Oracle® Zero Downtime Migration

Runbook: Move On-Premise Databases and Oracle Cloud Infrastructure Classic Instances to Exadata Cloud Service

Release 19c
F25675-02
December 2019

Oracle Zero Downtime Migration Runbook: Move On-Premise Databases and Oracle Cloud Infrastructure Classic Instances to Exadata Cloud Service, Release 19c

F25675-02

Primary Authors: Virginia Beecher, Giridhar Reddy Sangala

Contents

Preface

1 About Zero Downtime Migration

2 Prerequisites

3 Preparing for Database Migration

4 Migrate the Database

Preface

This document describes Zero Downtime Migration prerequisites, the steps to set up the Zero Downtime Migration service and connectivity, and migrate on-premises databases and Oracle Cloud Infrastructure Classic instances to Exadata Cloud Service without incurring any significant downtime, by leveraging technologies such as Oracle Active Data Guard. Zero Downtime Migration also supports offline migration using Backup and Recovery.

- Documentation Accessibility

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

1
About Zero Downtime Migration

Zero Downtime Migration gives you a way to simplify moving your on-premise databases and Oracle Cloud Infrastructure Classic instances to Oracle Cloud Infrastructure, Exadata Cloud at Customer, and Exadata Cloud Service, without incurring any significant downtime, by leveraging technologies such as Oracle Active Data Guard.

Zero Downtime Migration uses mechanisms such as backing up the source database to Oracle Cloud Infrastructure Object Storage, creating a standby database (with Oracle Data Guard Maximum Performance protection mode and asynchronous (ASYNC) redo transport mode) in the target environment from the backup, synchronizing the source and target databases, and switching over to the target database as the primary database.

2
Prerequisites

Satisfy the following requirements for Zero Downtime Migration.

- Zero Downtime Migration Service Host Requirements
- Source and Target Database Requirements
- Object Storage Requirements
- SQL*Net Connectivity Between Source and Target Database Servers
- Port Requirements

2.1 Zero Downtime Migration Service Host Requirements

Requirement	Description	Comments
Zero Downtime Migration service host	Zero Downtime Migration software requires a Linux host running on Oracle Linux 7 or later with 100 GB of free storage space. Zero Downtime Migration service host must be able to connect with the source and target database servers through either root user or SSH keys.	You cannot perform database migration without having established connectivity from the Zero Downtime Migration service host to the source database servers, and from the Zero Downtime Migration service host to the target database servers, so make sure that you meet the requirements in Configuring Connectivity From the Zero Downtime Migration Service Host to the Source and Target Database Servers.

2.2 Source and Target Database Requirements

Requirement	Description	Comments
Supported database releases	Zero Downtime Migration supports Oracle Database 11g Release 2 (11.2.0.4) or later.	Both the Source and Target database release versions should be the same.

Requirement	Description	Comments
Source database prerequisites	1. The source database must be running in archive log mode. 2. For Oracle Database 12c Release 2 and later, if the source database does not have Transparent Data Encryption (TDE) enabled, then it is mandatory that you configure the TDE wallet before migration begins. The `WALLET_TYPE` can be `AUTOLOGIN` (preferred) or `PASSWORD` based. 3. Ensure that the wallet `STATUS` is `OPEN` and `WALLET_TYPE` is `AUTOLOGIN` (for an `AUTOLOGIN` wallet type), or `WALLET_TYPE` is `PASSWORD` (for a `PASSWORD` based wallet type). For a multitenant database, ensure that the wallet is open on all PDBs as well as the CDB, and the master key is set for all PDBs and the CDB. `SQL> SELECT * FROM v$encryption_wallet;` 4. If the source is an Oracle RAC database, and `SNAPSHOT CONTROLFILE` is not on a shared location, configure `SNAPSHOT CONTROLFILE` to point to a shared location on all Oracle RAC nodes to avoid the ORA-00245 error during backups to Oracle Object Store. 5. Verify that port 22 on the source and target database server nodes allow incoming connections from Zero Downtime Migration service host. 6. Ensure that the scan listener ports (1521, for example) on the source database servers allow incoming connections from the target database servers and vice versa. Alternate SQL connectivity should be made available if a firewall blocks incoming remote connection using the SCAN listener port. 7. During the migration, disable all RMAN scripts, for example, cron jobs, to avoid multiple RMAN backups.	• If the source database is Oracle release 12.2 and later, TDE is not enabled, so see Setting Up the Transparent Data Encryption Wallet to enable TDE. • In step 4, if the database is deployed on ASM storage, for example: `$ rman target / RMAN> CONFIGURE SNAPSHOT CONTROLFILE NAME TO '+DATA/snapcf_matrix.f';` If the database is deployed on an ACFS file system, specify the shared ACFS location in the above command.

