ORACLE

OCI Database Migration Service Tutorial - Online Migration to Oracle AI Database@Azure

Aimed for scenarios where your application must remain online, your source database has a direct connection to OCI.

April 2026, Version 1.1

Public

Purpose statement

This document walks you through all the steps to get started using Oracle Cloud Infrastructure (OCI) Database Migration (DMS). You will provision a Virtual Cloud Network (VCN), an Oracle Database instance, and an Oracle Autonomous AI Database instance to perform an online database migration using DMS.

With DMS we make it quick and easy for you to migrate databases from on-premises, Oracle, or third-party cloud into Oracle databases on OCI.

Disclaimer

Table of contents

Introduction to OCI Database Migration – DMS

OCI Database Migration (DMS) provides a high performant, self-service experience to achieve migrations, which include:

Homogeneous migration of data from Oracle or databases into OCI.

Enterprise-level logical online and offline migrations with minimal downtime into OCI targets.

Based on industry leading GoldenGate for data replication.

DMS Documentation:

Please review the documentation here.

Prerequisites for this guide:

- Azure permissions to accept private offers on the Azure Marketplace
- Azure Virtual Network with a delegated subnet to the Oracle Database@Azure service: (Oracle.Database/networkAttachments)
- A deployed Oracle AI Database@Azure Autonomous AI Database Service
- Federated SSO user to sign in into OCI

More details can be found on Onboarding with Oracle Database@Azure.

Task 0 – Understand New DMS Concepts

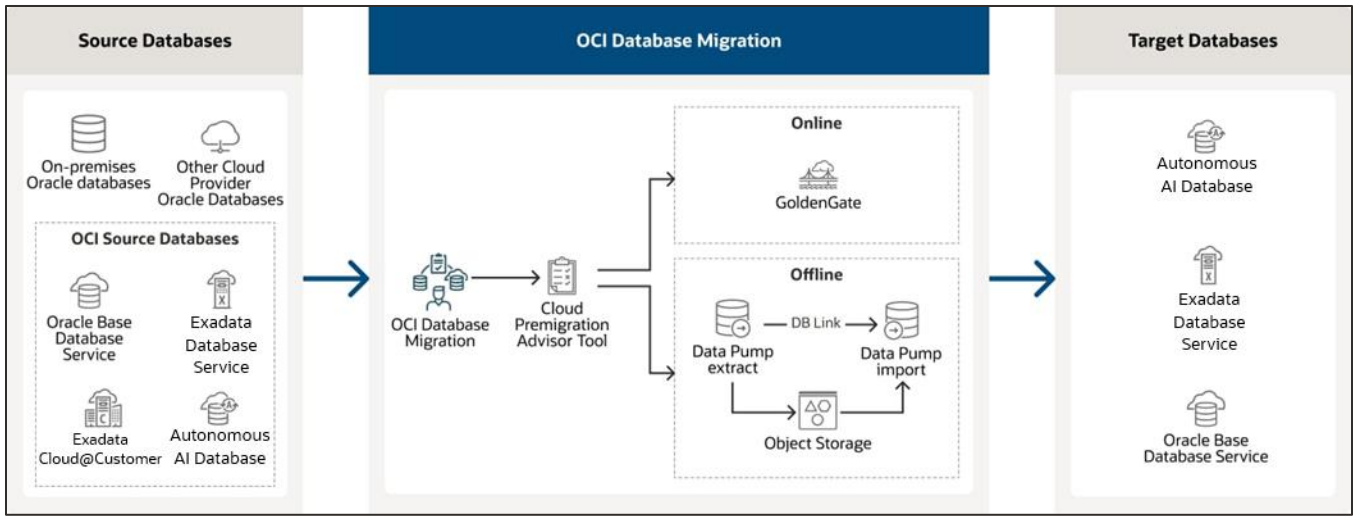
DMS provides a fully managed approach to migrating databases from various locations into OCI-hosted databases.

Migrations can be either one of the following modes:

- **Offline**: The Migration makes a point-in-time copy of the source to the target database. Any changes to the source database during migration are not copied, requiring any applications to stay **offline** for the duration of the migration.
- **Online**: The Migration makes a point-in-time copy and replicates all subsequent changes from the source to the target database. This allows applications to stay **online** during the migration and then be switched over from source to target database.

DMS supports both offline and online mode. For Oracle migrations the source databases can be located on-premises, in 3rd party clouds, or on Oracle OCI. The supported targets can be Oracle Autonomous AI Database Serverless or Dedicated, Oracle Base Database Service and Exadata Database Service on dedicated infrastructure.

The DMS service runs as a managed cloud service separate from the user's tenancy and resources. The service operates as a multi-tenant service in a DMS Service Tenancy and communicates with the user's resources using Private Endpoints (PEs). PEs are managed by DMS and are transparent to the user.



DMS Simplified Topology

Compartment: A compartment is a collection of related resources (such as cloud networks, compute instances, or block volumes) that can be accessed only by those groups that have been given permission by an administrator in your organization. For example, one compartment could contain all the servers and storage volumes that make up the production version of your company's Human Resources system. Only users with permission to that compartment can manage those servers and volumes.

Data region: A geographical region that's associated with one or more data centers. When you sign up for an Oracle Cloud account, you select a default data region, where your services will be hosted.

DMS Control Plane: Used by DMS end user to manage Migration and Database Connection objects. The control plane is exposed through the DMS Console UI as well as the Rest API.

DMS Data Plane: Managed by DMS Control Plane and transparent to the user. The GGS Data Plane manages ongoing migration jobs and communicates with the user's databases and GoldenGate instance using PEs. The DMS data plane does not store any customer data, as data flows through GoldenGate and Data Pump directly within the user's tenancy.

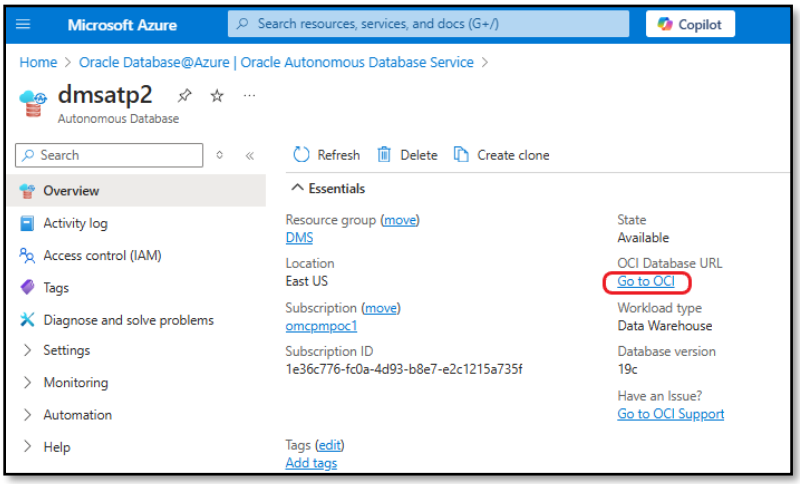
Migration: A Migration contains metadata for migrating one database. It contains information about source, target, and migration methods and is the central object for users to run migrations. After creating a migration, a user can validate the correctness of the environment and then run the migration to perform the copy of database data and schema metadata from source to target.

