

Oracle Zero Downtime Migration – Logical Offline Migration to ADB-S on Oracle Database@Azure

Technical Brief

October, 2024, Version [1.0]

Copyright © 2024, Oracle and/or its affiliates

Public

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, timing, and pricing 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

Purpose	4
Zero Downtime Migration	5
Supported Configurations	5
Architecture	6
Zero Downtime Migration Service Host	6
Zero Downtime Migration Service Host Requirements	6
Network and Connectivity	7
Source Database	7
Target Database	8
NFS File Share via Azure Files Configuration	9
Pre-Requisites	9
Additional Configuration	12
SSH Key	12
Authentication Token	12
OCI CLI Command Line Tool	12
API Signing Public Key and Configuration File	13
Database Migration Step by Step with ZDM	14
Step 1: Fill the response file	14
Step 2: Perform a migration in evaluation mode	14
Step 3: Migrate the Database	16
Troubleshooting Oracle ZDM & Other Resources	18

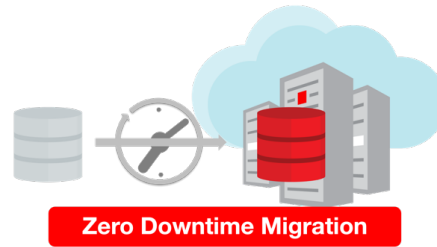


Figure 1. The Oracle Zero Downtime Migration Logo comprises a Database and a Clock with an arrow pointing to a Database deployed in the Cloud.

Purpose

Oracle customers are rapidly increasing their workload migration into the Oracle Cloud, Engineered Systems, and Oracle Database@Azure. 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 you to migrate your on-premises Oracle Databases directly and seamlessly to and between Oracle Database@Azure 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) 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 Oracle Autonomous Database Serverless (ADB-S) on Oracle Database@Azure, with ZDM's Logical Offline workflow.

Oracle ZDM will run on a separate node and connect to Source and Target to perform the migration. This guide will cover all requirements for installing the Oracle ZDM Service Host, the Source Database, the Target Database recipient of the migration process, and the 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.

For more information on Oracle Zero Downtime Migration, please visit ZDM's product website and Oracle Database@Azure product website.²

¹ <https://oracle.com/goto/maa>

² <https://www.oracle.com/goto/zdm>

<https://www.oracle.com/cloud/azure/oracle-database-at-azure/>

4 Oracle Zero Downtime Migration – Logical Offline Migration to ADB-S on Oracle Database@Azure / Version [1.0]

Zero Downtime Migration

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 to ensure the most negligible impact on production workloads. The Source Database to be migrated can be on-premises, deployed on Oracle Cloud Infrastructure, or a 3rd Party Cloud. The Target Database deployment can be in Oracle Autonomous Database or Oracle Exadata Database Service on Dedicated Infrastructure (ExaDB-D) on Oracle Database@Azure, Database Cloud Service on Oracle Cloud Infrastructure (OCI) Virtual Machine, Exadata Cloud Service, Exadata Cloud at Customer, and 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 Offline Migration Workflow.

A standard Logical Offline migration will take the following steps:

1. Download and Configure ZDM.
2. ZDM starts database migration.
3. ZDM starts a data pump export job.
4. ZDM transfers dump files to the backup location.
5. ZDM starts a data pump import job and instantiates the target database.
6. ZDM switches over.
7. ZDM validates, cleans up, and finalizes the database migration.

Supported Configurations

Oracle ZDM supports Oracle Database versions 11.2.0.4, 12.1.0.2, 12.2.0.1, 18c, 19c and 21c. ZDM's physical migration workflow requires the Source and Target Databases to be in the same database release.