Requirement	Description	Comments
Target database prerequisites	1. A placeholder target database must be created before database migration begins. Ensure that the shape chosen for creating the database from the console can accommodate the source database plus any future sizing requirements. A good guideline is to use a shape similar or larger in size than source database. The target database `db_name` should be same as the source database `db_name`, and the target database `db_unique_name` parameter value must be unique, to ensure that Oracle Data Guard can identify the target as a different database from the source database. 2. The SYS password must match the source database SYS password. 3. The target database version should be the same as the source database version. The target database patch level should also be the same as (or higher than) the source database. 4. Transparent Data Encryption (TDE) should be enabled and ensure that the wallet `STATUS` is `OPEN` and `WALLET_TYPE` is `AUTOLOGIN` (for an `AUTOLOGIN` wallet type), or `WALLET_TYPE` is `PASSWORD` (for a `PASSWORD` based wallet type). `SQL> SELECT * FROM v$encryption_wallet;` 5. If the target is an Oracle RAC database, then you must set up SSH connectivity without a passphrase between the Oracle RAC servers for the oracle user. 6. Check the size of the disk groups and usage on the target database (ASM disk groups or ACFS file systems) and make sure adequate storage is provisioned and available on the target database servers. 7. Make sure adequate storage is provisioned and available on the	• The placeholder target database is overwritten during migration, but it retains the overall configuration. • If the target database environment is at a higher patch level than the source database (for example, if the source database is at Oct 2018 PSU/BP and the target database is at Jan 2019 PSU/BP), then you must run datapatch after database migration. • For Exadata Cloud Service, the target environment must be installed with latest DBaaS Tooling RPM. • For Exadata Cloud Service, provision the target database from the console without enabling automatic backups (in the **Configure database backups** section do not select the **Enable automatic backups** option.

Requirement	Description	Comments
	object store to accommodate the source database backup. 8. Verify that ports 22 and 1521 on the target servers are open and not blocked by a firewall.	

2.3 Object Storage Requirements

Requirement	Description	Comments
Object Storage requirements	Access to Oracle Cloud Object Storage and permission to create the bucket to hold database backup are required. During the migration job submission, you must have the object store Swift token password.	Make sure adequate storage is provisioned and available on the object store to accommodate the source database backup.

2.4 SQL*Net Connectivity Between Source and Target Database Servers

Requirement	Description	Comments
SQL*Net connectivity between source and target database servers	For the Oracle Data Guard setup and sync, open SQL*Net (DB Ports, Scan ports) between the source (on-premises databases or Oracle Cloud Infrastructure Classic instances) and the target Exadata Cloud Service. For example, 1521 is the database port on the Exadata Cloud Service target.	none

2.5 Port Requirements

Initiator	Target	Protocol	Port	Purpose	Comments
Zero Downtime Migration service host	Source and Target	TCP	22	SSH	Authentication-based operations to run Zero Downtime Migration operational phases. Source and Target nodes should accept incoming connections from the Zero Downtime Migration service host.

Initiator	Target	Protocol	Port	Purpose	Comments
Source	Target	TCP	1521	SQL*Net	Should allow Oracle client connections to the database over Oracle's SQL*Net protocol to perform database queries, Data Guard sync and configuration. Note: If you are using a non-default port number (that is, something other than port 1521) for the local listener address, then the non-default port should allow connections.
Target	Source	TCP	1521	SQL*Net	Should allow Oracle client connections to the database over Oracle's SQL*Net protocol. Allows redo log shipping if on-premises database needs to be in sync with the new primary on Oracle Cloud after switchover. If there is no communication possible from Oracle Cloud to on-premises host then set `SKIP_FALLBACK` to `TRUE` in the response file to avoid this communication. Note: If you are using a non-default port number (that is, something other than port 1521) for the local listener address, then the non-default port should allow connections.
Source	Oracle Cloud Object Store Service	SSL	443	Create a backup of the source database to the specified Oracle Cloud Object store bucket.	If the chosen backup method involves Oracle Cloud Object Store Service as the backup medium, then access ports as documented Oracle Cloud Object Store Service applies.

