

Show Advanced Options

Previous Create DB System Cancel

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

```
-- TARGET Time zone check
SQL> SELECT * FROM v$timezone_file;
FILENAME          VERSION  CON_ID
-----
timezlrg_35.dat      35        0
```

```

-- SRVCTL Configuration Show current settings on Target
[oracle@target ~]$ srvctl config database -d TARGET_DB -a
Database unique name: TARGET_DB_nrtlwq
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.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=pdh1;
```

```
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
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

-- 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.

Protocol	Port	Purpose
TCP	22	SSH
TCP	1521, 2484 or a DB SCAN listener port	SQL*NET
SSL	443	Database Backup Store. OCI OSS.

You can find more information on the Oracle Zero Downtime Migration documentation section “*Zero Downtime Migration Port Requirements*”.¹⁰

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:qQd25G01EtiUCUwdGTYPHeeps6f2vOqJHV6cHBeUSWg zdmuser@zdmhost
The key's randomart image is:
+---[RSA 2048]----+
|      oo*OB..+oo|
|      o.*=.E.+ |
|      . . = o. |
|      o o o o .|
|      o S o o . .|
|      . + . = + |
|      . . o * |
|      . +.B |
|      .oBo+. |
+----[SHA256]-----+
```

¹⁰ <https://docs.oracle.com/en/database/oracle/zero-downtime-migration/21.1/zdmug/zero-downtime-migration-port-requirements.html#GUID-E6F8EF13-03A5-43DD-8F98-182632C83CB3>

```
[zdmuser@zdmhost ~]$ cd ~/.ssh
[zdmuser@zdmhost .ssh]$ cat id_rsa.pub >> authorized_keys
[zdmuser@zdmhost .ssh]$ chmod 600 authorized_keys
[zdmuser@zdmhost .ssh]$ ll
total 16
-rw-----. 1 zdmuser zdm 397 June 25 03:09 authorized_keys
-rw-----. 1 zdmuser zdm 1675 June 25 03:08 id_rsa
-rw-r--r--. 1 zdmuser zdm 397 June 25 03:08 id_rsa.pub
-rw-r--r--. 1 zdmuser zdm 178 June 25 06:57 known_hosts
```

You can find more information on ZDM Product's documentation section [Generating a Private SSH Key Without a Phrase](#)¹¹.

Before continuing with the migration environment setup, rename the id_rsa.pub file to <zdm_service_host_name>.ppk on the ZDM Service Host.

```
[zdmuser@zdmhost .ssh]$ cd /home/zdmuser/.ssh
[zdmuser@zdmhost .ssh]$ mv id_rsa zdm.ppk
[zdmuser@zdmhost .ssh]$ ll
total 16
-rw-----. 1 zdmuser zdm 397 Jul 7 03:09 authorized_keys
-rw-r--r--. 1 zdmuser zdm 397 Jul 7 03:08 id_rsa.pub
-rw-r--r--. 1 zdmuser zdm 178 Jul 7 06:57 known_hosts
-rw-----. 1 zdmuser zdm 1675 Jul 7 03:08 zdm.ppk
```

