ORACLE

PeopleSoft Application with Autonomous Database Dedicated

Migration Guide with Oracle ZDM Logical Offline Workflow

November 2024 | Version 1.0

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 remain 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

Figure 0 – Oracle PeopleSoft + ZDM + ADB Logo

INTRODUCTION

Oracle customers are moving Oracle workloads into the Oracle Cloud at a growingly rapid pace. However, migrating workloads has been a source of challenges for many years. 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 depending on the migration scenario. ZDM allows Oracle customers to migrate their on-premises Oracle Databases directly and seamlessly to Oracle Database@Azure, Oracle Database@Google Cloud, Oracle Database@AWS and any Oracle-owned infrastructure, including Exadata Database Machine on-premises, Exadata Cloud at Customer (ExaDB-C@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 migration workflows that leverage technologies such as the Recovery Manager, Data Pump, and Database Links.

Oracle PeopleSoft customers migrating to the Oracle Cloud can benefit from ZDM and its automation, having a more accessible, automated Cloud Journey.

This Migration guide will walk you through all the requirements, steps, and best practices for Migrating your Database and having your PeopleSoft environment leverage Oracle Autonomous Database and ZDM's Logical Offline Migration Workflow.

For more information on Oracle Zero Downtime Migration, please visit ZDM's product website.[1]
For more information on Oracle PeopleSoft, please visit PeopleSoft's product website.[2]
For more information on Oracle Autonomous Database, please visit Oracle Autonomous Database's website.[3]

[1] Http://www.oracle.com/goto/zdm

[2] https://www.oracle.com/applications/peoplesoft/

[3] https://www.oracle.com/autonomous-database/

PEOPLESOFT APPLICATION WITH AUTONOMOUS DATABASE

Architecture

This step-by-step guide starts with a full-tier source PeopleSoft HCM environment deployed on an Oracle Linux VM. This guide aims to migrate the database to an **A**utonomous **D**atabase on **Dedicated** Exadata Infrastructure (now on ADB-D)– configured for Autonomous Transaction Processing workloads. At a high level, we will use Oracle Zero Downtime Migration (now on ZDM) during this procedure. This document is based upon ZDM's offline logical migration methodology for migrating the on-premises database to ADB-D and leveraging Oracle Data Pump. The migrated database at ADB-D can be rewired with the Mid-Tier of PeopleSoft provisioned at OCI Infrastructure.

- **Offline Migration with Data Pump and Backup Location**
  - ZDM logical offline migration with Data Pump and Backup Location offers customers a simple yet efficient method to migrate their databases to the Oracle Cloud.

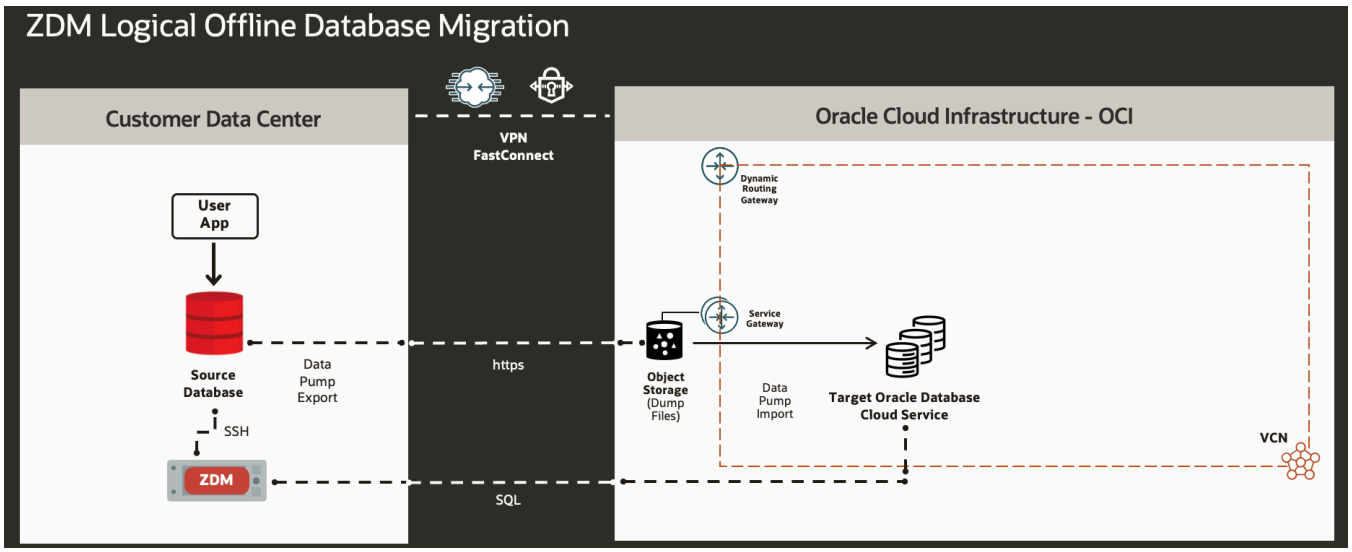

Figure 1 – Logical Offline Migration with ZDM, Architectural Diagram

Requirements for PeopleTools and Autonomous Database

- PeopleTools: For 8.57, it is 8.57.16 and above. For 8.58, it is 8.58.05 and above. For 8.59, 8.60 and 8.61, there is no minimum PeopleTools patch level required for ADB.[4]

- Database Client: The client must be updated by applying a Database Release Update (DBRU) patch on the middle tier to obtain the required Oracle Client levels. The minimum level required is 19.13 (October 2021), which can be found here:

  - Oracle Database 19c Release Update & Release Update Revision October 2021 Known Issues NOTE: 19202110.9
  - Supported Oracle Client version with TLS authentication without a wallet, based on: Oracle Client version supporting TLS authentication
    - Oracle Instant Client 19.13 - only on Linux x64
    - Oracle Instant Client 19.14 (or later) and 21.5 (or later) - all platforms
      - ** Selecting the latest version available during deployment is always recommended.*

[4] https://blogs.oracle.com/peoplesoft/post/now-supported%C2%A0-peoplesoft-applications-using-autonomous-database

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 7,8 (must be this OS version). For RHEL 8 or 9, please visit Oracle Zero Downtime Migration Documentation for specific requirements and installation instructions.
- 100 GB of free storage space. This space is required for all the logs that ZDM will generate.
- A `zdm` group and a `zdmuser` as part of this group.
- The following packages must be installed:
    - `glibc-devel`
    - `expect`
    - `unzip`
    - `libaio`
    - `oraclelinux-developer-release-el7`

- All hostnames and IP addresses to be used must be present as entries in the `/etc/hosts file`.

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*"[5] section.

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. For this guide a VM in OCI has been provisioned with an attached block volume of 100 GB.

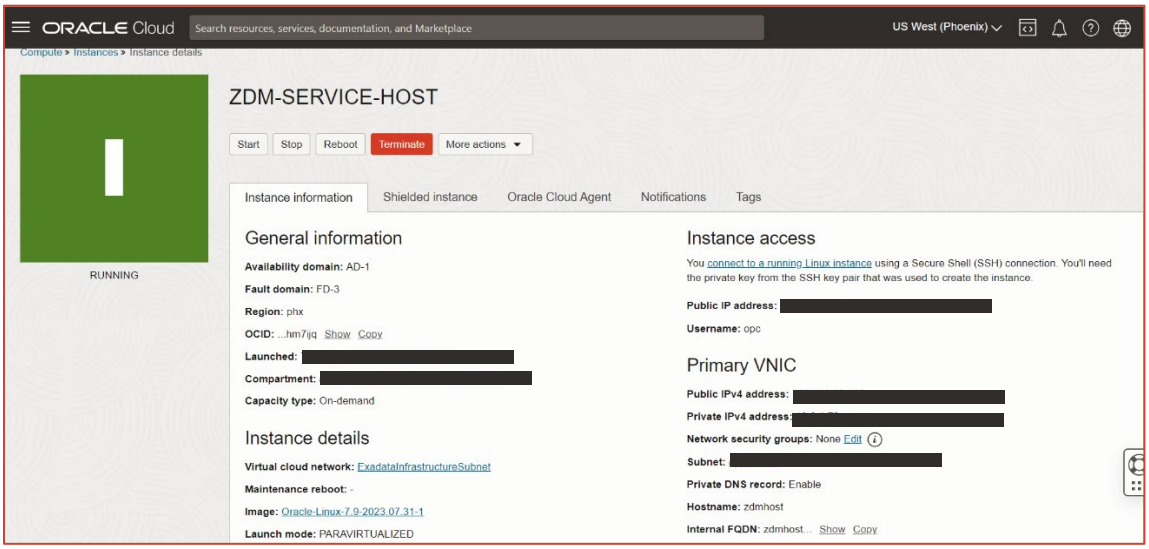

Figure 2 –ZDM Service-Host VM in Oracle Cloud Infrastructure

[5] https://docs.oracle.com/en/database/oracle/zero-downtime-migration/index.html

ZDM Service Host Installation

Log in to the ZDM Service Host via the terminal as root user:

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

```
[root@zdmhost]# groupadd zdm
[root@zdmhost]# useradd zdmuser -g zdm
[root@zdmhost]# mkdir -p /home/zdmuser/zdminstall
[root@zdmhost]# mkdir /home/zdmuser/zdmhome
[root@zdmhost]# mkdir /home/zdmuser/zdmbase
[root@zdmhost]# chown -R zdmuser:zdm /home/zdmuser/
```

2 Install the required software packages. As root user:

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

3 Download ZDM binaries to /home/zdmuser/zdminstall from www.oracle.com/database/technologies/rac/zdm-downloads.html. Change the owner of the zip file to zdmuser. As root user:

```
[[root@zdmhost]# cd /home/zdmuser/zdminstall
[root@zdmhost zdminstall]# chown zdmuser:zdm /home/zdmuser/zdminstall/zdm21.X.zip
```

4 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.x.zip
[zdmuser@zdmhost zdminstall]$ cd zdm21.x
-- Proceed to execute ZDM's installation script zdmuser:
[zdmuser@zdmhost zdm21.3]$ ./zdminstall.sh setup \
oraclehome=$ZDM_HOME \
oraclebase=$ZDM_BASE \
ziploc=./zdm_home.zip -zdm
```

5 Start ZDM and check the status. As zdmuser:

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

6 Install the OCL CLI on the ZDM Service host as 'root,' execute the following:

```
[root@zdmhost ]# yum install python36-oci-cli
```

Install SQL Client

Install the Oracle Database Client on the ZDM Service Host for establishing connectivity to the Source and Target Database.

Download the RPM packages of Oracle Client for using the installer available at Oracle Instant Client Downloads for Linux x86-64 (64-bit)

- Basic Package (RPM)
- SQL*Plus Package (RPM)
- Tools Package (RPM)

For this step-by-step guide, the version used was 19.20.

Install Client Packages

As a 'root' user, install the packages in the given order—First Basic, then SQL*Plus, and finally the Tools Package. Commands executed:

```
[root@zdmhost software]# yum install -y oracle-instantclient19.20-basic-19.20.0.0.0-1.x86_64.rpm
[root@zdmhost software]# yum install -y oracle-instantclient19.20-sqlplus-19.20.0.0.0-1.x86_64.rpm
[root@zdmhost software]# yum install -y oracle-instantclient19.20-tools-19.20.0.0.0-1.x86_64.rpm
```

Environment Variables

Update the environment variable for 'zdmuser' for TNS_ADMIN and PATH, as shown below.