Initiator	Target	Protocol	Port	Purpose	Comments
Target	Oracle Cloud Object Store Service	SSL	443	Restore backup of the source database from the specified Oracle Cloud Object store bucket to the target.	If the chosen backup method involves Oracle Cloud Object Store Service as the backup medium, then access ports as documented Oracle Cloud Object Store Service applies.

3 Preparing for Database Migration

The following topics describe how to prepare for database migration.

- Setting Up Zero Downtime Migration Software
- Configuring Connectivity From the Zero Downtime Migration Service Host to the Source and Target Database Servers
- Configuring Connectivity Between the Source and Target Database Servers
- Prepare the Response File Template
- Preparing for Automatic Application Switchover
- Customizing a Migration Job

3.1 Setting Up Zero Downtime Migration Software

Requirement	Description	Comments

Set up the Zero Downtime Migration software	1. You may use an existing user, or, on the Zero Downtime Migration service host, as root user, create a `zdm` group and add `zdmuser` user to the group.
	For example
	`root> groupadd zdm`
	`root> useradd -g zdm zdmuser`
	Note that you can use any non root user to set up Zero Downtime Migration. For the examples in this document all of the commands are shown run as `zdmuser`.
	2. Download the Zero Downtime Migration software kit from https://www.oracle.com/database/technologies/rac/zdm-downloads.html to the Zero Downtime Migration service host.
	3. Refer to the README in the download for the mandatory packages and install them.
	4. Change to the directory to where Zero Downtime Migration software is downloaded.
	`zdmuser> cd zdm_download_directory`
	Run the Zero Downtime Migration installation script.
	`zdmuser>./zdminstall.sh setup oraclehome=zdm_oracle_home oraclebase=zdm_base_directory ziploc=zdm_software_location -zdm`
	Where `zmdinstall.sh` is the installation script.
	`oraclehome` is the Oracle Home where the Zero Downtime Migration kit is installed.
	`oraclebase` is the base directory where all of the Zero Downtime Migration configuration files, logs, and other artifacts are stored.
	`ziploc` is the location of the compressed software file (ZIP) included in the Zero Downtime Migration kit.
		Ignore the following messages which are displayed on the terminal at the end of installation. There is no need to run these scripts.
		"As a root user, execute the following script(s):
		1. `/u01/app/zdmhome/inventory/orainstRoot.sh`
		2. `/u01/app/zdmhome/root.sh`"

Requirement	Description	Comments

	For example,
	`zdmuser>./zdminstall.sh setup oraclehome=/u01/app/zdmhome`
	`oraclebase=/u01/app/zdmbase ziploc=/u01/app/oracle/zdm/shiphome/zdm_home.zip -zdm`
	Hereafter, the `oraclehome` value is referred to as `ZDM_HOME`, and the `oraclebase` value is referred to as `ZDM_BASE`.
	5. Start the Zero Downtime Migration service as user zdmuser.
	`zdmuser> /u01/app/zdmhome/bin/zdmservice start`
	You must start `zdmservice` before you can migrate your databases using Zero Downtime Migration.
	6. To stop the Zero Downtime Migration service, run the following command.
	`zdmuser> /u01/app/zdmhome/bin/zdmservice stop`
	7. Verify that the Zero Downtime Migration service installation is successful.
	When you run the following command, the output should be similar to that shown here.
	`zdmuser> /u01/app/zdmhome/bin/zdmservice status`
	`---------------------------------------`
	`Service Status`
	`---------------------------------------`
	`Running: true`
	`Tranferport: 5000-7000`
	`Conn String: jdbc:derby:/u01/app/zdmbase/derbyRepo;create=true`
	`Repo Path: /u01/app/zdmbase/derbyRepo`
	`RMI port: 8895`	