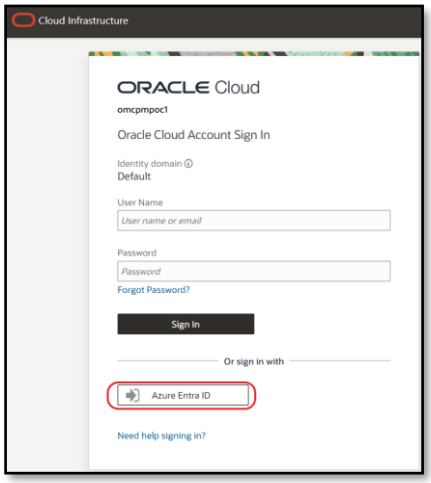
Migration Job: A Migration Job displays the state or a given Migration execution, either for validation or migration purposes. A job consists of several sequential phases, users can opt to wait after a given phase for user input to resume with the following phase.

Database Connection: A **Database Connection** represents information about a source or target database, such as connection and authentication credentials. DMS uses the OCI Vault to store credentials. A **Database Connection** is reusable across multiple Migrations.

Task 1 – Sign in to OCI and Open DMS Console

- On the Azure portal navigate to Home >Oracle Database@Azure | Oracle Autonomous AI Database Service > Select your database > Click on the Go to OCI link
- Log in using your Azure Entra id credentials.
- In the OCI console title bar change region if applicable.





Task 2 – Have the Administrator Set Required Permissions

The following permissions need to be set in OCI to have access to the necessary objects unless you have administrative privileges. The following permissions assume that the user is part of group DMS_LA and all resources are created in a compartment called DMS_LA. Have your tenancy administrator set these permissions. Review the following documentation on Required Policies if required

PERMISSIONS REQUIRED BY DMS TO USE DATABASES, VAULTS, AND NETWORKING
Allow group DMS_LA to manage virtual-network-family in compartment DMS_LA
Allow group DMS_LA to manage vaults in compartment DMS_LA
Allow group DMS_LA to manage keys in compartment DMS_LA
Allow group DMS_LA to manage database-family in compartment DMS_LA
Allow group DMS_LA to manage autonomous-database-family in compartment DMS_LA
Allow group DMS_LA to manage object-family in compartment DMS_LA
Allow group DMS_LA to manage secret-family in compartment DMS_LA
Allow group DMS_LA to manage goldengate-connections in compartment DMS_LA
Allow group DMS_LA to manage odms-connection in compartment DMS_LA
Allow group DMS_LA to manage odms-migration in compartment DMS_LA
Allow group DMS_LA to manage odms-job in compartment DMS_LA
Allow group DMS_LA to manage cloud-shell in compartment DMS_LA

Task 3 – Create Virtual Cloud Network

The following task is optional if a suitable VCN is already present.

In the OCI Console Menu, go to Networking > Virtual Cloud Networks

Pick a compartment on the left-hand side Compartment list. You need to have the necessary permissions for the compartment.

Click on Start VCN Wizard and pick VCN with Internet Connectivity.

Enter a VCN Name, such as VCN_DMS_LA. Leave CIDR block defaults, unless you need non-overlapping addresses for peering later. Press Next.

Review Summary and press Create.

Task 4 – Update Security List for Virtual Cloud Network Subnet

This task assumes default permissions in your public subnet. If you disabled or restricted your default permissions such as port 22 SSH access or restricted egress, please add default permissions as needed.

In the OCI Console Menu, go to **Networking > Virtual Cloud Networks** and pick your VCN.

In the Subnets list, pick Public Subnet-*VCN NAME*.

In the Security Lists list, pick Default Security List for *VCN NAME*.

In the Ingress Rules list press Add Ingress Rules.

Enter the following values, otherwise leave defaults:

- Source CIDR: 0.0.0.0/0
- Destination Port Range: 443
- Description: OGG HTTPS

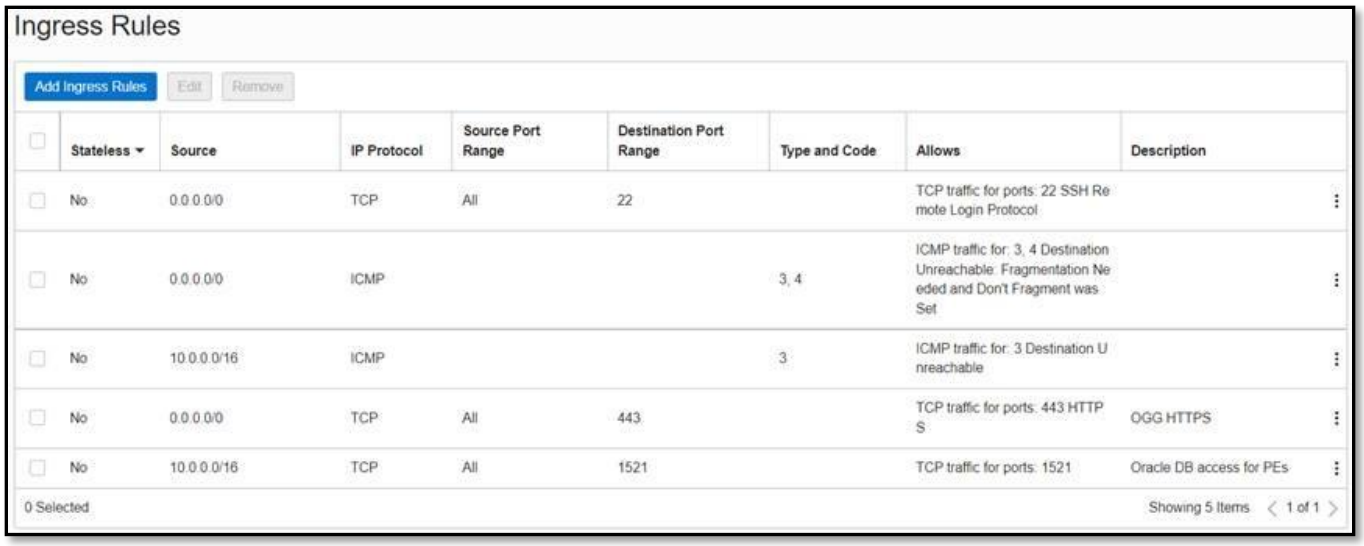
Close dialog by pressing **Add Ingress Rules**.

In the Ingress Rules list press Add Ingress Rules.

Enter the following values, otherwise leave defaults:

- Source CIDR: **10.0.0.0/16**
- Destination Port Range: **1521**
- Description: Oracle DB access for PEs

Close dialog by pressing **Add Ingress Rules**.



Task 5 – Create Vault

The following task is optional if a Vault is already present.

In the OCI Console Menu, go to **Identity & Security > Vault**.

Pick a compartment on the left-hand side **Compartment** list.

Press Create Vault.

In the **Create Vault** dialog, enter a Name such as **DMS_Vault**.

Close the dialog by pressing **Create Vault**.

Wait until the state of the new vault is **Active**.

Click on the new vault and press **Create Key** in the **Master Encryption Keys** list.

In the **Create Key** dialog, enter a Name such as **DMS_Key**.

Close the dialog by pressing **Create Key**.