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
10.0.0.30 source.publicsubnet.xyz.xyz.com source ##Source Database
10.0.0.49 target.publicsubnet.xyz.xyz.com target ##Target Database
10.0.0.158 ogg19cora.publicsubnet.xyz.xyz.com ogg19cora ##OGG Hub
10.0.0.225 zdmhost.publicsubnet.xyz.xyz.com zdmhost ##ZDM Host
```

¹¹ <https://docs.oracle.com/en/database/oracle/zero-downtime-migration/21.1/zdmug/preparing-for-database-migration.html#GUID-602216AD-4B59-441B-BA8D-AC7B333038FD>

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  
AAAAB3NzaC1yc2EAAAADAQABAAQDhdpp8nQznvsyTc7+9Lxq0xgUHnWYJmaJpHrG8jd4ke8EWaI5YULLvuak12Z4  
kawKfQALhTiImZ+RCzjUlwG7liqN/0n3pymV1pazN0vzdpnYZSfKL7SnXiB8ZcQTPeCTjPYFMcYofTq+9jDMwP0RVt  
0Td5sDud1Y3qZh3j0TlW1Qs2Xc67RI6TD7orv+XsdmGMXVHg48wlToQvXTuiebM7XW9M26vtBu3FWa0C21UmJaOcQK  
ND6rJxibFLH72MLvvXBGecU4qDL/QN/NEKXYXdgUUHdpv/mNp/JgMkFAdpsZEALqbL4cWrSlg7OiyjoX6uaaaGaJjfi  
hZf0NbUH 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  
AAAAB3NzaC1yc2EAAAADAQABAAQDhdpp8nQznvsyTc7+9Lxq0xgUHnWYJmaJpHrG8jd4ke8EWaI5YULLvuak12Z4  
kawKfQALhTiImZ+RCzjUlwG7liqN/0n3pymV1pazN0vzdpnYZSfKL7SnXiB8ZcQTPeCTjPYFMcYofTq+9jDMwP0RVt  
0Td5sDud1Y3qZh3j0TlW1Qs2Xc67RI6TD7orv+XsdmGMXVHg48wlToQvXTuiebM7XW9M26vtBu3FWa0C21UmJaOcQK  
ND6rJxibFLH72MLvvXBGecU4qDL/QN/NEKXYXdgUUHdpv/mNp/JgMkFAdpsZEALqbL4cWrSlg7OiyjoX6uaaaGaJjfi  
hZf0NbUH 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.

The screenshot illustrates the process of generating an authentication token in the OCI console. On the left, the 'Resources' sidebar shows 'Auth Tokens' selected. The main area displays the 'Auth Tokens' table with a 'Generate Token' button highlighted. Below this, the 'Generate Token' dialog box is shown with the description 'Token for Object Storage' and a 'Generate Token' button highlighted. Finally, the 'Generated Token' modal is shown, displaying the token 'Du>Bg3<vRF14y;0#WE...' with 'Hide' and 'Copy' buttons highlighted.

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]$ bash -c "$(curl -L https://raw.githubusercontent.com/oracle/oci-cli/master/scripts/install/install.sh)"

[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: ocidl.user.oc1..xyz
Enter a tenancy OCID: ocidl.tenancy.oc1..xyz

Enter a region by index or name(e.g.1: ap-chiyoda-1, 2: ap-chuncheon-1, 3: ap-hyderabad-1, 4: ap-melbourne-1, 5: ap-mumbai-1,6: ap-osaka-1, 7: ap-seoul-1, 8: ap-sydney-1, 9: ap-tokyo-1, 10: ca-montreal-1,11: ca-toronto-1, 12: eu-amsterdam-1, 13: eu-frankfurt-1, 14: eu-zurich-1, 15: me-dubai-1,16: me-jeddah-1, 17: sa-santiago-1, 18: sa-saopaulo-1, 19: uk-cardiff-1, 20: uk-gov-cardiff-1,21: uk-gov-london-1, 22: uk-london-1, 23: us-ashburn-1, 24: us-gov-ashburn-1, 25: us-gov-chicago-1,26: us-gov-phoenix-1, 27: us-langley-1, 28: us-luke-1, 29: us-phoenix-1, 30: us-sanjose-1): ap-tokyo-1

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
Fingerprint: 31:xx:tt:4r:yy:bb:aa:cc:dd:ee:vv:ll:mm:nn:00
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-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA3X1Lpl1267BfXFrIgtHE
A7o/hVo89AHKgnkz1eSHS55tfvuglk8SOXAhV+7gTvDurcQz5AfIB0Uq22EvrGLq
JWwjTf6Z/xi/tCBNgNdpUln4LAnC8D4PSgSXMJc6SFkkDXUliqqjnpSrKAOnavnt
0n5csuAyWjqxHMbS/NX0J+0UH4rJmhIoxayfTAHs9WPWg+Vg8OG/sxwfp5rhS1td
agLEKSYvE/Wy0hz/o9poBkoZ3ELrnWY9yBVW3Gw9fSEVMXxehAqofkrR6LilPbPv
eRZqbyxDUN8/C4P5q8Y23OCAsuH8bAH70rbJpQk0FvCQu6KnPVUDFF6G+GafRa7X
yQIDAQAB
-----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"**.

API Keys

Add API Key

Fingerprint

Add API Key

Public Key

```
-----BEGIN PUBLIC KEY-----
MIIBBgkqhkiG9w0BAQEFAAACQASMBKjCCKCAQEA3X1Plt2678fXfgrtHE
A7whU89AHKZKc1e5HS55fivugk850XANV+7gtDwrcQZ5ARBU0q2ZEwrlQ
VJwvjtRZxHCElNgdpUdALnCD4P5g5XJMjcdSFxXUkqagiparKA0navnt
0n5c5uayVjXp0tMB5NDQJ+0uHxJmibxayTAH9wPWiV-Vjg0G0a9wPfn5t1d
yQ5Sc3GhY9h-zip0BqBzE1eM1Y9-BVVOQ5w5REMIkxehqRlRwPnRP-
rE20y0zQUN8/C4PsgP+230CAshl8BAH70rbJpQMfVCu6KnPvUDFF6G-GaRtX
yQDAQAB
-----END PUBLIC KEY-----
```

Add **Cancel**

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
fingerprint=31:xx:tt:4r:yy:bb:aa:cc:dd:ee:vv:ll:mm:nn:00
```