Requirement	Description	Comments

	`HTTP port: 8896`
	`Wallet path: /u01/app/zdmbase/crsdata/fopds/security`	

3.2 Configuring Connectivity From the Zero Downtime Migration Service Host to the Source and Target Database Servers

Requirement	Description	Comments

Configure connectivity from the Zero Downtime Migration service host to the source and target database servers	1. On the Zero Downtime Migration service host, verify that the authentication key pairs are available without a passphrase for the Zero Downtime Migration software installed user.
	If a new key pair must be generated without the passphrase, then, as a Zero Downtime Migration software installed user, generate new key pairs as described in Generating a Private SSH Key Without a Passphrase.
	2. Rename the private key file.
	Rename the *ZDM_installed_user_home*/.ssh/id_rsa file name to *ZDM_installed_user_home*/.ssh/*ZDM_service_host_name*.ppk
	3. Add the contents of the *ZDM_installed_user_home*/.ssh/id_rsa.pub file to the *opc_user_home*/.ssh/authorized_keys file on all of the source and target database servers.
	Note that the `opc` user is a standard Oracle Cloud user that is used to access the Oracle Cloud Infrastructure Classic and Exadata Cloud Service database servers.
	4. Add the source and target database server names and IP address details to the Zero Downtime Migration service host `/etc/hosts` file.
	For example,
	`#OCI-C public IP of two node RAC server details`
	`192.0.2.1 zdm122011`
	`192.0.2.2 zdm122012`
	`#ExaCS public IP of two node RAC server details`
	`192.0.2.3 exacstarget1`
	`192.0.2.4 exacstarget2`
	5. Make certain that port 22 in the source and target database servers accept incoming connections from the Zero Downtime Migration service host.
		Note 1: If you have root access to source database server, then configuring connectivity through SSH keys for the source database server is not required.
		Note 2: If you configure connectivity through SSH keys, then the Zero Downtime Migration service host should be able to connect to the source and target database servers using a private key file, without prompting for any password.
		Note 3: If your Zero Downtime Migration service host needs proxy details to connect to the target server, then along with SSH keys you also need to configure the *ZDM_installed_user_home*/.ssh/config with the following details, otherwise configuring the *ZDM_installed_user_home*/.ssh/config file is not required.
		`cat ZDM_installed_user_home/.ssh/config`
		`Host *`
		`ServerAliveInterval 10`
		`ServerAliveCountMax 2`
		Host *Target_server_name*
		HostName *Target_server_IP_address*
		IdentityFile *Private_key_file_location*
		User *Target_user_login*
		ProxyCommand /usr/bin/nc --proxy *proxy_url*:*port* %h %p
		For example
		`Host *`
		`ServerAliveInterval 10`
		`ServerAliveCountMax 2`
		`Host exacstarget1`
		`HostName 192.0.2.3`
		`IdentityFile /home/zdmuser/.ssh/zdm_service_host.ppk`
		`User opc`
		`ProxyCommand /usr/bin/nc --proxy www-proxy-example.com:80 %h %p`

Requirement	Description	Comments

	6. Test the connectivity from the Zero Downtime Migration service host to all source and target database servers.
	`zdmuser> ssh -i ZDM_service_node_private_key_file_location user@source/target_database_server_name`
	For example,
	`zdmuser> ssh -i /home/zdmuser/.ssh/zdm_service_node.ppk opc@zdm122011`
	`zdmuser> ssh -i /home/zdmuser/.ssh/zdm_service_node.ppk opc@zdm122012`
	`zdmuser> ssh -i /home/zdmuser/.ssh/zdm_service_node.ppk opc@exacstarget1`
	`zdmuser> ssh -i /home/zdmuser/.ssh/zdm_service_node.ppk opc@exacstarget2`
		Note 4: The `authorized_keys` file permissions should be similar to the following.
		`/home/opc/.ssh>ls -l authorized_keys`
		`-rw------- 1 opc opc 1679 Oct 16 10:05 authorized_keys`

3.3 Configuring Connectivity Between the Source and Target Database Servers

Requirement	Description	Comments

Configure connectivity between the source and target database servers	You can configure connectivity between the source and target database servers using one of two options.
	Option 1
	The source database server specified in the `ZDMCLI` command `-sourcenode` parameter can connect to the target database instance over target SCAN through the respective SCAN port and vice versa. The SCAN of the target should be resolvable from the source database server, and the SCAN of the source should resolve from the target server.
	Having connectivity from both sides, you can synchronize between the source database and target database from either side. If the source database server SCAN cannot be resolved from the target database server, then the `SKIP_FALLBACK` parameter in the response file must be set to `TRUE`, and you cannot synchronize between the target database and source database.
	To test the connectivity for Option 1:
	1. Test connectivity from the source to target environments.
	Add the TNS entry of the target database to the source database server `tnsnames.ora` file.
	`[oracle@zdm122011 ~]tnsping target-tns-string`
	2. Test connectivity from the target to the source environment.
	Add the TNS entry of the source database to the target database server `tnsnames.ora` file.
	`[oracle@exacstarget1~] tnsping source-tns-string`
	Option 2
	If connectivity through SCAN and the SCAN port is not possible between the source and target database servers, set up an SSH tunnel from the source database server to the target database server using the procedure below.
	Using this option, you will not be able to synchronize between the target database and source database.
	To set up an SSH tunnel on the source database servers for the root user:
		Ensure that the source database server is able to connect to target database server without prompting for any password, otherwise Data Guard Sync cannot occur from the source to the target side.

