

# Oracle® Cloud

## Exadata Cloud Infrastructure Migration Automation Utility



G45773-01  
November 2025



Oracle Cloud Exadata Cloud Infrastructure Migration Automation Utility,

G45773-01

Copyright © 2025, Oracle and/or its affiliates.

Primary Authors: Douglas Williams, Nirmal Kumar, Jean-Francois VERRIER

Contributing Authors: Giridhar Reddy Sangala

Contributors: Jigar Pathak, Lingam Krishna Mohan, Bob Thome, Prakash Choudhary, Atharva Kitkaru, Ramasubramanian Athmanathan, Sanjay Singh

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

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

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

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

|          |                                                                                                                             |    |
|----------|-----------------------------------------------------------------------------------------------------------------------------|----|
| <b>1</b> | <b>Introduction, Prerequisites, and Preparation for Using the Exadata Cloud Infrastructure Migration Automation Utility</b> |    |
|          | Introduction to the Exadata Cloud Infrastructure Migration Automation Utility                                               | 1  |
|          | Prerequisites for Using the Exadata Cloud Infrastructure Migration Automation Utility                                       | 8  |
|          | Preparing to Use the Exadata Cloud Infrastructure Migration Automation Utility                                              | 9  |
| <b>2</b> | <b>Exadata Cloud Infrastructure Migration Automation Utility Functionality</b>                                              |    |
|          | Exadata Cloud Infrastructure Migration Automation Utility Operations                                                        | 1  |
|          | Exadata Cloud Infrastructure Migration Automation Utility Directory Structure                                               | 3  |
|          | Input template file format                                                                                                  | 3  |
|          | Generate input templates                                                                                                    | 3  |
|          | Generate input templates for Infrastructure, VM cluster(s) and database(s)                                                  | 4  |
|          | Generate input template for Infrastructure only                                                                             | 9  |
|          | Generate input template for VM cluster(s) only                                                                              | 10 |
|          | Generate input template for database(s) only                                                                                | 11 |
|          | Generate patches file                                                                                                       | 11 |
|          | Validate and dry run                                                                                                        | 13 |
|          | Create Exadata Infrastructure                                                                                               | 14 |
|          | Create VM cluster(s)                                                                                                        | 15 |
|          | Create database dbHome software image                                                                                       | 16 |
|          | Migrate Exadata Infrastructure                                                                                              | 18 |
|          | Migrate VM cluster(s)                                                                                                       | 20 |
|          | Migrate DBHome(s)                                                                                                           | 22 |
|          | Database wallet operations                                                                                                  | 24 |
|          | Option 1: exacloudmigration generate_dbwallet                                                                               | 25 |
|          | Option 1: exacloudmigration edit_dbwallet                                                                                   | 26 |
|          | Option 1: exacloudmigration list_wallet_dbs                                                                                 | 26 |
|          | Option 2: ecmasshutility to generate database password wallet                                                               | 27 |
|          | Option 2: ecmasshutility generate_dbwallet                                                                                  | 28 |
|          | Option 2: ecmasshutility edit_dbwallet                                                                                      | 29 |
|          | Option 2: ecmasshutility list_wallet_dbs                                                                                    | 30 |
|          | Migrate database(s)                                                                                                         | 30 |

|                                                                   |    |
|-------------------------------------------------------------------|----|
| Enable Data Guard association between source and target databases | 31 |
| Migrate database switchover                                       | 33 |
| Migrate database failover                                         | 35 |
| Database switchover                                               | 38 |
| Database failover                                                 | 39 |
| Reinstate standby                                                 | 41 |
| Delete standby                                                    | 42 |
| List all Data Guard associations for given VM cluster             | 44 |
| List VM cluster Networks                                          | 45 |
| List the jobs history                                             | 45 |
| Get job details and status                                        | 46 |
| Create Database from existing database backup                     | 46 |

### 3 Post Migration

---

### 4 Logging and Troubleshooting

---

|                                                                |   |
|----------------------------------------------------------------|---|
| Exadata Cloud Infrastructure Migration Automation Utility logs | 1 |
| Review the logs                                                | 1 |
| Upload the logs to Service Request                             | 2 |
| Create a Service Request                                       | 2 |
| Upload Logs to a Service Request                               | 3 |
| Troubleshooting and diagnostics                                | 4 |

### 5 Known Issues

---

|                                                                 |   |
|-----------------------------------------------------------------|---|
| Creation of Data Guard association fails                        | 1 |
| Create database from backup or Data Guard association fails     | 2 |
| When to configure manual Data Guard                             | 2 |
| Does the utility support cross-tenancy migration?               | 3 |
| Cluster verification failures due to hostname resolution issues | 3 |

# 1

## Introduction, Prerequisites, and Preparation for Using the Exadata Cloud Infrastructure Migration Automation Utility

The Exadata Cloud Infrastructure Migration Automation Utility simplifies and automates migration between Exadata Cloud Infrastructure environments.

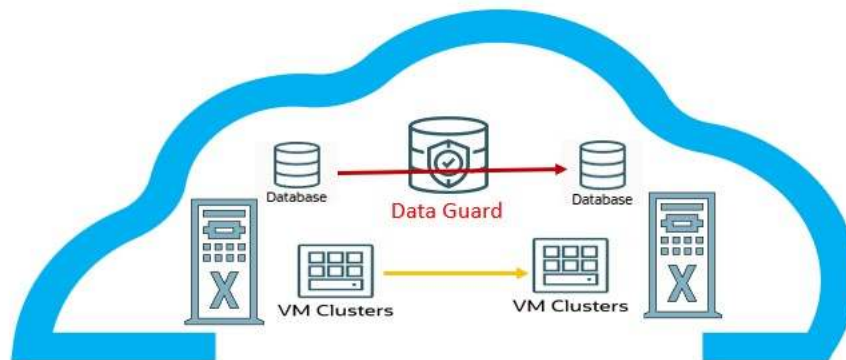
The utility will discover and capture the configuration of the current environment and create input templates for the Oracle Exadata Cloud Infrastructure, cloud VM clusters, and databases. These templates can be modified as required to accommodate the intended migration target. The utility will validate the template, checking for any potential problems in the setup and configuration. Upon successful validation, the utility will provision a new Oracle Exadata Cloud Infrastructure, create desired cloud VM cluster, and migrate the databases. Databases are migrated with no downtime by instantiating standby databases on the target infrastructure and switching over to the standby on the target infrastructure.

The utility simplifies hardware refreshes, shape transitions, and cross-service migrations while ensuring accuracy and full OCI auditability.

- [Introduction to the Exadata Cloud Infrastructure Migration Automation Utility](#)
- [Prerequisites for Using the Exadata Cloud Infrastructure Migration Automation Utility](#)
- [Preparing to Use the Exadata Cloud Infrastructure Migration Automation Utility](#)

## Introduction to the Exadata Cloud Infrastructure Migration Automation Utility

**Figure 1-1 Exadata Cloud Infrastructure Migration Automation Utility**



The Exadata Cloud Infrastructure Migration Automation Utility is a lightweight tool that can be installed on any Linux client machine with access to the OCI Control Plane.

Using the specified source Infrastructure OCID, the utility will discover and capture the configuration of the current environment and create input templates for the Infrastructure, VM Cluster(s) and database(s). These templates can be modified as required to accommodate the intended migration target. The utility will validate the template checking for any potential problems in the setup and configuration. Upon successful validation, the utility will provision a new Exadata Infrastructure, create desired VM Cluster(s) and migrate the databases. Databases are migrated with no downtime by instantiating standby databases on the target infrastructure and switching over to the standby on the target infrastructure.

### When to Use the Utility

This utility is particularly useful in scenarios such as:

- Performing hardware refreshes
- Migrating fixed-shape systems to elastic systems (for example, X7 to X11M, X8 to X11M, or Base System to X11M)
- Migrating between elastic systems
- Migrating between fixed-shape Base Systems
- Redistributing databases across clusters
- Retiring old systems

### Key Capabilities

- The Exadata Cloud Infrastructure Migration Automation Utility supports:
  - Migration between ExaDB-D systems
  - Migration between ExaDB-C@C systems
  - Migration from ExaDB-C@C to ExaDB-D systems
- Supported Target Exadata Infrastructure models:
  - For ExaDB-D : Base system, X8M, X9M, and X11M
  - For ExaDB-C@C: X8M, X9M, X10M, and X11M

### Key Benefits

- Validates configurations before execution to prevent migration errors
- Simplifies migration of multiple VM clusters and hundreds of databases with end-to-end automation
- Supports parallel migrations, significantly reducing the time to move large database fleets
- Provides fine-grained control to migrate entire infrastructures or selected components
- Uses standard OCI APIs for all actions, ensuring full auditing within OCI
- Enables monitoring of progress for all migration operations

Table 1-1    Functionality of Exadata Cloud Infrastructure Migration Automation Utility

| Functionality Description                                                                                                                       | Migration from ExaDB-D to ExaDB-D | Migration from ExaDB-C@C to ExaDB-C@C | Migration from ExaDB-C@C to ExaDB-D |
|-------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|---------------------------------------|-------------------------------------|
| Generates and validates the input templates and database wallet file required for migrating the Infrastructure, VM cluster(s), and database(s). | Yes                               | Yes                                   | Yes                                 |

① Note

Generates and validates the input templates and database wallet file required for migrating the Infrastructure, VM cluster(s), and database(s).

Table 1-1 (Cont.) Functionality of Exadata Cloud Infrastructure Migration Automation Utility

| Functionality Description | Migration from ExaDB-D to ExaDB-D | Migration from ExaDB-C@C to ExaDB-C@C | Migration from ExaDB-C@C to ExaDB-D                           |
|---------------------------|-----------------------------------|---------------------------------------|---------------------------------------------------------------|
|                           |                                   |                                       | o<br>t<br>a<br>p<br>p<br>l<br>i<br>c<br>a<br>b<br>l<br>e<br>. |

Table 1-1 (Cont.) Functionality of Exadata Cloud Infrastructure Migration Automation Utility

| Functionality Description                                                                                                                                                                                                          | Migration from ExaDB-D to ExaDB-D | Migration from ExaDB-C@C to ExaDB-C@C | Migration from ExaDB-C@C to ExaDB-D |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|---------------------------------------|-------------------------------------|
| Migrates the Exadata Infrastructure to a target region or compartment using the specified source infrastructure details. This includes creating the Infrastructure, VM cluster(s), Application VIPs, and Database Homes (dbHomes). | Yes                               | No                                    | Yes                                 |

**Note** Migration of Application VIPs is not applicable.

Table 1-1 (Cont.) Functionality of Exadata Cloud Infrastructure Migration Automation Utility

| Functionality Description                                                                                                                                                                    | Migration from ExaDB-D to ExaDB-D | Migration from ExaDB-C@C to ExaDB-C@C | Migration from ExaDB-C@C to ExaDB-D |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|---------------------------------------|-------------------------------------|
| Migrates VM cluster(s) to the target infrastructure using the specified source VM cluster details. This includes creating the VM cluster(s), Application VIPs, and Database Homes (dbHomes). | Yes                               | Yes                                   | Yes                                 |

**Note** Migration of Application VIPs is not applicable.

**Table 1-1 (Cont.) Functionality of Exadata Cloud Infrastructure Migration Automation Utility**

| Functionality Description                                                                                                                                                                                                                               | Migration from ExaDB-D to ExaDB-D | Migration from ExaDB-C@C to ExaDB-C@C | Migration from ExaDB-C@C to ExaDB-D |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|---------------------------------------|-------------------------------------|
| Migrates Database Homes (dbHomes) from the source VM cluster to the specified target VM cluster.                                                                                                                                                        | Yes                               | Yes                                   | Yes                                 |
| Migrates database(s) from the source VM cluster to the specified target VM cluster. The process includes enabling Data Guard association between the source and target databases and provides the option to perform a switchover or failover operation. | Yes                               | Yes                                   | No                                  |
| Creates Exadata Infrastructure in the target region or compartment from the specified source infrastructure details.                                                                                                                                    | Yes                               | No                                    | Yes                                 |
| Creates VM cluster(s) in the target infrastructure from the specified source VM cluster(s) details.                                                                                                                                                     | Yes                               | Yes                                   | Yes                                 |
| Creates Database Homes (dbHomes) software image(s) for the specified database(s).                                                                                                                                                                       | Yes                               | Yes                                   | Yes                                 |
| Reinstates the standby database.                                                                                                                                                                                                                        | Yes                               | Yes                                   | No                                  |
| Deletes the standby database.                                                                                                                                                                                                                           | Yes                               | Yes                                   | No                                  |
| Lists all Data Guard associations.                                                                                                                                                                                                                      | Yes                               | Yes                                   | No                                  |
| Lists and fetches migration job status and details.                                                                                                                                                                                                     | Yes                               | Yes                                   | Yes                                 |
| Creates database(s) from the existing database backup(s).                                                                                                                                                                                               | Yes                               | No                                    | No                                  |

For more information on supported systems, shape configuration options, database editions and versions for Exadata Cloud Infrastructure, see:

- ExaDB-D: [System and Shape Configuration Options](#)
- ExaDB-C@C: [System and Shape Configuration Options](#)

- Supported Database Edition and Versions for Exadata Cloud Infrastructure ExaDB-D and ExaDB-C@C:
  - Exadata Database Service Software Versions ([Doc ID 2333222.1](#))
  - Exadata Database Machine and Exadata Storage Server Supported Versions ([Doc ID 888828.1](#))
- ExaDB-D: [Scaling Options](#)
- ExaDB-D: [Overview of X8M, X9M, and X11M Scalable Exadata Infrastructure](#)
- ExaDB-C@C: [Overview of Elastic Storage Expansion](#)

For more information on managing ExaDB-D and ExaDB-C@C, see:

- ExaDB-D: [Manage Exadata Cloud Infrastructure](#)
- ExaDB-C@C: [Manage Oracle Exadata Database Service on Cloud@Customer Infrastructure](#)

## Prerequisites for Using the Exadata Cloud Infrastructure Migration Automation Utility

- Create or use an existing user.  
For more information on how to create a new user, see [Creating a User](#).
- Add the user to a group  
For more information, see [Adding a User to a Group](#).
- Assign the following IAM policies by specifying the user's group name and compartment name.

```
Allow group '<group name>' to use database-family in compartment
<compartment name>
Allow group '<group name>' to use virtual-network-family in compartment
<compartment name>
Allow group '<group name>' to manage db-nodes in compartment <compartment
name>
Allow group '<group name>' to manage db-homes in compartment <compartment
name>
Allow group '<group name>' to manage databases in compartment <compartment
name>
Allow group '<group name>' to manage database-software-image in
compartment <compartment name>
Allow group '<group name>' to use compartments in compartment <compartment
name>
Allow group '<group name>' to use resource-availability in compartment
<compartment name>
```

Additionally, assign the following policies to ExaDB-D only:

```
Allow group '<group name>' to manage cloud-exadata-infrastructures in
compartment <compartment name>
Allow group '<group name>' to manage cloud-vmclusters in compartment
<compartment name>
```

Additionally, assign the following policies to ExaDB-C@C only

Allow group '<group name>' to manage exadata-infrastructures in compartment <compartment name>  
Allow group '<group name>' to manage vmclusters in compartment <compartment name>

**Note**

If the target is in a different compartment from the source, assign the above policies to the target compartment as well.