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

Architecture

An architectural overview of the ZDM server, the source database on-premises, the target database on Oracle Autonomous Database Serverless (ADB-S) on Oracle Database@Azure, and all networks and components required are described in the diagram below:

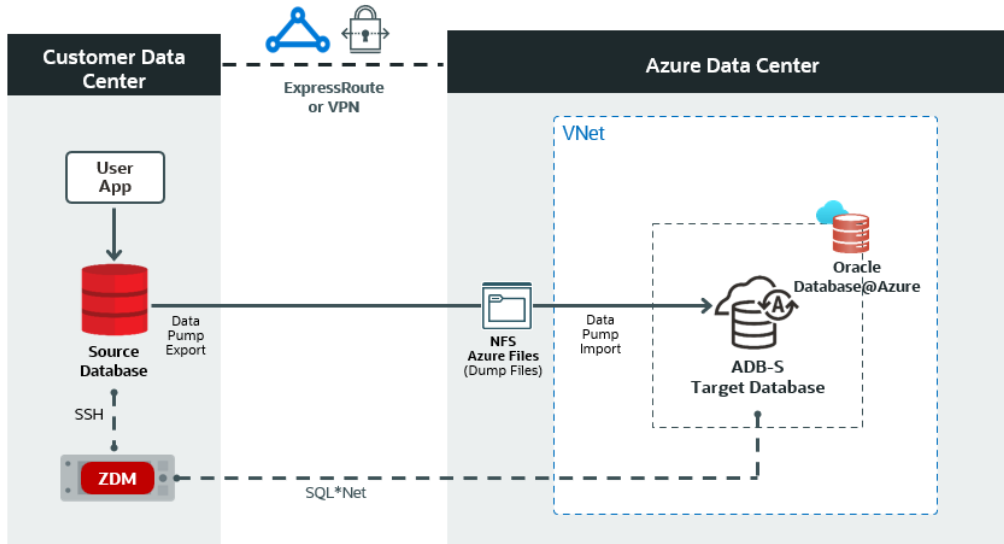


Figure 2. This is a High-Level Architectural overview showcasing the customer data center where the source database and ZDM's server reside. It also shows all connectivity to the target Oracle Autonomous Database Serverless (ADB-S) on Oracle Database@Azure.

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, or RHEL 8 (only these OS platforms/versions are supported).
- 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 and setting up ZDM on RHEL platforms, please refer to Oracle ZDM's product documentation, specifically "Setting Up Zero Downtime Migration Software" section³.

For this step-by-step guide, the ZDM Service Host runs on-premises on an Oracle Linux Server 8.9. The host private IP is masked for this guide, but as an example we will use the fictional `zz.dd.mm.hh` and the hostname is `zdmhost`.

³ <https://docs.oracle.com/en/database/oracle/zero-downtime-migration/index.html>

Network and Connectivity

Region

An Oracle Cloud Infrastructure region is a localized geographic area that contains one or more data centers, called availability domains. Regions are independent of other regions, and vast distances can separate them (across countries or continents).

Virtual Cloud Network (VCN) and subnet

A VCN is a customizable, software-defined network that you set up in an Oracle Cloud Infrastructure region. Like traditional data center networks, VCNs give you complete control over your network environment. A VCN can have multiple non-overlapping CIDR blocks that you can change after you create the VCN. You can segment a VCN into subnets, which can be scoped to a region or an availability domain. Each subnet consists of a contiguous range of addresses that don't overlap with the other subnets in the VCN. You can change the size of a subnet after creation. A subnet can be public or private.

OCI Network Security Group (NSG)

A network security group (NSG) provides a virtual firewall for cloud resources with the same security posture. For example, a group of compute instances performs the same tasks and thus needs to use the same set of ports.

Azure VNet

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many Azure resources, such as Azure virtual machines (VM), to securely communicate with each other, the internet, and on-premises networks.

Azure Delegated Subnet

Subnet delegation is Microsoft's ability to inject a managed service, specifically a platform-as-a-service service, directly into your virtual network. This means you can designate or delegate a subnet to be a home for an externally managed service inside your virtual network. In other words, that external service will act as a virtual network resource, even though technically it is an external platform-as-a-service service.

Virtual network interface card (vNIC)

The services in Azure data centers have physical network interface cards (NICs). Virtual machine instances communicate using virtual NICs (vNICs) associated with the physical NICs. Each instance has a primary vNIC that's automatically created and attached during launch and is available during the instance's lifetime.

Azure Route table (User Defined Route – UDR)

Virtual route tables contain rules to route traffic from subnets to destinations outside a VNet, typically through gateways. Route tables are associated with subnets in a VNet.