Requirement	Description	Comments

	1. Generate a private SSH key file without a passphrase for the opc user on the target server.
	See Generating a Private SSH Key Without a Passphrase.
	If the target is an Oracle RAC database, then generate a private SSH key file without a passphrase from the first Oracle RAC server.
	2. Add the contents of the target server `opc_user_home/.ssh/id_rsa.pub` file into the source server `opc_user_home/.ssh/authorized_keys` file.
	If the source is an Oracle RAC database, add the contents of the target server `opc_user_home/.ssh/id_rsa.pub` file into the `opc_user_home/.ssh/authorized_keys` file on all Oracle RAC source servers.
	3. Copy the target server private SSH key file onto the source server in the `/root/.ssh/` directory.
	If the source is an Oracle RAC database, copy the file into all of the source servers. For better manageability, keep the private SSH key file name the same as the target server name, and keep the .ppk extension. For example, `exacstarget1.ppk` (where `exacstarget1` is the target server name).
	The file permissions should be similar to the following.
	`/root/.ssh>ls -l exacstarget1t.ppk`
	`-rw------- 1 root root 1679 Oct 16 10:05 exacstarget1.ppk`
	4. Put the following entries in the source server /root/.ssh/config file.
	`Host *`
	`ServerAliveInterval 10`
	`ServerAliveCountMax 2`
	`Host Target_server_name`
	`HostName Target_server_IP_address`	

Requirement	Description	Comments
	IdentityFile *Private_key_file_location* User *Target_user_login* Where *Target_server_name* is the target database server name without the domain name. For an Oracle RAC database, use the first Oracle RAC server name without the domain name. *Target_server_IP_address* is the target database server IP address. For an Oracle RAC database use the first Oracle RAC server IP address. *Private_key_file_location* is the location of the private key file. *Target_user_login* is the OS user used to access the target database servers. For example, after specifying the relevant values, the /root/.ssh/config file contents similar to the following. Host * ServerAliveInterval 10 ServerAliveCountMax 2 Host exacstarget1 HostName 192.0.2.3 IdentityFile ~/.ssh/exacstarget1.ppk User opc The file permissions should be similar to the following. /root/.ssh>ls -l config -rw------- 1 root root 1679 Oct 16 10:05 config In the above example, the target server name is exacstarget1, and the target server public IP address is 192.0.2.3. If the source is an Oracle RAC database, then copy the same /root/.ssh/config file onto all of the source Oracle RAC database servers. Make sure that you can SSH to first target server from the source server before you enable the SSH tunnel.	

Requirement	Description	Comments

	For an Oracle RAC database, test the connection from all of the source servers to the first target server.
	Test the connectivity using the private key file and ensure source database server is able to connect to target database server without prompting for any password.
	`[root@zdm122011 ~] ssh -i /root/.ssh/exacstarget1.ppk opc@exacstarget1`
	`[root@zdm122012 ~] ssh -i /root/.ssh/exacstarget1.ppk opc@exacstarget1`
	5. Run the following command on the source server to enable the SSH tunnel.
	ssh -f *Target_hostname_without_domain_name* -L \ *ssh_tunnel_port_number*:*Target_server_IP_address*:*Target_server_listener_port* -N
	Where *Target_hostname_without_domain_name* is the target database server name without a domain name. For an Oracle RAC database use the first Oracle RAC server name without domain name.
	The *ssh_tunnel_port_number* is any available ephemeral port in the range (1024-65545). Make sure that the SSH tunnel port is not used by any other process in the server before using it.
	Target_server_listener_port is the target database listener port number. The listener port must be open between the source database servers and target servers.
	Target_server_IP_address is configured based on database architecture. For a single instance database, specify the Target server IP address. For an Oracle RAC database, specify the Target scan name with the domain name.
	If the scan name with domain name is not resolvable or not working,	