```
TNS_ADMIN=$ORACLE_HOME/network/admin; export TNS_ADMIN
PATH=/usr/lib/oracle/19.20/client64/bin:$PATH; export PATH
echo "PATH=/usr/lib/oracle/19.20/client64/bin:$PATH;
export PATH" >> ~/.bashrc
echo "LD_LIBRARY_PATH=/usr/lib/oracle/19.20/client64/lib:$LD_LIBRARY_PATH;
export LD_LIBRARY_PATH" >> ~/.bashrc
```

PRE-REQUISITES

Network Connectivity

Please ensure the network connectivity is met as per the table below.

Initiator	Target	Protocol	Port	Purpose
ZDM Service Host	Source Database Server	TCP	22	SSH
ZDM Service Host	Source Database Server	TCP	1521	SQL*Net
ZDM Service Host	Target Database Server	TCP	1521	SQL*Net
Source Database Server	Oracle Cloud Object Store Service	HTTPS	443	Database backup store
Target Database Server	Oracle Cloud Object Store Service	HTTPS	443	Database backup store

SSH Connectivity

Configure SSH Connectivity from the ZDM host as 'zdmuser' to the Source Database host based on SSH Keys without a passphrase

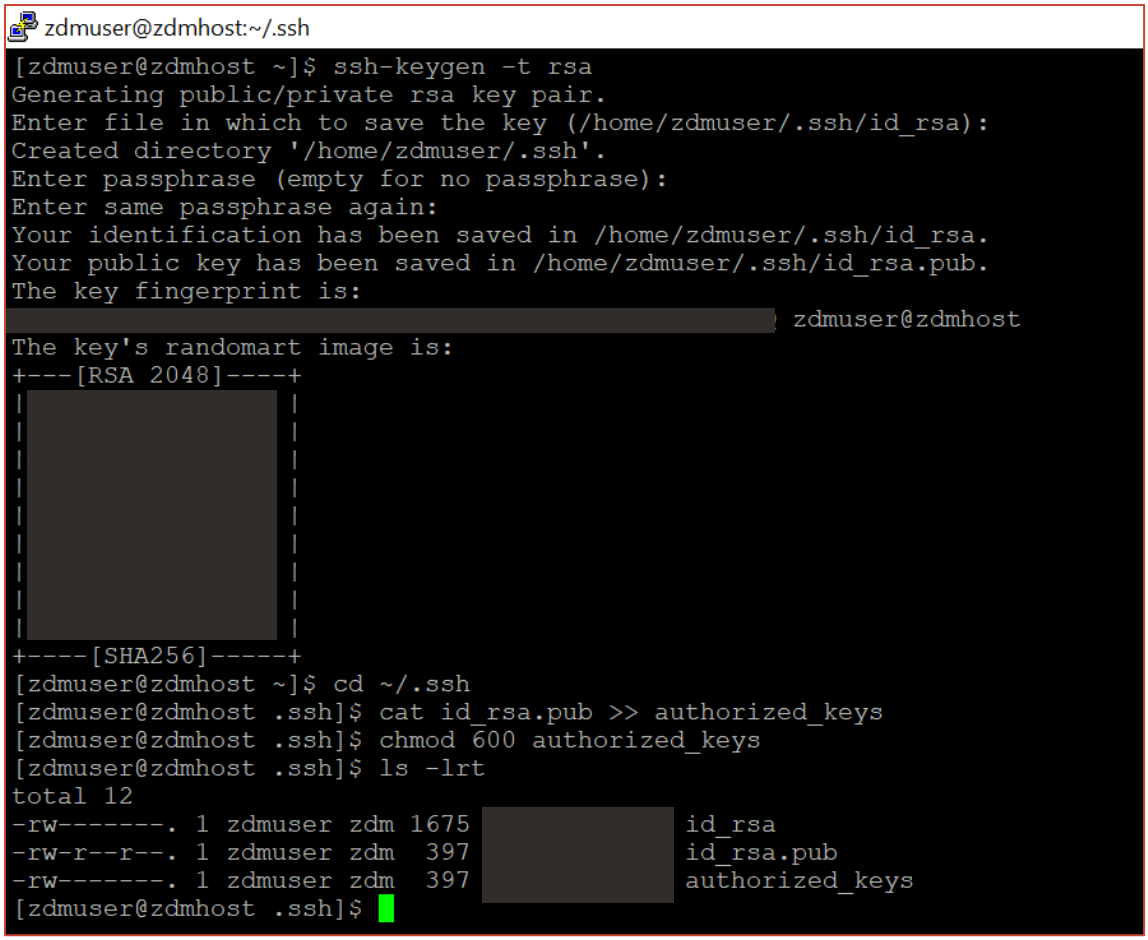

Figure 3 – Screenshot for ssh configuration.

```
ssh-keygen -t rsa
cd ~/.ssh
cat id_rsa.pub >> authorized_keys
chmod 600 authorized_keys
```

The next step is to update the authorized_keys file; this is achieved by adding the contents of the `/home/zdmuser/.ssh/id_rsa.pub` file into the `opc_user_home/.ssh/authorized_keys` file on the source database server. Finally, validate the SSH connectivity between the ZDM host and the PSFT host by executing `ssh opc@source_hostname` (in this case the source hostname is hr9246.)

API Signing Public Key and Configuration File

API Gateway Service is needed to create governed HTTP/S interfaces for other OCI Services; to set this up, please follow the instructions below:

1 Run as 'zdmuser' and copy the contents of the public key:

```
[zdmuser@zdmhost ]# mkdir zdmhome/.oci

[zdmuser@zdmhost ]# cd zdmhome/.oci

[zdmuser@zdmhost .oci]# openssl genrsa -out /home/zdmuser/zdmhome/.oci/oci_api_key.pem
2048
```

```
[zdmuser@zdmhost .oci]# openssl rsa -pubout -in
/home/zdmuser/zdmhome/.oci/oci_api_key.pem -out
/home/zdmuser/zdmhome/.oci/oci_api_key_public.pem

Writing RSA key
[zdmuser@zdmhost .oci]#cat oci api_key_public.pem
------BEGIN PUBLIC KEY--------
XXXXXXXXXXXX
------END PUBLIC KEY------
[zdmuser@zdmhost .oci]#
```

2 Go to the OCI Dashboard, navigate to the top right, click on your user profile icon, and select the top option representing your user. Select **API Keys** and **Add API Key**, copy the content of the saved public key from step 1 above:

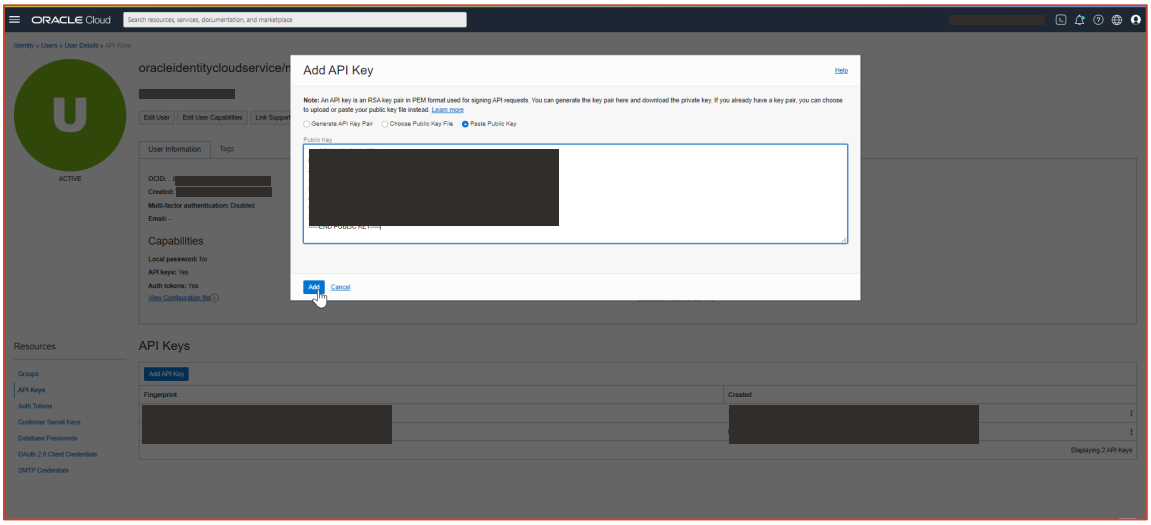

Figure 4 –Add API Key in Oracle Cloud Infrastructure

3 You will see a configuration file preview. Copy its contents, which you will be using to populate your configuration file later:

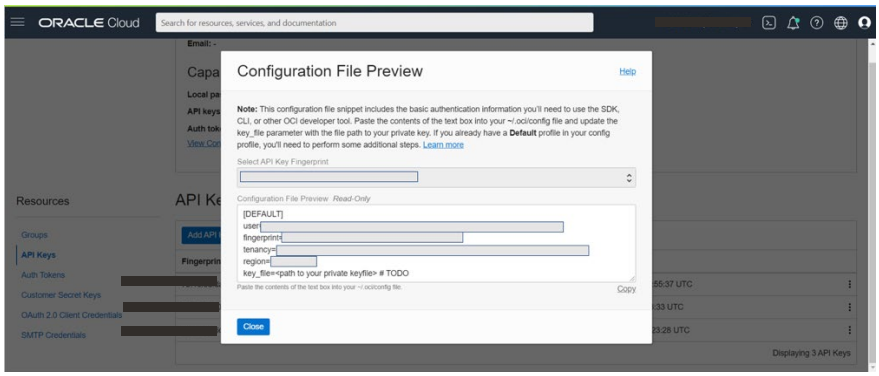

Figure 5 – Screenshot for "Configuration File Preview" window on Oracle Cloud.

4 As the zdmuser in the ZDM Service Host, create a configuration file in the command prompt; you can use vi/vim or any editor you prefer. In the empty file, paste the configuration file contents copied from above. Replace < path to your private keyfile > # TODO with the line above; once done, save the file and quit the editor:

```
/u01/app/zdmhome/.oci/oci_api_key.pem
```

Verify Virtual Cloud Network (VCN) Configuration

In your Oracle Cloud account, click on the top left *hamburger* menu and select the displayed "***Networking***" option. A new pane will appear; click "***Virtual Cloud Networks.***"

Follow these steps to access the Security List of your subnet:

- Click on the name of the Virtual Cloud Network (VCN) you will use for the migration. Your browser will refresh the view to show information relevant to the chosen VCN.
- Scroll down to Subnets and click on the subnet in your compartment used for the migration. Your browser will refresh the view to show information relevant to the chosen subnet.
- Scroll down to Security Lists and click on the default security list in your compartment. Your browser will refresh the view to show information relevant to the chosen security list.
- Scroll down to the Ingress Rules sections.

If there are no rules for **Port 443** and **Port 1521**, they must be added. Click the "Add Ingress Rules" button to add the rules.

An **Add Ingress Rules** pane will pop up. Enter the following parameters:

- **Stateless**: Left Unchecked
- **Source Type**: CIDR
- **Source CIDR**: *Enter valid subnet CIDR as per your environment requirements*
- **IP Protocol**: TCP
- **Source Port Range**: Left as is
- **Destination Port Range**: 443
- **Description**: OGG HTTPS

Click on **+ Another Ingress Rule**:

Enter the following parameters:

- **Stateless**: Left Unchecked
- **Source Type**: CIDR
- **Source CIDR**: *Enter valid subnet CIDR as per your environment requirements*
- **IP Protocol**: TCP
- **Source Port Range**: Left as is
- **Destination Port Range**: 1521
- **Description**: Oracle DB

Click on **Add Ingress Rules** to add the ingress rules for Ports **443** & **1521**.

Hosts file

Configure the /etc/hosts file at ZDM Service Host as a first step to ensure connectivity between the ZDM Service Host and the Source Database. As a root user on the ZDM Service Host, adding the Source Database information:

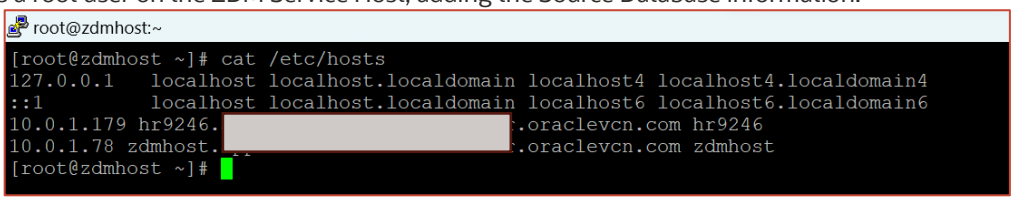

Figure 6 –ZDM Service Host /etc/hosts file

Source Database

The source database for this step-by-step guide is configured on Oracle Linux 7 VM as a PUM Database with HRMS 9.2 Image 46 on PeopleTools 8.60.05. The Source database runs with Oracle Database Version 19.18.0.0 and PSU Patch January 2023.

Property	Source Database
Hostname	hr9246.appsubnet.exadatainfrastr.oraclevcn.com
Operating System	OL 7.9
DB Version	19.18.0.0
Patch	19.18.0.0.230117
File System	Standard
CDB Name	CDBHCM
PDB Name	HR9246
NLS_CHARACTERSET	AL32UTF8
NLS_NCHAR_CHARACTERSET	UTF8

Source Database's stream pool must be configured with the initialization parameter STREAMS_POOL_SIZE:

```
SQL> alter system set streams_pool_size=512M scope=spfile sid='*';
SQL> shutdown immediate;
SQL> startup open
```

**Restart the Database upon setting the parameter*

Ensure the DATAPUMP_EXP_FULL_DATABASE role is assigned to the specified source database user.
At ADB-D, SELECT is no longer allowed on system objects. All SELECT grants on system objects must be replaced with READ grants. For additional details, please refer to Oracle Support Document ID 1911151.1. To Revoke 'SELECT' privileges from the source database, execute the following at the source database:

```
[oracle2@hr924 ~]$ export ORACLE_SID=CDBHCM
[oracle2@hr924 ~]$ sqlplus / as sysdba
SQL> alter session set container=hr9246;
SQL> REVOKE SELECT ON SYS.V_$IM_COLUMN_LEVEL from PSADMIN;
SQL> REVOKE SELECT ON SYS.V_$IM_USER_SEGMENTS from PSADMIN;
SQL> REVOKE SELECT ON SYS.V_$MYSTAT from PSADMIN;
SQL> REVOKE SELECT ON USER_AUDIT_POLICIES from PSADMIN;
SQL> REVOKE SELECT ON DBA_AUDIT_POLICY_COLUMNS from PSADMIN;
```

To Grant 'READ' privileges from the source database, execute the following at the source database:

```
SQL> GRANT READ ON SYS.V_$IM_COLUMN_LEVEL to PSADMIN;
SQL> GRANT READ ON SYS.V_$IM_USER_SEGMENTS to PSADMIN;
SQL> GRANT READ ON SYS.V_$MYSTAT to PSADMIN;
SQL> GRANT READ ON USER_AUDIT_POLICIES to PSADMIN;
SQL> GRANT READ ON DBA_AUDIT_POLICY_COLUMNS to PSADMIN;
```

Target Database

The Target Database will be an Autonomous Database on Dedicated Exadata Infrastructure for Online Transaction Processing, referred to as ADB-D in this document. This Database has been provisioned on OCI using the steps described below:

Database Provisioning

a. Change the compartment of your choice and create an Autonomous Database via the menu option of Autonomous Transaction Processing:

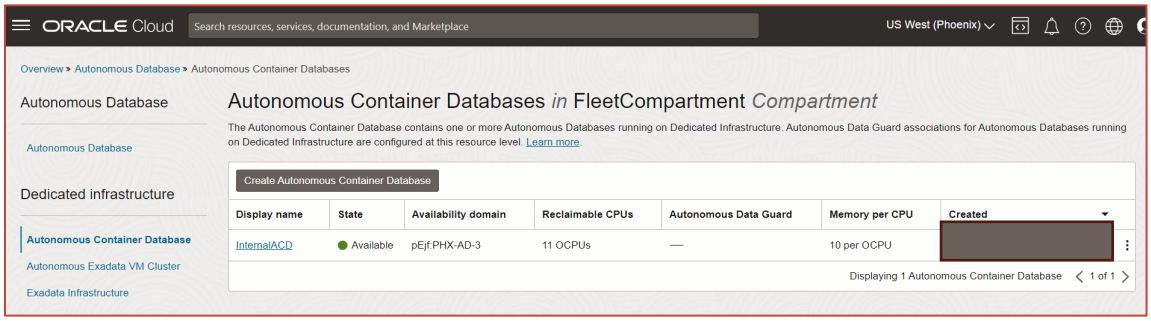

Figure 7 – Autonomous Database menu in Oracle Cloud Infrastructure

b. Database Options required for provisioning:

- **Compartment**: 'Compartment to be used for Workload'
- **Workload Type**: Transaction Processing
- **Deployment Type**: Dedicated Infrastructure
- **Choose Autonomous Container Database**: 'Compartment for Autonomous Container Database'
- **Network Access**: We can control and restrict access to ADB-D by specifying network access control lists (ACLs)
- **Character Set and National Character Set**: Character sets used by ADB-D.
- **PeopleSoft Application Tag**: To ensure the ADB instance is configured optimally for PeopleSoft workloads

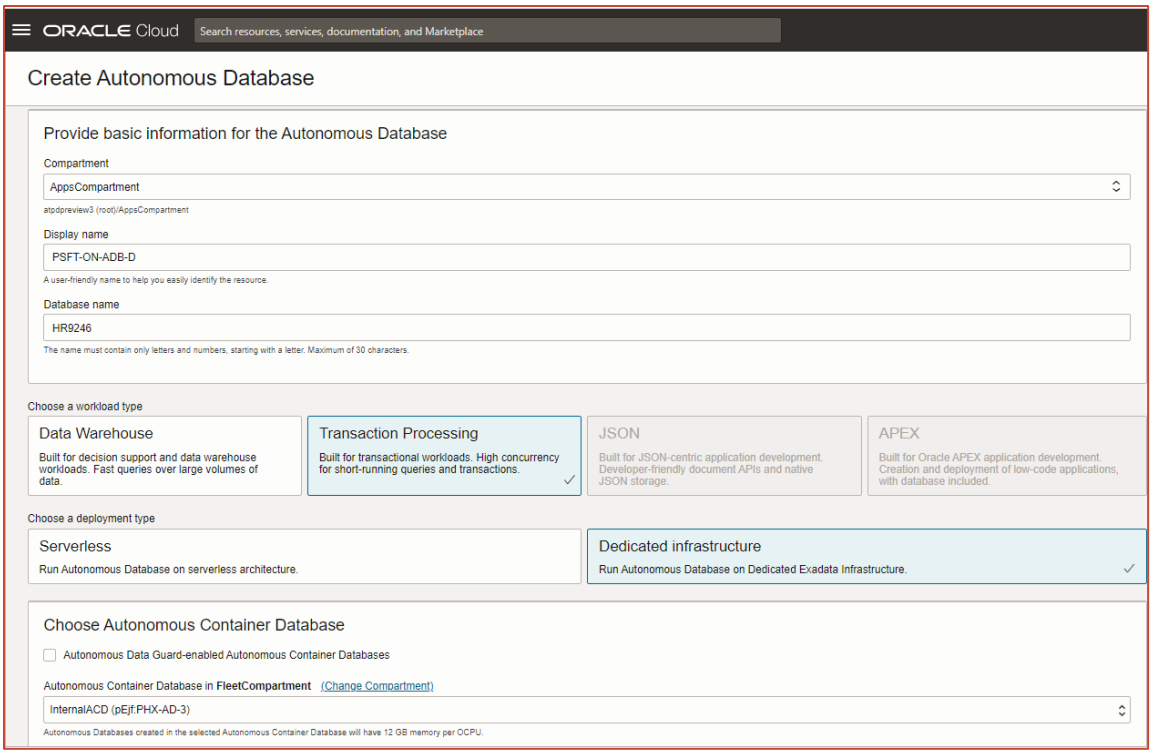

Figure 8 – Create an Autonomous Database menu in Oracle Cloud Infrastructure

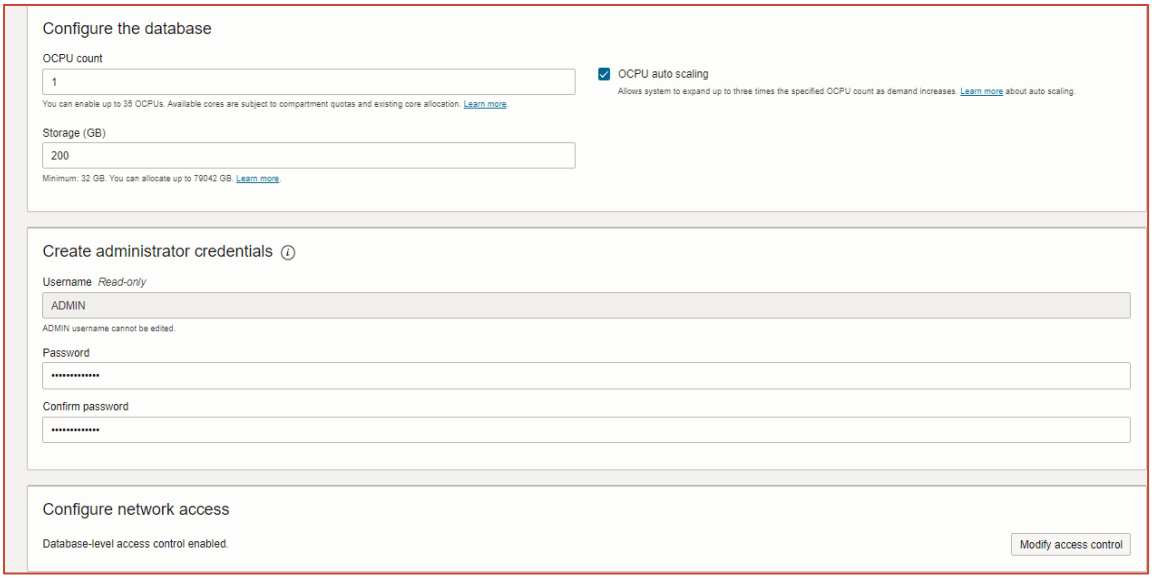

Figure 9 – Create an Autonomous Database menu in Oracle Cloud Infrastructure

Figure 10 – Create an Autonomous Database menu in Oracle Cloud Infrastructure

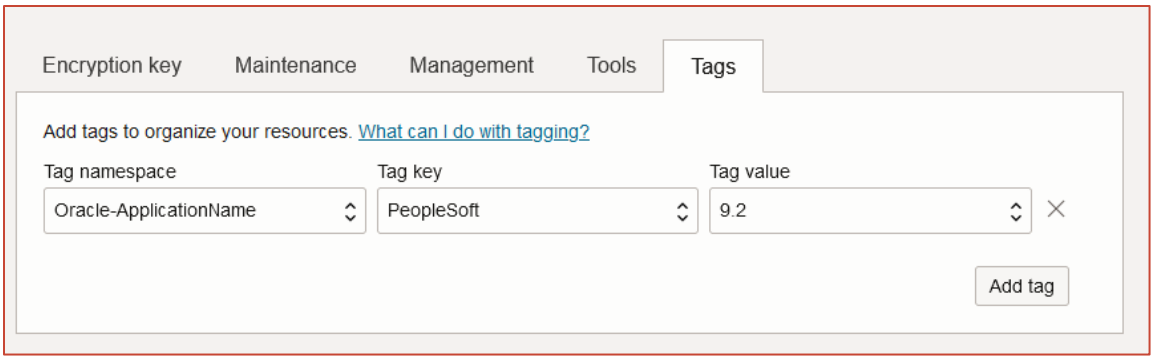

Figure 11 – Create an Autonomous Database menu in Oracle Cloud Infrastructure.

Be sure to set up the PeopleSoft tag as outlined above. Adding this tag is important, since it adds a key required performance setting to the PSFT environment.[6]

Validate the Pluggable Database (PDB) from OCI Console after it gets provisioned:

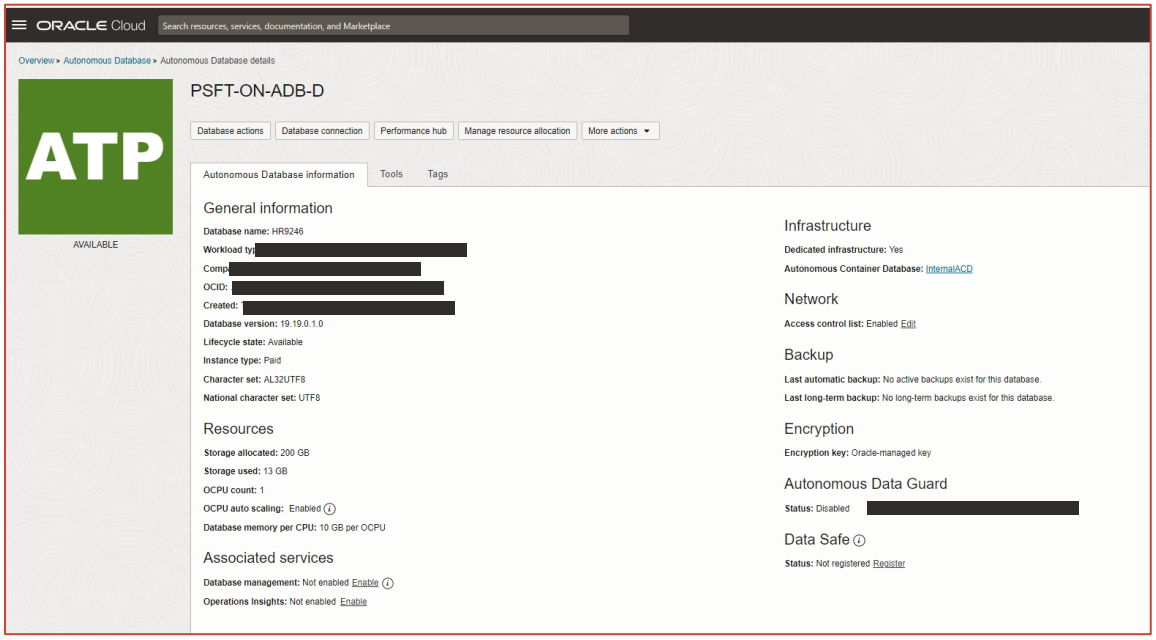

Figure 12 –Autonomous Database menu in Oracle Cloud Infrastructure

Summary of Source and Target Environments

Property	Source Database	Target Database
Hostname	hostname-lnfxt-database.test	-
Operating System	OL 7.9	-
DB Version	19.18.0.0	19.19.0.1.0
Patch	19.18.0.0.230117	-

[6] https://docs.oracle.com/en/cloud/paas/autonomous-database/dedicated/myyyc/index.html#articletitle

File System	Standard	-
CDB Name	CDBHCM	-
PDB Name	HR9246	HR9246
NLS_CHARACTERSET	AL32UTF8	AL32UTF8
NLS_NCHAR_CHARACTERSET	UTF8	UTF8

Architecture Changes with ADB-D

Database Account – ADMIN

ADMIN is the predefined administrative user in Oracle Autonomous Database on Dedicated Exadata Infrastructure. Due to ADB-D security controls and its ability to perform administrative database tasks autonomously, the ADMIN user does not have as many privileges as the SYS user. For details, please refer to the Autonomous Database Documentation here.[7]

Database Character Set

Autonomous Database on dedicated Exadata infrastructure has AL32UTF8 as the default database character set and AL16UTF16 as the default national character set. As part of ZDM's prerequisites, the character set on the source database must be the same as the target database. For further information on Character Set Migration, please refer to *Character Set Migration and Support Note 788156.1.*

For this step-by-step guide, the source database character set is Unicode AL32UTF8.

Database Time Zone

The Autonomous VM Cluster OS Timezone drives the default time zone for the Autonomous Database. For the current use case, the time zone will be Coordinated Universal Time (UTC), and by default, any calls to SYSDATE and SYSTIMESTAMP will return the date and time in UTC.

Password Policy for Database Users

An Autonomous Database requires strong passwords; the password user specified for a Database User must meet the following default password complexity rules:

- The password must be between 12 and 30 characters long and must include at least one uppercase letter, one lowercase letter, and one numeric character.
- The password cannot contain the username.
- The password cannot be one of the last four passwords used for the same username.
- The password cannot contain the double quote (") character.
- The password must differ from the one set less than 24 hours ago.

To change the password complexity rules and parameter values, you can alter the default profile or create a new one and assign it to users. For more information, see the link below.[8]

---

[7] https://docs.oracle.com/en/cloud/paas/autonomous-database/dedicated/adbdk/#GUID-798FB413-6160-4EEC-93D1-6D2B996046CE
[8] https://docs.oracle.com/en/cloud/paas/autonomous-database/dedicated/adbcu/#GUID-0E019845-31AE-44D7-B55C-9BCBA7E1377F

Remember that you can create a Password Verify Function (PVF) and associate the PVF with a profile to manage the complexity of user passwords. For details, please refer to the link below.[9]

Automatic Indexing

Automatic indexing automates the index management tasks in the Autonomous Database. It is turned off by default. For PeopleSoft, it is recommended that it rely upon application-provided indexes.

Optimizer Hints & Statistics

Autonomous Database honors optimizer hints and PARALLEL hints in SQL statements by default.
Autonomous Database gathers optimizer statistics automatically so that the user does not need to perform this task manually, which also helps to ensure database statistics are current.

Data Encryption

Autonomous Database uses always-on encryption that protects data at rest and in transit. All data stored in and network communication with Oracle Cloud is encrypted by default. Encryption cannot be turned off.

Certificate Management

Oracle Autonomous Database on Dedicated Exadata Infrastructure uses standard TLS 1.2 certificate-based authentication for client connections. Regardless of whether the client attempts to connect through a TCPS or TCP database connection service, the access the client has to the database is restricted by the access rights of the database user used by the client. By default, Autonomous Database uses self-signed certificates. However, you can install your CA-signed server-side certificate from the Oracle Cloud Infrastructure (OCI) console.

Database Features Not Supported

To ensure ADB-D's security and performance, ADB-D does not support some PeopleSoft-relevant Oracle Database features, options, and packs used with On-Premises Databases. Among them:

- Root container (CDB$ROOT) access.
- Clusters (groups of tables).
- Common users.
- Manual undo management.
- Dictionary-managed tablespaces.
- Manual segment space management.
- Transportable tablespaces.
- Logical standby databases.
- Database access: Users do not have direct access to the database node, local file system, SYSTEM, or SYSAUX tablespaces.

For more details, please refer to the link below. [10]

Database Features with Limited Support

In addition to the restrictions mentioned above, ADB-D comes with certain limitations required for security and performance in Autonomous Database on Dedicated Exadata Infrastructure.
For more details, please refer to the link below.[11]

---

[9] https://docs.oracle.com/en/cloud/paas/autonomous-database/dedicated/mudad/#GUID-81E6B578-C942-4755-A693-33773350B0DA

[10] https://docs.oracle.com/en/cloud/paas/autonomous-database/dedicated/adbdg/#articletitle

[11] https://docs.oracle.com/en/cloud/paas/autonomous-database/dedicated/adbdl/#articletitle

Target Database Required Settings

The Target Database needs to be prepared accordingly before starting the migration process. Please follow the steps as described below.

Target Database Parameters

PeopleSoft Unicode databases require NLS_LENGTH_SEMANTICS=CHAR. Update the NLS_LENGTH_SEMANTICS parameter as recommended for the PeopleSoft Database following these My Oracle Support notes:

- https://support.oracle.com/epmos/faces/DocContentDisplay?id=1986664.1
- https://support.oracle.com/epmos/faces/DocContentDisplay?id=2626966.1