For more information, see [Create Identity and Access Management \(IAM\) Groups and Policies for IAM Users](#).

4. To use the OCI SDK for Java, configure the key pair for the user. This key pair is used to authenticate API requests.  
For more information, see:
  - [Generating an API Signing Key \(Console\)](#)
  - [SDK and CLI Configuration File](#)
5. Setup Virtual Cloud Network (VCN).
  - a. The existing VCN may be used if the target is in the same region or compartment as the source.
  - b. If the requirement is to use a new VCN or the target is going to be in cross region, then a new VCN setup is required.
    - i. The new VCN must have a sufficient number of IP addresses available for creating the VM cluster.  
For more information, see [Requirements for IP Address Space](#).
    - ii. Ensure that the new VCN has required security lists or rules configured as same as the source VCN.
    - iii. For setting up Oracle Data Guard, ensure that the connectivity between the source and the target VCNs is established and if a new VCN is to be used, then ensure that the peering is done for the two VCNs allowing the Data Guard association communication.
6. Request Service Limits and ensure that sufficient resources are available.  
For more information, see [Request a Service Limit Increase](#).
7. This utility can be run from any Linux x86-64 machine that can make API calls.  
Supported client machine OS version: Oracle Linux 7, Oracle Linux 8, and Oracle Linux 9.
8. Port requirement  
To make API calls to OCI endpoints ensure that the port 443 (egress) is opened for outgoing connections from the client machine.

## Preparing to Use the Exadata Cloud Infrastructure Migration Automation Utility

1. Download the p37344128\_240000\_Linux-x86-64.zip file for the platform Linux-x86-64 from the [My Oracle Support portal](#).

**Note**

As a best practice, always download the latest version of the patch.

2. Copy the patch file to the client machine.  
Log in to the client machine as the opc user.

**Note**

For example, in this procedure, the username opc is used as the example user in subsequent steps.

Unzip the utility patch:

```
/home/opc> unzip p37344128_240000_Linux-x86-64.zip
```

After the utility is unzipped, the following files are created:

```
/home/opc> ls -lrth
-rwxr-xr-x opc opc 154M Mar 12 18:13 exacloudmigration
-rw-r--r-- opc opc 13K Mar 12 18:28 README.txt
-rw-r--r-- opc opc 37M Mar 12 21:21 ecmaasshutility
```

3. On the client machine, if the resource access requires a proxy, set the following parameters. Otherwise, skip this step.

For proxy without user authentication, set:

```
/home/opc> export ECMA_PROXY_HOST=<Proxy host IP address>
```

```
/home/opc> export ECMA_PROXY_PORT=<Proxy port>
```

For example:

```
/home/opc> export ECMA_PROXY_HOST=a.b.c.d
```

```
/home/opc> export ECMA_PROXY_PORT=80
```

For proxy with user authentication, additionally set:

```
/home/opc> ECMA_PROXY_USER=<Proxy user name>
```

```
/home/opc> ECMA_PROXY_PASSWORD=<Proxy user password>
```

For example:

```
/home/opc> ECMA_PROXY_USER=proxyuser
```

```
/home/opc> ECMA_PROXY_PASSWORD=*****
```

4. Collect the following information to use as input when running the utility in the next steps:
  - a. Log in to the Oracle Cloud Console.
  - b. Obtain the source Exadata Infrastructure OCID to be used for migration or for creating the Exadata Infrastructure and Exadata VM clusters.

In the Console, go to:

Dashboard → Navigation Menu → Oracle AI Database → Exadata Database Service on Dedicated Infrastructure → Exadata Infrastructures → Infrastructure name → Exadata Infrastructure Details → Copy OCID

- c. Obtain the source VM Cluster OCID for migrating databases.

In the Console, go to:

Dashboard → Navigation Menu → Oracle AI Database → Exadata Database Service on Dedicated Infrastructure → Exadata VM Clusters → Cluster name → VM Cluster Information → Copy OCID

# 2

## Exadata Cloud Infrastructure Migration Automation Utility Functionality

- [Exadata Cloud Infrastructure Migration Automation Utility Operations](#)
- [Exadata Cloud Infrastructure Migration Automation Utility Directory Structure](#)
- [Input template file format](#)
- [Generate input templates](#)
- [Generate patches file](#)
- [Validate and dry run](#)
- [Create Exadata Infrastructure](#)
- [Create VM cluster\(s\)](#)
- [Create database dbHome software image](#)
- [Migrate Exadata Infrastructure](#)
- [Migrate VM cluster\(s\)](#)
- [Migrate DBHome\(s\)](#)
- [Database wallet operations](#)
- [Migrate database\(s\)](#)
- [Reinstate standby](#)
- [Delete standby](#)
- [List all Data Guard associations for given VM cluster](#)
- [List VM cluster Networks](#)
- [List the jobs history](#)
- [Get job details and status](#)
- [Create Database from existing database backup](#)

## Exadata Cloud Infrastructure Migration Automation Utility Operations

### Syntax

```
exacloudmigration [-hV] [command]
```

**Table 2-1 Exadata Cloud Infrastructure Migration Automation Utility Options**

| Option        | Description                                                  |
|---------------|--------------------------------------------------------------|
| -h, --help    | Displays help information about the command and its options. |
| -V, --version | Displays the version of the utility.                         |

**Table 2-2 Exadata Cloud Infrastructure Migration Automation Utility Commands**

| Command              | Description                                                                                                                                                    |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| generate_template    | Generates input templates for the Exadata Cloud Infrastructure Migration Automation Utility.<br>Options: all, infra, vmcluster, database                       |
| generate_dbwallet    | Generates database wallet file.                                                                                                                                |
| edit_dbwallet        | Adds or updates password for a database in the wallet file.                                                                                                    |
| list_wallet_dbs      | Displays the list of database names with passwords stored in the wallet file.                                                                                  |
| create_infra         | Creates Exadata infrastructure in the target region/compartment from the given source infrastructure details.                                                  |
| create_vmcluster     | Creates VM cluster(s) in target infrastructure from the given source VM cluster(s) details.                                                                    |
| create_dbhomeimage   | Creates dbHome software image(s) for the given database(s).                                                                                                    |
| migrate_infra        | Migrates Exadata infrastructure from the given source infrastructure details.<br>Includes creation of Infrastructure, VM cluster(s), and database dbHome(s).   |
| migrate_vmcluster    | Migrates VM cluster(s) to the target infrastructure from the given source VM cluster(s) details.<br>Includes creation of VM cluster(s) and database dbHome(s). |
| migrate_dbhomes      | Migrates dbHome(s) from the source VM cluster to the target VM cluster.                                                                                        |
| migrate_database     | Migrates database(s) from the source VM cluster to the target VM cluster.<br>Options: dgassociation, switchover, failover                                      |
| switchover           | Performs a role transition, making the standby database the primary database, while the old primary database becomes the standby.                              |
| failover             | Performs a role transition, making the standby database the primary database, while the old primary database becomes the disabled standby.                     |
| reinstate            | Makes disabled standby database as standby database.                                                                                                           |
| delete_standby       | Deletes the standby database(s).                                                                                                                               |
| list_dg_associations | Lists all Data Guard associations for a given VM cluster.                                                                                                      |

**Table 2-2 (Cont.) Exadata Cloud Infrastructure Migration Automation Utility Commands**

| Command              | Description                                                       |
|----------------------|-------------------------------------------------------------------|
| list_vm_networks     | Lists all VM cluster networks for a given Exadata infrastructure. |
| list_jobs            | Lists the job history.                                            |
| get_job_status       | Displays details and status of the job.                           |
| create_db_frombackup | Creates database(s) from the existing database backup(s).         |

## Exadata Cloud Infrastructure Migration Automation Utility Directory Structure

For example, utility is unzipped into the /home/opc directory

**Table 2-3 Exadata Cloud Infrastructure Migration Automation Utility Directory Structure**

| Directory Path                                                                        | Description                                                                |
|---------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| /home/opc/inputs/                                                                     | Directory where the template files are generated.                          |
| /home/opc/inputs/<br>infra_<source_infra_name>/                                       | Contains the generated Infrastructure and VM cluster input template files. |
| /home/opc/inputs/<br>infra_<source_infra_name>/<br>vmcluster_<source_vmcluster_name>/ | Contains the generated Database input template file.                       |
| /home/opc/exacloudmigration_data/<br>logs/                                            | Directory where job log files are generated.                               |

## Input template file format

Template file will contain two types of parameters:

- Global parameters between (START and END)
- And, resource specific parameters between (START and END) will contain the following parameters details:
  - Details of the Infrastructure
  - Details of each VM cluster
  - Details of each database

## Generate input templates

### Syntax

```
exacloudmigration generate_template [-hv] [command]
```

The Exadata Cloud Infrastructure Migration Automation Utility discovers and captures the configuration of the source environment as input templates for the Exadata Infrastructure, VM clusters, and databases. These templates are then used to create or migrate the infrastructure, VM clusters, and databases in the target environment.

**Table 2-4 Exadata Cloud Migration Automation Generate Template Options**

| Option        | Description                                                  |
|---------------|--------------------------------------------------------------|
| -h, --help    | Displays help information about the command and its options. |
| -V, --version | Displays the version of the utility.                         |

**Table 2-5 Exadata Cloud Migration Automation Generate Template Commands**

| Command   | Description                                                                            |
|-----------|----------------------------------------------------------------------------------------|
| all       | Generates input template for Infrastructure, VM cluster(s) and Data Guard association. |
| infra     | Generates input template for Exadata infrastructure.                                   |
| vmcluster | Generates input template for VM cluster(s)                                             |
| database  | Generates input template for database(s).                                              |

- [Generate input templates for Infrastructure, VM cluster\(s\) and database\(s\)](#)
- [Generate input template for Infrastructure only](#)
- [Generate input template for VM cluster\(s\) only](#)
- [Generate input template for database\(s\) only](#)

## Generate input templates for Infrastructure, VM cluster(s) and database(s)

### Syntax

```
exacloudmigration generate_template all [-hV] -t <targetType> -s
<sourceExadataInfraOCID> [-i <infraInputFileName>] [-v
<vmClusterInputFileName>] [-d <dbInputFileName>] [-p <profile>]
```

**Table 2-6 Exadata Cloud Migration Automation Generate Template All Commands**

| Command                      | Description                                                                         |
|------------------------------|-------------------------------------------------------------------------------------|
| -t, --targetType             | Target System Type (exadb, exadbcc).                                                |
| -s, --sourceExadataInfraOCID | Source Exadata Infrastructure OCID.                                                 |
| -i, --infraInputFileName     | Sets the infrastructure input template file name.<br>Default: infrastructure_inputs |
| -v, --vmClusterInputFileName | Sets the VM cluster input template file name.<br>Default: cloudvmcluster_inputs     |
| -d, --dbInputFileName        | Sets the database input template file name.<br>Default: database_inputs             |

**Table 2-6 (Cont.) Exadata Cloud Migration Automation Generate Template All Commands**

| Command       | Description                                                  |
|---------------|--------------------------------------------------------------|
| -p, --profile | OCI user profile name.<br>Default profile: DEFAULT           |
| -h, --help    | Displays help information about the command and its options. |
| -V, --version | Displays the version of the utility.                         |

Examples:

- To generate all input template files (Infrastructure, VM cluster(s) and database(s)) if the migration target is ExaDB-D:  
  

```
/home/opc>./exacloudmigration generate_template all -t exadbd -s <OCID>
```
- To generate all input template files (VM cluster(s) and database(s)) if the migration target is ExaDB-C@C:  
  

```
/home/opc>./exacloudmigration generate_template all -t exadbcc -s <OCID> -t exadbcc
```

**Sample contents of input template for ExaDB-D Infrastructure (/home/opc/inputs/infra\_phxref/infrastructure\_inputs)**

**Note**

phxref refers to the infrastructure name

```
!----- CLOUD-EXADATA-INFRASTRUCTURE-INPUTS -----!
## For details on these parameters refer readme which is shipped with this
utility
## Modify these values for new infrastructure creation with different
specifications
===== START: GLOBAL PARAMETERS =====
sourceProfile=DEFAULT
targetProfile=
sourceExadataInfraOCID=<OCID>
===== END: GLOBAL PARAMETERS =====

## The below values are retrieved based on the source infrastructure ocid.
## By default it will create new infra having same region, compartment,
availability domain, display name, exadata infra model, compute count and
storage count.
## Allowed values for exadataInfraModel: Exadata.X8M | Exadata.X9M |
Exadata.X11M | Exadata.Base.48 and default is Exadata.X11M
## Allowed values for databaseServerType: X11M (required only for creating
X11M infrastructure)
## Allowed values for storageServerType: X11M-HC (required only for creating
```

```
X11M infrastructure)
## computeCount and storageCount is required only for Elastic models
===== START: phxref =====
compartmentOCID=<OCID>
region=us-phoenix-1
displayName=phxref
availabilityDomain=OLou:PHX-AD-2
exadataInfraModel=Exadata.X11M
databaseServerType=X11M
computeCount=2
storageServerType=X11M-HC
storageCount=3
===== END: phxref =====
```