Source Database

For this step-by-step guide, the source database runs on-premises on an Oracle Linux Server 7.7. The host's private IP is masked for this guide, but as an example, we will use the fictional **aa.bb.sr.db** address, and the hostname is **oradbonpremises**.

The source Oracle database is a single-instance Enterprise Edition database version 19.22 with multitenant architecture. The Oracle SID is **zdmdb**, and the PDB name is **orclpdb**.

Target Database

Oracle Database@Azure offers the following products:

- **Oracle Exadata Database Service on Dedicated Infrastructure**
 - You can provision flexible Exadata systems that allow you to add database compute servers and storage servers to your system anytime after provisioning.

- **Oracle Autonomous Database Serverless**
 - Autonomous Database provides an easy-to-use, fully autonomous database that scales elastically, delivers fast query performance, and requires no database administration.

Oracle Database@Azure integrates Oracle Exadata Database Service, Oracle Real Application Clusters (Oracle RAC), and Oracle Data Guard technologies into the Azure platform. The Oracle Database service runs on Oracle Cloud Infrastructure (OCI) and is co-located in Microsoft Azure data centers. The service offers features and price parity with OCI. Users purchase the service on Azure Marketplace.

Oracle Database@Azure service offers the same low latency as other Azure-native services and meets mission-critical workloads and cloud-native development needs. Users manage the service on the Azure console and with Azure automation tools. The service is deployed in Azure Virtual Network (VNet) and integrated with the Azure identity and access management system. The OCI and Oracle Database metrics and audit logs are natively available in Azure. The service requires that users have an Azure tenancy and an OCI tenancy.

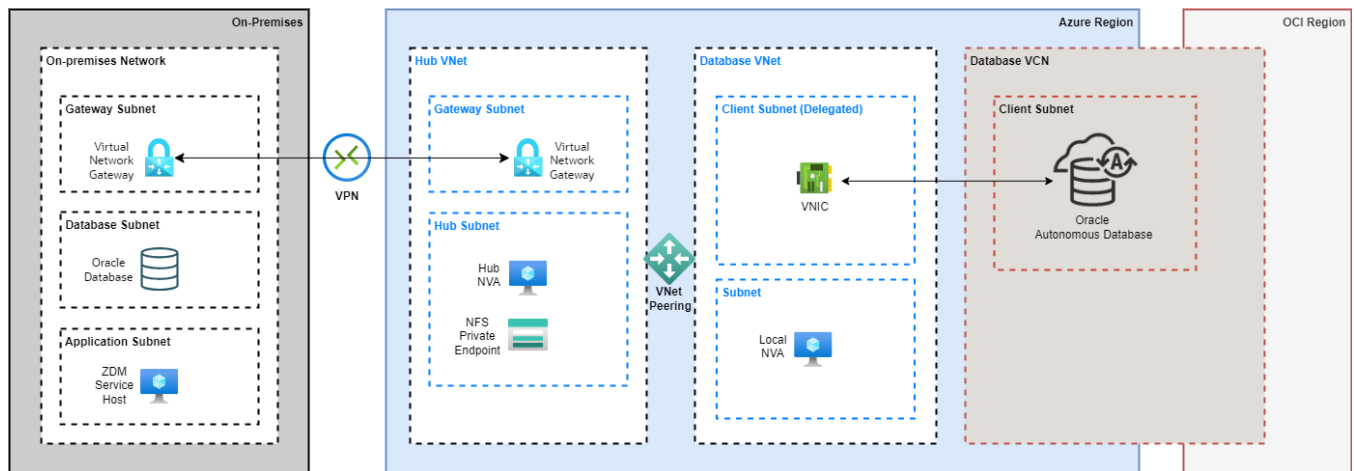


Figure 3 Network and Architectural Overview

For this step-by-step guide, the target platform is Oracle Autonomous Database Serverless (ADB-S) on Oracle Database@Azure. ZDM requires configuring a placeholder database target environment before beginning the migration process.