```
SQL> show parameter nls_length_semantics

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
nls_length_semantics                 string      BYTE
SQL> alter system set nls_length_semantics=CHAR;

System altered.

SQL>
```

Figure 13 –Target Autonomous Database nls_length_semantics parameter update

```
SQL> alter system set nls_length_semantics=CHAR;
```

* This is to ensure that the length of CHAR and VARCHARs is measured in characters and not bytes to address the multibyte characteristics of AL32UTF8. For more details, please refer to Oracle Documentation for NLS_LENGTH_SEMANTICS

Create Tablespace, Roles, and PeopleSoft Users

Since PeopleSoft Users for Database, i.e., PS, PEOPLE, and SYSADM, have interdependencies, building users and its prerequisites, such as Default Tablespace and Roles/Privileges following PeopleSoft delivered scripts is recommended. Those scripts need an update to fit in the use case for ADB-D. For example:

- Instead of the SYSTEM user, ADB-D will have an ADMIN user.
- There is no need to switch Pluggable Databases as ADB-D only connects to Pluggable Databases by default.
- Creation of Public Synonyms.
- Running the Data Dictionary Scripts.

PeopleSoft scripts are located in the Source Environment, and they are available in $PS_HOME/scripts/unix/pdb folder as shown below:

```
psadm2@hr9246:/u01/app/oracle/product/pt/ps_home/scripts/unix/pdb
[opc@hr9246 ~]$ sudo su - psadm2
Last login: Fri Sep 15 06:21:28 GMT 2023 on pts/0
[psadm2@hr9246 ~]$ cd $PS_HOME/scripts/unix/pdb
[psadm2@hr9246 pdb]$ ls -lrt
total 56
-rwxr-xr-x. 1 psadm1 oinstall 1484 Apr 25  2022 ptperlcat.sh
-rwxr-xr-x. 1 psadm1 oinstall 2509 Apr  4 09:15 utlspace.sql
-rwxr-xr-x. 1 psadm1 oinstall 1617 Apr  4 09:15 upggrant.sql
-rwxr-xr-x. 1 psadm1 oinstall 3318 Apr  4 09:15 ptddlupg.sql
-rwxr-xr-x. 1 psadm1 oinstall 8084 Apr  4 09:15 ptddl.sql
-rwxr-xr-x. 1 psadm1 oinstall 3613 Apr  4 09:15 psroles2.sql
-rwxr-xr-x. 1 psadm1 oinstall 2083 Apr  4 09:15 psroles.sql
-rwxr-xr-x. 1 psadm1 oinstall 3278 Apr  4 09:15 psadmin.sql
-rwxr-xr-x. 1 psadm1 oinstall 2118 Apr  4 09:15 dbowner.sql
-rwxr-xr-x. 1 psadm1 oinstall 1929 Apr  4 09:15 createpdb.sql
-rwxr-xr-x. 1 psadm1 oinstall 3254 Apr  4 09:15 createdbcdb.sql
-rwxr-xr-x. 1 psadm1 oinstall 3449 Apr  4 09:15 createdb18cdb.sql
-rwxr-xr-x. 1 psadm1 oinstall 2011 Apr  4 09:15 connect.sql
[psadm2@hr9246 pdb]$
```

Figure 14 –PeopleSoft Scripts

Script Editing

Create a copy of the files listed below and execute them against the target ADB-D Database. This can be run from the Source Database or the ZDM Service Host. These scripts have been run from the ZDM Service Host for this guide.

utlspace.sql: This script builds the default tablespace for PeopleSoft Users. Below are the updates required before executing it against ADB-D:

- Comment the following lines:
  - ALTER SESSION SET CONTAINER = <PDB_SERVICE_NAME>: Connect with PDB as it is not required for ADB-D.
  - @$ORACLE_HOME/rdbms/admin/catblock.sql: Creates views that can dynamically display lock dependency graphs

- Update the following lines:
  - CREATE TEMPORARY TABLESPACE PSTEMP: Update file location as per Database File Path
  - CREATE TABLESPACE PSDEFAULT: Update file location as per Database File Path

To get the file location, execute the following commands:

```
SQL> set linesize 200
SQL> col name format a120
SQL> select FILE# , NAME from  v$datafile;
SQL> select FILE# , NAME from  v$tempfile;
```

dbowner.sql: This script creates PeopleSoft's PSDBOWNER Owner ID. Below are the updates required before executing it against ADB-D:

- Comment the following lines:
  - CREATE PUBLIC SYNONYM PRODUCT_PROFILE: operation not allowed from within a pluggable database
  - CREATE PUBLIC SYNONYM PRODUCT_USER_PROFILE: operation not allowed from within a pluggable database
  - @$ORACLE_HOME/sqlplus/admin/pupbld.sql: PUBBLD stands for "Product User Profile Build." This is not required for ADB-D.

- Update the following line:
  - CONNECT system/&SYSTEMPWD@<PDB_SERVICE_NAME>: Update the user to 'ADMIN' instead of 'SYSTEM' along with the correct PDB Service Name.

connect.sql: This script sets up the PeopleSoft Connect ID. No update is required for this script.

psroles.sql: This script provides the privileges required to run the PeopleSoft Application. One update is necessary before executing it against ADB-D.

- Comment the following lines:
  - ALTER SESSION SET CONTAINER = <PDB_SERVICE_NAME>: Connect with PDB as it is not required for ADB-D.

psroles2.sql: This script provides additional privileges required for the PSADMIN role. Below are the updates required before executing it against ADB-D:

- Comment the following line:
  - ALTER SESSION SET CONTAINER = <PDB_SERVICE_NAME>: Connect with PDB as it is not required for ADB-D.
- Update the following lines:
  - GRANT SELECT ON SYS.V_$MYSTAT to PSADMIN: Update SELECT to READ permission for ADB-D
  - GRANT SELECT ON USER_AUDIT_POLICIES to PSADMIN: Update SELECT to READ permission for ADB-D

- GRANT SELECT ON DBA_AUDIT_POLICY_COLUMNS to PSADMIN: Update SELECT to READ permission for ADB-D
- GRANT EXECUTE ON DBMS_FGA to PSADMIN: Update SELECT to READ permission for ADB-D

upggrant.sql: This script grants additional privileges required for the existing PSADMIN role. No update is needed for this script.

psadmin.sql: This script creates PeopleSoft's PSDBOWNER Owner ID. Below are the updates required before executing it against ADB-D:

- Comment the following lines:
  - ALTER SESSION SET CONTAINER = <PDB_SERVICE_NAME>: Connect with PDB as it is not required for ADB-D.
  - @$ORACLE_HOME/rdbms/admin/catdbsyn: Script for catalog dba synonyms. This is not required for ADB-D.
  - @$ORACLE_HOME/sqlplus/admin/pupbld: Script for Product User Profile BuiLD. This is not required for ADB-D.
  - @$ORACLE_HOME/rdbms/admin/utlxmv: creates a table named mv_capabilities_table that is required by the dbms_mview.explain_mview procedure. This is not necessary for ADB-D.
- Update the following line:
  - CONNECT system/&SYSTEMPWD@<PDB_SERVICE_NAME>: Update the user to 'ADMIN' instead of 'SYSTEM' along with the correct PDB Service Name.

Once the scripts have been updated accordingly, they **must** be run against the ADB-D in the following order:

- utlspace.sql
- dbowner.sql
- connect.sql
- psroles.sql
- psroles2.sql
- upggrant.sql
- psadmin.sql

SQL*Net Connectivity

PeopleSoft Applications and tools connect to ADB-D using Oracle Net Services (SQL*Net). Oracle Net Services enables a network session from the client application to an Oracle Database server through the network defined by the dedicated infrastructure hosting the database. SQL*NET supports various connection types to the Autonomous Database, including Oracle Call Interface (OCI). Wallet files, Database user ID, and password provide access to data in the Autonomous Database. Users should store wallet files in a secure location.

Ports and TLS/mTLS

While provisioning an Autonomous Exadata VM Cluster (AVMC) resource, the user can:

- Customize the Single Client Access Name (SCAN) listener port for Transport Layer Security (TLS) and non-TLS from a range of available ports (1024 - 8999). Users can also choose mutual TLS (mTLS) authentication by selecting the **Enable mutual TLS (mTLS) authentication** checkbox.

- Choose between one-way TLS and mutual TLS (mTLS) authentication modes. This applies only to Database TLS certificates because ORDS certificates are one-way TLS certificates.

Database Wallet

Certification authentication uses an encrypted key stored in a wallet on both the client (where the application is running) and the server (where ADB-D is running) to provide a secure connection. The wallet must be downloaded from the OCI ADB-D console, as shown below. Once downloaded, it must be transferred to the client host, where the ZDM Service Host and the PeopleSoft Application Server Host are running.

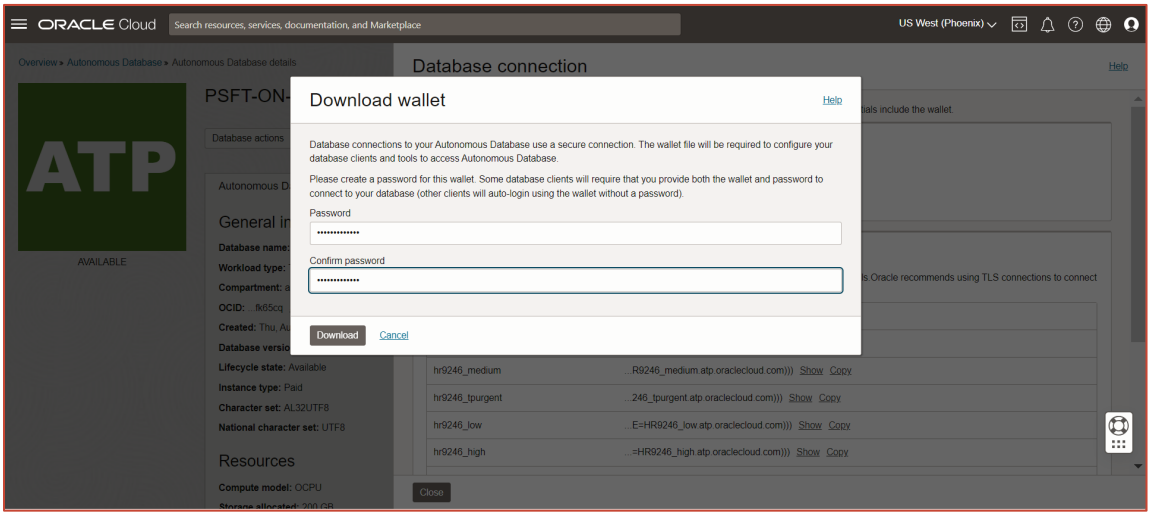

Figure 15 –Wallet

Extract Wallet and Update TNSNAME.ORA and SQLNET.ORA Files

Once the wallet is transferred to the Client Host, extract the zip file as downloaded from the OCI Console, then proceed to update the files as listed below:

```
[zdmuser@zdmhost ~]$ echo $TNS_ADMIN
[zdmuser@zdmhost ~]$ cd $TNS_ADMIN
[zdmuser@zdmhost admin]$ ls -lrt
[zdmuser@zdmhost admin]$ cp /tmp/Wallet_hostname.zip
[zdmuser@zdmhost admin]$ unzip Wallet_hostname.zip
```

Using the Source and the Target Database details, update the tnsnames.ora file, available under the $TNS_ADMIN folder. Refer to the connection string for ADB-D using your environment-specific details; the examples below are specific to this step-by-step guide.

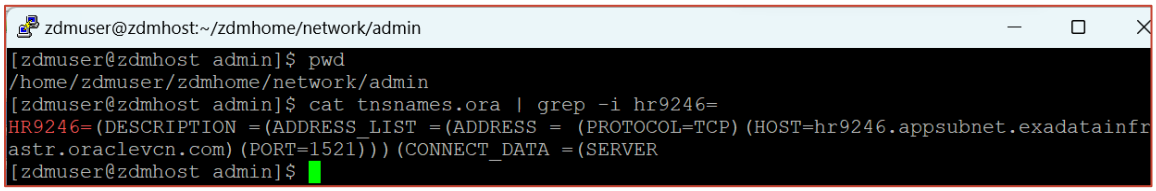

Figure 16 –Tnsnames.ora file on ZDM Server Host update for the Source Database.

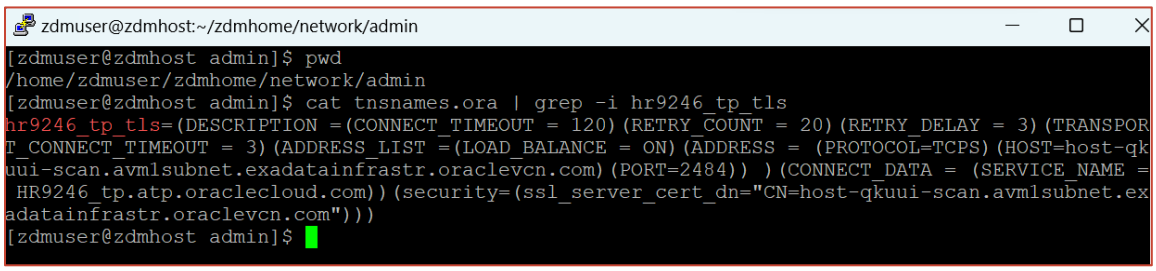

Figure 17 – Tnsnames.ora file on ZDM Server Host update for the Target Database.

Using the Target Database details, update the sqlnet.ora file, available under the $TNS_ADMIN folder. Refer to the folder location for ADB-D wallet files using your environment-specific details; the example below is specific to this step-by-step guide.