**Sample contents of input template for ExaDB-D VM cluster(s) (/home/opc/inputs/infra\_phxref/cloudvmcluster\_inputs)**

**Note**

phxref refers to the infrastructure name

```
!----- CLOUD-VMCLUSTER-INPUTS -----!
## For details on these parameters refer readme which is shipped with this
utility.
## Modify these values for new cloud VM cluster creation with different
specifications.
===== START: GLOBAL PARAMETERS =====
sourceProfile=DEFAULT
targetProfile=
sourceExadataInfraOCID=<OCID>
targetExadataInfraOCID=
===== END: GLOBAL PARAMETERS =====

## The below values are retrieved based on the source infrastructure ocid.
## By default it will create new cluster having values same as source VM
cluster.
===== START: vmcluster1 =====
create=no
createAppVIP=no
sourceVmClusterOCID=<OCID>
compartmentOCID=<OCID>
displayName=examig
giVersion=19.0.0.0
exadataImageVersion=23.1.13.0.0.240410.1
dbServers=dbServer-1, dbServer-2
cpuModel=ocpu
cpuCountPerVM=6
memorySizeInGBsPerVM=720
u02SizeInGBsPerVM=175
fileSystemSizesInGBsPerVM={"/=15, /var=5, /home=4, /tmp=3, /u01=20, /var/
log=18, /var/log/audit=3}
dataStorageSizeInTBs=180
dataStoragePercentage=80
```

```

clientSubnetOCID=<OCID>
backupSubnetOCID=<OCID>
hostnamePrefix=examig
===== END: vmcluster1 =====

```

**Sample contents of input template for ExaDB-C@C VM cluster(s) (/home/opc/inputs/infra\_phxref/cloudvmcluster\_inputs)**

**Note**

phxref refers to the infrastructure name

```

!----- VMCLUSTER-INPUTS -----!
## For details on these parameters refer readme which is shipped with this
utility.
## Modify these values for new ExaDb-C@C VM cluster creation with different
specifications.
===== START: GLOBAL PARAMETERS =====
sourceProfile=DEFAULT
targetProfile=
sourceExadataInfraOCID=<OCID>
targetExadataInfraOCID=
===== END: GLOBAL PARAMETERS =====

## The below values are retrieved based on the source infrastructure ocid.
## By default it will create new cluster having values same as source VM
cluster.
===== START: vmcluster1 =====
create=no
sourceVmClusterOCID=<OCID>
compartmentOCID=<OCID>
displayName=examig
giVersion=19.0.0.0
exadataImageVersion=23.1.13.0.0.240410.1
dbServers=dbServer-1, dbServer-2
cpuModel=ocpu
cpuCountPerVM=6
memorySizeInGBsPerVM=720
u02SizeInGBsPerVM=175
fileSystemSizesInGBsPerVM={"/=15, /var=5, /home=4, /tmp=3, /u01=20, /var/
log=18, /var/log/audit=3}
dataStorageSizeInTBs=180
dataStoragePercentage=80
vmClusterNetworkOCID=
===== END: vmcluster1 =====

```

**Sample contents of input template for ExaDB-D database(s) (/home/opc/inputs/infra\_phxref/vmcluster\_vmcluster1/database\_inputs)****Note**

phxref refers to the infrastructure name and vmcluster1 refers to the VM cluster name

```
!----- DATABASE-INPUTS -----!
## For details on these parameters refer readme which is shipped with this
utility.
## Modify these values for new dg association creation with different
specifications.
===== START: GLOBAL PARAMETERS =====
sourceProfile=DEFAULT
targetProfile=
sourceVmClusterOCID=<OCID>
targetVmClusterOCID=
===== END: GLOBAL PARAMETERS =====

## The below values are retrieved based on the source infrastructure or VM
cluster ocid.
## By default it will create dg association having default values.
## Allowed values for dataGuardType: dataGuard | activeDataGuard and default
is dataGuard
## Multidg supports only MAXIMUM PERFORMANCE for protection mode
===== START: DBNAME1 =====
create=no
sourceDatabaseOCID=<OCID>
dbName=DBNAME1
dataGuardType=dataGuard
dbUniqueName=
oracleSidPrefix=DBNAME1
standbyDatabaseOCID=
===== END: DBNAME1 =====

===== START: DBNAME2 =====
create=no
sourceDatabaseOCID=<OCID>
dbName=DBNAME2
dataGuardType=dataGuard
dbUniqueName=
oracleSidPrefix=DBNAME2
standbyDatabaseOCID=
===== END: DBNAME2 =====
```

Sample contents of input template for ExaDB-C@C database(s) (/home/opc/inputs/infra\_phxref/vmcluster\_vmcluster1/database\_inputs)

### ① Note

phxref refers to the infrastructure name and vmcluster1 refers to the VM cluster name

```
!----- DATABASE-INPUTS -----!
## For details on these parameters refer readme which is shipped with this
utility.
## Modify these values for new dg association creation with different
specifications.
===== START: GLOBAL PARAMETERS =====
sourceProfile=DEFAULT
targetProfile=
sourceVmClusterOCID=<OCID>
targetVmClusterOCID=
===== END: GLOBAL PARAMETERS =====

## The below values are retrieved based on the source infrastructure or VM
cluster ocid.
## By default it will create dg association having default values.
## Allowed values for dataGuardType: dataGuard | activeDataGuard and default
is dataGuard
## Multidg supports only MAXIMUM PERFORMANCE for protection mode

===== START: DBNAME1 =====
create=no
sourceDatabaseOCID=<OCID>
dbName=DBNAME1
dataGuardType=dataGuard
dbUniqueName=
oracleSidPrefix=DBNAME1
standbyDatabaseOCID=
===== END: DBNAME1 =====

===== START: DBNAME2 =====
create=no
sourceDatabaseOCID=<OCID>
dbName=DBNAME2
dataGuardType=dataGuard
dbUniqueName=
oracleSidPrefix=DBNAME2
standbyDatabaseOCID=
===== END: DBNAME2 =====
```

## Generate input template for Infrastructure only

**Syntax**

```
exacloudmigration generate_template infra [-hV] -s <sourceExadataInfraOCID> [-i <inputFileName>] [-p <profile>]
```

**Table 2-7 Exadata Cloud Migration Automation Generate Template Infrastructure Commands**

| Command                      | Description                                                                         |
|------------------------------|-------------------------------------------------------------------------------------|
| -s, --sourceExadataInfraOCID | Source Exadata Infrastructure OCID.                                                 |
| -i, --inputFileName          | Sets the infrastructure input template file name.<br>Default: infrastructure_inputs |
| -p, --profile                | OCI user profile name.<br>Default profile: DEFAULT                                  |
| -h, --help                   | Displays help information about the command and its options.                        |
| -V, --version                | Displays the version of the utility.                                                |

Examples:

- To generate input template for ExaDB-D infrastructure only:

```
/home/opc>./exacloudmigration generate_template infra -s <OCID>
```

## Generate input template for VM cluster(s) only

**Syntax**

```
exacloudmigration generate_template vmcluster [-hV] -t <targetType> -s <sourceExadataInfraOCID> [-i <inputFileName>] [-p <profile>]
```

**Table 2-8 Exadata Cloud Migration Automation Generate Template VM Clusters Commands**

| Command                      | Description                                                                     |
|------------------------------|---------------------------------------------------------------------------------|
| -t, --targetType             | Target System Type (exadb, exadbcc).                                            |
| -s, --sourceExadataInfraOCID | Source Exadata Infrastructure OCID.                                             |
| -i, --inputFileName          | Sets the VM cluster input template file name.<br>Default: cloudvmcluster_inputs |
| -p, --profile                | OCI user profile name.<br>Default profile: DEFAULT                              |
| -h, --help                   | Displays help information about the command and its options.                    |
| -V, --version                | Displays the version of the utility.                                            |

Examples:

- To generate input template for VM cluster(s) if the migration target is ExaDB-D:

```
/home/opc>./exacloudmigration generate_template vmcluster -t exadbd -s
<OCID>
```

- To generate input template for VM cluster(s) if the migration target is ExaDB-C@C:

```
/home/opc>./exacloudmigration generate_template vmcluster -t exadbcc -s
<OCID>
```

## Generate input template for database(s) only

### Syntax

```
exacloudmigration generate_template database [-hv] -s <sourceVmClusterOCID> [-i
<inputFileName>] [-p <profile>]
```

**Table 2-9 Exadata Cloud Migration Automation Generate Template Database(s) Commands**

| Command                   | Description                                                               |
|---------------------------|---------------------------------------------------------------------------|
| -s, --sourceVmClusterOCID | Source VM cluster OCID.                                                   |
| -i, --inputFileName       | Set the database(s) input template file name.<br>Default: database_inputs |
| -p, --profile             | OCI user profile name.<br>Default profile: DEFAULT                        |
| -h, --help                | Displays help information about the command and its options.              |
| -V, --version             | Displays the version of the utility.                                      |

Examples:

- To generate input template for ExaDB-D database(s) only:

```
/home/opc>./exacloudmigration generate_template database -s <OCID>
```

- To generate input template for ExaDB-C@C database(s) only:

```
/home/opc>./exacloudmigration generate_template database -s <OCID>
```

## Generate patches file

### Note

Generate patches file is required only if the source environment is ExaDB-C@C.

Running this step requires access to a database server in the source ExaDB-C@C VM cluster.

1. Transfer the `ecmasshutility` binary from the client machine to any database server of the source VM cluster.

For example:

```
scp /home/opc/ecmasshutility opc@[ip address of db server ]:/tmp
```

2. Log in to the source VM Cluster database server where the utility is copied.
3. Grant the required permissions on the `/tmp/ecmasshutility` directory to the `oracle` user.
4. Switch to the `oracle` user and navigate to the directory where the utility was copied (for example, `/tmp`).

The `ecmasshutility get_patches` supports the following options:

### Syntax

```
ecmasshutility get_patches [-p <patchesFileName>]
```

**Table 2-10** `ecmasshutility get_patches` Commands

| Command                            | Description                                                  |
|------------------------------------|--------------------------------------------------------------|
| <code>-p, --patchesFileName</code> | Sets the patches file name.                                  |
| <code>-h, --help</code>            | Displays help information about the command and its options. |
| <code>-V, --version</code>         | Displays the version of the utility.                         |

For example, to generate the patches file: `./ecmasshutility get_patches`.

**Note**

- By default, the patches file is created with the name `patches_file`. This file contains details of the one-off patches and Release Updates (RU) applied to each `ORACLE_HOME` in the cluster.  
For example:
  - `/u02/app/oracle/product/23.0.0.0/  
dbhome_1=35221462, 37537949, 37701424, 37701421`
  - `/u02/app/oracle/product/19.0.0.0/  
dbhome_2=34672698, 34697081, 37499406, 37542054, 37654975, 37642901`
- If the patches file is generated using the steps above, ensure that it is copied to the client machine in the respective VM cluster input directory. This is required for the creation of DB Home software images, DB Homes, and database migration operations.
- Generation of the patches file is required only when the source is ExaDB-C@C and for the following commands:
  - `migrate_infra`
  - `migrate_vmcluster`
  - `create_dbhomeimage`
  - `migrate_dbhome`
  - `migrate_database dgassociation`
  - `migrate_database switchover`
  - `migrate_database failover`

## Validate and dry run

The utility supports the `--dry-run` option, which performs a validation (precheck) of the input template parameters and runs an OCI dry run without making any changes to the source or target environments. This option helps identify potential issues in the setup or configuration before executing actual operations.

It is recommended to use the `--dry-run` option to validate the input templates and ensure a successful precheck before running any create or migrate commands. You can safely run the `--dry-run` option multiple times until all validations are successful.

Ensure that the input template(s) are generated before validating them.

**Note**

- Refer to the README.txt file that is included with the patch for a detailed description of the input file parameters. Use this file to modify parameter values in the input template files before performing any utility operations.
- In the input template files, the parameter create=yes|no determines whether a resource is included in the migration. However, this parameter is ignored if the command is run with the -a or -l options:
  - -a option: Migrates all resources listed in the input templates, regardless of the create parameter setting.
  - -l option: Migrates only the specific resources provided in the command, ignoring other entries in the templates.
  - No -a or -l option: The utility uses the create=yes setting in the input template files to determine which specific resources to migrate.

## Create Exadata Infrastructure

Create the Exadata Infrastructure in the target region or compartment using the specified source infrastructure details.

Ensure that the input template for the infrastructure is generated and modified according to the target platform requirements.