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

```
[root@source opc]# cat /etc/hosts
10.0.0.30    source.publicsubnet.xyz.xyz.com    source
192.168.16.18  source-priv.publicsubnet.xyz.xyz.com  source-priv
10.0.0.30    source-vip.publicsubnet.xyz.xyz.com  source-vip
10.0.0.30    source-scan.publicsubnet.xyz.xyz.com  source-scan
10.0.0.49    target.publicsubnet.xyz.xyz.com      target
10.0.0.49    target-scan.publicsubnet.xyz.xyz.com  target-scan
```

Target Server

```
[root@target opc]# cat /etc/hosts
10.0.0.49 target.publicsubnet.xyz.xyz.com target
192.168.16.18 target-priv.publicsubnet.xyz.xyz.com target-priv
10.0.0.49 target-vip.publicsubnet.xyz.xyz.com target-vip
10.0.0.49 target-scan.publicsubnet.xyz.xyz.com target-scan
10.0.0.30 source.publicsubnet.xyz.xyz.com source
10.0.0.30 source-scan.publicsubnet.xyz.xyz.com source-scan
```

Testing Connectivity between Source and Target via SCAN

```
--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=10.0.0.49) (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=10.0.0.30) (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 Tunnel](#)¹².

¹² <https://docs.oracle.com/en/database/oracle/zero-downtime-migration/21.1/zdmug/preparing-for-database-migration.html#GUID-808F9EBE-036E-48CF-B55F-396C3681754E>

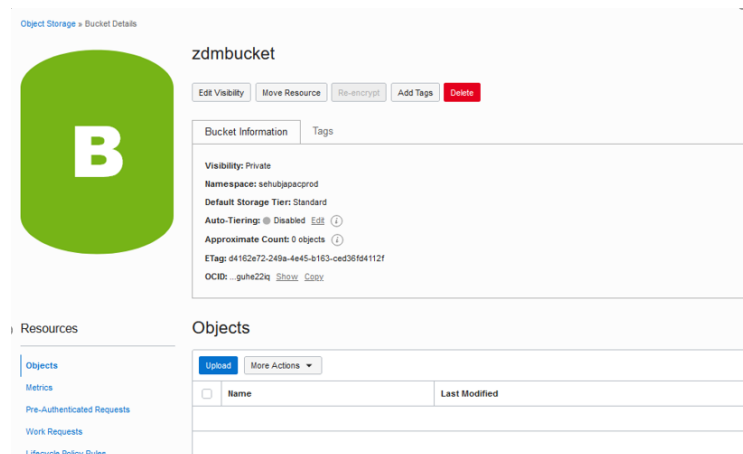
BACKUP LOCATION

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`:

```
[zdmuser@zdmhost lib]$ oci os bucket create --compartment-id ocidl.compartment.oc1..xyz
--name zdmbucket
{
  "data": {
    "approximate-count": null,
    "approximate-size": null,
    "auto-tiering": null,
    "compartment-id":
Enter a user OCID: ocidl.user.oc1..xyz
Enter a tenancy OCID: ocidl.tenancy.oc1..xyz
"ocidl.compartment.oc1..xyz",
    "created-by": "ocidl.user.oc1..xyz",
    "defined-tags": {
      "default_tags": {
        "CreatedBy": "xyz@zdmtesting.xyzxyz",
        "CreatedOn": "2021-07-05T05:08:28.372Z"
      }
    },
    "etag": "d4162e72-249a-4e45-b163-ced36fd4112f",
    "freeform-tags": {},
    "id": "ocidl.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": "d4162e72-249a-4e45-b163-ced36fd4112f"
}
```

Next, open the Object Storage Page from within your Oracle Cloud Account; it should show that recently created bucket as per below:



GOLDENGATE HUB

GoldenGate Hub Requirements

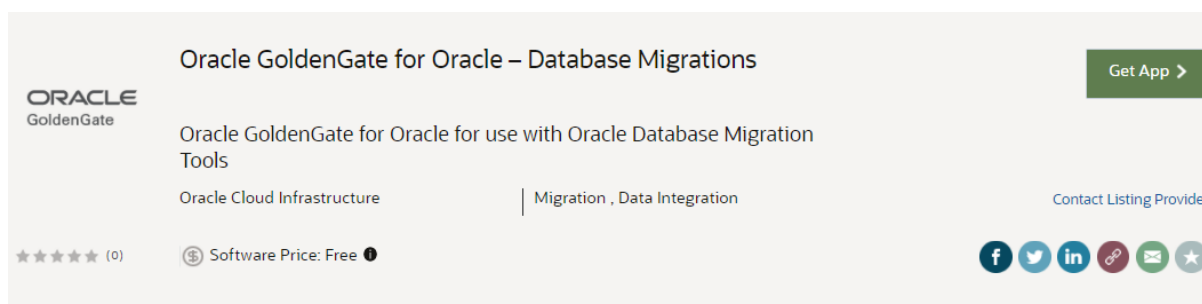
Logical Online migration to both DBCS and ExaCS requires an Oracle GoldenGate Microservices hub. To set this up, please follow the instructions as per:

- [Deploying Oracle GoldenGate Microservices](#)

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

- [Oracle GoldenGate for Oracle – Database Migrations](#)

Please visit the link and from the main menu, choose “Oracle GoldenGate for Oracle-Database Migrations”, and continue with the default version “19.1.0.0.201013 (Microservices Edition)”.



Provide the required information and keep the default deployment names “Source” and “Target”. Create Oracle GoldenGate Deployments for source and target as follows:

- **Source Database:** Oracle 11g (11.2.0.4)
- **Target Database:** Oracle 19c (19.x)

Create Stack

Stack Information

Availability Domain: [Redacted]

Compute Shape: VM.Standard2.1

Assign Public IP: ☒ Includes if the new VM should have a public IP address.

Custom Volume Sizes: ☐ Use this field to customize the size of new block storage volumes.

Create OGG Deployments

Deployment 1 - Name: Source

Deployment 1 - Database: Oracle 11g (11.2.0.4)

Deployment 1 - Oracle RCU Version for deployment 1: [Redacted]

Deployment 2 - Name: Target

Deployment 2 - Database: Oracle 19c (19.4)

Deployment 2 - Oracle RCU Version for deployment 2: [Redacted]

Deployment 2 - Autonomous Database: ☐ Select if deployment 2 replicates to an Autonomous Database.

Shell Access

SSH Public Key: [Redacted]

Public key for allowing SSH access to the VM will be:

Previous Next Cancel

Upon creation, the compute VM is now active.

Compute » Instances » Instance Details

Oracle GoldenGate Microservices Edition for Oracle 19.1.0.0.201013-ZDM

Start Stop Reboot Edit More Actions

Instance Information Oracle Cloud Agent Tags

General Information

Availability Domain: AD-1

Fault Domain: FD-1

Region: [Redacted]

OCPU: [Redacted] (Status: Green)

Launch Mode: [Redacted]

Compartment: [Redacted]

Capacity type: On-demand

Instance Details

Virtual Cloud Network: [Redacted]

Maintenance Reboot: -

Image: ogg-19.1.0.0.201013-Microservices-Oracle

Launch Mode: NATIVE

Instance Metadata Service: Versions 1 and 2 [Edit](#)

Live Migrations: Use recommended default

Maintenance Recovery Action: Restore Instance

Shape Configuration

Shape: VM.Standard2.1

OCPU Cores: 1

Network Bandwidth (Gbps): 1

Memory (GB): 15

Local Disk: Block Storage Only

Instance Access

You [connect to a Compute VM instance](#) using a Secure Shell (SSH) connection. You'll need the private key from the SSH key pair that was used to create the instance.

Public IP Address: [IP address](#) [Copy](#)

Username: opc

Primary VNIC

Private IP Address: 10.0.0.158

Network Security Groups: None [Edit](#)

Subnet: [Public_subnet](#)

Private DNS record: Enable

Hostname: ogg19cora

Internal FQDN: ogg19cora... [Show](#) [Copy](#)

Launch Options

HC Attachment Type: RAW/VIRTUALIZED

Remote Data Volume: RAW/VIRTUALIZED

Firmware: UEFI_64

Boot Volume Type: RAW/VIRTUALIZED

In-transit Encryption: Disabled

Resources

Metrics

[Metrics](#) are measurements that let you monitor the health, capacity, and performance of your resources.

Connect to the VM via SSH using its Public IP address in order to get the oggadmin user password:

```
-bash-4.2$ cat /home/opc/ogg-credentials.json
{"username": "oggadmin", "credential": "JjWjigAx._e_3u_e"}
```

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

```
-bash-4.2$ cat /etc/hosts
10.0.0.158 ogg19cora.x.y.z.com ogg19cora
```

The GoldenGate Hub uses a Ngnix Reverse Proxy which requires a self-signed certificate; hence, it is necessary to add the self-signed certificate used by the GoldenGate Market place image as a trusted certificate to the ZDM Service host.

To achieve this, follow the instructions on [Zero Downtime Migration - GoldenGate Hub Certificate Known Issues \(Doc ID 2768483.1\)](#)¹³. Once you have copied the provided script and saved it as `zdm_add_cert.sh`, execute the following:

```
[zdmuser@zdmhost ~]$ chmod a+x zdm_add_cert.sh
[zdmuser@zdmhost ~]$ echo $ORACLE_HOME
/home/zdmuser/zdmhome
[zdmuser@zdmhost ~]$ ./zdm_add_cert.sh ogg19cora
stopping zdmserver
spawn /home/zdmuser/zdmhome/mysql/server/bin/mysqladmin
--defaults-file=/home/zdmuser/zdmhome/crsdata/zdmhost/rhp/conf/my.cnf -u root -p shutdown
WARNING: oracle.jwc.rest does not exist in the configuration file. It will be FALSE by default.
WARNING: oracle.jwc.jmx does not exist in the configuration file. It will be TRUE by default.
[jwcctl debug] Environment ready to start JWC
[jwcctl debug] Return code of initialization: [0]
[jwcctl debug] ... BEGIN_DEBUG [Action= stop] ...
Stop JWC
[jwcctl debug] Get JWC PIDs
[jwcctl debug] Done Getting JWC PIDs
[jwcctl debug] ... JWC Container (pid=19859) ...
[jwcctl debug] Stop command:-Dcatalina.base=/home/zdmuser/zdmhome/crsdata/zdmhost/rhp -
Doracle.tls.enabled=false -Doracle.wlm.dbwlmlogger.logging.level=FINEST -
Doracle.jwc.client.logger.file.name=/home/zdmuser/zdmhome/crsdata/zdmhost/rhp/logs/jwc_shutter_s
tdout_err %g.log -Doracle.jwc.client.logger.file.number=10 -
Doracle.jwc.client.logger.file.size=1048576 -
Doracle.jwc.wallet.path=/home/zdmuser/zdmhome/crsdata/zdmhost/security -
Doracle.jmx.login.credstore=WALLET -classpath /home/zdmuser/zdmhome/jlib/jwc-
logging.jar:/home/zdmuser/zdmhome/jlib/jwc-client.jar:/home/zdmuser/zdmhome/jlib/jwc-
security.jar:/home/zdmuser/zdmhome/jdk/lib/tools.jar:/home/zdmuser/zdmhome/tomcat/lib/tomcat-
juli.jar oracle.cluster.jwc.tomcat.client.ShutdownContainer 19859
[jwcctl debug] Get JWC shutter PIDs
[jwcctl debug] Done getting JWC shutter PIDs
[jwcctl debug] ... JWC shutter command (pid=21113) ...
[jwcctl debug] ... Initial Check - JWC Shutter JVM waiting (pid=21113) ...
[jwcctl debug] ... Sleep for 1 seconds ...
[jwcctl debug] ... Iteration 0 Check for JWC Shutter ...
[jwcctl debug] Get JWC shutter PIDs
[jwcctl debug] Done getting JWC shutter PIDs
[jwcctl debug] ... JWC shutter command (pid=21113) ...
[jwcctl debug] ... Iteration 0 JWC Shutter waiting (pid=21113) ...
[jwcctl debug] ... Sleep for 1 seconds ...
[jwcctl debug] ... Iteration 1 Check for JWC Shutter ...
```

¹³ <https://support.oracle.com/epmos/faces/DocContentDisplay?id=2768483.1>

```
[jwcctl debug] Get JWC shutter PIDs
[jwcctl debug] Done getting JWC shutter PIDs
[jwcctl debug] ... JWC shutter command (pid=21113) ...
[jwcctl debug] ... Iteration 1 JWC Shutter waiting (pid=21113) ...
[jwcctl debug] ... Sleep for 1 seconds ...
[jwcctl debug] ... Iteration 2 Check for JWC Shutter ...
[jwcctl debug] Get JWC shutter PIDs
[jwcctl debug] Done getting JWC shutter PIDs
[jwcctl debug] ... JWC shutter command not found ...
[jwcctl debug] ... Iteration 2 Check for JWC Container ...
[jwcctl debug] Get JWC PIDs
[jwcctl debug] Done Getting JWC PIDs
[jwcctl debug] ... JWC containers not found ...
[jwcctl debug] ... JWC Container is stopped ...
[jwcctl debug] ... STOP - Return code = 0 ...
[jwcctl debug] ... END_DEBUG [Action=stop] ...
[jwcctl debug] Return code of AGENT: [0]
```

Return code is 0

zdm service stopped successfully.

Adding certs to /home/zdmuser/zdmhome/jdk/jre/lib/security/cacerts

extracting certificate from /tmp/ogg19cora_443.pem

Certificate was added to keystore

Certificate was added to keystore

starting zdmserver

No instance detected, starting zdm service

mysqld will log errors to /home/zdmuser/zdmhome/crsdata/zdmhost/rhp/mysql/metadata/mysql-error.log

mysqld is running as pid 21237

WARNING: oracle.jwc.rest does not exist in the configuration file. It will be FALSE by default.

WARNING: oracle.jwc.jmx does not exist in the configuration file. It will be TRUE by default.

```
[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:
/home/zdmuser/zdmhome/crsdata/zdmhost/rhp/conf/jwc.properties
```

```
[jwcctl debug] oracle.jmx.login.credstore = CRSCRED
```

```
[jwcctl debug] oracle.jmx.login.args = DOMAIN=rhp CACHE_ENABLED=true CACHE_EXPIRATION=180
```

```
[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.tls.rmi.clientfactory = RELOADABLE
[jwcctl debug] oracle.jwc.lifecycle.start.log.fileName = JWCStartEvent.log
[jwcctl debug] oracle.jwc.http.connector.ssl.protocols = TLSv1.2,TLSv1.3
[jwcctl debug] Get JWC PIDs
[jwcctl debug] Done Getting JWC PIDs
[jwcctl debug] ... JWC containers not found ...