Task 6 – Create Source Database

The following task is optional if a source database is already present. In this example the source database is a Base Database with Oracle Database 19c.

In the OCI Console Menu, go to Oracle Database > Oracle Base Database Service.

Press Create DB System.

Enter the following values, otherwise leave defaults. You can adjust shapes and storage to your use case requirements and available quota.

- Name: SourceDB
- Leave VM.Standard.E5.Flex as default shape.

- Select generate SSH key pair, you need to save the private and public keys.
- Choose a license type as applicable: BYOL
- Virtual cloud network: VCN_DMS_LA *(Or your VCN name)*
- Client subnet: Public Subnet-VCN_DMS_LA *(Or your subnet name)*
- Hostname prefix: sourcedb

Press **Next**

Enter the following values, otherwise leave defaults.

- Database name: sourcedb
- PDB name: pdb
- Create administrator credentials – Password: *password of your choice*

Press Create DB System

The provisioning of the database can take 30 or more minutes. Wait for the Lifecycle State of the database to change to Active.

Open the database system SourceDB in the DB Systems table

Open the database sourcedb in the databases table

Press DB Connection

Press Show next to the Easy Connect Connection String. A string like:

`sourcedb.sub12062328210.vcndmsla.oraclevcn.com:1521/`**`sourcedb_iad158.sub12062328210.vcndmsla.oraclevcn.com`**

should be shown. Copy the string after the /, in this case:

`sourcedb_iad158.sub12062328210.vcndmsla.oraclevcn.com`

This is the service name of your CDB, you will need this string later for accessing your database and creating migrations. Close the dialog.

For online migrations and when the source database is multitenant architecture, and the database version is < 19.23 then one connection to the CDB and one for the PDB are required. When the database version is >= 19.23 then only one connection to the PDB is required.

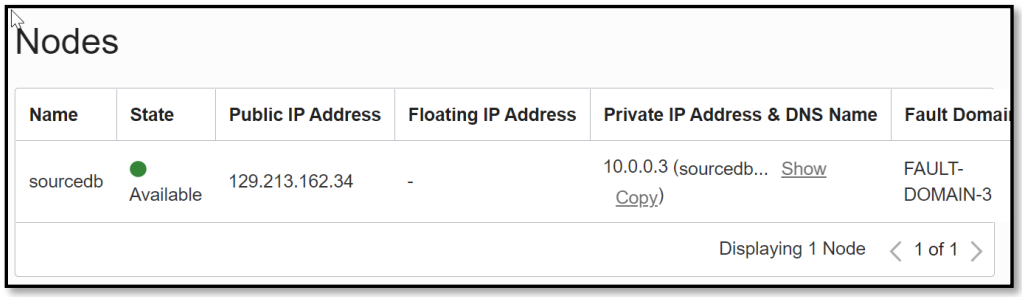
Click on Pluggable Databases link on the left side under Resources section and click on pdb.
Press DB Connection. Like with CDB steps copy the string after the /, this is the service name of your PDB a string like:

`pdb.sub12062328210.vcndmsla.oraclevcn.com`

Go back to the DB Systems Details page of your database and select Nodes on the left-hand side Resources list.

The Nodes list shows the sourcedb node. Note the Public IP Address and Private IP Address of the node, in this case 129.213.162.34 and 10.0.0.3.

These values **can be used** later during database connection creation.



Task 7 – Create an Oracle AI Database@Azure as Target

The following task is optional since an Autonomous AI database should already be present.

1. On the Azure portal navigate to Home> Oracle Database@Azure | Oracle Autonomous AI Database Click on Create
2. Select the subscription, resource group and give a name to your instance.
3. For this testing we are leaving the default values on the configuration tab: Data Warehouse as workload type and database version 19c.
4. Provide a password for the Admin user
5. On the Networking tab, select the virtual network and a subnet with the required subnet delegation.
6. Click on Review + create

Task 8 – Prepare Source

This task prepares the required user accounts and settings for Migration in the Source DB. It assumes default settings in the database. If you changed default settings, further settings might be necessary.

Open an SSH terminal to the source database instance. The instructions are for Unix-style ssh command:

```
ssh -i <private_key_file> opc@<dbnode_public_ip>
```

Create a new directory in the user volume, this directory will be used to temporary store the database export files:

```
sudo su - oracle
mkdir /u01/app/oracle/dumpdir
```

In this guide for your Oracle Base Database Service source no SSH details will be provided during the creation of the **database connection**, to achieve HTTPS connectivity, the following steps need to be followed:

a. Create a new directory: `mkdir /u01/app/oracle/dumpdir/wallet`

b. Download a pre created SSL wallet: `curl -o walletSSL.zip` https://objectstorage.us-phoenix-1.oraclecloud.com/p/YYkalHlLbbrfOAMIor-Mzl1qcFxaAZOvrYABKzRQYPErFQdzJrVjma1cUg4SIXEu/n/axsdric7bk0y/b/SSL-Wallet-For-No-SSH-Migrations-Setup/o/walletSSL.zip

i. This link Is also available in the official documentation in the “Managing migrations section”

c. Unzip the files: `unzip walletSSL.zip`

d. **Make sure these files are present in your desired directory path**:

- 2022 ewallet.p12.lck
- cwallet.sso.lck
- ewallet.p12
- cwallet.sso
- addedCertificates.txt
- Save this path location, you will need it during the **migration creation** to populate the **SSL Wallet Path** with it, i.e: **/u01/app/oracle/dumpdir/wallet**

The user performing the export or import requires the necessary network ACL to be granted to access the network from the source and target database host. Create the script file acl.sql with the following content, for this guide, run the following script as SYS. Run the script connected to the pluggable database and not to CDB$ROOT . Replace `clouduser` and `sslwalletdir` accordingly with GGADMIN.

```
define clouduser='system';/*user performing export at source or import at target*/
define sslwalletdir='/u01/app/oracle/dumpdir/wallet';/* OCI wallet path*/
begin
dbms_network_acl_admin.append_host_ace(
        host =>'*',
        lower_port => 443,
        upper_port => 443,
        ace => xs$ace_type(
                privilege_list => xs$name_list('http', 'http_proxy'),
                principal_name => upper('&clouduser'),
                principal_type => xs_acl.ptype_db));
dbms_network_acl_admin.append_wallet_ace(
    wallet_path => 'file:&sslwalletdir',
        ace => xs$ace_type(privilege_list =>
                xs$name_list('use_client_certificates', 'use_passwords'),
                principal_name => upper('&clouduser'),
                principal_type => xs_acl.ptype_db));
end;
/
```

Enter the following commands:

```
. oraenv
ORACLE_SID enter your database details.
sqlplus sys/<db password>@<db private ip>/<db cdb service> as sysdba
```

In SQL Plus enter the following commands:

```
SQL> @acl.sql
PL/SQL procedure successfully completed.
```

Once the connect privilege is granted, connect as the relevant user such as, SYSTEM, and verify if the privilege is granted using the following query:

```
SELECT host, lower_port, upper_port, privilege, status
FROM user_network_acl_privileges;
```

	HOST	LOWER_PORT	UPPER_PORT	PRIVILEGE	STATUS
1	*	443	443	http	GRANTED
2	*	443	443	http-proxy	GRANTED

Follow the next link for a reference to the documentation.