The create\_infra command supports the following options:

**Syntax**

```
exacloudmigration create_infra [-hV] -i <inputFile> [--dry-run]
```

**Table 2-11 exacloudmigration create\_infra Commands**

| Command         | Description                                                          |
|-----------------|----------------------------------------------------------------------|
| -i, --inputFile | Specifies the infrastructure input file name with the complete path. |
| --dry-run       | Performs dry run without creating the infrastructure.                |
| -h, --help      | Displays help information about the command and its options.         |
| -V, --version   | Displays the version of the utility.                                 |

Examples:

- To validate the infrastructure input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration create_infra -i inputs/infra_phxref/  
infrastructure_inputs --dry-run
```

- To create the infrastructure in the target region or compartment using the specified source infrastructure details, following successful validation and dry run:

```
/home/opc>./exacloudmigration create_infra -i inputs/infra_phxref/
infrastructure_inputs
```

#### ① Note

- Creation of the infrastructure alone in the target environment takes approximately 3–5 minutes.
- When creating the infrastructure, associated settings such as free-form tags, customer contacts, and the maintenance window are also replicated to the target.

## Create VM cluster(s)

Create VM cluster(s) in the target infrastructure using the specified source VM cluster(s) details.

Ensure that the target infrastructure is in an available state, and that the input template for VM cluster(s) has been generated and modified according to the target platform requirements.

The `create_vmcluster` command supports the following options:

### Syntax

```
exacloudmigration create_vmcluster [-ahV] -i <inputFile> [-a|-l
<vmClusterNames>] [--dry-run]
```

**Table 2-12 exacloudmigration create\_vmcluster Commands**

| Command              | Description                                                                                                                                                                                                                                                                        |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a, --all            | Creates VM clusters in the target environment for all VM clusters in the source.                                                                                                                                                                                                   |
| -i, --inputFile      | Specifies the VM cluster input file name with the complete path.                                                                                                                                                                                                                   |
| -l, --vmClusterNames | Specifies a comma-delimited list of source VM cluster names to create.<br>Examples: <ul style="list-style-type: none"> <li>To create a VM cluster: <code>-l vmcluster1</code></li> <li>To create multiple VM clusters: <code>-l vmcluster1,vmcluster2,vmcluster3</code></li> </ul> |
| --dry-run            | Performs dry run without creating the VM cluster(s).                                                                                                                                                                                                                               |
| -h, --help           | Displays help information about the command and its options.                                                                                                                                                                                                                       |
| -V, --version        | Displays the version of the utility.                                                                                                                                                                                                                                               |

Examples:

- To validate the VM clusters input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration create_vmcluster -a -i inputs/infra_phxref/  
cloudvmcluster_inputs --dry-run
```

- To validate comma-delimited list of VM clusters input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration create_vmcluster -a -i inputs/infra_phxref/  
cloudvmcluster_inputs -l vmcluster1,vmcluster2 --dry-run
```

- To create VM clusters in the target environment corresponding to all VM clusters in the source, following successful validation and dry run:

```
/home/opc>./exacloudmigration create_vmcluster -a -i inputs/infra_phxref/  
cloudvmcluster_inputs
```

- To create a comma-delimited list of source VM clusters in the target, following successful validation and dry run:

```
/home/opc>./exacloudmigration create_vmcluster -i inputs/infra_phxref/  
cloudvmcluster_inputs -l vmcluster1,vmcluster2
```

**Note**

- Creation of a single VM cluster in the target environment takes approximately 6 hours.
- For ExaDB-D and ExaDB-C@C, multiple VM clusters are created in parallel.
- When creating the VM cluster, associated metadata including free-form tags, SSH keys, and other configuration details are also replicated to the target environment.

## Create database dbHome software image

Create database dbHome software image(s) using the specified database(s) details

Ensure that the target VM cluster(s) is in an available state, and that the input template for database has been generated and modified according to the target platform requirements.

The create\_dbhomeimage command supports the following options:

### Syntax

```
exacloudmigration create_dbhomeimage [-ahV] -i <inputFile> [-a|-l <dbNames>]  
[-p <patchesFile>] [--dry-run]
```

**Table 2-13 exacloudmigration create\_dbhomeimage Commands**

| Command                  | Description                                                                                                                                                                                                                                                                                                                            |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a, --all                | Creates software image for all databases in the source VM cluster.                                                                                                                                                                                                                                                                     |
| -i, --inputFile          | Specifies the database input file name with the complete path.                                                                                                                                                                                                                                                                         |
| -i, --infraInputFileName | Specifies a comma-delimited list of database(s) to create the database software image.<br>Examples: <ul style="list-style-type: none"> <li>To create the database software image for a database: -l <i>dbname1</i></li> <li>To create the database software image for multiple databases: -l <i>dbname1,dbname2,dbname3</i></li> </ul> |
| -p, --patchesFile        | Specifies the one-off patches file with the complete path. This file is generated by the <code>ecmasshutility</code> and is applicable only when the source is ExaDB-C@C.<br>Use the <code>nopatch</code> option to create all target database home images with the source home base RU. For example, <code>-p nopatch</code> .        |
| --dry-run                | Performs dry run without creating the database software image(s).                                                                                                                                                                                                                                                                      |
| -h, --help               | Displays help information about the command and its options.                                                                                                                                                                                                                                                                           |
| -V, --version            | Displays the version of the utility.                                                                                                                                                                                                                                                                                                   |

**Examples:**

- To validate database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration create_dbhomeimage -a -i inputs/infra_phxref/vmcluster_vmcluster1/database_inputs --dry-run
```
- To validate comma-delimited list of dbnames of database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration create_dbhomeimage -i inputs/infra_phxref/vmcluster_vmcluster1/database_inputs -l dbname1,dbname2 --dry-run
```
- To create a software image(s) for each database home of the source ExaDB-D VM cluster, following successful validation and dry run:

```
/home/opc>./exacloudmigration create_dbhomeimage -a -i inputs/infra_phxref/vmcluster_vmcluster1/database_inputs
```
- To create specific Database homes software image(s) of source ExaDB-D VM cluster, following successful validation and dry run:

```
/home/opc>./exacloudmigration create_dbhomeimage -i inputs/infra_phxref/vmcluster_vmcluster1/database_inputs -l dbname1,dbname2
```

- To create a software image(s) for every database home of the source ExaDB-C@C VM cluster, following successful validation and dry run:

```
/home/opc>./exacloudmigration create_dbhomeimage -a -i inputs/infra_phxref/
vmcluster_vmcluster1/database_inputs -p /home/opc/patches_file
```

Generate the patches\_file using the [Generate patches file](#) step and copy it to the client machine.

- To create specific Database homes software images of source ExaDB-C@C VM cluster, following successful validation and dry run:

```
/home/opc>./exacloudmigration create_dbhomeimage -i inputs/infra_phxref/
vmcluster_vmcluster1/database_inputs -l dbname1,dbname2 -p /home/opc/
patches_file
```

Generate the patches\_file using the [Generate patches file](#) step and copy it to the client machine.

## Migrate Exadata Infrastructure

Migrate the Exadata infrastructure using the specified details, including the creation of the infrastructure, VM cluster(s), and database dbHome(s).

Ensure that the input templates for the Infrastructure, VM cluster(s) and database(s) have been generated and modified according to the target platform requirements.

The migrate\_infra command supports the following options:

### Syntax

```
exacloudmigration migrate_infra [-ahVd] -i <inputFileInfra> -c
<inputFileVmCluster> [-a|-l <vmClusterNames>] [-p <includePatches>] [--dry-
run]
```

**Table 2-14 exacloudmigration migrate\_infra Commands**

| Command                  | Description                                                                                                                                                                                                                                          |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a, --all                | Migrates all VM clusters in the newly created target infrastructure from the source infrastructure.                                                                                                                                                  |
| -i, --inputFileInfra     | Specifies the infrastructure input file name with the complete path.                                                                                                                                                                                 |
| -c, --inputFileVmCluster | Specifies the VM cluster input file name with the complete path.                                                                                                                                                                                     |
| -l, --vmClusterNames     | Specifies a comma-delimited list of VM cluster names to migrate.<br>Examples: <ul style="list-style-type: none"> <li>To migrate a VM cluster: -l vmcluster1</li> <li>To migrate multiple VM clusters: -l vmcluster1,vmcluster2,vmcluster3</li> </ul> |
| -d, --includeDbInputs    | Specifies to include database inputs (need database_inputs inside each vm_cluster folder).                                                                                                                                                           |

**Table 2-14 (Cont.) exacloudmigration migrate\_infra Commands**

| Command              | Description                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -p, --includePatches | Specifies to consider One-off patches file. This option is applicable only when the source is ExaDB-C@C.<br>Options: <ul style="list-style-type: none"> <li>-p yes: Specify to consider patches_file inside the input template vmcluster_&lt;source_vmcluster_name&gt; directory.</li> <li>-p nopatch: Specify to create all target database homes with the source home base RU.</li> </ul> |
| --dry-run            | Performs dry run without creating the infrastructure.                                                                                                                                                                                                                                                                                                                                       |
| -h, --help           | Displays help information about the command and its options.                                                                                                                                                                                                                                                                                                                                |
| -V, --version        | Displays the version of the utility.                                                                                                                                                                                                                                                                                                                                                        |

**Examples:**

- To validate the Infrastructure, VM cluster and database input template files and perform a --dry-run:

```
/home/opc>./exacloudmigration migrate_infra -a -i inputs/infra_phxref/
infrastructure_inputs -c inputs/infra_phxref/cloudvmcluster_inputs --dry-
run
```
- To migrate the infrastructure, all VM clusters, and database dbhome(s) in the source (ExaDB-D) to the target (ExaDB-D), following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_infra -a -i inputs/infra_phxref/
infrastructure_inputs -c inputs/infra_phxref/cloudvmcluster_inputs
```
- To migrate the infrastructure, all VM clusters, and the selected database dbhome(s) in the source (ExaDB-D) to the target (ExaDB-D) by specifying create=yes in database input file, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_infra -a -i inputs/infra_phxref/
infrastructure_inputs -c inputs/infra_phxref/cloudvmcluster_inputs -d
```
- To migrate the infrastructure, a comma-delimited list of VM clusters, and the corresponding database dbhome(s) in the source (ExaDB-D) to the target (ExaDB-D), following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_infra -i inputs/infra_phxref/
infrastructure_inputs -c inputs/infra_phxref/cloudvmcluster_inputs -l
vmcluster1,vmcluster2
```

- To migrate the infrastructure, all VM clusters, and database dbhome(s) in the source (ExaDB-C@C) to the target (ExaDB-D), following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_infra -a -i inputs/infra_phxref/
infrastructure_inputs -c inputs/infra_phxref/cloudvmcluster_inputs -p yes
```

Generate the patches\_file using the [Generate patches file](#) step, and copy it to the input template directory (vmcluster\_<source\_vmcluster\_name>) on the client machine for each VM cluster.

- To migrate the infrastructure, a comma-delimited list of VM clusters, and the corresponding database dbhome(s) in the source (ExaDB-C@C) to the target (ExaDB-D), following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_infra -i inputs/infra_phxref/
infrastructure_inputs -c inputs/infra_phxref/cloudvmcluster_inputs -l
vmcluster1,vmcluster2 -p yes
```

Generate the patches\_file using the [Generate patches file](#) step, and copy it to the input template directory (vmcluster\_<source\_vmcluster\_name>) on the client machine for each VM cluster.

- To migrate the infrastructure, all VM clusters and database dbhome(s) in the source (ExaDB-C@C) to the target (ExaDB-D) without the patch file, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_infra -a -i inputs/infra_phxref/
infrastructure_inputs -c inputs/infra_phxref/cloudvmcluster_inputs -p
nopatch
```

#### Note

Along with migrating the infrastructure, VM cluster(s), and database dbHome(s), associated metadata is also replicated to the target environment.

## Migrate VM cluster(s)

Migrate VM cluster(s) in the target infrastructure using the specified source VM cluster details, including the creation of VM cluster(s) and database dbHome(s).

Ensure that the target infrastructure is in available state, input template for VM cluster(s) has been generated and modified according to the target platform requirements.

The migrate\_vmcluster command supports the following options:

### Syntax

```
exacloudmigration migrate_vmcluster [-ahVd] -i <inputFile> [-a|-l
<vmClusterNames>] [-p <includePatches>] [--dry-run]
```

**Table 2-15 exacloudmigration migrate\_vmcluster Commands**

| Command               | Description                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a, --all             | Migrates all VM clusters to the target infrastructure from the source infrastructure.                                                                                                                                                                                                                                                                                                       |
| -i, --inputFile       | Specifies the VM cluster input file name with the complete path.                                                                                                                                                                                                                                                                                                                            |
| -l, --vmClusterNames  | Specifies a comma-delimited list of VM cluster names to migrate.<br>Examples: <ul style="list-style-type: none"> <li>To migrate a VM cluster: -l vmcluster1</li> <li>To migrate multiple VM clusters: -l vmcluster1,vmcluster2,vmcluster3</li> </ul>                                                                                                                                        |
| -d, --includeDbInputs | Specifies to include database inputs (need database_inputs inside each vm_cluster folder).                                                                                                                                                                                                                                                                                                  |
| -p, --includePatches  | Specifies to consider One-off patches file. This option is applicable only when the source is ExaDB-C@C.<br>Options: <ul style="list-style-type: none"> <li>-p yes: Specify to consider patches_file inside the input template vmcluster_&lt;source_vmcluster_name&gt; directory.</li> <li>-p nopatch: Specify to create all target database homes with the source home base RU.</li> </ul> |
| --dry-run             | Performs dry run without creating the VM cluster(s).                                                                                                                                                                                                                                                                                                                                        |
| -h, --help            | Displays help information about the command and its options.                                                                                                                                                                                                                                                                                                                                |
| -V, --version         | Displays the version of the utility.                                                                                                                                                                                                                                                                                                                                                        |

**Examples:**

- To validate the VM cluster and database input template files and perform a --dry-run:

```
/home/opc>./exacloudmigration migrate_vmcluster -a -i inputs/infra_phxref/cloudvmcluster_inputs --dry-run
```
- To validate the comma-delimited list of VM clusters and database input template files and perform a --dry-run:

```
/home/opc>./exacloudmigration migrate_vmcluster -i inputs/infra_phxref/cloudvmcluster_inputs -l vmcluster1,vmcluster2 --dry-run
```
- To migrate all VM clusters and database dbhome(s) in the source (ExaDB-D) infrastructure to the target (ExaDB-D) infrastructure, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_vmcluster -a -i inputs/infra_phxref/cloudvmcluster_inputs
```

- To migrate a comma-delimited list of VM clusters and database dbhome(s) in the source (ExaDB-D) infrastructure to the target (ExaDB-D) infrastructure, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_vmcluster -i inputs/infra_phxref/
cloudvmcluster_inputs -l vmcluster1,vmcluster2
```

- To migrate all VM clusters and database dbhome(s) in the source (ExaDB-C@C) infrastructure to the target (ExaDB-C@C) infrastructure or source (ExaDB-C@C) infrastructure to the target (ExaDB-D) infrastructure, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_vmcluster -a -i inputs/infra_phxref/
cloudvmcluster_inputs -p yes
```

Generate the patches\_file using the [Generate patches file](#) step and copy it to the input template directory (vmcluster\_<source\_vmcluster\_name>) on the client machine for each VM cluster.