[jwcctl debug] Start command:-server -Xms2048M -Xmx4096M -Djava.awt.headless=true -
Ddisable.checkForUpdate=true -
Djava.util.logging.config.file=/home/zdmuser/zdmbase/crsdata/zdmhost/rhp/conf/logging.properties
-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -DTRACING.ENABLED=true -
DTRACING.LEVEL=2 -Doracle.wlm.dbwlmlogger.logging.level=FINEST -Duse_scan_IP=true -
Djava.rmi.server.hostname=zdmhost -Doracle.http.port=8896 -Doracle.jmx.port=8895 -
Doracle.tls.enabled=false -Doracle.jwc.tls.http.enabled=false -
Doracle.rhp.storagebase=/home/zdmuser/zdmbase -Djava.security.manager -
Djava.security.policy=/home/zdmuser/zdmbase/crsdata/zdmhost/rhp/conf/catalina.policy -
Djava.security.egd=file:/dev/urandom -
Doracle.jwc.wallet.path=/home/zdmuser/zdmbase/crsdata/zdmhost/security -
Doracle.jmx.login.credstore=WALLET -Doracle.rest.enabled=false -Doracle.jmx.enabled=true -
Dcatalina.home=/home/zdmuser/zdmhome/tomcat -
Dcatalina.base=/home/zdmuser/zdmbase/crsdata/zdmhost/rhp -
Djava.io.tmpdir=/home/zdmuser/zdmbase/crsdata/zdmhost/rhp/temp -
Doracle.home=/home/zdmuser/zdmhome -Doracle.jwc.mode=STANDALONE -classpath
/home/zdmuser/zdmhome/jlib/cryptoj.jar:/home/zdmuser/zdmhome/jlib/oraclepki.jar:/home/zdmuser/zd
mhome/jlib/osdt_core.jar:/home/zdmuser/zdmhome/jlib/osdt_cert.jar:/home/zdmuser/zdmhome/tomcat/1
ib/tomcat-
juli.jar:/home/zdmuser/zdmhome/tomcat/lib/bootstrap.jar:/home/zdmuser/zdmhome/jlib/jwc-
logging.jar org.apache.catalina.startup.Bootstrap start

[jwcctl debug] Get JWC PIDs
[jwcctl debug] Done Getting JWC PIDs
[jwcctl debug] ... JWC Container (pid=21306) ...
[jwcctl debug] ... JWC Container running (pid=21306) ...

[jwcctl debug] Check command:-Djava.net.preferIPv6Addresses=true -
Dcatalina.base=/home/zdmuser/zdmbase/crsdata/zdmhost/rhp -
Doracle.wlm.dbwlmlogger.logging.level=FINEST -
Doracle.jwc.client.logger.file.name=/home/zdmuser/zdmbase/crsdata/zdmhost/rhp/logs/jwc_checker_s
tdout_err_%g.log -Doracle.jwc.client.logger.file.number=10 -
Doracle.jwc.client.logger.file.size=1048576 -
Doracle.jwc.wallet.path=/home/zdmuser/zdmbase/crsdata/zdmhost/security -
Doracle.jmx.login.credstore=WALLET -Doracle.tls.enabled=false -
Doracle.jwc.tls.http.enabled=false -classpath /home/zdmuser/zdmhome/jlib/jwc-
logging.jar:/home/zdmuser/zdmhome/jlib/jwc-security.jar:/home/zdmuser/zdmhome/jlib/jwc-
client.jar:/home/zdmuser/zdmhome/jlib/jwc-
cred.jar:/home/zdmuser/zdmhome/jlib/srvn.jar:/home/zdmuser/zdmhome/jlib/srvnhas.jar:/home/zdmuse
r/zdmhome/jlib/cryptoj.jar:/home/zdmuser/zdmhome/jlib/oraclepki.jar:/home/zdmuser/zdmhome/jlib/o
sdt_core.jar:/home/zdmuser/zdmhome/jlib/osdt_cert.jar:/home/zdmuser/zdmhome/tomcat/lib/tomcat-
juli.jar oracle.cluster.jwc.tomcat.client.JWCChecker localhost 8896 -1

[jwcctl debug] ... JWC Container is ready ...
[jwcctl debug] ... START - Return code = 0 ...
[jwcctl debug] ... END_DEBUG [Action=start] ...
[jwcctl debug] Return code of AGENT: [0]