Enhanced Security for Outbound Connections with Private Endpoints

Setting the ROUTE_OUTBOUND_CONNECTIONS database property to the value PRIVATE_ENDPOINT enforces that all outgoing connections to a target host are subject to and limited by the private endpoint's egress rules.

```
ALTER DATABASE PROPERTY SET ROUTE_OUTBOUND_CONNECTIONS = 'PRIVATE_ENDPOINT';
```


NFS File Share via Azure Files Configuration

ZDM Logical Offline migration workflow uses Oracle Data Pump export and import to migrate the data from the source to the target database. An NFS file share is provided through the Azure Files service to store the Data Pump dump files.

The private endpoint of the storage account needs to be in a different VNet than the Autonomous Database⁴.

For this step-by-step guide, the file share path is

odaamigration.file.core.windows.net:/odaamigration/testmigration. The NFS private endpoint IP is masked for this guide, but as an example, we will use the fictional **aa.an.fs.pe** address.

The NFS share must be mounted on both the source database host and the target Autonomous Database.

To mount the NFS Share on the source database server:

As root:

```
mkdir /nfstest
```

```
mount -t nfs odaamigration.file.core.windows.net:/odaamigration/testmigration
/nfstest -o vers=4,minorversion=1,sec=sys
```

or using IP:

```
mount -t nfs aa.an.fs.pe:/odaamigration/testmigration /nfstest -o
vers=4,minorversion=1,sec=sys
```

Make sure the *oracle* user has access to the NFS mount

```
chown oracle:oinstall /nfstest
```

As *oracle* user:

```
touch /nfstest/test.txt
```

On the source PDB:

```
SQL> create directory DATA_PUMP_DIR_NFS as '/nfstest';
```

Pre-Requisites

ZDM Service Host

On the ZDM host, as root, add the Autonomous Database Private Endpoint URL **sample.adb.us-region-1.oraclecloud.com** to the `/etc/hosts` file to be resolved to the Autonomous Database Private Endpoint IP **aa.dd.bb.ss**.

Access Network File System from Autonomous Database

You can attach a Network File System to a directory location in your Autonomous Database⁵. This allows you to load data from Oracle Cloud Infrastructure File Storage in your Virtual Cloud Network (VCN), Azure Files NFS, or any other Network File System in on-premises data centers. Depending on the Network File System version you want to access, both **NFSv3** and **NFSv4** are supported.

⁴ <https://learn.microsoft.com/en-us/azure/oracle/oracle-db/oracle-database-network-plan>

⁵ <https://docs.oracle.com/en/cloud/paas/autonomous-database/serverless/adbsb/load-oci-file-storage.html#GUID-7C396A7A-D20A-40F7-99D7-50B85B9B18DC>

⁹ Oracle Zero Downtime Migration – Logical Offline Migration to ADB-S on Oracle Database@Azure / Version [1.0]

Step 1: Add NFS mount point FQDN to OCI DNS VCN Resolver

Bear in mind that if the OCI tenancy is a new tenancy created within the Oracle Database@Azure provisioning process, there might be a need to increase the limits for OCI private DNS and A records. To increase the limits, open a Service Request with Oracle Support. A limit of at least three records is needed.

Follow these steps to create an A-record in OCI DNS to resolve the FQDN of the Azure NFS mount point:

1. From the Oracle Autonomous Database details page in OCI, click on the virtual cloud network in the Network section.
2. On the network details page, click on the DNS Resolver.
3. On the private resolver details page, click on the default private view.
4. Click the create zone button and create a new zone using the name file.core.windows.net.

Create private zone

i You can only view or manage a zone when working in the region where it was created. This zone will not be visible when working from another region.

Zone type *Read-only* *i*

Primary

Zone name *i*

file.core.windows.net

Create in compartment

641531f9-2e12-430b-be35-0d8eba8d957c

orpsandbox6 (root)/MulticloudLink_ODBAA_20231213011510/641531f9-2e12-430b-be35-0d8eba8d957c

[Show advanced options](#)

5. Click on the newly created zone, manage records, and add a record. Replace aa.an.fs.pe by the actual IP address.

Add record

There is one answer (RDATA) for each record. Records of the same type that have the same name must also have the same TTL. For easier management, answers shown in this page are grouped by record type. Records that have the same name, type, and TTL are displayed as a single RRSet in the Zone Records list.

Record information

Name *Optional*

odaamigration .file.core.windows.net

Type

A - IPv4 address *↕*

Host record, used to point a hostname to an IPv4 address.