- To migrate all VM clusters and database dbhome(s) in the source (ExaDB-C@C) infrastructure to the target (ExaDB-C@C) infrastructure or source (ExaDB-C@C) infrastructure to the target (ExaDB-D) infrastructure without patch file, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_vmcluster -a -i inputs/infra_phxref/
cloudvmcluster_inputs -p nopatch
```

- To migrate a comma-delimited list of VM clusters and database dbhome(s) in the source (ExaDB-C@C) infrastructure to the target (ExaDB-C@C) infrastructure or source (ExaDB-C@C) infrastructure to the target (ExaDB-D) infrastructure, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_vmcluster -i inputs/infra_phxref/
cloudvmcluster_inputs -l vmcluster1,vmcluster2 -p yes
```

Generate the patches\_file using the [Generate patches file](#) step and copy it to the input template directory (vmcluster\_<source\_vmcluster\_name>) on the client machine for each VM cluster.

#### Note

Along with migrating VM cluster(s) and database dbHome(s), free-form tags, SSH keys, and other VM cluster metadata are also replicated to the target.

## Migrate DBHome(s)

Migrate DBHome(s) from source VM cluster using the specified target VM cluster details.

Ensure that the target VM cluster(s) is in available state, the database input file has been generated and modified according to the target platform requirement.

The migrate\_dbhomes command supports the following options:

**Syntax**

```
exacloudmigration migrate_dbhomes [-hV] [-a|-l <dbNames>] -i <inputFile> [-p
<patchesFile>] [--dry-run]
```

**Table 2-16 Exadata exacloudmigration migrate\_dbhomes Commands**

| Command           | Description                                                                                                                                                                                                                                                                                                                  |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a, --all         | Migrates all dbhomes to the target VM cluster from the source VM cluster.                                                                                                                                                                                                                                                    |
| -i, --inputFile   | Specifies the database input file name with the complete path.                                                                                                                                                                                                                                                               |
| -l, --dbNames     | Specifies a comma-delimited list of source database names to migrate the database home. Examples: <ul style="list-style-type: none"> <li>To migrate a database: <i>-l dbname1</i></li> <li>To migrate multiple databases: <i>-l dbname1,dbname2,dbname3</i></li> </ul>                                                       |
| -p, --patchesFile | Specifies the one-off patches file with the complete path. This file is generated by the <code>ecmasshutility</code> and is applicable only when the source is ExaDB-C@C. Use the <code>nopatch</code> option to create all target database home images with the source home base RU. For example, <code>-p nopatch</code> . |
| --dry-run         | Performs dry run without creating the dbhome(s).                                                                                                                                                                                                                                                                             |
| -h, --help        | Displays help information about the command and its options.                                                                                                                                                                                                                                                                 |
| -V, --version     | Displays the version of the utility.                                                                                                                                                                                                                                                                                         |

**Examples:**

- To validate the database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration migrate_dbhomes -a -i inputs/infra_phxref/
vmcluster_vmcluster1/database_inputs --dry-run
```

- To validate a comma-delimited list of dbnames of the database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration migrate_dbhomes -i inputs/infra_phxref/
vmcluster_vmcluster1/database_inputs -l dbname1,dbname2 --dry-run
```

- To migrate all dbhome(s) in source (ExaDB-D) VM cluster to target (ExaDB-D) VM cluster, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_dbhomes -a -i inputs/infra_phxref/
vmcluster_vmcluster1/database_inputs
```

- To migrate a comma-delimited list of dbhome(s) in the source (ExaDB-D) VM cluster to the target (ExaDB-D) VM cluster, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_dbhomes -i inputs/infra_phxref/  
vmcluster_vmcluster1/database_inputs -l dbname1,dbname2
```

- To migrate all dbhomes in the source (ExaDB-C@C) VM cluster to the target (ExaDB-C@C) VM cluster or source (ExaDB-C@C) VM cluster to the target (ExaDB-D) VM cluster, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_dbhomes -a -i inputs/infra_phxref/  
vmcluster_vmcluster1/database_inputs -p /home/opc/patches_file
```

Generate the patches\_file using the [Generate patches file](#) step and copy it to the client machine.

- To migrate comma separated list of dbhomes in the source (ExaDB-C@C) VM cluster to the target (ExaDB-C@C) VM cluster or source (ExaDB-C@C) VM cluster to the target (ExaDB-D) VM cluster, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_dbhomes -i inputs/infra_phxref/  
vmcluster_vmcluster1/database_inputs -l dbname1,dbname2 -p /home/opc/  
patches_file
```

Generate the patches\_file using the [Generate patches file](#) step and copy it to the client machine.

#### Note

Along with the migration, free-form tags of database dbHome(s) are also replicated to the target.

## Database wallet operations

Generates a database wallet file for a specified database or for all databases in the source VM cluster. This eliminates the need to provide the database SYS and TDE passwords during the following operations: migrate database, switchover, failover, and reinstate.

There are two methods available to generate the database wallet file. Option 2 requires access to any database server in the source VM cluster.

- Option 1: exacloudmigration generate\_dbwallet
- Option 2: ecmasshutility generate\_dbwallet

#### Note

The generated wallet file, created using either option, securely stores both the SYS and TDE passwords for the databases and is protected with a file password.

- [Option 1: exacloudmigration generate\\_dbwallet](#)
- [Option 2: ecmasshutility to generate database password wallet](#)

## Option 1: exacloudmigration generate\_dbwallet

Ensure that the input template for database(s) has been generated and modified according to the target platform requirements.

The generate\_dbwallet command supports the following options:

### Syntax

```
exacloudmigration generate_dbwallet [-ahV] -i <inputFile> [a | -l <dbNames>]
```

**Table 2-17 exacloudmigration generate\_dbwallet Commands**

| Command         | Description                                                                                                                                                                                                                                                                                                                   |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a, --all       | Generates password wallet for all databases in the source VM cluster.                                                                                                                                                                                                                                                         |
| -i, --inputFile | Specifies the database input file name with the complete path.                                                                                                                                                                                                                                                                |
| -l, --dbNames   | Specifies a comma-delimited list of database names to generate the password wallet.<br>Examples: <ul style="list-style-type: none"> <li>• To generate the password wallet for a database: -l <i>dbname1</i></li> <li>• To generate the password wallet for multiple databases: -l <i>dbname1, dbname2, dbname3</i></li> </ul> |
| -h, --help      | Displays help information about the command and its options.                                                                                                                                                                                                                                                                  |
| -V, --version   | Displays the version of the utility.                                                                                                                                                                                                                                                                                          |

Examples:

- To generate password wallet for all databases in source VM cluster:

```
/home/opc>./exacloudmigration generate_dbwallet -a -i inputs/infra_phxref/vmcluster_vmcluster1/database_inputs
```

Specify the SYS and TDE passwords for the databases along with a secure password for the wallet file and file will be generated with the default file name ewallet.p12.

- To generate password wallet using a comma-delimited list of Database names:

```
/home/opc>./exacloudmigration generate_dbwallet -i inputs/infra_phxref/vmcluster_vmcluster1/database_inputs -l dbname1,dbname2
```

Specify the SYS and TDE passwords for the *dbname1*, *dbname2*, along with a secure password for the wallet file and file will be generated with the default file name ewallet.p12.

- [Option 1: exacloudmigration edit\\_dbwallet](#)

- [Option 1: exacloudmigration list\\_wallet\\_dbs](#)

## Option 1: exacloudmigration edit\_dbwallet

If database wallet is already generated using the option `generate_dbwallet`, then this helps in adding or updating password for a given database in database wallet file

The `edit_dbwallet` command supports the following options:

### Syntax

```
exacloudmigration edit_dbwallet [-hV] -a <add>|-u <update> -w <walletFile> -d <dbName>
```

**Table 2-18 exacloudmigration edit\_dbwallet Commands**

| Command          | Description                                                   |
|------------------|---------------------------------------------------------------|
| -a, --add        | Adds a password for a given database in the wallet file.      |
| -u, --update     | Updates the password for a given database in the wallet file. |
| -w, --walletFile | Specifies the database wallet file name with the path.        |
| -d, --dbName     | Specifies the database name to perform edit operation.        |
| -h, --help       | Displays help information about the command and its options.  |
| -V, --version    | Displays the version of the utility.                          |

Examples:

- To add password for a specified database in the wallet file:

```
/home/opc>./exacloudmigration edit_dbwallet -a -w ewallet.p12 -d dbnamenew
```

Specify wallet file password to unlock the file for adding *dbnamenew* SYS and TDE password to the wallet.

- To update password for a specified database in the wallet file:

```
/home/opc>./exacloudmigration edit_dbwallet -u -w ewallet.p12 -d dbname1
```

Specify wallet file password to unlock the file for updating *dbname1* SYS and TDE passwords.

## Option 1: exacloudmigration list\_wallet\_dbs

Display the list of database names having password stored in database wallet file.

The `list_wallet_dbs` command supports the following options:

**Syntax**

```
exacloudmigration list_wallet_dbs [-hV] -w <walletFile>
```

**Table 2-19 exacloudmigration list\_wallet\_dbs Commands**

| Command          | Description                                                  |
|------------------|--------------------------------------------------------------|
| -w, --walletFile | Specifies the database wallet file name with the path.       |
| -h, --help       | Displays help information about the command and its options. |
| -V, --version    | Displays the version of the utility.                         |

Examples:

- To display the list of database names having password stored in the password wallet file:

```
/home/opc>./exacloudmigration list_wallet_dbs -w ewallet.p12
```

## Option 2: ecmasshutility to generate database password wallet

Running this utility requires access to any database server of the source VM cluster:

1. Transfer the ecmasshutility binary from client machine to any database server of the source VM cluster.

For example:

```
scp /home/opc/ecmasshutility opc@[ip address of db server ]:/tmp
```

2. Log in to the source VM Cluster database server where the utility is copied.
3. Grant the required permissions on the /tmp/ecmasshutility directory to the oracle user.
4. Switch to the oracle user and navigate to the directory where the utility was copied (for example, /tmp).

The ecmasshutility command supports the following operations:

**Syntax**

```
ecmasshutility [-hV] [command]
```

**Table 2-20 ecmasshutility Commands**

| Command           | Description                                                                         |
|-------------------|-------------------------------------------------------------------------------------|
| generate_dbwallet | Specifies to create exacloud migration password wallet file for database migration. |
| edit_dbwallet     | Specifies to edit exacloud migration password wallet file for database migration.   |

**Table 2-20 (Cont.) ecmasshutility Commands**

| Command         | Description                                                                    |
|-----------------|--------------------------------------------------------------------------------|
| list_wallet_dbs | Displays the list of database names having password stored in the wallet file. |
| get_patches     | Creates patches file for the exacloudmigration utility.                        |
| -h, --help      | Displays help information about the command and its options.                   |
| -V, --version   | Displays the version of the utility.                                           |

- [Option 2: ecmasshutility generate\\_dbwallet](#)
- [Option 2: ecmasshutility edit\\_dbwallet](#)
- [Option 2: ecmasshutility list\\_wallet\\_dbs](#)

## Option 2: ecmasshutility generate\_dbwallet

The generate\_dbwallet command supports the following options:

### Syntax

```
ecmasshutility generate_dbwallet [-hV] [-l <dbNames>]
```

**Table 2-21 ecmasshutility generate\_dbwallet Commands**

| Command       | Description                                                                                                                                                                                                                                                                                                |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -l, --dbNames | Specifies a comma-delimited list of database(s) to generate the password wallet.<br>Examples: <ul style="list-style-type: none"> <li>• To generate the password wallet for a database: -l dbname1</li> <li>• To generate the password wallet for multiple databases: -l dbname1,dbname2,dbname3</li> </ul> |
| -h, --help    | Displays help information about the command and its options.                                                                                                                                                                                                                                               |
| -V, --version | Displays the version of the utility.                                                                                                                                                                                                                                                                       |

Examples:

- To generate password wallet for all databases in source VM cluster:

```
/tmp>./ecmasshutility generate_dbwallet
```

Specify secure password for the wallet file and generates with name ewallet.p12.

- To generate wallet for selected database(s) in Source VM cluster:

```
/tmp>./ecmasshutility generate_dbwallet -l dbname1,dbname2
```

Specify secure password for the wallet file and file will be generated with the default file name `ewallet.p12`.

### Note

The `/tmp/ewallet.p12` file will be generated. Copy this file to the utility location on the client machine.

## Option 2: `ecmasshutility edit_dbwallet`

If database wallet is already generated using the `generate_dbwallet` option, then this helps in add or update password for a given database in database wallet file

The `edit_dbwallet` command supports the following options:

### Syntax

```
ecmasshutility edit_dbwallet [-hV] -a <add>|-u <update> -w <walletFile> -d <dbName>
```

**Table 2-22** `ecmasshutility edit_dbwallet` Commands

| Command          | Description                                                          |
|------------------|----------------------------------------------------------------------|
| -a, --add        | Specifies to add a password for a given database in the wallet file. |
| -u, --update     | Updates password for a given database in the wallet file.            |
| -w, --walletFile | Specifies the database wallet file name with the path.               |
| -d, --dbName     | Specifies the database name to perform edit operation.               |
| -h, --help       | Displays help information about the command and its options.         |
| -V, --version    | Displays the version of the utility.                                 |

Examples:

- To add password for a given database in wallet file:

```
/tmp>./ecmasshutility edit_dbwallet -a -w ewallet.p12 -d dbnamenew
```

Specify database wallet file password to unlock the file for adding *dbnamenew* database password.