The next step will prepare the source database. It will create the DMSROLE and create or unlock GGADMIN user in the PDB and will provide all the required grants, this user will be provided during **source database connection creation**:

1) Download the preparation script from database creation connection screen.

2) Locate the file and run it dms-userprep-analyze.sql
- Provide a password for the GGADMIN user.

You should see an output like this, this ran in SQL Developer.

```
-- Privilege CREATE ANY SEQUENCE already granted TO DMSROLE
-- Privilege CREATE ANY TRIGGER already granted TO DMSROLE
-- Privilege CREATE ANY TYPE already granted TO DMSROLE
-- Privilege CREATE ANY VIEW already granted TO DMSROLE
-- Privilege ALTER ANY TABLE already granted TO DMSROLE
-- Privilege ALTER ANY INDEX already granted TO DMSROLE
-- Privilege ALTER ANY CLUSTER already granted TO DMSROLE
-- Privilege ALTER ANY INDEXTYPE already granted TO DMSROLE
-- Privilege ALTER ANY OPERATOR already granted TO DMSROLE
-- Privilege ALTER ANY PROCEDURE already granted TO DMSROLE
-- Privilege ALTER ANY SEQUENCE already granted TO DMSROLE
-- Privilege ALTER ANY TRIGGER already granted TO DMSROLE
-- Privilege ALTER ANY TYPE already granted TO DMSROLE
-- Privilege CREATE DATABASE LINK already granted TO DMSROLE
-- Privilege ALTER SYSTEM already granted TO DMSROLE
-- Privilege ALTER DATABASE already granted TO DMSROLE
GRANT DMSROLE TO GGADMIN;
-- Privilege SELECT ON V_$SESSION already granted TO GGADMIN
-- Privilege SELECT ON V_$TRANSACTION already granted TO GGADMIN
-- Privilege SELECT ON V_$DATABASE already granted TO GGADMIN
--
--
-- Execution finished. See summary below.
--
--##################################################################################################
--                Execution Summary
--##################################################################################################
-- Total attempted: 1
-- Succeeded     : 1
-- Skipped       : 0
-- Failed        : 0
-- End of execution summary


PL/SQL procedure successfully completed.
```

The next steps add a user HR01 with a sample table and data. If your database already contains data for migration, you can skip these steps.

Create the script file `create_hr01.sql` with the following content:

```
DROP USER HR01 CASCADE;
CREATE USER HR01 IDENTIFIED BY HR##hr01123;
GRANT CONNECT,RESOURCE,CREATE TABLE,CREATE SEQUENCE to HR01;
GRANT CREATE ANY PROCEDURE to HR01;
ALTER USER HR01 quota unlimited on users;
CREATE TABLE HR01.EMPL (col1 number, col2 varchar2(9), col3
varchar2(100), col4 timestamp);
ALTER TABLE HR01.EMPL ADD CONSTRAINT EMPL_i1 PRIMARY KEY
(col1,col2);
```

Create the script file `data_hr01.sql` with the following content:

```
SET ECHO OFF;
SET HEADING OFF;
SET FEEDBACK OFF;
SET SERVEROUTPUT ON;
DECLARE
  SCN     HR01.EMPL.COL1%TYPE;
  RND1    HR01.EMPL.COL2%TYPE;
  RND2    HR01.EMPL.COL3%TYPE;
  RND3    HR01.EMPL.COL4%TYPE;
  ROWSNUM NUMBER;
  DBNAME VARCHAR2(60);
  i       INTEGER;
BEGIN
  i := 0;
  LOOP
    SELECT COUNT(*) INTO ROWSNUM FROM HR01.EMPL;
    SELECT DBMS_RANDOM.STRING('P', 9) INTO RND1 FROM DUAL;
    SELECT DBMS_RANDOM.STRING('P', 10) INTO RND2 FROM DUAL;
    SELECT TO_DATE(TRUNC (DBMS_RANDOM.VALUE (2451545, 5373484)),
'J') INTO RND3 FROM DUAL;
    INSERT INTO HR01.EMPL(col1, col2, col3, col4) VALUES (ROWSNUM,
RND1, RND2, RND3);
    COMMIT;
    DBMS_OUTPUT.PUT_LINE('Number of rows = ' || ROWSNUM);
    IF ( i >= 1000 ) THEN
      EXIT;
    END IF;
    i := i + 1;
  END LOOP;
END;
/
```

Enter the following commands:

```
sqlplus sys/<db password>@<db private ip>/<db pdb service>  as sysdba
```

In SQL Plus enter the following commands:

```
SQL> @create_hr01.sql
DROP USER HR01 CASCADE   (You can ignore this error)
          *
ERROR at line 1:
ORA-01918: user 'HR01' does not exist

SQL> @data_hr01.sql
Number of rows = 0
[…]
Number of rows = 1000
SQL> quit
```

Your source DB now has a user HR01 with a table EMPL that has 1000 rows.

Task 9 – Prepare Target

The next steps will connect to the target Oracle Autonomous AI Database instance, enable, and prepare the standard GGADMIN user. Like in Task 8, **the preparation script needs to be downloaded** and placed wherever accessible.

You need to download the wallet, modify the sqlnet.ora so that it points to the wallet location, this step is not covered in this guide.

Make sure that your Autonomous AI Database mTLS authentication option is marked as 'Not required', you can check this in the **network** details of your Autonomous AI Database:

Click Database connection/ Connection settings section and select TLS from the TLS authentication list of values, then copy the connection string for one of the TNS names.

Connect to the **source** system as in Task 8 (will be used as jump host), once connected we should be able to connect to the target and run SQL commands:

```
ssh -i <private_key_file> opc@<dbnode_public_ip>
sudo su – oracle
export TNS_ADMIN=/home/oracle/<Directory where you placed the wallet>
```

Now connect to sqlplus:

```
sqlplus admin/ <ATP password>@ ATP connection string
```

In SQL Plus enter the following commands:

```
SQL> @dms-userprep-analyze.sql

SQL> PL/SQL procedure successfully completed.

SQL> quit
```

Task 10 – Create Object Store Bucket for Data Pump Storage

Object Store is used as temporary storage between source and target databases with Data Pump. This task is creating an empty bucket for use in the migration.

In the OCI Console Menu, go to Storage > Object Storage & Archive…
Press Create Bucket.

On the page Create Bucket, fill in the following entries, otherwise leave defaults:

- Bucket Name: **DMSStorage**

Press Create Bucket

Task 11 – Create Database Connection for Source PDB

In the OCI Console Menu, go to Migration & Disaster Recovery> Database Migration > Database Connections.

Press Create connection.