```
zdmuser@zdmhost:~/zdmhome/network/admin
[zdmuser@zdmhost admin]$ pwd
/home/zdmuser/zdmhome/network/admin
[zdmuser@zdmhost admin]$ cat sqlnet.ora
SQLNET.IGNORE_ANO_ENCRYPTION_FOR_TCPS=TRUE
SSL_SERVER_DN_MATCH=ON
SQLNET.EXPIRE_TIME = 10
SQLNET.WALLET_OVERRIDE = FALSE
WALLET_LOCATION = (SOURCE=(METHOD=FILE)(METHOD_DATA=(DIRECTORY="/home/zdmuser/zdmhome/network/admin")))
SSL_VERSION = 1.2
[zdmuser@zdmhost admin]$
```

Figure 18 – sqlnet.ora file on the ZDM Server Host update for the Target Database wallet files.

Test Database Connectivity

Connect to both Source and Target Database and validate the SQL connection from the ZDM Server Host:

- `[zdmuser@zdmhost]$ sqlplus SYSADM@HR9246`
  - Enter the password and verify connectivity, then proceed to exit.
- `[zdmuser@zdmhost]$ sqlplus SYSADM@hr9246_tp_tls`
  - Enter the password and verify connectivity, then proceed to exit.

Database Backup Location

Create an Object Storage Bucket for Database Backup: Create a standard bucket, named ZDMBucket, for storage of Database Backup.

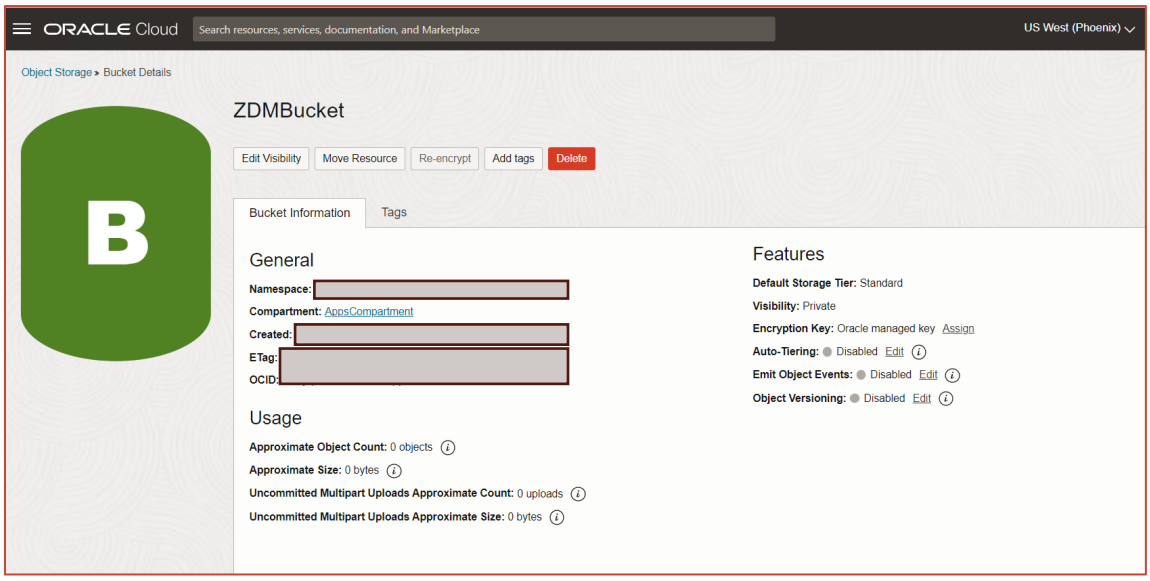

Figure 19 – Object Storage Bucket

Shutdown PeopleSoft Application Gracefully

Before executing any database migration activity, as a best practice, proceed gracefully to shut down the PeopleSoft Application Domain, including the Web Server, the Elastic Search Domain, etc. You may lock the environment and take additional precautions. To shut down the application gracefully, please execute as shown below:

```
[psadm2@hr9246 ~]$ psadmin stop -d *all;
```

MIGRATING TO AUTONOMOUS DATABASE

Preparing the Response File

Oracle Zero Downtime Migration leverages a response file that is fully customizable by the customer. A wide array of parameters for the logical migration methodology allows the customer to configure the migration according to the appropriate use case. 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**[12].

A response file template has been provided for each installation. As a 'zdmuser,' copy the template file to update parameters based on the environment:

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

The template contains parameters to handle all supported methodologies. For this step-by-step guide, **Offline Logical** migration methodology was selected; please proceed to update the response file based on this.

ZDM will migrate users, their Roles, Privileges, and the Tablespace. You must update the response file with all the parameters relevant to your environment. For this step-by-step guide, this is the response file used:

```
MIGRATION_METHOD=OFFLINE_LOGICAL
DATA_TRANSFER_MEDIUM=OSS
TARGETDATABASE_ADMINUSERNAME=ADMIN
SOURCEDATABASE_ADMINUSERNAME=SYSTEM
SOURCEDATABASE_CONNECTIONDETAILS_HOST=hr9246.xxxxx.yyyyyy.oraclevcn.com
SOURCEDATABASE_CONNECTIONDETAILS_PORT=1521
SOURCEDATABASE_CONNECTIONDETAILS_SERVICENAME=HR9246
TARGETDATABASE_OCID=ocid1.autonomousdatabase.oc1.zzz.yyyyyyyyyy
TARGETDATABASE_CONNECTIONDETAILS_HOST=host-xyxy-scan.yyyysubnet.aaaaaaa.oraclevcn.com
TARGETDATABASE_CONNECTIONDETAILS_PORT=2484
TARGETDATABASE_CONNECTIONDETAILS_SERVICENAME=hr9246_tp_tls
DATAPUMPSETTINGS_JOBMODE=SCHEMA
DATAPUMPSETTINGS_DELETEDUMPSINOSS=FALSE
DATAPUMPSETTINGS_DATABUCKET_NAMESPACENAME=namespace
DATAPUMPSETTINGS_DATABUCKET_BUCKETNAME=ZDMBucket
DATAPUMPSETTINGS_EXPORTDIRECTORYOBJECT_NAME=DATA_PUMP_DIR
DATAPUMPSETTINGS_EXPORTDIRECTORYOBJECT_PATH=/u01/app/oracle/product/db/oracle-
server/19.3.0.0/rdbms/log
OCIAUTHENTICATIONDETAILS_REGIONID=us-phoenix-1
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_TENANTID=ocid1.tenancy.oc1.aaaxxxyyyy
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_USERID=ocid1.user.oc1.aaaabbbbbbccccccc
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_FINGERPRINT=xx:11:22:33
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_PRIVATEKEYFILE=/home/zdmuser/zdmhome/.oci/oci_api_key.
pem
```

Performing a Test Database Migration in 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 that no issues will be encountered. 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. ZDM also provides a tool (Cloud Premigration Advisor Tool, CPAT) that performs analysis of the source database, looking for uses of database features and constructs that are problematic when migrating to one of Oracle's Autonomous Cloud offerings before you run it against the production database.

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

To start the evaluation of the source database, do as follows:

Run the EVAL Job

Run the job as 'zdmuser,' which needs the credentials for the Source and Target databases. ZDM will then request the different required passwords and generate a job id. The generated job id can be queried for progress using the `zdmcli query job -jobid job_id` command.

```
[zdmuser@zdmhost ~]$ $ZDM_HOME/bin/zdmcli migrate database -rsp
/home/zdmuser/template/zdm_logical_offline_pdb19c.rsp -sourcenode hostname-lnfxt-database.test -
sourcesid CM92PUM -srcauth zdmauth -srcarg1 user:opc -srcarg2 identity_file:/home/zdmuser/.ssh/id_rsa -
srcarg3 sudo_location:/usr/bin/sudo -eval
```

Monitor the Job

Use the provided Job ID to find the status of the job. You can do this by querying the ZDM server using the `zdmcli query job -jobid job_id` command.

```
[zdmuser@zdmhost ~]$ $ZDM_HOME/bin/zdmcli query job -jobid 1
```

Proceed to review the log file mentioned under "Result file path." This log file contains any warnings or showstoppers for the migration. Each check successfully executed by the migration advisor tool (CPAT) will result in **PASS**, **INFORMATIONAL**, **WARNING**, or **BLOCKER**.

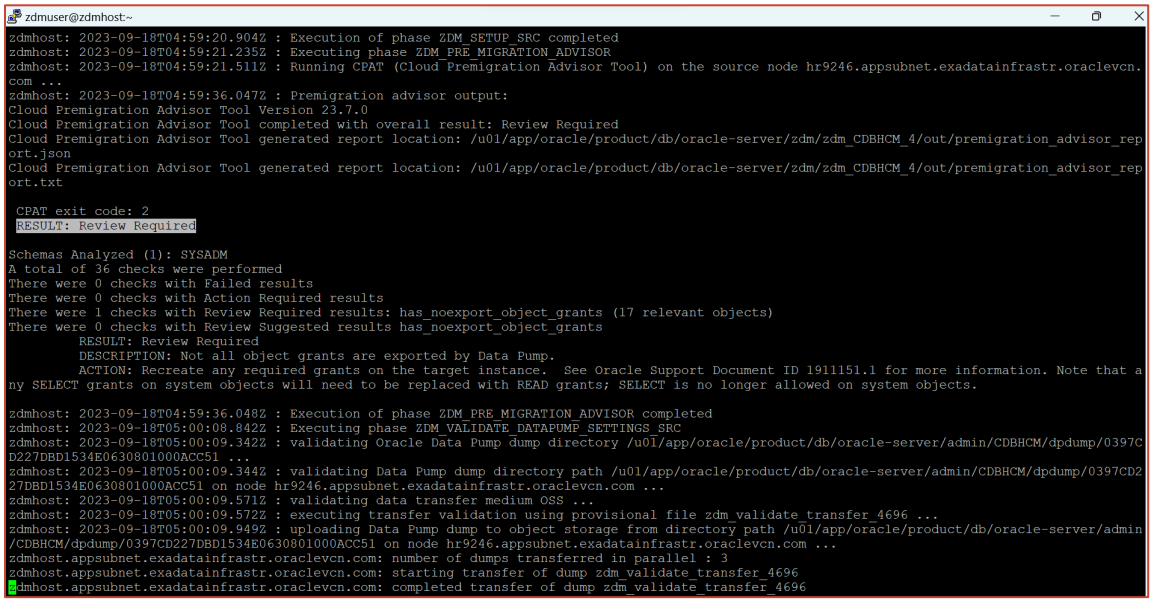

Figure 20 – Screenshot of a CPAT report

For more information on the Cloud Pre-migration Advisor Tool, please visit My Oracle Support and review Doc ID 2758371.1
https://support.oracle.com/rs?type=doc&id=2758371.1

Performing a Database Migration

Run the Migration Job

As the 'zdmuser,' submit the migration job as described below:

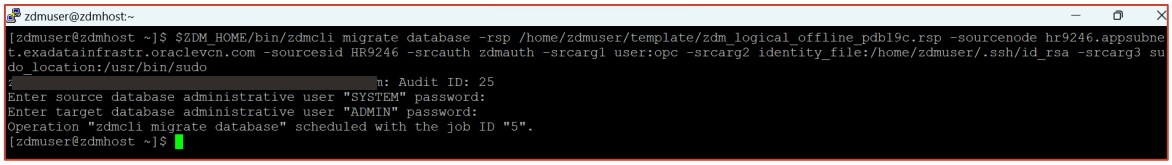

Figure 21 – Screenshot of a ZDM Migration job