- To update password for a given database in wallet file:

```
/tmp>./ecmasshutility edit_dbwallet -u -w ewallet.p12 -d dbname1
```

Specify database wallet file password to unlock the file for updating *dbname1* password.

## Option 2: ecmasshutility list\_wallet\_dbs

Display the list of database names having password stored in the database wallet file.

The `list_wallet_dbs` command supports the following options:

### Syntax

```
ecmasshutility list_wallet_dbs [-hV] -w <walletFile>
```

**Table 2-23 ecmasshutility list\_wallet\_dbs Commands**

| Command          | Description                                                  |
|------------------|--------------------------------------------------------------|
| -w, --walletFile | Specifies the database wallet file name with the path.       |
| -h, --help       | Displays help information about the command and its options. |
| -V, --version    | Displays the version of the utility.                         |

Examples:

- To show the list of database names having password stored in the database wallet file:

```
/tmpc>./ecmasshutility list_wallet_dbs -w ewallet.p12
```

## Migrate database(s)

Migrate database(s) from the source VM cluster to the target VM cluster.

### Note

- If the migration is between ExaDB-C@C and ExaDB-D models, this utility does not support database migration. In such cases, alternative methods such as Zero Downtime Migration (ZDM) or `dbaascli` commands are recommended.
- If the wallet file (`ewallet.p12`) is generated using the [Database wallet operations](#) step, ensure that it is copied to the client machine, for example, to `/home/opc`, for database migration operations.

The `migrate_database` command supports the following options:

### Syntax

```
exacloudmigration migrate_database [-hV] [command]
```

**Table 2-24 exacloudmigration migrate\_database Options**

| Option        | Description                                                  |
|---------------|--------------------------------------------------------------|
| -h, --help    | Displays help information about the command and its options. |
| -V, --version | Displays the version of the utility.                         |

**Table 2-25 exacloudmigration migrate\_database Commands**

| Command       | Description                                                                                                                                                          |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dgassociation | Enables Data Guard association between source and target databases.                                                                                                  |
| switchover    | Enables Data Guard association between source and target databases.<br>Makes standby database as primary database and old primary database becomes standby database  |
| failover      | Enables Data Guard association between source and target databases.<br>Makes standby database as primary database and old primary database becomes disabled standby. |

**Note**

During database migration, if the source database has automatic backups enabled, the same setting will be applied to the target database.

- [Enable Data Guard association between source and target databases](#)
- [Migrate database switchover](#)
- [Migrate database failover](#)
- [Database switchover](#)
- [Database failover](#)

## Enable Data Guard association between source and target databases

Ensure that the target VM cluster(s) is in an available state, and that the input template for database has been generated and modified according to the target platform requirements.

Ensure that both source and target database servers are having the same dbaastools and dbcs-agent versions.

The migrate\_database dgassociation command supports the following options:

**Syntax**

```
exacloudmigration migrate_database dgassociation [-ahV] -i <inputFile> [-w <walletFile>] [-a|-l <dbNames>] [-p <patchesFile>] [--dry-run]
```

**Table 2-26 exacloudmigration migrate\_database dgassociation Commands**

| Command           | Description                                                                                                                                                                                                                                                                                                                  |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a, --all         | Creates Data Guard association for all source databases.                                                                                                                                                                                                                                                                     |
| -i, --inputFile   | Specifies the database inputs file name with complete path.                                                                                                                                                                                                                                                                  |
| -l, --dbNames     | Specifies a comma-delimited list of databases to create Data Guard associations.<br>Examples: <ul style="list-style-type: none"> <li>To create Data Guard association for a database: -l <i>dbname1</i></li> <li>To create Data Guard association for multiple databases: -l <i>dbname1,dbname2,dbname3</i></li> </ul>       |
| -w, --walletFile  | Specifies the wallet file name with the path.                                                                                                                                                                                                                                                                                |
| -p, --patchesFile | Specifies the one-off patches file with the complete path. This file is generated by the <code>ecmasshutility</code> and is applicable only when the source is ExaDB-C@C. Use the <code>nopatch</code> option to create all target database home images with the source home base RU. For example, -p <code>nopatch</code> . |
| --dry-run         | Performs dry run without creating Data Guard associations.                                                                                                                                                                                                                                                                   |
| -h, --help        | Displays help information about the command and its options.                                                                                                                                                                                                                                                                 |
| -V, --version     | Displays the version of the utility.                                                                                                                                                                                                                                                                                         |

**Examples:**

- To validate the database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration migrate_database dgassociation -a -i inputs/infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12 --dry-run
```

Specify the password for reading the wallet file.

- To validate a comma-delimited list of dbnames of database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration migrate_database dgassociation -i inputs/infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12 -l dbname1,dbname2 --dry-run
```

Specify the password for reading the wallet file.

- To create Data Guard association for all ExaDB-D source databases, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_database dgassociation -a -i inputs/infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12
```

Specify the password for reading the wallet file.

- To create Data Guard association(s) using a comma-delimited list of ExaDB-D source database(s) provided, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_database dgassociation -i inputs/  
infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12 -l  
dbname1,dbname2
```

Specify the password for reading the wallet file.

- To create Data Guard association for all ExaDB-C@C source databases, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_database dgassociation -a -i inputs/  
infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12 -p /  
home/opc/patches_file
```

Generate the patches\_file using the [Generate patches file](#) step and copy it to the client machine.

Specify the password for reading the wallet file.

- To create Data Guard association(s) using the comma-delimited list of ExaDB-C@C source database(s) provided, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_database dgassociation -i inputs/  
infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12 -l  
dbname1,dbname2 -p /home/opc/patches_file
```

Generate the patches\_file using the [Generate patches file](#) step and copy it to the client machine.

Specify the password for reading the wallet file.

#### Note

- If the database wallet file is not used, the utility will prompt for the SYS and TDE passwords for each database.
- If the database wallet file is used but a password entry for any database is missing, the utility execution will stop and display an error indicating that the password does not exist in the wallet.
- In a source VM cluster where multiple databases share the same dbHome, enabling a Data Guard association between the source and target databases will occur sequentially for those databases.

## Migrate database switchover

Enables the Data Guard association and performs a role transition, making the standby database the primary database, while the old primary database becomes the standby.

Ensure that the database input file has been generated and modified according to the target platform requirements.

The `migrate_database switchover` command supports the following options:

### Syntax

```
exacloudmigration migrate_database switchover [-ahV] -i <inputFile> [-w
<walletFile>] [-a|-l <dbNames>] [-p <patchesFile>] [--dry-run]
```

**Table 2-27 exacloudmigration migrate\_database switchover Commands**

| Command           | Description                                                                                                                                                                                                                                                                                                                  |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a, --all         | Performs migrate switchover for all source databases.                                                                                                                                                                                                                                                                        |
| -i, --inputFile   | Specifies the database input file name with complete path.                                                                                                                                                                                                                                                                   |
| -l, --dbNames     | Specifies a comma-delimited list of databases to perform migrate switchover.<br>Examples: <ul style="list-style-type: none"> <li>To perform migrate switchover for a database: <code>-l dbname1</code></li> <li>To perform migrate switchover for multiple databases: <code>-l dbname1, dbname2, dbname3</code></li> </ul>   |
| -w, --walletFile  | Specifies the wallet file name with the path.                                                                                                                                                                                                                                                                                |
| -p, --patchesFile | Specifies the one-off patches file with the complete path. This file is generated by the <code>ecmasshutility</code> and is applicable only when the source is ExaDB-C@C. Use the <code>nopatch</code> option to create all target database home images with the source home base RU. For example, <code>-p nopatch</code> . |
| --dry-run         | Performs dry run without creating the Data Guard associations.                                                                                                                                                                                                                                                               |
| -h, --help        | Displays help information about the command and its options.                                                                                                                                                                                                                                                                 |
| -V, --version     | Displays the version of the utility.                                                                                                                                                                                                                                                                                         |

### Examples:

- To validate the database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration migrate_database switchover -a -i inputs/
infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12 --dry-run
```

Specify the password for reading the wallet file.

- To validate a comma-delimited list of dbnames of database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration migrate_database switchover -i inputs/
infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12 -l
dbname1,dbname2 --dry-run
```

Specify the password for reading the wallet file.

- To perform migrate switchover for all ExaDB-D source databases, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_database switchover -a -i inputs/  
infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12
```

Specify the password for reading the wallet file.

- To perform migrate switchover using the comma-delimited list of ExaDB-D source databases provided, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_database switchover -i inputs/  
infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12 -l  
dbname1,dbname2
```

Specify the password for reading the wallet file.

- To perform migrate switchover for all ExaDB-C@C source databases, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_database switchover -a -i inputs/  
infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12 -p /  
home/opc/patches_file
```

Generate the patches\_file using the [Generate patches file](#) step and copy it to the client machine.

Specify the password for reading the wallet file.

- To perform migrate switchover using the comma-delimited list of ExaDB-C@C source database(s), following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_database switchover -i inputs/  
infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12 -l  
dbname1,dbname2 -p /home/opc/patches_file
```

Generate the patches\_file using the [Generate patches file](#) step and copy it to the client machine.

Specify the password for reading the wallet file.

#### Note

- If the database wallet file is not used, the command prompts for the SYS and TDE passwords for each database.
- If the database wallet file is used but an entry for any database is missing, the utility stops execution and displays an error indicating that the password does not exist in the wallet.

## Migrate database failover

Enables the Data Guard association and performs a role transition, making the standby database the primary database, while the old primary database becomes the disabled standby.

Ensure that the database input file has been generated and modified according to the target platform requirements.

The `migrate_database failover` command supports the following options:

### Syntax

```
exacloudmigration migrate_database failover [-ahV] -i <inputFile> [-w
<walletFile>] [-a|-l <dbNames>] [-p <patchesFile>] [--dry-run]
```

**Table 2-28 exacloudmigration migrate\_database failover Commands**

| Command           | Description                                                                                                                                                                                                                                                                                                                  |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a, --all         | Performs migrate failover for all source databases.                                                                                                                                                                                                                                                                          |
| -i, --inputFile   | Specifies the database input file name with complete path.                                                                                                                                                                                                                                                                   |
| -l, --dbNames     | Specifies a comma-delimited list of databases to perform migrate failover.<br>Examples: <ul style="list-style-type: none"> <li>To perform migrate failover for a database: -l <i>dbname1</i></li> <li>To perform migrate failover for multiple databases: -l <i>dbname1,dbname2,dbname3</i></li> </ul>                       |
| -w, --walletFile  | Specifies the wallet file name with the path.                                                                                                                                                                                                                                                                                |
| -p, --patchesFile | Specifies the one-off patches file with the complete path. This file is generated by the <code>ecmasshutility</code> and is applicable only when the source is ExaDB-C@C. Use the <code>nopatch</code> option to create all target database home images with the source home base RU. For example, -p <code>nopatch</code> . |
| --dry-run         | Performs dry run without creating the Data Guard associations.                                                                                                                                                                                                                                                               |
| -h, --help        | Displays help information about the command and its options.                                                                                                                                                                                                                                                                 |
| -V, --version     | Displays the version of the utility.                                                                                                                                                                                                                                                                                         |

Examples:

- To validate the database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration migrate_database failover -a -i inputs/
infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12 --dry-run
```

Specify the password for reading the wallet file.

- To validate a comma-delimited list of dbnames of database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration migrate_database failover -i inputs/
infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12 -l
dbname1,dbname2 --dry-run
```

Specify the password for reading the wallet file.

- To perform migrate failover for all ExaDB-D source databases, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_database failover -a -i inputs/
infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12
```

Specify the password for reading the wallet file.

- To perform migrate failover using the comma-delimited list of ExaDB source database(s) provided, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_database failover -i inputs/
infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12 -l
dbname1,dbname2
```

Specify the password for reading the wallet file.

- To perform migrate failover for all ExaDB-C@C source databases, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_database failover -a -i inputs/
infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12 -p /
home/opc/patches_file
```

Generate the patches\_file using the [Generate patches file](#) step and copy it to the client machine.

Specify the password for reading the wallet file.

- To perform migrate failover using the comma-delimited list of ExaDB-C@C source database(s) provided, following successful validation and dry run:

```
/home/opc>./exacloudmigration migrate_database failover -i inputs/
infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12 -l
dbname1,dbname2 -p /home/opc/patches_file
```

Generate the patches\_file using the [Generate patches file](#) step and copy it to the client machine.

Specify the password for reading the wallet file.

**Note**

- If the database wallet file is not used, the command prompts for the SYS and TDE passwords for each database.
- If the database wallet file is used, but a password entry for any database is missing, the utility stops execution and displays an error indicating that the password does not exist in the wallet.

# Database switchover

Makes the standby database the primary database, and the old primary database the standby database.

Ensure that the Data Guard association is enabled between source and target database.

Ensure that the database input file has been generated and modified according to the target platform requirements.

The switchover command supports the following options:

**Syntax**

```
exacloudmigration switchover [-ahV] -i <inputFile> [-w <walletFile>] [-a|-l <dbNames>] [--dry-run]
```

**Table 2-29 exacloudmigration switchover Commands**

| Command          | Description                                                                                                                                                                                                                                                                             |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a, --all        | Performs switchover for all databases in the source VM cluster.                                                                                                                                                                                                                         |
| -i, --inputFile  | Specifies the database input file name with complete path.                                                                                                                                                                                                                              |
| -l, --dbNames    | Specifies a comma-delimited list of databases to perform switchover.<br>Examples: <ul style="list-style-type: none"><li>• To perform switchover for a database: -l <i>dbname1</i></li><li>• To perform switchover for multiple databases: -l <i>dbname1, dbname2, dbname3</i></li></ul> |
| -w, --walletFile | Specifies the wallet file name with the path.                                                                                                                                                                                                                                           |
| --dry-run        | Performs dry run without switchover of Data Guard associations.                                                                                                                                                                                                                         |
| -h, --help       | Displays help information about the command and its options.                                                                                                                                                                                                                            |
| -V, --version    | Displays the version of the utility.                                                                                                                                                                                                                                                    |

Examples:

- To validate the database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration switchover -a -i inputs/infra_phxref/  
vmcluster_vmcluster1/database_inputs -w ewallet.p12 --dry-run
```

Specify the password for reading the wallet file.