Return code is 0

Server started successfully.

```

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 allow 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](#)¹⁴.

Response File Parameters used in this Guide

PARAMETER	DESCRIPTION
MIGRATION_TYPE	<ul style="list-style-type: none">• ONLINE_LOGICAL → ZDM will leverage Data Pump and Oracle GoldenGate for the migration process. <i>(Used in this guide)</i>• OFFLINE_LOGICAL → ZDM will leverage Data Pump only for the migration process
DATA_TRANSFER_MEDIUM	<ul style="list-style-type: none">• OSS → Object Storage Service. <i>(Used in this guide)</i>• NFS → Network File System (for ExaC@C targets only)• DBLINK → Direct transfer of data over database link• COPY → secure copy (for user managed targets only)
TARGETDATABASE_OCID	<ul style="list-style-type: none">• Specifies the connection details for the Oracle Cloud resource identifier for the target database.
TARGETDATABASE_ADMINUSERNAME	<ul style="list-style-type: none">• Specifies the Target Database administrator's user name.
SOURCEDATABASE_ADMINUSERNAME	<ul style="list-style-type: none">• Specifies the Source Database administrator's user name.
SOURCEDATABASE_CONNECTIONDETAILS_HOST	<ul style="list-style-type: none">• Specifies the listener host name or IP address. This is only for DBCS/ExaCS/ExaC@C targets.
SOURCEDATABASE_CONNECTIONDETAILS_PORT	<ul style="list-style-type: none">• Specifies the listener port number. This is only for DBCS/ExaCS/ExaC@C targets.
TARGETDATABASE_CONNECTIONDETAILS_SERVICENAME	<ul style="list-style-type: none">• Specifies the fully qualified service name
TARGETDATABASE_CONNECTIONDETAILS_HOST	<ul style="list-style-type: none">• Specifies the listener host name or IP address.

¹⁴ <https://docs.oracle.com/en/database/oracle/zero-downtime-migration/21.1/zdmug/zero-downtime-migration-logical-migration-response-file-parameters-reference.html#GUID-D580AD1C-C209-4F0F-A630-863D206FF0E5>

TARGETDATABASE_CONNECTIONDETAILS_PORT	<ul style="list-style-type: none"> Specifies the listener port number.
TARGETDATABASE_CONNECTIONDETAILS_SERVICENAME	<ul style="list-style-type: none"> Specifies the fully qualified service name.
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_TENANTID	<ul style="list-style-type: none"> Specifies the OCI of the OCI tenancy.
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_USERID	<ul style="list-style-type: none"> Specifies the OCID of the IAM user.
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_FINGERPRINT	<ul style="list-style-type: none"> Specifies the fingerprint of the public API key.
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_PRIVATEKEYFILE	<ul style="list-style-type: none"> Specifies the absolute path of the API private key file.
OCIAUTHENTICATIONDETAILS_REGIONID	<ul style="list-style-type: none"> Specifies the OCI region identifier.
SOURCEDATABASE_GGADMINUSERNAME	<ul style="list-style-type: none"> Specifies the GoldenGate administrator's user name.
TARGETDATABASE_GGADMINUSERNAME	<ul style="list-style-type: none"> Specifies the GoldenGate administrator's user name for the PDB.
GOLDENGATEHUB_ADMINUSERNAME	<ul style="list-style-type: none"> Specifies the GoldenGate Hub administrator's user name.
GOLDENGATEHUB_URL	<ul style="list-style-type: none"> Specifies the GoldenGate Hub's REST endpoint.
GOLDENGATEHUB_SOURCEDEPLOYMENTNAME	<ul style="list-style-type: none"> Specifies the name of the GoldenGate Microservices deployment to operate on the Source Database.
GOLDENGATEHUB_TARGETDEPLOYMENTNAME	<ul style="list-style-type: none"> Specifies the name of the GoldenGate Microservices deployment to operate on the Target Database.
GOLDENGATEHUB_COMPUTEID	<ul style="list-style-type: none"> Specifies the Oracle Cloud identifier of the VM.

DATAPUMPSETTINGS_JOBMODE	<ul style="list-style-type: none"> Specifies the Data Pump export mode: <ul style="list-style-type: none"> FULL SCHEMA TABLE TABLESPACE TRANSPORTABLE (not supported by ZDM)
INCLUDEOBJECTS-N	<ul style="list-style-type: none"> 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_DATAPUMPPARAMETERS_IMPORTPARALLELISMDEGREE	<ul style="list-style-type: none"> Specifies the maximum number of worker processes that a Data Pump import job can use.
DATAPUMPSETTINGS_DATAPUMPPARAMETERS_EXPORTPARALLELISMDEGREE	<ul style="list-style-type: none"> Specifies the maximum number of worker processes that a Data Pump export job can use.
DATAPUMPSETTINGS_DATABUCKET_NAME SPACENAME	<ul style="list-style-type: none"> Specifies the object storage bucket namespace.
DATAPUMPSETTINGS_DATABUCKET_BUCKETNAME	<ul style="list-style-type: none"> Specifies the object storage bucket name.
DATAPUMPSETTINGS_EXPORTDIRECTORYOBJECT_NAME	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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_OMITENCRYPTIONCLAUSE	<ul style="list-style-type: none"> When enabled, this parameter sets <code>TRANSFORM=OMIT_ENCRYPTION_CLAUSE</code>, 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 bin]$ cp /home/zdmuser/zdmhome/rhp/zdm/template/zdm_logical_template.rsp
/home/zdmuser

[zdmuser@zdmhost bin]$ 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 zdbase
drwxr-xr-x. 47 zdmuser zdm  4096 Jul  7 02:47 zdmhome
drwxr-xr-x.  3 zdmuser zdm    40 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


# migration method
MIGRATION_METHOD=ONLINE_LOGICAL
DATA_TRANSFER_MEDIUM=OSS


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


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


# target db (PDB)
TARGETDATABASE_CONNECTIONDETAILS_HOST=target
TARGETDATABASE_CONNECTIONDETAILS_PORT=1521
TARGETDATABASE_CONNECTIONDETAILS_SERVICENAME=pdb1.publicsubnet.xyz.xyz.com
```

```

# oci cli OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_TENANTID=ocidl.tenancy.oc1.xyz.xyz
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_USERID=ocidl.user.oc1..xyz
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_FINGERPRINT=331:xx:tt:4r:yy:bb:aa:cc:dd:ee:vv:ll:mm:
nn:00
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=ocidl.instance.oc1.xyz.yyx.xyz


# data pump
DATAPUMPSETTINGS_JOBMODE=SCHEMA
INCLUDEOBJECTS-1=owner:TEST
DATAPUMPSETTINGS_DATAPUMPPARAMETERS_IMPORTPARALLELISMDEGREE=2
DATAPUMPSETTINGS_DATAPUMPPARAMETERS_EXPORTPARALLELISMDEGREE=2
DATAPUMPSETTINGS_DATABUCKET_NAMESPACENAME=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