Requirement	Description	Comments

	then specify the IP address obtained using the `LSNRCTL STATUS` command output from the target server. For example
	`Listening Endpoints Summary...`
	`(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER)))`
	`(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.0.2.9)(PORT=1521)))`
	`(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.0.2.10)(PORT=1521)))`
	The following is an example of the command run to enable the SSH tunel.
	`[root@zdm122011 ~]ssh -f exacstarget1 -L 9001:192.0.2.9:1521 -N`
	For an Oracle RAC database, this step must be repeated on all of the source servers.
	6. Test the connectivity with SSH tunnel for Option 2.
	For Oracle RAC, this step must be repeated on all of the source servers.
	Log in to source server, switch to the oracle user and source the database environment and run the following command.
	`tnsping localhost:`*`ssh_tunnel_port`*
	For example
	`[oracle@zdm122011 ~] tnsping localhost:9001`
	The command output is similar to the following.
	`TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 22-JAN-2019 05:41:57`
	`Copyright (c) 1997, 2014, Oracle. All rights reserved.`
	`Used parameter files:`	

Requirement	Description	Comments

	`Used HOSTNAME adapter to resolve the alias`
	`Attempting to contact(DESCRIPTION=(CONNECT_DATA= (SERVICE_NAME=))(ADDRESS=(PROTOCOL=TCP)(HOST=127.0.0.1)(PORT=9001)))`
	`OK (50 msec)`
	If `tnsping` does not work, then the SSH tunnel is not enabled.	

3.4 Prepare the Response File Template

Prepare the response file template according to whether you will perform the migration online or offline.

- Preparing a Response File for Online Migration
- Preparing a Response File for Offline Migration (Backup and Recovery)

3.4.1 Preparing a Response File for Online Migration

Requirement	Description	Comments

Prepare the response file for online migration	Get the response file template from `$ZDM_HOME/rhp/zdm/template/zdm_template.rsp`, and edit the folowing settings.
	• Set `TGT_DB_UNIQUE_NAME` to the target database `db_unique_name` value.
	• Set `MIGRATION_METHOD` to `DG_OSS`, where `DG` stands for Data Guard and `OSS` stands for for Object Storage service.
	• Set `PLATFORM_TYPE` to `EXACS`.
	• If an SSH proxy is required to access the source database server from the Zero Downtime Migration service host, set `SRC_HTTP_PROXY_URL` and `SRC_HTTP_PROXY_PORT`.
	• If an SSH proxy is required to access the target database server from the Zero Downtime Migration service host, set `TGT_HTTP_PROXY_URL` and `TGT_HTTP_PROXY_PORT`.
	• If SSH tunneling is set up, set the `TGT_SSH_TUNNEL_PORT` parameter.
	• Specify the target database data files storage (ASM or ACFS) properties as appropriate for (`TGT_DATADG`, `TGT_REDODG`, and `TGT_RECODG`) or (`TGT_DATAACFS`, `TGT_REDOACFS`, and `TGT_RECOACFS`).
	• Set `HOST` and `OPC_CONTAINER` with the Object Store URL and bucket name.
	• Set `SKIP_FALLBACK=TRUE` if you do not want to ship redo logs from the target to the source standby, either voluntarily or because there is no connectivity between the target and the source.
	• Set `SHUTDOWN_SRC=TRUE` if, after the database migration, you want to shut down the source database.
	• Set `SRC_RMAN_CHANNELS` with the number of RMAN channels to be allocated at the source and used to perform RMAN backups. The default is 10.
	• Set `TGT_RMAN_CHANNELS` with the number of RMAN channels to be
		For example, copy the response file to `$ZDM_HOME/rhp/zdm/template/zdm_template_ZDM12201.rsp` and add values to the settings based on source and target as shown below.
		`TGT_DB_UNIQUE_NAME=ZDM12201_phx1xx`
		`MIGRATION_METHOD=DG_OSS`
		`PLATFORM_TYPE=EXACS`
		`TGT_HTTP_PROXY_URL=www-proxy-example.com`
		`TGT_HTTP_PROXY_PORT=80`
		`TGT_SSH_TUNNEL_PORT=9001`
		`TGT_DATADG=+DATAC1`
		`TGT_REDODG=+RECOC1`
		`TGT_RECODG=+RECOC1`
		`HOST=https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/tenancyname`
		`OPC_CONTAINER=DEMOZDM`
		`SKIP_FALLBACK=TRUE`
		`SHUTDOWN_SRC=TRUE`
		`SRC_RMAN_CHANNELS=8`
		You can find a sample response file in `/$ZDM_HOME/rhp/zdm/template/zdm_template.rsp`.