- To validate a comma-delimited list of dbnames to validate the database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration switchover -i inputs/infra_phxref/  
vmcluster_vmcluster1/database_inputs -w ewallet.p12 -l dbname1,dbname2 --  
dry-run
```

Specify the password for reading the wallet file.

- To perform switchover for all source databases, following successful validation and dry run:

```
/home/opc>./exacloudmigration switchover -a -i inputs/infra_phxref/  
vmcluster_vmcluster1/database_inputs -w ewallet.p12
```

Specify the password for reading the wallet file.

- To perform switchover using the comma-delimited list of source database(s), following successful validation and dry run:

```
/home/opc>./exacloudmigration switchover -i inputs/infra_phxref/  
vmcluster_vmcluster1/database_inputs -w ewallet.p12 -l dbname1,dbname2
```

Specify the password for reading the wallet file.

#### Note

- If the database wallet file is not used, the command prompts for the SYS password for each database.
- If the database wallet file is used, but a password entry for any database is missing, the utility stops execution and displays an error indicating that the password does not exist in the wallet.

## Database failover

Makes the standby database the primary database, and the old primary database becomes disabled standby database.

Ensure that the Data Guard association is enabled between source and target database.

Ensure that the database input file has been generated and modified according to the target platform requirements.

The failover command supports the following options:

**Syntax**

```
exacloudmigration failover [-ahV] -i <inputFile> [-w <walletFile>] [-a|-l <dbNames>] [--dry-run]
```

**Table 2-30 exacloudmigration failover Commands**

| Command          | Description                                                                                                                                                                                                                                                                    |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a, --all        | Performs failover for all databases in the source VM cluster.                                                                                                                                                                                                                  |
| -i, --inputFile  | Specifies the database input file name with complete path.                                                                                                                                                                                                                     |
| -l, --dbNames    | Specifies a comma-delimited list of databases to perform failover.<br>Examples: <ul style="list-style-type: none"> <li>To perform failover for a database: -l <i>dbname1</i></li> <li>To perform failover for multiple databases: -l <i>dbname1,dbname2,dbname3</i></li> </ul> |
| -w, --walletFile | Specifies the wallet file name with the path.                                                                                                                                                                                                                                  |
| --dry-run        | Performs dry run without failover of Data Guard associations.                                                                                                                                                                                                                  |
| -h, --help       | Displays help information about the command and its options.                                                                                                                                                                                                                   |
| -V, --version    | Displays the version of the utility.                                                                                                                                                                                                                                           |

**Examples:**

- To validate the database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration failover -a -i inputs/infra_phxref/
vmcluster_vmcluster1/database_inputs -w ewallet.p12 --dry-run
```

Specify the password for reading the wallet file.

- To validate a comma-delimited list of dbnames to verify the database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration failover -i inputs/infra_phxref/
vmcluster_vmcluster1/database_inputs -w ewallet.p12 -l dbname1,dbname2 --
dry-run
```

Specify the password for reading the wallet file.

- To perform failover for all source databases, following successful validation and dry run:

```
/home/opc>./exacloudmigration failover -a -i inputs/infra_phxref/
vmcluster_vmcluster1/database_inputs -w ewallet.p12
```

Specify the password for reading the wallet file.

- To perform failover using the comma-delimited list of source database(s) provided, following successful validation and dry run:

```
/home/opc>./exacloudmigration failover -i inputs/infra_phxref/
vmcluster_vmcluster1/database_inputs -w ewallet.p12 -l dbname1,dbname2
```

Specify the password for reading the wallet file.

#### ① Note

- If the database wallet file is not used, the command prompts for the SYS password for each database.
- If the database wallet file is used, but a password entry for any database is missing, the utility stops execution and displays an error indicating that the password does not exist in the wallet.

## Reinstate standby

Makes disabled standby database as standby database.

Ensure that the database input file has been generated and modified according to the target platform requirements.

The reinstate command supports the following options:

### Syntax

```
exacloudmigration reinstate [-ahV] -i <inputFile> [-w <walletFile>] [-a|-l <dbNames>] [--dry-run]
```

**Table 2-31 exacloudmigration reinstate Commands**

| Command          | Description                                                                                                                                                                                                                                                                       |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a, --all        | Performs reinstate for all databases in the source VM cluster.                                                                                                                                                                                                                    |
| -i, --inputFile  | Specifies the database input file name with complete path.                                                                                                                                                                                                                        |
| -l, --dbNames    | Specifies a comma-delimited list of databases to perform reinstate.<br>Examples: <ul style="list-style-type: none"> <li>To perform reinstate for a database: -l <i>dbname1</i></li> <li>To perform reinstate for multiple databases: -l <i>dbname1,dbname2,dbname3</i></li> </ul> |
| -w, --walletFile | Specifies the wallet file name with the path.                                                                                                                                                                                                                                     |
| --dry-run        | Performs dry run without reinstate of Data Guard associations.                                                                                                                                                                                                                    |
| -h, --help       | Displays help information about the command and its options.                                                                                                                                                                                                                      |
| -V, --version    | Displays the version of the utility.                                                                                                                                                                                                                                              |

Examples:

- To validate the database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration reinstate -a -i inputs/infra_phxref/  
vmcluster_vmcluster1/database_inputs -w ewallet.p12 --dry-run
```

Specify the password for reading the wallet file.

- To validate a comma-delimited list of dbnames to verify the database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration reinstate -i inputs/infra_phxref/  
vmcluster_vmcluster1/database_inputs -w ewallet.p12 -l dbname1,dbname2 --  
dry-run
```

Specify the password for reading the wallet file.

- To perform reinstate for all databases, following successful validation and dry run:

```
/home/opc>./exacloudmigration reinstate -a -i inputs/infra_phxref/  
vmcluster_vmcluster1/database_inputs -w ewallet.p12
```

Specify the password for reading the wallet file.

- To perform reinstate using the comma-delimited list of database(s), following successful validation and dry run:

```
/home/opc>./exacloudmigration reinstate -i inputs/infra_phxref/  
vmcluster_vmcluster1/database_inputs -w ewallet.p12 -l dbname1,dbname2
```

Specify the password for reading the wallet file.

#### Note

- If the database wallet file is not used, the command prompts for the SYS password for each database.
- If the database wallet file is used, but a password entry for any database is missing, the utility stops execution and displays an error indicating that the password does not exist in the wallet.

## Delete standby

Delete standby database(s).

Ensure that the database input file has been generated and modified according to the requirements.

The delete\_standby command supports the following options:

**Syntax**

```
exacloudmigration delete_standby [-hV] -i <inputFile> -l <dbNames> [--dry-run]
```

**Table 2-32 exacloudmigration delete\_standby Commands**

| Command         | Description                                                                                                                                                                                                                                  |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -i, --inputFile | Specifies the database input file name with complete path.                                                                                                                                                                                   |
| -l, --dbNames   | Specifies a comma-delimited list of databases to perform delete operation.<br>Examples: <ul style="list-style-type: none"> <li>To delete a database: -l dbname1</li> <li>To delete multiple databases: -l dbname1,dbname2,dbname3</li> </ul> |
| --dry-run       | Performs dry run without deleting the standby databases.                                                                                                                                                                                     |
| -h, --help      | Displays help information about the command and its options.                                                                                                                                                                                 |
| -V, --version   | Displays the version of the utility.                                                                                                                                                                                                         |

Examples:

- To validate the database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration delete_standby -i inputs/infra_phxref/
vmcluster_vmcluster1/database_inputs -l dbname1 --dry-run
```

Confirm the name of the database to terminate.

- To validate a comma-delimited list of dbnames to verify the database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration delete_standby -i inputs/infra_phxref/
vmcluster_vmcluster1/database_inputs -l dbname1,dbname2 --dry-run
```

Confirm the name of the database to terminate.

- To delete standby for the specified database, following successful validation and dry run:

```
/home/opc>./exacloudmigration delete_standby -i inputs/infra_phxref/
vmcluster_vmcluster1/database_inputs -l dbname1
```

Confirm the name of the database to terminate.

- To delete a comma-delimited list of standby databases, following successful validation and dry run:

```
/home/opc>./exacloudmigration delete_standby -i inputs/infra_phxref/
vmcluster_vmcluster1/database_inputs -l dbname1,dbname2
```

Confirm the name of the database to terminate.

## List all Data Guard associations for given VM cluster

List all Data Guard associations for the specified VM cluster.

Ensure that the database input file has been generated and modified according to the requirements.

The `list_dg_associations` command supports the following options:

### Syntax

```
exacloudmigration list_dg_associations [-ahV] -i <inputFile> | -s
<sourceVmClusterOCID>
```

**Table 2-33 exacloudmigration list\_dg\_associations Commands**

| Command                   | Description                                                                                                                                                                                                                                                                                                                         |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a, --all                 | Lists Data Guard associations for all databases in the source VM cluster.                                                                                                                                                                                                                                                           |
| -i, --inputFile           | Specifies the database input file name with complete path.                                                                                                                                                                                                                                                                          |
| -s, --sourceVmClusterOCID | Specifies the source VM cluster OCID.                                                                                                                                                                                                                                                                                               |
| -l, --dbNames             | Specifies a comma-delimited list of database names to list the Data Guard associations.<br>Examples: <ul style="list-style-type: none"> <li>To list the Data Guard associations for a database: -l <i>dbname1</i></li> <li>To list the Data Guard associations for multiple databases: -l <i>dbname1,dbname2,dbname3</i></li> </ul> |
| -u, --profile             | Specifies the OCI user profile name (Default profile is DEFAULT).                                                                                                                                                                                                                                                                   |
| -h, --help                | Displays help information about the command and its options.                                                                                                                                                                                                                                                                        |
| -V, --version             | Displays the version of the utility.                                                                                                                                                                                                                                                                                                |

Examples:

- To list all Data Guard associations for all databases:

```
/home/opc>./exacloudmigration list_dg_associations -a -i inputs/infra_phxref/vmcluster_vmcluster1/database_inputs
```
- To list Data Guard associations for a comma-delimited list of database(s) (ExaDB-D):

```
/home/opc>./exacloudmigration list_dg_associations -s ocid1.cloudvmcluster.oc1.phx.anyhq5ftfhjtqccxyzabc -l dbname1,dbname2
```
- To list Data Guard associations for a comma-delimited list of database(s) (ExaDB-C@C):

```
/home/opc>./exacloudmigration list_dg_associations -s ocid1.vmcluster.oc1.phx.anyhq5ftfhjtqccxyzabc -l dbname1,dbname2
```

## List VM cluster Networks

### Note

This command is applicable only to ExaDB-C@C.

List all VM cluster networks for the specified Exadata Infrastructure.

The `list_vm_networks` command supports the following options:

### Syntax

```
exacloudmigration list_vm_networks [-hV] -s <sourceExadataInfraOCID> [-p <profile>]
```

**Table 2-34** exacloudmigration list\_vm\_networks Commands

| Command                      | Description                                                       |
|------------------------------|-------------------------------------------------------------------|
| -s, --sourceExadataInfraOCID | Specifies the source Exadata Infrastructure OCID.                 |
| -p, --profile                | Specifies the OCI user profile name (Default profile is DEFAULT). |
| -h, --help                   | Displays help information about the command and its options.      |
| -V, --version                | Displays the version of the utility.                              |

Examples:

- To list all VM cluster networks for the specified Exadata Infrastructure:

```
/home/opc>./exacloudmigration list_vm_networks -s <OCID>
```

## List the jobs history

List the jobs history.

The `list_jobs` command supports the following options:

### Syntax

```
exacloudmigration list_jobs [-hV] -c <count>
```

**Table 2-35** exacloudmigration list\_jobs Commands

| Command     | Description                                                        |
|-------------|--------------------------------------------------------------------|
| -c, --count | Specifies the count of latest jobs to be displayed (Default is 5). |

**Table 2-35 (Cont.) exacloudmigration list\_jobs Commands**

| Command       | Description                                                  |
|---------------|--------------------------------------------------------------|
| -h, --help    | Displays help information about the command and its options. |
| -V, --version | Displays the version of the utility.                         |

Examples:

- To list the history of the last 10 jobs:

```
/home/opc>./exacloudmigration list_jobs -c 10
```

## Get job details and status

Displays the job details and status.

The get\_job\_status command supports the following options:

### Syntax

```
exacloudmigration get_job_status [-hV] -i <jobid> [-p <profile>]
```

**Table 2-36 exacloudmigration get\_job\_status Commands**

| Command       | Description                                                       |
|---------------|-------------------------------------------------------------------|
| -i, --jobid   | Specifies the ID of the previous job (Default is latest job ID).  |
| -p, --profile | Specifies the OCI user profile name (Default profile is DEFAULT). |
| -h, --help    | Displays help information about the command and its options.      |
| -V, --version | Displays the version of the utility.                              |

Examples:

- To display the job details and status:

```
/home/opc>./exacloudmigration get_job_status -i 1
```

## Create Database from existing database backup

### Note

This command is applicable only to ExaDB-D.

Create Database from existing database backup.

The `create_db_frombackup` command supports the following options:

### Syntax