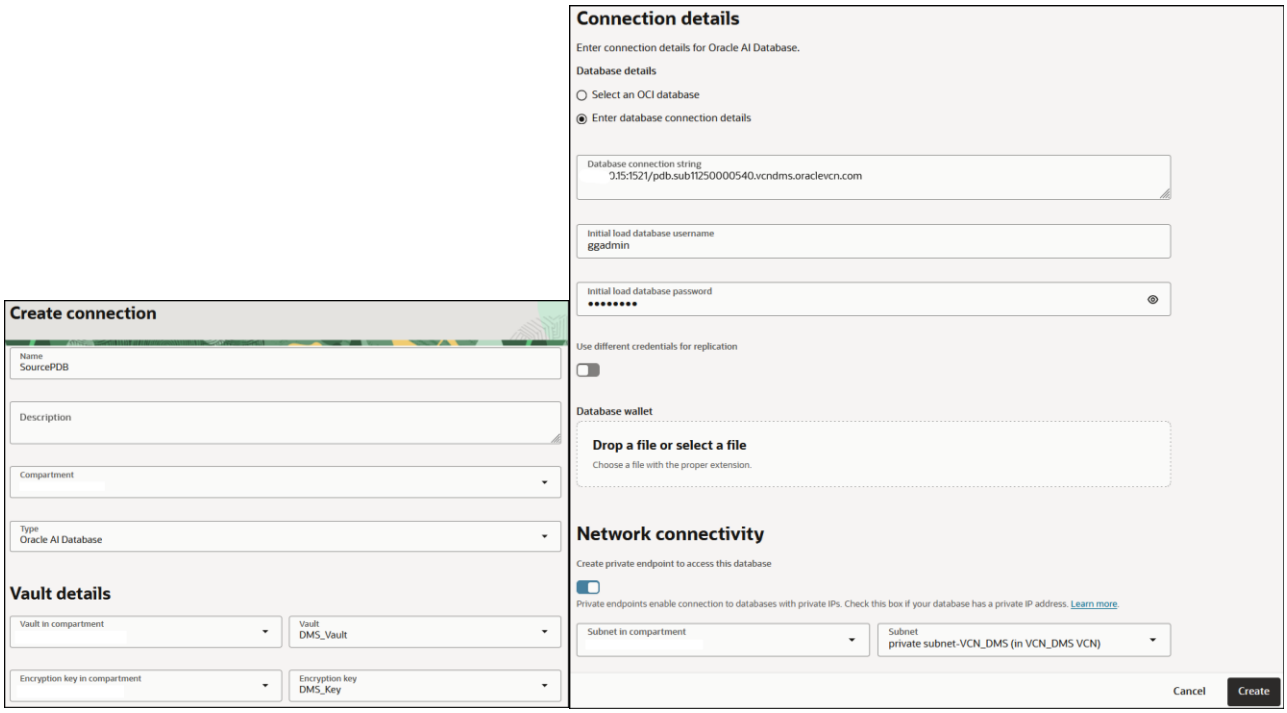
On the section Database Details, fill in the following entries, otherwise leave defaults:

- Name: **SourcePDB**
- Type: **Oracle AI Database**
- Vault: **DMS_Vault**
- Encryption Key: **DMS_Key**

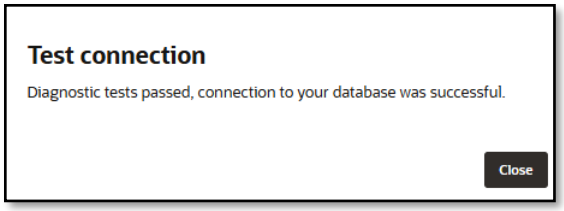
Select Database details: **Enter database connection details**

- Database connection string (Private IP:port/service name): **10.0.0.15:1521/pdb.sub11250000540.vcndms.oraclevcn.com**
- Initial load database username: **ggadmin**
- Initial load database password: < **password**>
- Create private endpoint to access this database : **Enable**
- Subnet: Select your private subnet as previously created.

Press Create



Once your newly created connection is in Active state, test it by clicking “Actions >Test connection” :



Task 12 – Create Database Connection for Target

In the OCI Console Menu, go to Migration & Disaster Recovery> Database Migration > Database Connections.

Press Create connection.

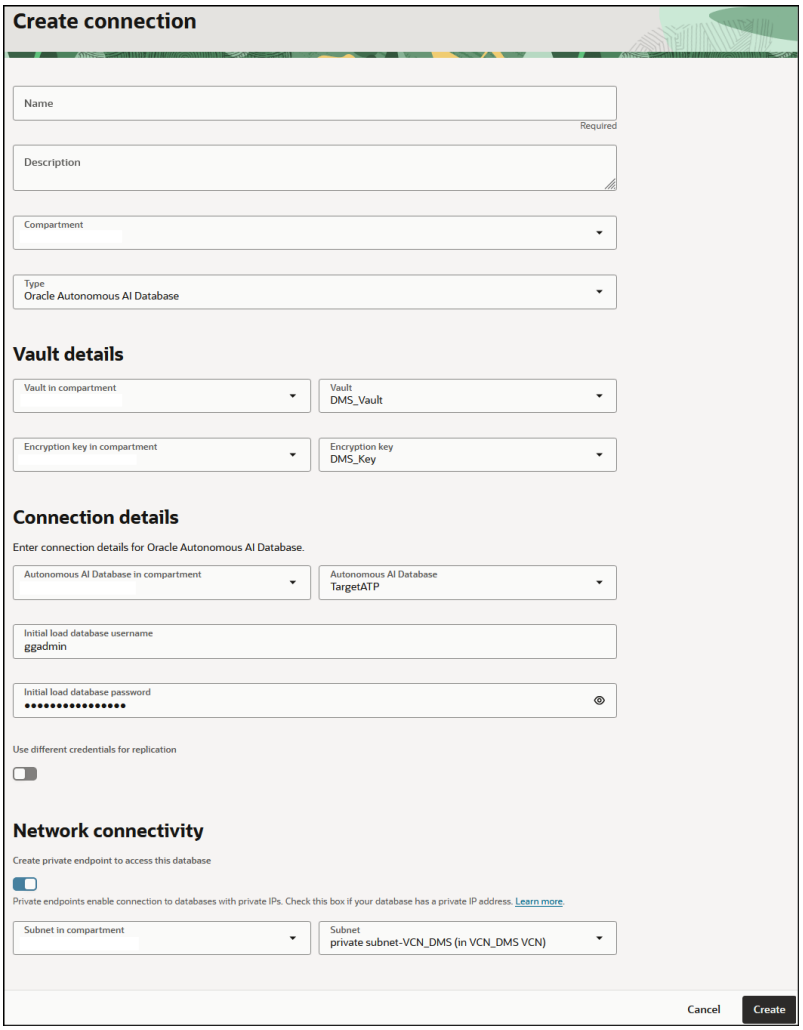
On the section Database Details, fill in the following entries, otherwise leave defaults:

- Name: **TargetATP**
- Type: **Oracle Autonomous AI Database**
- Vault: **DMS_Vault**
- Encryption Key: **DMS_Key**

  Select the Autonomous AI Database name in your compartment i.e: **dmsatp2**
- Initial load database username: **GGADMIN**
- Initial load database password: <**Admin password**>
- Network connectivity: **Create** private endpoint to access this database

- Subnet: Select your previously created private subnet

Press **Create**



Test your connection as in the previous task.

Task 13 – Create Migration

In the OCI Console Menu, go to Migration & Disaster Recovery> Database Migration > Migrations.

Press **Create Migration**.