TTL in seconds

3600

RDATA/Answers *i*

RDATA mode

Basic

Address

aa.an.fs.pe *×*

6. Publish the changes.

Domain	Type	TTL ⓘ	RDATA
file.core.windows.net	NS 🔒	86400	vcn-dns.oraclevcn.com.
file.core.windows.net	SOA 🔒	86400	vcn-dns.oraclevcn.com. hostmaster.oracle.com.
odaamigration.file.core.windows.net	A	3600	

7. Update the Network Security Group (NSG) in OCI to allow network traffic flow from the VNet where the NFS private endpoint resides.

On the target database:

Step 2: Add the NFS Mount FQDN to the Access Control List (ACL)

```
SQL> exec DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(host =>
'odaamigration.file.core.windows.net', ace => xs$ace_type(privilege_list =>
xs$name_list('connect', 'resolve'), principal_name => 'ADMIN', principal_type =>
xs_acl.ptype_db));
```

PL/SQL procedure successfully completed.

Step 3: Create a Directory on the Autonomous Database

```
SQL> CREATE or replace DIRECTORY FSS_DIR AS 'fss';
```

Directory created.

Step 4: Attach NFS to Autonomous Database

Use "**params => JSON_OBJECT('nfs_version' value 4)**", otherwise NFS cannot be mounted.

```
SQL> BEGIN
DBMS_CLOUD_ADMIN.ATTACH_FILE_SYSTEM(
file_system_name => 'AZUREFILES',
file_system_location =>
'odaamigration.file.core.windows.net:/odaamigration/testmigration',
directory_name => 'FSS_DIR',
description => 'Attach Azure Files',
params => JSON_OBJECT('nfs_version' value 4)
);
END;
/
```

PL/SQL procedure successfully completed.

```
SQL> SELECT object_name FROM DBMS_CLOUD.LIST_FILES('FSS_DIR');
```

```
OBJECT_NAME
```

```
-----
test.txt
```

Additional Configuration

SSH Key

ZDM connects via SSH to the Source Database servers; hence, an SSH key pair for the zdmuser is required. As zdmuser, run 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:keyfingerprintsample zdmuser@zdmhost
[zdmuser@zdmhost ~]$ cd ~/.ssh
[zdmuser@zdmhost .ssh]$ cat id_rsa.pub >> authorized_keys
[zdmuser@zdmhost .ssh]$ chmod 600 authorized_keys
```

You can find more information on ZDM Product’s documentation section, “Generating a Private SSH Key Without a Passphrase.”⁶

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
```

Authentication Token

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

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

```
[zdmuser@zdmhost ~]$ sudo yum install python36-oci-cli
```

⁶ <https://docs.oracle.com/en/database/oracle/zero-downtime-migration/index.html>

API Signing Public Key and Configuration File

ZDM uses an API Signing Public Key to call REST APIs. First, you need to create the API Keys. Do so by accessing the terminal on the ZDM Service Host, and as the `zdmuser`, run the following:

```
[zdmuser@zdmhost ~]$ mkdir .oci
[zdmuser@zdmhost ~]$ cd .oci
[zdmuser@zdmhost ~]$ openssl genrsa -out /u01/app/zdmhome/.oci/oci_api_key.pem 2048
[zdmuser@zdmhost ~]$ openssl rsa -pubout -in /u01/app/zdmhome/.oci/oci_api_key.pem -out
/u01/app/zdmhome/.oci/oci_api_key_public.pem
[zdmuser@zdmhost ~]$ cat oci_api_key_public.pem
```

Copy the catted '`oci_api_key_public.pem`' file and save it; you will need it in the next step. Include the "Begin Public Key" and "End Public Key" lines during the copy. Go to your Oracle Cloud 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. Paste the public OCI API key file you copied above and click Add Key.

You will see a configuration file preview. Copy its contents; you will use them to populate your configuration file in the following step.