```
exacloudmigration create_db_frombackup [-ahV] -i <inputFile> [-w
<walletFile>] [-a|-l <dbNames>] [--dry-run]
```

**Table 2-37 exacloudmigration create\_db\_frombackup Commands**

| Command          | Description                                                                                                                                                                                                                                                                                                                               |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a, --all        | Creates new databases from the backups of all existing databases.                                                                                                                                                                                                                                                                         |
| -i, --inputFile  | Specifies the database input file name with complete path.                                                                                                                                                                                                                                                                                |
| -l, --dbNames    | Specifies a comma-delimited list of databases to create databases from the existing backup.<br>Examples: <ul style="list-style-type: none"> <li>To create a database from the existing backup: <code>-l dbname1</code></li> <li>To create multiple databases from the existing backup: <code>-l dbname1,dbname2,dbname3</code></li> </ul> |
| -w, --walletFile | Specifies the wallet file name with the path.                                                                                                                                                                                                                                                                                             |
| --dry-run        | Performs dry run without creating databases.                                                                                                                                                                                                                                                                                              |
| -h, --help       | Displays help information about the command and its options.                                                                                                                                                                                                                                                                              |
| -V, --version    | Displays the version of the utility.                                                                                                                                                                                                                                                                                                      |

Examples:

- To validate the database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration create_db_frombackup -a -i inputs/
infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12 --dry-run
```

- To validate a comma-delimited list of dbnames to verify the database input template file and perform a --dry-run:

```
/home/opc>./exacloudmigration create_db_frombackup -i inputs/infra_phxref/
vmcluster_vmcluster1/database_inputs -l dbname1,dbname2 -w ewallet.p12 --
dry-run
```

- To create databases from the existing backups, following successful validation and dry run:

```
/home/opc>./exacloudmigration create_db_frombackup -a -i inputs/
infra_phxref/vmcluster_vmcluster1/database_inputs -w ewallet.p12
```

- To create databases from the existing backups for a comma-delimited list of databases, following successful validation and dry run:

```
/home/opc>./exacloudmigration create_db_frombackup -i inputs/infra_phxref/
vmcluster_vmcluster1/database_inputs -l dbname1,dbname2 -w ewallet.p12
```

**Note**

- If the database wallet file is not used, the command prompts for the SYS and TDE passwords for each database.
- If the database wallet file is used, but a password entry for any database is missing, the utility stops execution and displays an error indicating that the password does not exist in the wallet.
- All the PDBs will be restored.
- If the source database is created using Customer-Managed Keys, creating the database requires only the SYS password for the new database.
- If the source database is created using Oracle-Managed Keys, creating the database requires two passwords: the TDE password and the SYS password for the new database. By default, the utility uses the password stored in the wallet for both.
- If the TDE password and the new database SYS password need to be different, the recommended approach is to create the new database using the same password and then update the SYS password of the database afterward.

# 3

## Post Migration

Update the connect string with the new SCAN name so that applications and clients can connect to the migrated databases. If SSL/TLS connections are used, update the client wallets accordingly.

# 4

## Logging and Troubleshooting

- [Exadata Cloud Infrastructure Migration Automation Utility logs](#)
- [Review the logs](#)
- [Upload the logs to Service Request](#)
- [Troubleshooting and diagnostics](#)

### Exadata Cloud Infrastructure Migration Automation Utility logs

The Exadata Cloud Infrastructure Migration Automation Utility automatically generates log files during each job execution.

If the utility is extracted to the /home/opc directory, the log files are created under the following location:

```
/home/opc/exacloudmigration_data/logs
```

For example:

```
/home/opc/exacloudmigration_data/logs> ls
job_<jobid>_main.log
job_<jobid>_debug.log
```

Log file descriptions:

- `job_<jobid>_main.log`: Contains details of each step executed for a given operation.
- `job_<jobid>_debug.log`: Contains detailed Java SDK call information and is primarily intended for developer-level debugging.

For example, if the job ID is 1, the log file names appear as follows:

```
/home/opc/exacloudmigration_data/logs> ls
job_1_main.log
job_1_debug.log
```

### Review the logs

If you encounter any issues during the migration process, review the logs in the following order:

1. Check the job status:  
Run the following command to view the current status and details of the job:

```
/home/opc> exacloudmigration get_job_status -i <job number>
```

For example:

```
/home/opc> exacloudmigration get_job_status -i 1
```

2. Locate and review the main utility log file for the job:

```
/home/opc> ./exacloudmigration_data/logs/job_<jobid>_main.log
```

For example:

```
/home/opc/exacloudmigration_data/logs/job_1_main.log
```

3. Refer to the [Known Issues](#) section.  
If the issue is not listed or remains unresolved, proceed to the next step.
4. From the utility log file (for example, /home/opc/exacloudmigration\_data/logs/job\_1\_main.log), locate the Work Request ID entry.  
For example:

Work request ID:

```
ocid1.coreservicesworkrequest.region1.sea.abzwljsbsmatss5hee5nftkft2ex4dal  
erypxsdquw6mvzs222c1ls44ozq
```

5. Review the Work Request in One View.
6. Review DBaaS logs (for Data Guard failures)  
If the failure occurred during Data Guard operations such as dgassociation, switchover, or failover, collect and review DBaaS diagnostic logs for one hour before and after the issue:

```
dbaascli diag collect --startTime YYYY-MM-DDTHH24:MM:SS --endTime YYYY-MM-DDTHH24:MM:SS --dbNames <value> --waitForCompletion false
```

If you need to collect logs for multiple databases, specify a comma-separated list for the --dbNames parameter.

For example:

```
--dbNames db1,db2,db3
```

## Upload the logs to Service Request

If the issue persists after reviewing the logs, create a Service Request (SR) and attach the required details and log files.

- [Create a Service Request](#)
- [Upload Logs to a Service Request](#)

## Create a Service Request

When raising a Service Request (SR) for issues related to the Exadata Cloud Infrastructure Migration Automation Utility, use the following details to ensure your request is routed

**Table 4-1 Service Request details**

| Field            | Value                                                                                                                                                             |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Service Group    | Oracle Database Service                                                                                                                                           |
| Service          | Oracle Cloud Infrastructure – Exadata Cloud Service                                                                                                               |
| Service Category | ExaCloud Infra Migration Automation Utility                                                                                                                       |
| Subcategory      | <ul style="list-style-type: none"><li>• Infrastructure Migration</li><li>• VM Cluster Migration</li><li>• Database Migration</li><li>• Other Migrations</li></ul> |

**Note**

Select the subcategory that best matches the type of migration operation where the issue occurred.

## Upload Logs to a Service Request

If a job fails during the migration process, collect and upload the following information to your Service Request (SR) to assist with troubleshooting.

1. Check the Utility version.  
Run the following command to determine the version of the Exadata Cloud Infrastructure Migration Automation Utility:

```
/home/opc>./exacloudmigration -V
```

2. Retrieve failed job details.  
Use the following command to view the details and status of the failed job:

```
/home/opc>./exacloudmigration get_job_status -i <jobid>
```

For example, failed job ID is 1:

```
/home/opc>./exacloudmigration get_job_status -i 1
```

3. Identify the Template Files used for the job.  
Template files used during job execution are located under:

```
/home/opc> ./inputs/infra_<source_infra_name>/
```

The template file names are typically:

- infrastructure\_inputs
- cloudvmcluster\_inputs
- database\_inputs

You can also identify the specific template file name from the `get_job_status` command output. Look for the line containing the text **Command Executed**.

For example:

```
2025-03-12T10:37:09.893Z [main] INFO - Fetching the status of the JobCommand
Executed : /home/opc>./exacloudmigration delete_standby -i inputs/infra_infrax9m/
vmcluster_ecmvm1/database_inputs -l testdb
Utility version
```

4. Collect Job log files

Compress the contents of the following log directory:

```
/home/opc>./exacloudmigration_data/logs/
```

Upload the zipped log folder to your Service Request.

5. Collect additional logs (if applicable)

If the failure occurred during Data Guard operations such as `dgassociation`, `switchover`, or `failover`, collect additional diagnostic logs from the target cluster.

Run the following command to collect diagnostic logs for one hour before and after the issue:

```
dbaascli diag collect --startTime YYYY-MM-DDTHH24:MM:SS --endTime YYYY-MM-DDTHH24:MM:SS --dbNames <value> --waitForCompletion false
```

If you need to collect logs for multiple databases, specify a comma-separated list of database names for the `--dbNames` parameter.

For example:

```
--dbNames db1,db2,db3
```

## Troubleshooting and diagnostics

For more information on troubleshooting and diagnostics for ExaDB-D and ExaDB-C@C, see:

- ExaDB-D: [Collecting Oracle Diagnostics](#)  
ExaDB-C@C: [Collecting Oracle Diagnostics](#)
- ExaDB-D: [Troubleshooting Exadata Cloud Infrastructure Systems](#)  
ExaDB-C@C: [Troubleshooting Oracle Exadata Database Service on Cloud@Customer Systems](#)

# 5

## Known Issues

This section describes known issues and provides troubleshooting information for the Exadata Cloud Infrastructure Migration Automation Utility.

- [Creation of Data Guard association fails](#)
- [Create database from backup or Data Guard association fails](#)
- [When to configure manual Data Guard](#)
- [Does the utility support cross-tenancy migration?](#)
- [Cluster verification failures due to hostname resolution issues](#)

### Creation of Data Guard association fails

**Description:** DBAAS-70658:[FATAL] [DBAAS-70658] Inconsistent rpm versions detected in the Data Guard clusters for the RPM 'dbcsagent' .

**Cause:** The primary database cluster and the standby cluster have mismatched versions of the dbcs-agent RPM.

For example:

- Primary database cluster: dbcs-agent-update-2.24.0.0.0-241009.1500.x86\_64
- Standby cluster node (exa-dbhost1): dbcs-agent-update-24.3.2.0.0-240828.1206.x86\_64

**Action:** Ensure that all nodes across both the primary and standby clusters have the same versions of the dbcs-agent and dbaastools RPMs installed.

1. Check the current RPM versions.  
Log in to the database server, switch to the root user, and run the following commands:

```
rpm -qa | grep agent
dbcs-agent-2.80L7-22.3.1.0.1_220721.1435.x86_64
dbcs-agent-update-25.1.1.0.0-250107.0908.x86_64
```

```
rpm -qa | grep dbaas
dbaastools_exa-1.0-25.1.1.0.0_250107.0908.x86_64
```

If the versions differ between nodes within the same cluster or across primary and standby clusters, perform the following updates.

2. Update dbcs agent RPMs
  - a. Contact Oracle Support or your cloud operations team to obtain the latest dbcs-agent RPM packages.
  - b. Copy the RPMs to all database servers

- c. Log in to each database server, switch to the root user, and run:

```
dbaascli admin updateAgent --imageTag <image_tag_of_agent_update_rpm>
```

For example:

```
dbaascli admin updateAgent --imageTag dbcs-agent-  
update-24.3.2.0.0-240828.1206.x86_64
```

3. Update the admin agent.

```
dbaascli admin updateAgent --adminAgent --imageTag  
<image_tag_of_admin_update_rpm>
```

For example:

```
dbaascli admin updateAgent --adminAgent --imageTag dbcs-admin-  
update-1.2.0.0.0_240429.1351.x86_64
```

4. Update dbaastools.

```
dbaascli admin updateStack -latest
```

For more information, see [dbaascli admin updateStack](#).

## Create database from backup or Data Guard association fails

**Description:** [FATAL] [DBT-09101] Target environment does not meet some mandatory requirements.

**Cause:** Some mandatory prerequisites are not met.

**Action:** Review the configuration details in the following log file to identify and resolve any missing prerequisites: `/var/opt/oracle/log/testdb/database/createFromBackup/trace.log_2025-03-27_04-21-50AM_276192`. Refer to the installation guide if needed, and after resolving the issues, reattempt the operation.

## When to configure manual Data Guard

In scenarios such as migrating databases between ExaDB-D and ExaDB-C@C systems (for example, from ExaDB-D to ExaDB-C@C or vice versa), the utility does not support enabling Data Guard between the source and target databases. This limitation arises because the migration occurs across different service types. Consequently, the utility also does not support switchover or failover operations in these cases.

In such situations, customers must manually enable Data Guard using one of the following methods:

- **Manual method:** Copy the source database to the target using the `dbaascli database duplicate` command, and then enable Data Guard using the Data Guard Broker.
- **Automated method:** Use Zero Downtime Migration (ZDM) to enable Data Guard.

## Does the utility support cross-tenancy migration?

Yes, the utility supports cross-tenancy migration, including migration of infrastructure, VM clusters, and database homes (dbHomes).

However, due to current limitations of Oracle Exadata Database Service, Data Guard associations between source and target databases are not supported in cross-tenancy migration scenarios.

## Cluster verification failures due to hostname resolution issues

**Description:** The cluster verification utility (CVU) reports failures in several prerequisite checks due to inconsistent hostname resolution across cluster nodes when both IPv4 and IPv6 are enabled.

### List of Failed Tasks:

- **ASM Integrity:** Checks the integrity of Oracle Automatic Storage Management (ASM) across the cluster nodes.
  - Severity: Critical
  - Status: Verification failed
- **Node Connectivity:** Tests connectivity between all cluster nodes.
  - Subnets tested: IPv4 subnet1, IPv4 subnet2
  - Severity: Critical
  - Status: Verification failed
- **Hosts File:** Checks the consistency of the hosts file across cluster nodes.
  - Severity: Critical
  - Status: Verification failed

### Error:

PRVG-13632 : Host "<host-name>" resolves to IP address "<IPv4 address>" on the majority of cluster nodes but to address "<IPv6 address>" on nodes "<host-name>" .

PRVG-13632 : Host "<host-name>" resolves to IP address "<IPv6 address>" on the majority of cluster nodes but to address "<IPv4 address>" on nodes "<host-name>" .

**Cause:** With dual IP stack enabled, hostnames resolve to both IPv4 and IPv6 addresses, which are not consistent across all cluster nodes. This inconsistency causes the CVU checks to fail.

**Solution:** This is a known issue (Bug 37015893). The fix is included in the 19.26.0.0.250121 (January 2025) Oracle Clusterware (OCW) Release Update (RU) and applies to both Grid Infrastructure (GI) and RDBMS homes.

### Workaround:

- Recommended: Upgrade the database to version 19.26 or later to include the fix, and then retry the operation.
- Alternative (not recommended): Disable the dual stack on the admin network of the compute nodes and retry the operation.