On the page **Add Details**, fill in the following entries, otherwise leave defaults:

- Name: **TestMigration**

On the **Source Database**, fill in the following entries:

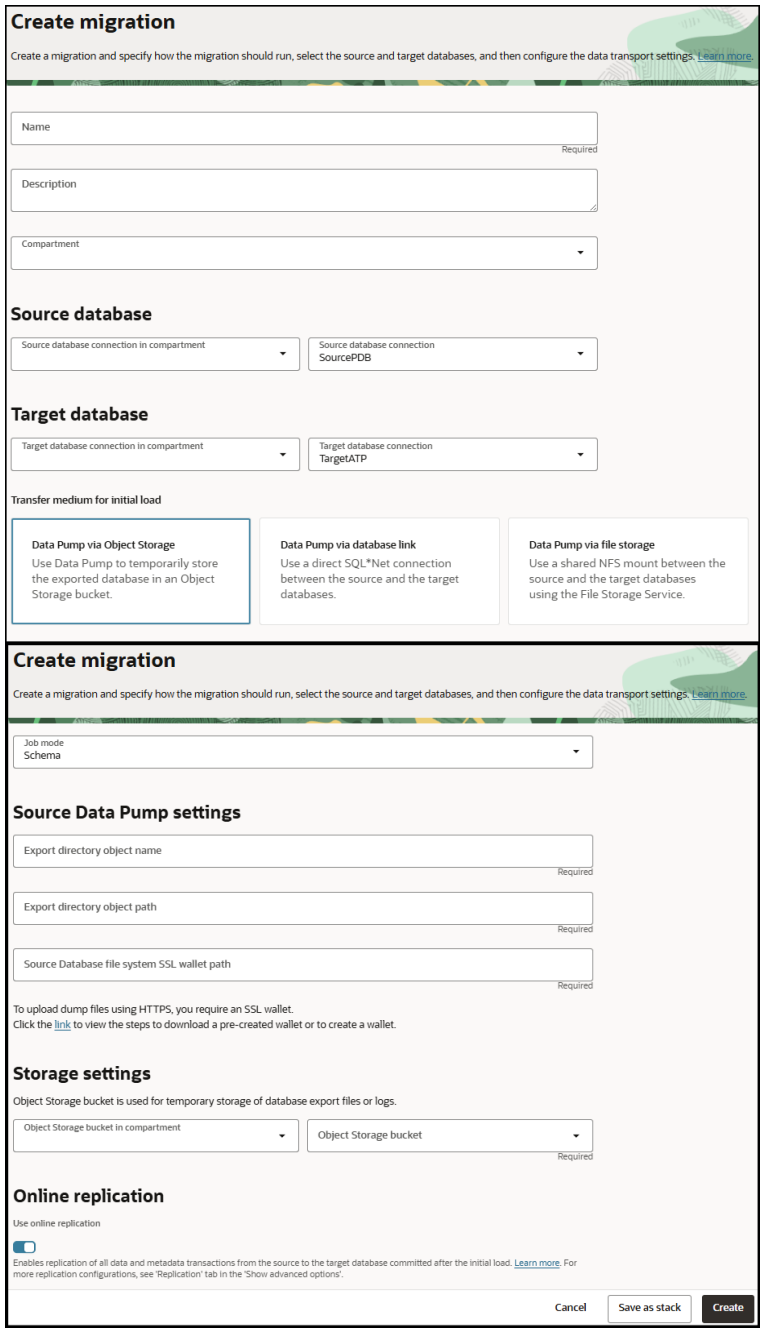
- Database connection in the compartment: **SourcePDB**
- Target Database: **TargetATP**

On the page **Migration Options**, fill in the following entries, otherwise leave defaults:

- In Initial Load: Datapump via Object Storage

- Export Directory Object:
  Name: **dumpdir**
  Path: **/u01/app/oracle/dumpdir**
- Source database file system SSL wallet path: **/u01/app/oracle/dumpdir/wallet**
- Object Storage Bucket: **DMSStorage**
- **Check** Use Online Replication

Click **Create**



Task 14 – Validate Migration

In this step you will validate a migration prior to running it. It will check the connections and settings for the source and target. Cloud premigration advisor (CPAT) will look for source and target incompatibilities.

In the OCI Console Menu, go to Migration & Disaster Recovery > Database Migration > Migrations.

Select TestMigration.

If the migration is still being created, wait until Lifecycle State is Active.