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 key file > # TODO` with the line above; once done, save the file and quit the editor:

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

Database Migration Step by Step with ZDM

Step 1: Fill the response file

```
vi /home/zdmuser/logical_offline_adb_nfs/logical_offline_adb_nfs.rsp

# migration method
MIGRATION_METHOD=OFFLINE_LOGICAL
DATA_TRANSFER_MEDIUM=NFS

# data pump
DATAPUMPSETTINGS_JOBMODE=SCHEMA
DATAPUMPSETTINGS_METADATAREMAPS-1=type:REMAP_TABLESPACE,oldValue:USERS,newValue:DATA
INCLUDEOBJECTS-1=owner:HR
DATAPUMPSETTINGS_EXPORTDIRECTORYOBJECT_NAME=DATA_PUMP_DIR_NFS
DATAPUMPSETTINGS_IMPORTDIRECTORYOBJECT_NAME=FSS_DIR

# source db
SOURCEDATABASE_CONNECTIONDETAILS_HOST=oradbonpremises
SOURCEDATABASE_CONNECTIONDETAILS_PORT=1521
SOURCEDATABASE_CONNECTIONDETAILS_SERVICENAME=orclpdb
SOURCEDATABASE_ADMINUSERNAME=SYSTEM

# target db
TARGETDATABASE_OCID=ocid1.autonomousdatabase.oc1.iad.aaaa.bbb.cccc.ddddd
TARGETDATABASE_ADMINUSERNAME=ADMIN

# oci cli
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_USERID=ocid1.user.oc1..aaaa.bbb.ccccc.ddddd
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_TENANTID=ocid1.tenancy.oc1.aaa.bbbbbb
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_FINGERPRINT=12:ac:34:cc:aa
OCIAUTHENTICATIONDETAILS_USERPRINCIPAL_PRIVATEKEYFILE=/home/zdmuser/.oci/oci_api_key.pem
OCIAUTHENTICATIONDETAILS_REGIONID=us-ashburn-1
```

Step 2: Perform a migration in evaluation mode

Execute the following command on the ZDM host as `zdmuser` to evaluate the migration. ZDM will check the source and target database configurations, but the actual migration will not be started. On the ZDM host as `zdmuser`

```
$ZDMHOME/bin/zdmcli migrate database \
-rsp /home/zdmuser/logical_offline_adb_nfs/logical_offline_adb_nfs.rsp \
-sourcenode oradbonpremises \
-sourcesid zmdmb \
-srcauth zdmauth \
-srcarg1 user:azureuser \
-srcarg2 identity_file:/home/zdmuser/.ssh/id_rsa \
-srcarg3 sudo_location:/usr/bin/sudo \
-eval
```

```
$ZDMHOME/bin/zdmcli query job -jobid 1

[zdmuser@zdmoprem logical_offline_adb_nfs]$ $ZDMHOME/bin/zdmcli query job -jobid 1
zdmoprem.internal.cloudapp.net: Audit ID: 1824
Job ID: 1
User: zdmuser
Client: zdmoprem
Job Type: "EVAL"

Scheduled job command: "zdmcli migrate database -rsp
/home/zdmuser/logical_offline_adb_nfs/logical_offline_adb_nfs.rsp -sourcename
oradbonpremise -sourcesid zmdmb -srcauth zdmauth -srcarg1 user:azureuser -srcarg2
identity_file:/home/zdmuser/.ssh/id_rsa -srcarg3 sudo_location:/usr/bin/sudo -eval"

Scheduled job execution start time: 2024-09-01T04:39:32-04. Equivalent local time: 2024-
09-0104:39:32

Current status: SUCCEEDED

Result file path: "/home/zdmuser/zdm/zdmbase/chkbase/scheduled/job-1-2024-09-01-
04:40:01.log"
Metrics file path: "/home/zdmuser/zdm/zdmbase/chkbase/scheduled/job-1-2024-09-01-
04:40:01.json"
Excluded objects file path: "/home/zdmuser/zdm/zdmbase/chkbase/scheduled/job-1-filtered-
objects-2024-09-01T04:40:17.309.json"

Job execution start time: 2024-09-01 04:40:01
Job execution end time: 2024-09-01 04:41:37
Job execution elapsed time: 1 minutes 35 seconds
ZDM_VALIDATE_TGT ..... COMPLETED
ZDM_VALIDATE_SRC ..... COMPLETED
ZDM_SETUP_SRC ..... COMPLETED
ZDM_PRE_MIGRATION_ADVISOR ..... COMPLETED
ZDM_VALIDATE_DATAPUMP_SETTINGS_SRC .... COMPLETED
ZDM_VALIDATE_DATAPUMP_SETTINGS_TGT .... COMPLETED
ZDM_PREPARE_DATAPUMP_SRC ..... COMPLETED
ZDM_DATAPUMP_ESTIMATE_SRC ..... COMPLETED
ZDM_CLEANUP_SRC ..... COMPLETED
```