```
[zdmuser@zdmhost ~]$ $ZDM_HOME/bin/zdmcli migrate database -rsp
/home/zdmuser/template/zdm_logical_offline_pdb19c.rsp -sourcenode
hr9246.appsubnet.exadatainfrastr.oraclevcn.com -sourcesid HR9246 -srcauth zdmauth -
srcarg1 user:opc -srcarg2 identity_file:/home/zdmuser/.ssh/id_rsa -srcarg3
sudo_location:/usr/bin/sudo
```

Check the Migration Job Status

After submitting the migration job, ZDM will return a JOB ID, which helps track the job status using the zmdcli query job command.

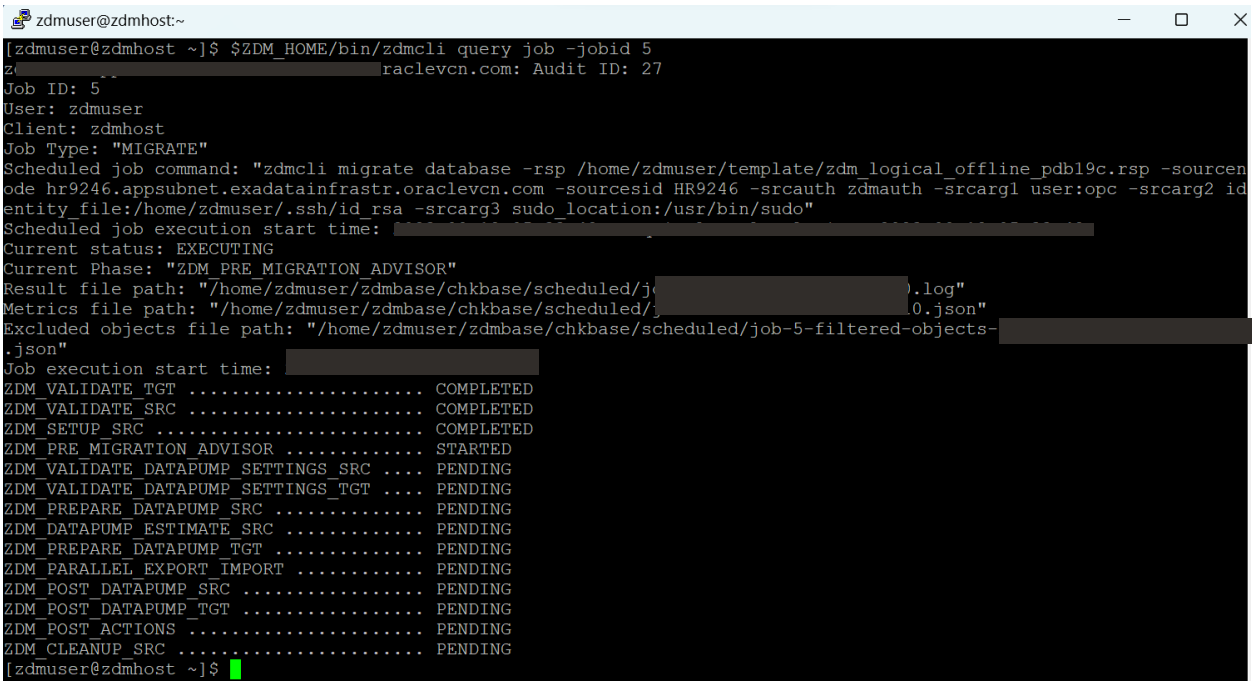

Figure 22 – Screenshot of a ZDM Migration job

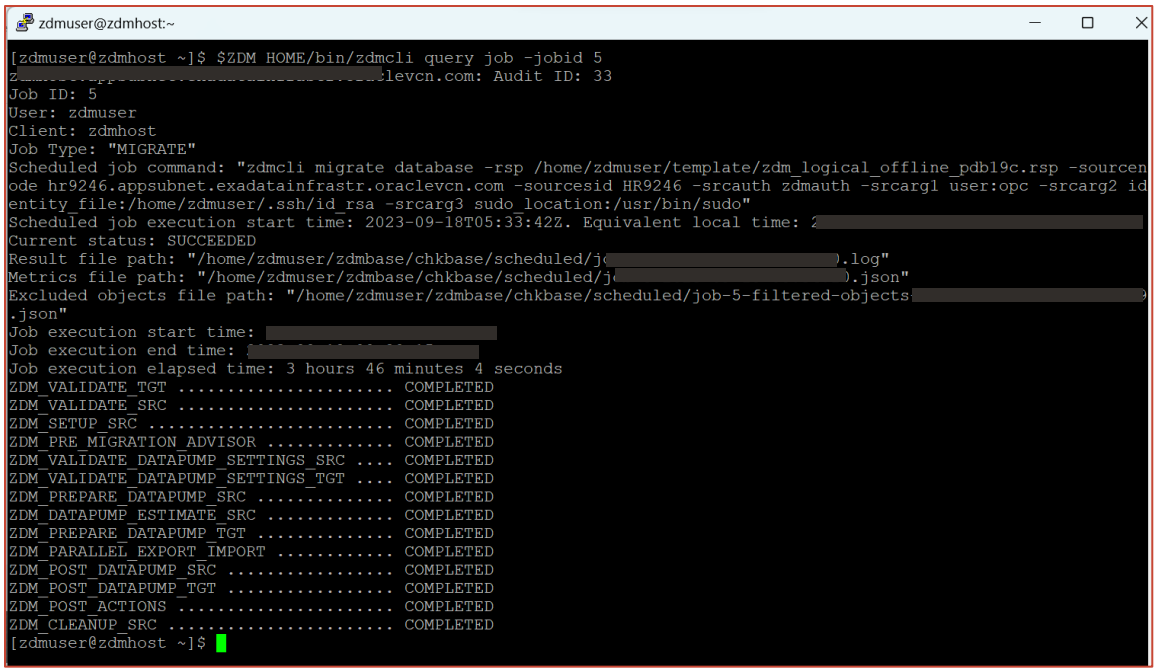

Figure 23 – Screenshot of a ZDM Migration job

```
[zdmuser@zdmhost ~]$ $ZDM_HOME/bin/zdmcli query job -jobid 5
```

POST MIGRATION DATABASE ACTIVITIES

After completing the ZDM Migration Job successfully, please follow these steps as part of the required post-migration activities. These steps are unique for the migration described in this step-by-step guide, where a PeopleSoft environment is present.

Validate PSDBOWNER Table

Validate the PSDBOWNER Table for DB Name: If there is a change in DB Name from source to target, updating the PSDBOWNER table is required.

```
[zdmuser@zdmhost ~]$sqlplus admin@hr9246_tp_tls
SQL> SELECT * FROM PS.PSDBOWNER;
SQL> INSERT INTO PS.PSDBOWNER VALUES ('HR9246', 'SYSADM');
col DBNAME format a30
col OWNERID format a20
SELECT * FROM PS.PSDBOWNER;
```

Validation of PeopleSoft Schema Objects

Validate the object count of PeopleSoft Schemas by running the object count at Source and Target.

Source Database - Objects Count

```
[oracle2@hr9246  ~]$export ORACLE_SID=CDBHCM
[oracle2@hr9246  ~]$sqlplus / as sysdba
alter session set container=hr9246;
col OWNER format a20
col OBJECT_TYPE format a40
SELECT OWNER, OBJECT_TYPE, COUNT(*)
FROM ALL_OBJECTS
WHERE OWNER IN ('PS','PEOPLE','SYSADM') GROUP BY  OWNER, OBJECT_TYPE ORDER BY 1,2;
```

Target Database - Object Count