Press **Validate** button

Click on the **Jobs** tab

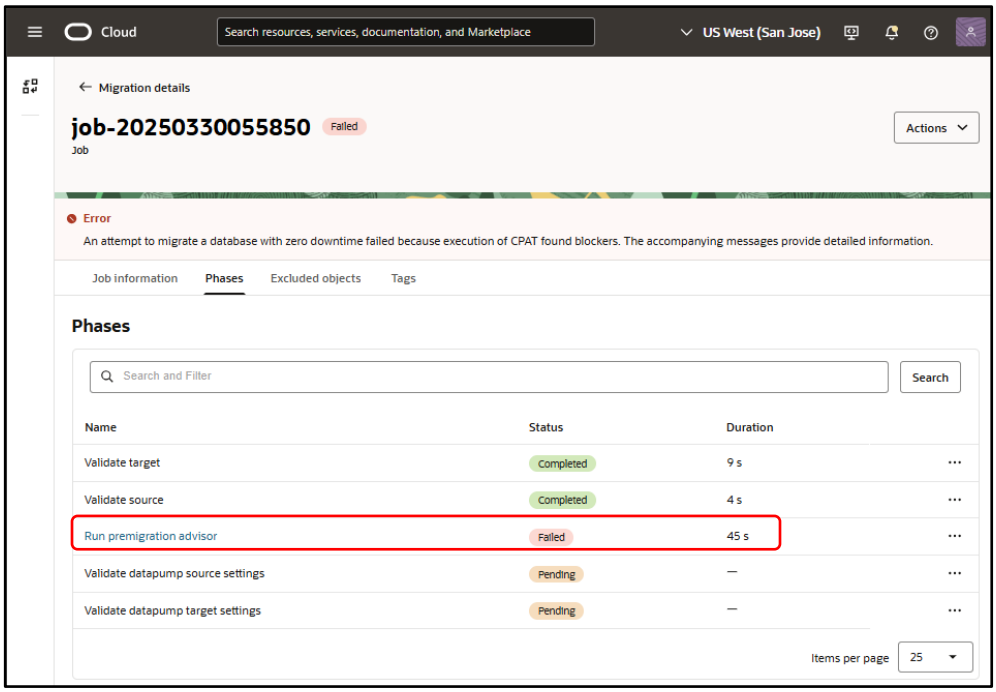
Click on most recent Evaluation Job

Click on the **Phases** tab

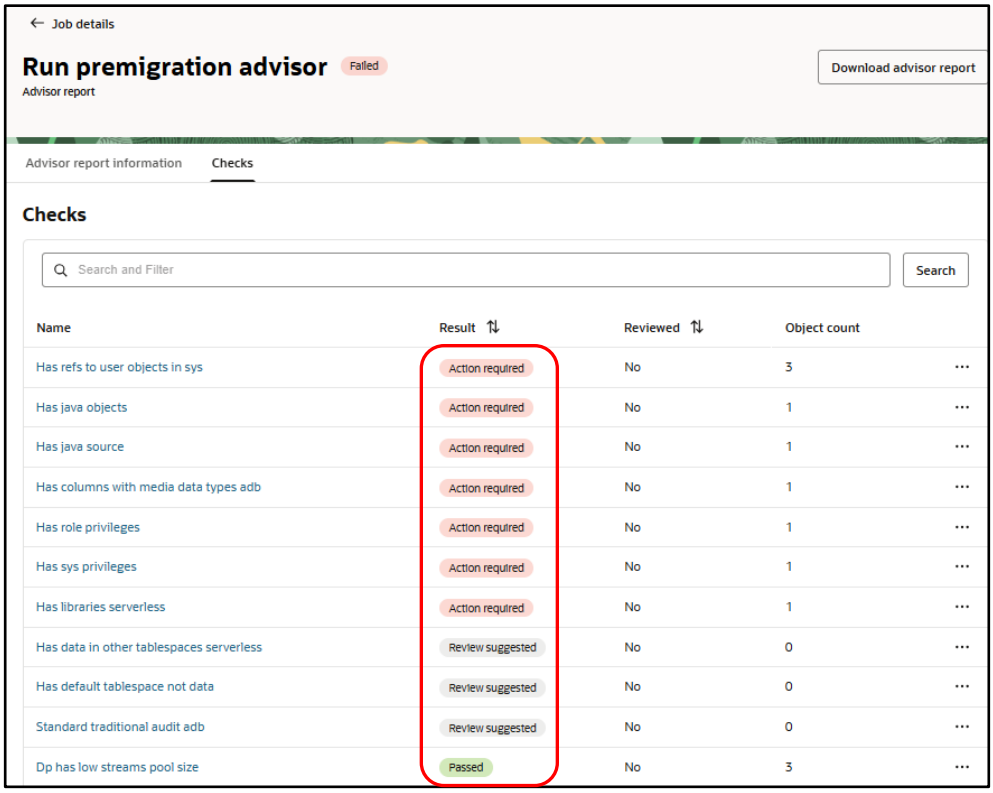
Phases will be shown, and status will be updated as phases are completed. It can take 2 minutes before the first phase is shown.

If a phase has failed, it will show with status **Failed**. In this case press Actions>**Download Log** to learn more about the reason of failure.

Click **Run premigration advisor** phase name to open the Validation premigration advisor detail page (You should not find issues in this exercise but below lines would walk you thru an event when the phase fails). From this page you can download the CPAT report, view the report statistics, and drill down in the Checks list as shown:

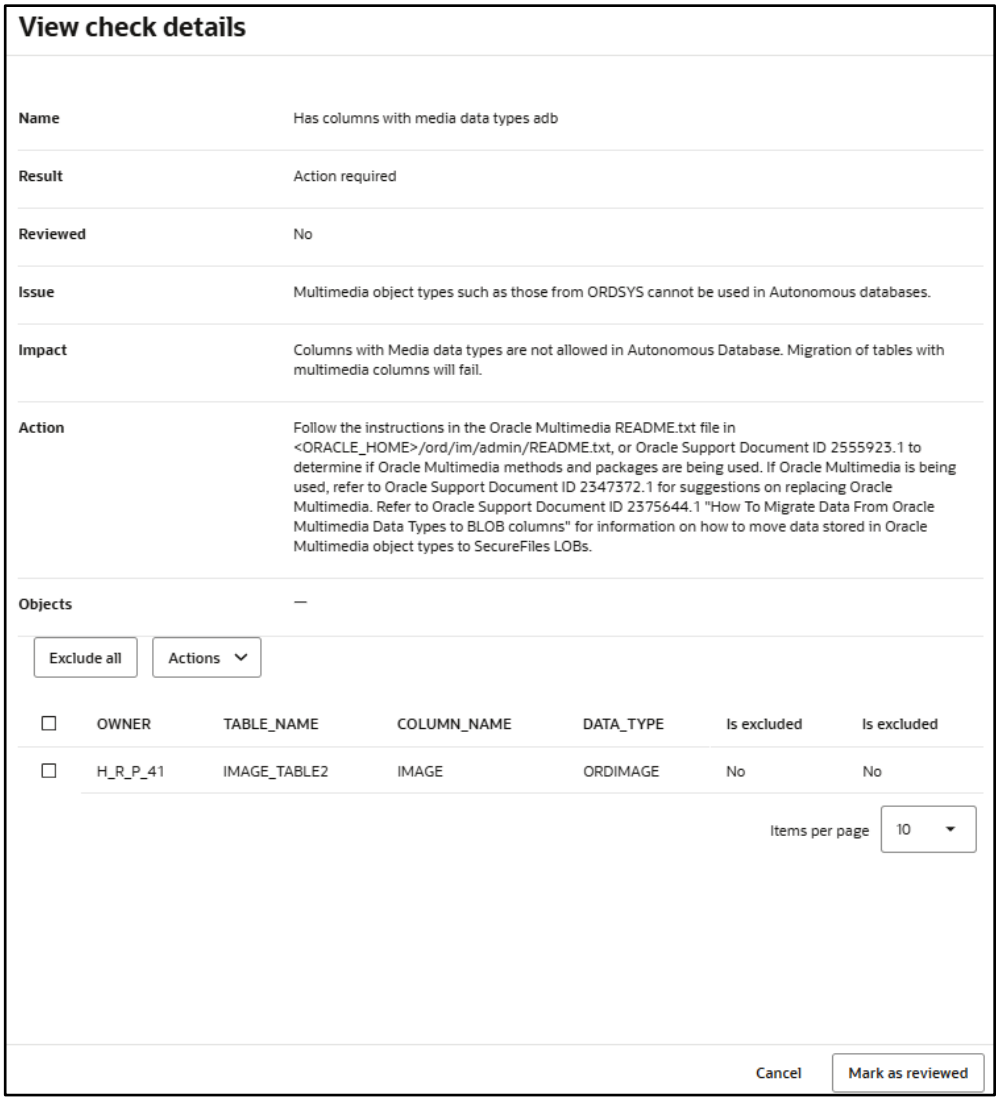


You can still download the advisor report as a text file, but now you can also navigate through the different checks. The summary view is displayed as follows:

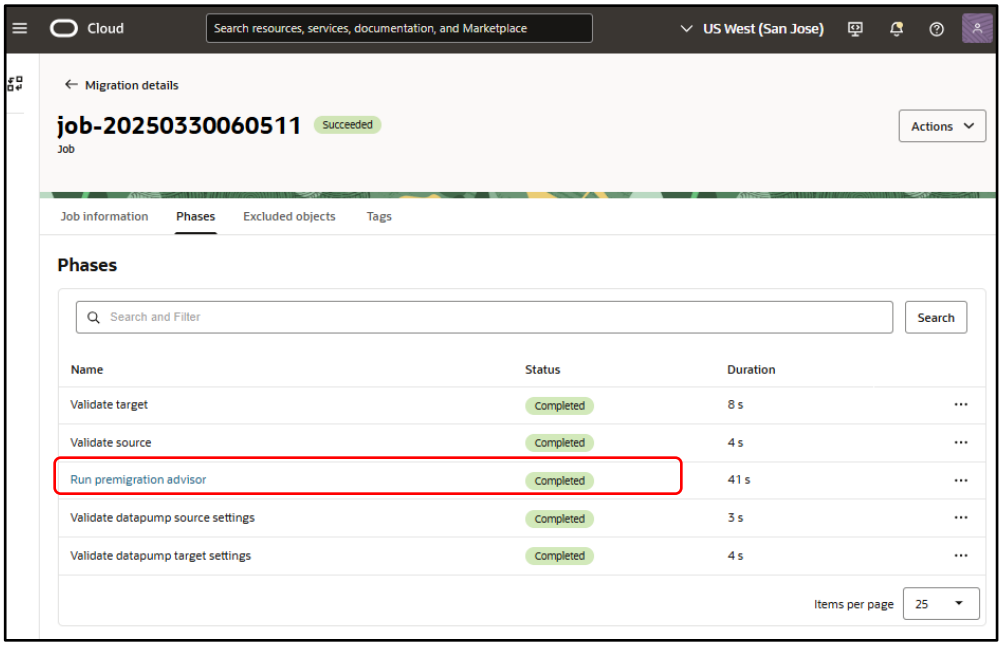


You can click a check name in the list to display details about a specific check from the CPAT report. You can mark a check as **Reviewed** or **Unreviewed**, this state is only for your convenience to track each check. For certain checks, CPAT generates a remedial script on the file system of the source database server. You can run the script on the source database to resolve the issue identified by the check. The checks page will also let you filter by this state (left side of screen):

The **View check details** panel is displayed as follows:



Once you have cleared all “Action Required” checks then the validation Job can be run again. Repeat the process until **Validate premigration advisor** phase completes with no error as shown:



Task 15 – Run Migration

After successful validation, a Migration can be run to perform the data transfer.

In the OCI Console Menu, go to Migration & Disaster Recovery > Database Migration > Migrations.

Select TestMigration.

Press Start to begin the migration.

The Start Migration dialog is shown. Select the phase Monitor replication Lag in the require user input after list, this will cause the replication to run continuously until the migration is resumed.
Press Start to begin the Migration.

Click on the **Jobs** tab

Click on the most recent migration Job

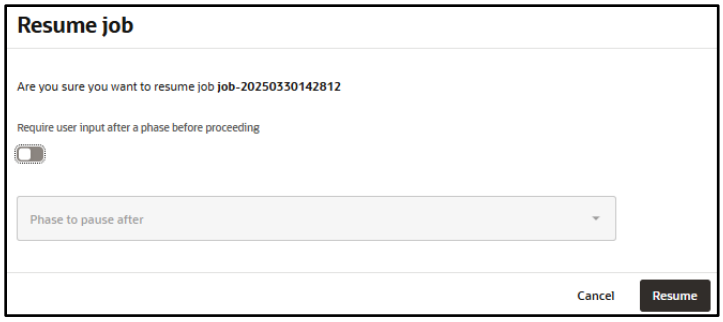
Click on the **Phases** tab

Job phases are updated as the migration progresses

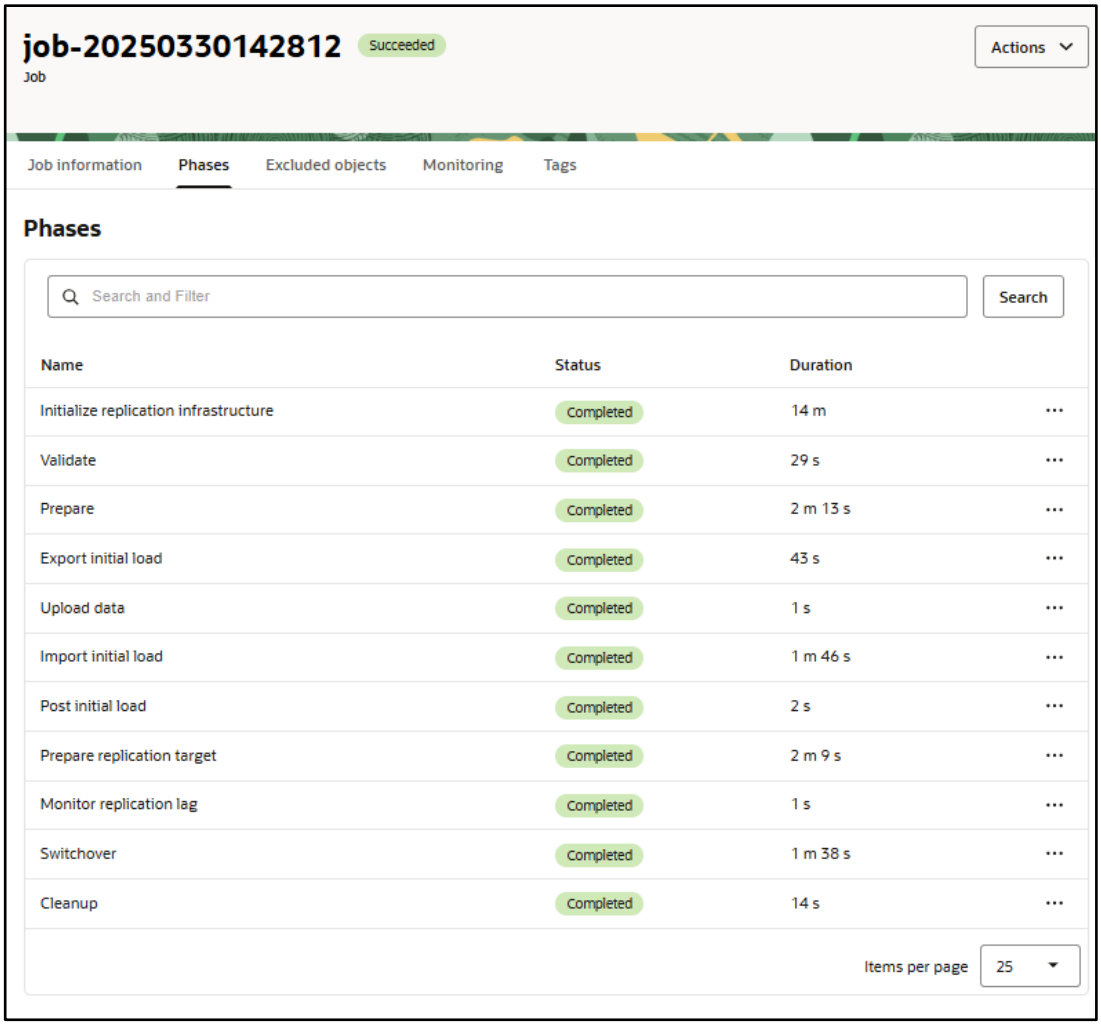
When the migration has reached the state to wait for user input, the migration job changes to **Waiting** state. This is the point where a migration user would stop the source application so that no more transactions are applied to the source DB. You can now press **Resume** on the job to complete replication.

In the Resume Job dialog, chose the **Switchover** phase and press **Resume**. The Switchover phase will gracefully stop replication and signal the target application to initiate transactions to the target DB. Find more information about the switchover phase in our documentation.

After Job resumes and waits after Switchover phase, press Resume. Select the last phase **Cleanup** and press **Resume**.



The migration runs the final cleanup phases and shows as Succeeded when finished.



Your migration is now completed.!

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