Step 3: Migrate the Database

To initiate the actual migration, execute the same command for evaluation, but this time without the `-eval` parameter. On the ZDM host as `zdmuser`:

```
$ZDMHOME/bin/zdmcli migrate database \  
-rsp /home/zdmuser/logical_offline_adb_nfs/logical_offline_adb_nfs.rsp \  
-sourcenode oradbbonpremises \  
-sourcesid zmdmb \  
-srcauth zdmauth \  
-srcarg1 user:azureuser \  
-srcarg2 identity_file:/home/zdmuser/.ssh/id_rsa \  
-srcarg3 sudo_location:/usr/bin/sudo
```

```
[zdmuser@zdmmonprem logical_offline_adb_nfs]$ $ZDMHOME/bin/zdmcli query job -jobid 2  
zdmmonprem.internal.cloudapp.net: Audit ID: 1825
```

```
Job ID: 2
```

```
User: zdmuser
```

```
Client: zdmmonprem
```

```
Job Type: "MIGRATE"
```

```
Scheduled job command: "zdmcli migrate database -rsp  
/home/zdmuser/logical_offline_adb_nfs/logical_offline_adb_nfs.rsp -sourcenode  
oradbbonpremises -sourcesid zmdmb -srcauth zdmauth -srcarg1 user:azureuser -srcarg2  
identity_file:/home/zdmuser/.ssh/id_rsa -srcarg3 sudo_location:/usr/bin/sudo"
```

```
Scheduled job execution start time: 2024-09-01T04:42:02-04. Equivalent local time: 2024-  
09-0104:42:02
```

```
Current status: SUCCEEDED
```

```
Result file path: "/home/zdmuser/zdm/zdmbase/chkbase/scheduled/job-02-2024-09-01-  
04:42:31.log"
```

```
Metrics file path: "/home/zdmuser/zdm/zdmbase/chkbase/scheduled/job-02-2024-09-01-  
04:42:31.json"
```

```
Excluded objects file path: "/home/zdmuser/zdm/zdmbase/chkbase/scheduled/job-02-  
filtered-objects-2024-09-01T04:42:46.247.json"
```

```
Job execution start time: 2024-09-01 04:42:31
```

```
Job execution end time: 2024-09-01 05:39:12
```

```
Job execution elapsed time: 56 minutes 40 seconds
```

```
ZDM_VALIDATE_TGT ..... COMPLETED  
ZDM_VALIDATE_SRC ..... COMPLETED  
ZDM_SETUP_SRC ..... COMPLETED  
ZDM_PRE_MIGRATION_ADVISOR ..... COMPLETED  
ZDM_VALIDATE_DATAPUMP_SETTINGS_SRC .... COMPLETED  
ZDM_VALIDATE_DATAPUMP_SETTINGS_TGT .... COMPLETED  
ZDM_PREPARE_DATAPUMP_SRC ..... COMPLETED
```



```
ZDM_DATAPUMP_ESTIMATE_SRC ..... COMPLETED
ZDM_PREPARE_DATAPUMP_TGT ..... COMPLETED
ZDM_DATAPUMP_EXPORT_SRC ..... COMPLETED
ZDM_TRANSFER_DUMPS_SRC ..... COMPLETED
ZDM_DATAPUMP_IMPORT_TGT ..... COMPLETED
ZDM_POST_DATAPUMP_SRC ..... COMPLETED
ZDM_POST_DATAPUMP_TGT ..... COMPLETED
ZDM_POST_ACTIONS ..... COMPLETED
ZDM_CLEANUP_SRC ..... COMPLETED
```

Troubleshooting Oracle ZDM & Other Resources

For Oracle ZDM log review:

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

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

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

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 © 2024, Oracle and/or its affiliates. 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, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.