```
[zdmuser@zdmhost ~]$]$sqlplus admin@hr9246_tp_tls
col OWNER format a20
col OBJECT_TYPE format a40
SELECT OWNER, OBJECT_TYPE, COUNT(*)
FROM ALL_OBJECTS
WHERE OWNER IN ('PS','PEOPLE','SYSADM') GROUP BY  OWNER, OBJECT_TYPE ORDER BY 1,2;
```

MID-TIER CONFIGURATION AT OCI

Mid-Tier Instance at Oracle Cloud Infrastructure

There are multiple ways to migrate the mid-tier to OCI:

- Using a tar ball backup
- Provisioning a new mid-tier using PUM Images
- PeopleSoft Cloud Manager

The existing Mid-Tier is being re-wired with ADB-D DataBase on OCI for this step-by-step guide.

Update TNS Entry and Test Database Connectivity

Validate the tnsnames.ora file as psadm2 user and test the connectivity. For the tnsnames.ora file, copy the connect string of service name <db_name>_tpurgent and make another service name with eight characters or shorter service name; this is a PeopleSoft App Server requirement. For instance, the service name 'HR9246' will be utilized to re-wire Mid-Tier with the database.

Configure PeopleSoft Server

Update Password for SYSADM User

PeopleSoft Users, PS, PEOPLE, and SYSADM have been created at ADB-D. Since ATP-D has a restricted policy for user passwords, The password must be updated in the PeopleSoft Application. The following MOS Note explains the process for changing the password:

- MOS: E-SEC: How To Change The Access ID (SYSADM) Password? (Doc ID 609603.1)
- MOS: E-ORA: How to change the PS database user password? (Doc ID 2398975.1)

To update the password for SYSADM, run the update statement for PSACCESSPROFILE with SYSADM Password and then encrypt the using Data Mover utility as shown below:

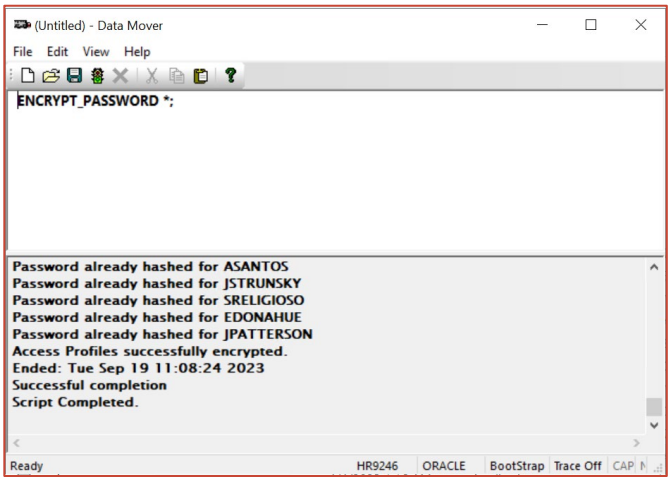

Figure 24 – Screenshot of Password Encryption

Configure App Server

Configure the Application Server Domain with the new password and start it:

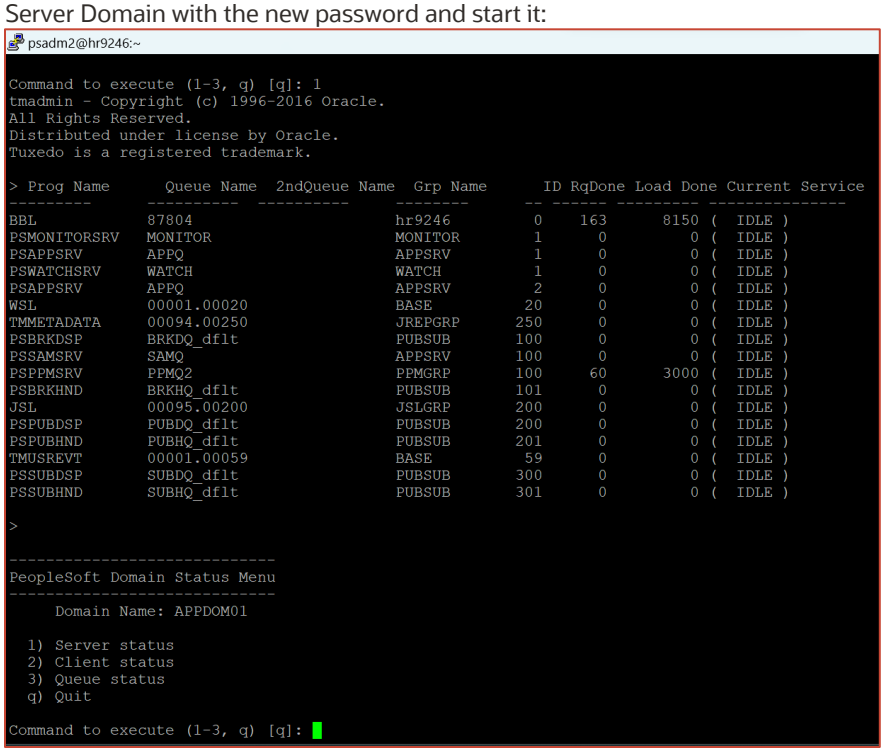

Figure 25 – App Server Configuration

Configure Process Scheduler

Similarly, configure the Process Scheduler Domain with the new password and start it:

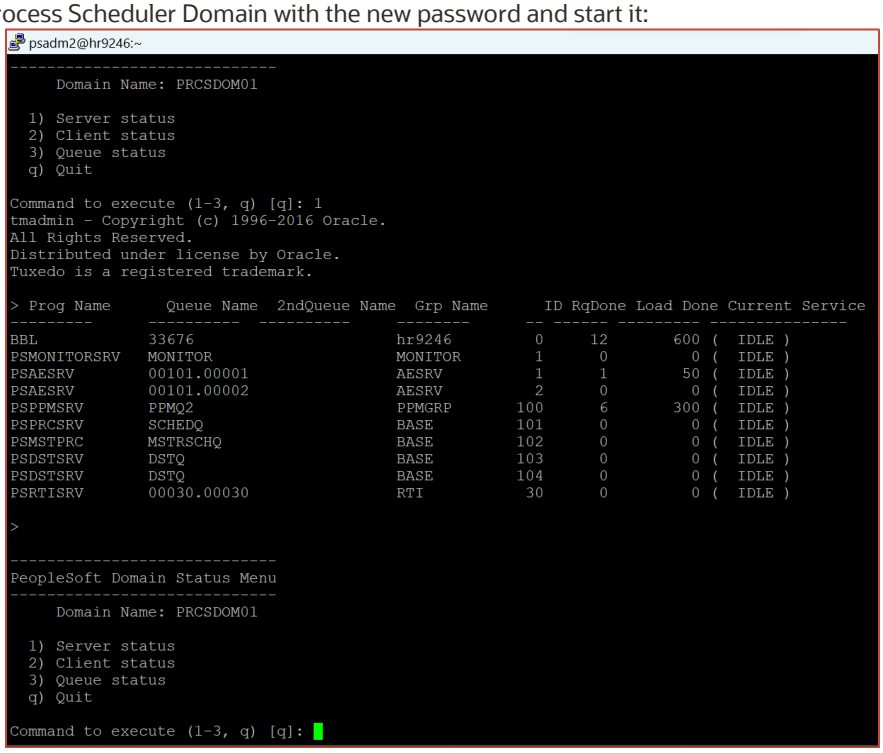

Figure 26 – Process Scheduler Configuration

Configure Web Server

Configure a new Web Server Domain and start it:

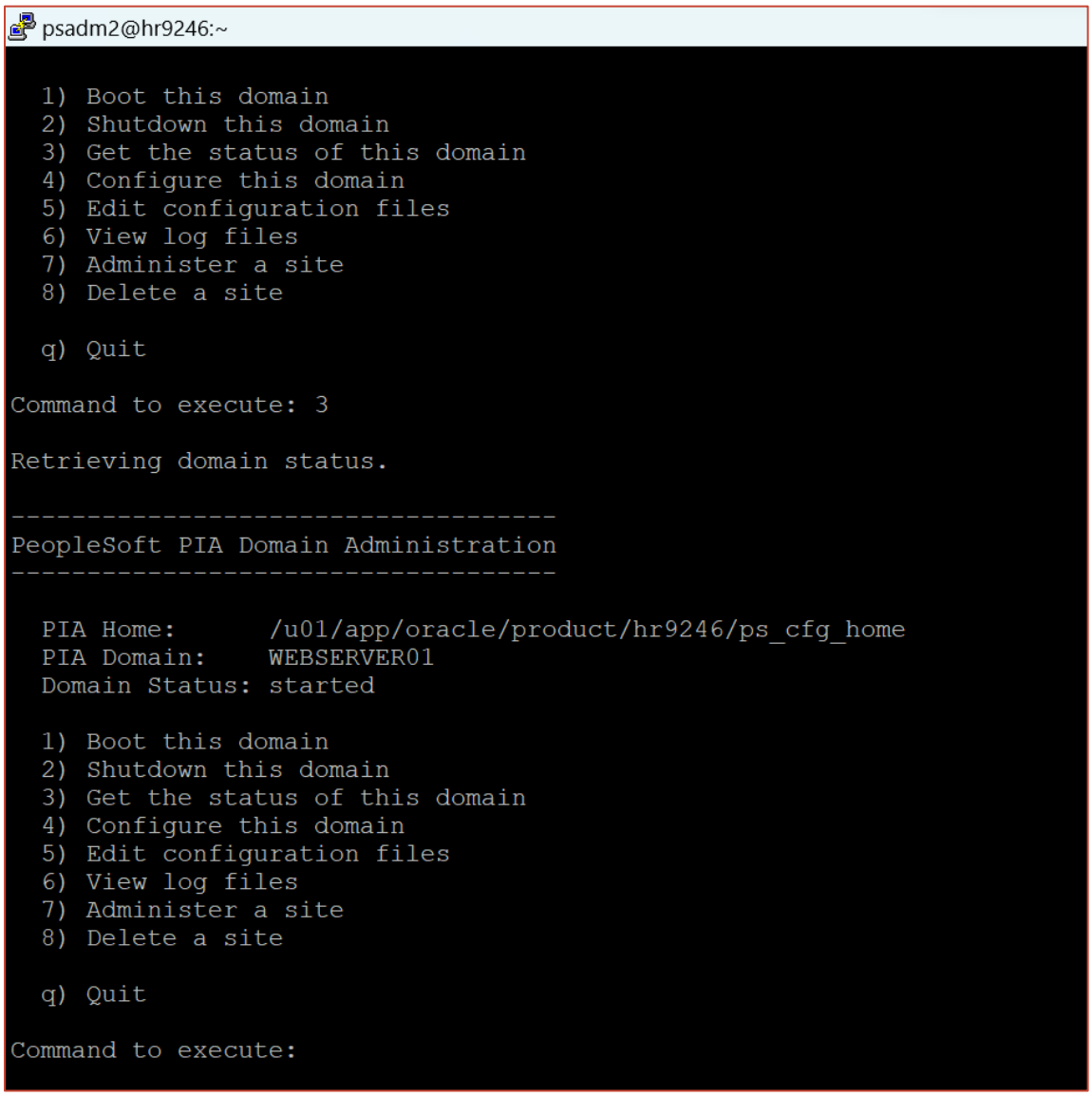

Figure 27 – Web Server Configuration

Configure PeopleSoft Components

Configure IB, Nodes, Report Repository, Printers, etc., as part of the post configuration of the PeopleSoft Application.

Validate PeopleSoft Application with ADB-D

Login via PIA of OCI Target Application and validate system health and performance.

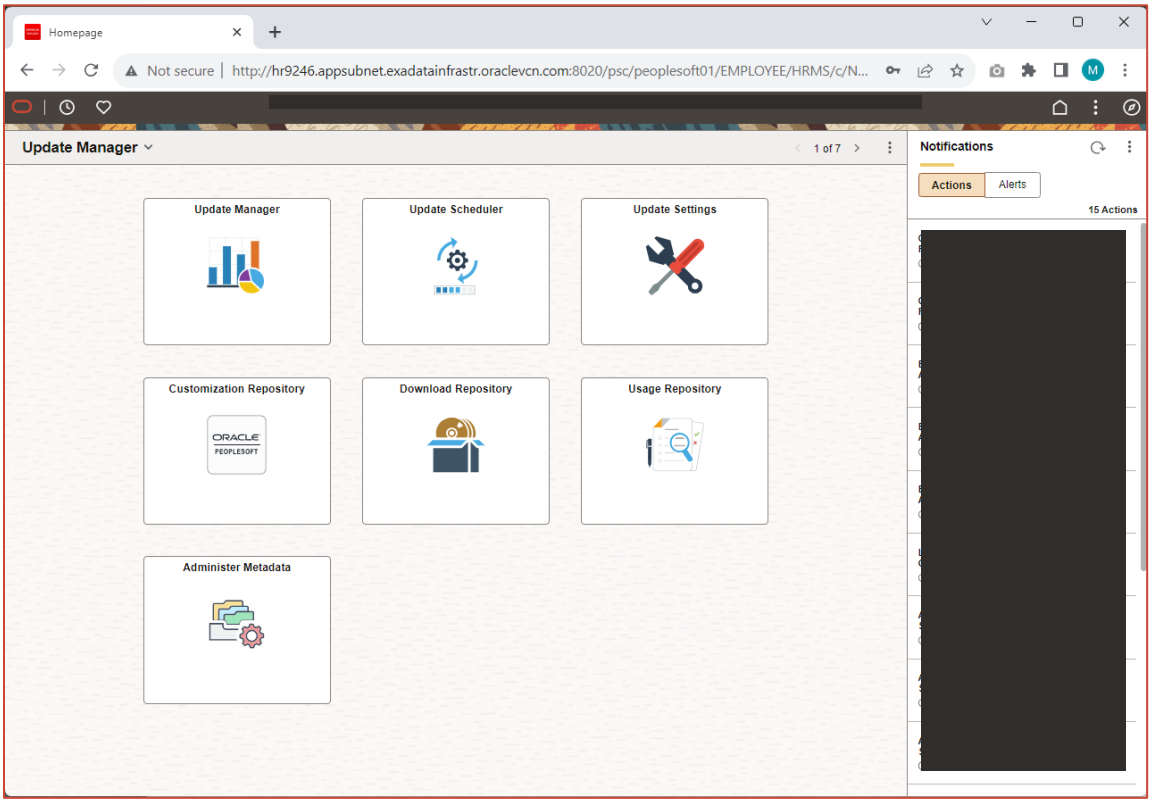

Figure 28 – PeopleSoft Application Validation

REFERENCES

My Oracle Support Articles

- ZDM: How To Install And Uninstall Zero Downtime Migraton(ZDM) Software (Doc ID 2630479.1)
  CPAT: Cloud Premigration Advisor Tool (CPAT) Analyzes Databases for Suitability of Cloud Migration (Doc ID 2758371.1)
- Required Interim Patches for the Oracle Database with PeopleSoft (Doc ID 1100831.1)
- E-SEC: How To Change The Access ID (SYSADM) Password? (Doc ID 609603.1)
- E-ORA: How to change the PS database user password? (Doc ID 2398975.1)

OCI Documentation:

- OCI Documentation: https://docs.cloud.oracle.com/en-us/iaas/Content/services.htm
- OCI CLI: https://docs.cloud.oracle.com/en-us/iaas/Content/API/SDKDocs/cliinstall.htm
- Compute: https://docs.cloud.oracle.com/en-us/iaas/Content/Compute/Concepts/computeoverview.htm
- Block Volume: https://docs.cloud.oracle.com/en-us/iaas/Content/Block/Concepts/overview.htm
- OCI Network: https://docs.cloud.oracle.com/en-us/iaas/Content/Network/Concepts/overview.htm
- ADB: https://docs.cloud.oracle.com/en-us/iaas/Content/Database/Concepts/adboverview.htm
- ADB-D: https://docs.oracle.com/en/cloud/paas/autonomous-database/dedicated/adbaa/index.html#articletitle
- Access Control Within Autonomous Database on Dedicated Exadata Infrastructure: https://docs.oracle.com/en/cloud/paas/autonomous-database/dedicated/adbcx/#articletitle

ZDM Documentation:

- ZDM for migration to ADB: https://www.oracle.com/a/tech/docs/oracle-zdm-logical-migration-to-autonomous-guide.pdf
- ZDM Product Documentation: https://docs.oracle.com/en/database/oracle/zero-downtime-migration/index.html

BLOG:

- Now Supported: PeopleSoft Applications Using Autonomous Database: https://blogs.oracle.com/peoplesoft/post/now-supported%C2%A0-peoplesoft-applications-using-autonomous-database
- Connecting ADB-D using the client authenticates the server (one-way TLS or simply TLS): https://blogs.oracle.com/datawarehousing/post/connecting-your-autonomous-database-has-never-been-easier.

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

PeopleSoft Application with Automomous Database – Migration Guide with Oracle ZDM || Technical Brief
November, 2024
Author: Mukul Prasad
Consulting Authors: Oracle ZDM Development Team, Oracle ZDM Product Management Team, Oracle Autonomous Database Product Management Team