```

LOGICAL MIGRATION TO DBCS AND EXACS WITH ZDM

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 \
-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: JjWjigAx._e_3u_e
Enter Authentication Token for OCI user "ocid1.user.oc1..xyz": Du>:Bg3<vRft4y;0#WE_
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
```

```

Current status: SUCCEEDED
Result file path: "/home/zdmuser/zdmbase/chkbase/scheduled/job-30-2021-07-08-06:32:09.log"
Job execution start time: 2021-07-08 06:32:09
Job execution end time: 2021-07-08 06:34:33
Job execution elapsed time: 2 minutes 23 seconds
ZDM_VALIDATE_TGT ..... COMPLETED
ZDM_VALIDATE_SRC ..... 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_CLEANUP_SRC ..... COMPLETED

```

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 \
-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 \

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: JjWjigAx._e_3u_e
Enter Authentication Token for OCI user "ocidl.user.ocl..xyz": Du>:Bg3<vRFt4y;0#WE_
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:

```
[zdmuser@zdmhost ~]$ $ZDM_HOME/bin/zdmcli query job -jobid 34
zdmhost.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 -tgtarg1 user:opc -tgtarg2
identity_file:/home/zdmuser/.ssh/zdm.ppk -tgtarg3 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
```

```
ZDM_SWITCHOVER_APP ..... COMPLETED
ZDM_RM_GG_EXTRACT_SRC ..... COMPLETED
ZDM_RM_GG_REPLICAT_TGT ..... COMPLETED
ZDM_DELETE_SCHEMA_TRANDATA_SRC ..... COMPLETED
ZDM_RM_HEARTBEAT_SRC ..... COMPLETED
ZDM_RM_CHECKPOINT_TGT ..... COMPLETED
ZDM_RM_HEARTBEAT_TGT ..... COMPLETED
ZDM_CLEAN_GG_HUB ..... COMPLETED
ZDM_POST_ACTIONS ..... COMPLETED
ZDM_CLEANUP_SRC ..... COMPLETED
```

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:

- <https://docs.oracle.com/en/database/oracle/zero-downtime-migration/21.1/zdmrn/#GUID-1F4D8423-BCF5-4B97-972E-9C97C99F9482>

TROUBLESHOOTING & OTHER RESOURCES

For Oracle ZDM log review:

- 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
- OGG hub logs:
 - /u02/deployments/<ogg_deployment_name>/var/log

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com.
Outside North America, find your local office at oracle.com/contact.



blogs.oracle.com



facebook.com/oracle



twitter.com/oracle

Copyright © 2021, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

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

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0120

Oracle Zero Downtime Migration – Logical Migration and In-Flight Upgrade from On-Premises to DBCS and ExaCS Step-by-Step Guide
August, 2021

Authors: Ning Xu, Rex Zheng, Ricardo Gonzalez

Contributing Authors: Hiromi Sakuragi, Oracle ZDM Development Team, Oracle RACPack Team, Oracle MAA Team