Requirement	Description	Comments
	allocated at the target and used to perform RMAN restore. The default is 10.	

3.4.2 Preparing a Response File for Offline Migration (Backup and Recovery)

Requirement	Description	Comments

Prepare the response file for offline migration (using Backup and Recovery)	Get the response file template from `$ZDM_HOME/rhp/zdm/template/zdm_template.rsp`, and edit the following settings.
	• Set `TGT_DB_UNIQUE_NAME` to the target database `db_unique_name` value.
	• Set `PLATFORM_TYPE` to the appropriate value, depending on your target environment.
	• For Exadata Cloud Service, set `PLATFORM_TYPE=EXACS`.
	• Set `MIGRATION_METHOD` to `BACKUP_RESTORE_OSS`, where OSS stands for Object Storage Service.
	• Specify the target database data files storage (ASM or ACFS) properties as appropriate. For ASM, set `TGT_DATADG`, `TGT_REDODG`, and `TGT_RECODG`. For ACFS set `TGT_DATAACFS`, `TGT_REDOACFS`, and `TGT_RECOACFS`.
	• If an SSH proxy is required to access the source database server from the Zero Downtime Migration service host, set `SRC_HTTP_PROXY_URL` and `SRC_HTTP_PROXY_PORT`.
	• If an SSH proxy is required to access the target database server from the Zero Downtime Migration service host, set `TGT_HTTP_PROXY_URL` and `TGT_HTTP_PROXY_PORT`.
	• Set `HOST` and `OPC_CONTAINER` with the Object Store URL and bucket name.
	• Set `SHUTDOWN_SRC=TRUE` if, after the database migration, you want to shut down the source database.
	• Set `SRC_RMAN_CHANNELS` with the number of RMAN channels to be allocated at the source and used to perform RMAN backups. The default is 10.
	• Set `TGT_RMAN_CHANNELS` with the number of RMAN channels to be allocated at the target and used to perform RMAN restore. The default is 10.
		For example, copy the response file template to `$ZDM_HOME/rhp/zdm/template/zdm_template_ZDM12201.rsp` and enter values for the settings based on the source and target as shown here.
		`TGT_DB_UNIQUE_NAME=ZDM12201_phx1xx`
		`MIGRATION_METHOD=BACKUP_RESTORE_OSS`
		`PLATFORM_TYPE=EXACS`
		`TGT_HTTP_PROXY_URL=www-proxy-example.com`
		`TGT_HTTP_PROXY_PORT=80`
		`TGT_DATADG=+DATAC1`
		`TGT_REDODG=+RECOC1`
		`TGT_RECODG=+RECOC1`
		`HOST=https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/tenancyname`
		`OPC_CONTAINER=DEMOZDM`
		`SHUTDOWN_SRC=TRUE`
		`SRC_RMAN_CHANNELS=8`
		You can find a sample response file in `$ZDM_HOME/rhp/zdm/template/zdm_template.rsp`.

3.5 Preparing for Automatic Application Switchover

Requirement	Description	Comments

Prepare for automatic application switchover	To minimize or eliminate service interruptions on the application after you complete the database migration and switchover, prepare your application to automatically switch over connections from the source database to the target database.	In the following example connect string, the application connects to the source database, and when it is not available the connection is switched over to the target database.
		`(DESCRIPTION=(FAILOVER=on)(LOAD_BALANCE=on)(CONNECT_TIMEOUT=3)(RETRY_COUNT=3)(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=source_database_scan)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=target_database_scan)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=zdm_prod_svc)))`
		On the source database, create the service, named `zdm_prod_svc` in the examples.
		`srvctl add service -db clever -service zdm_prod_svc -role PRIMARY -notification TRUE -session_state dynamic -failovertype transaction -failovermethod basic -commit_outcome TRUE -failoverretry 30 -failoverdelay 10 -replay_init_time 900 -clbgoal SHORT -rlbgoal SERVICE_TIME -preferred clever1,clever2 -retention 3600 -verbose`

3.6 Customizing a Migration Job

Requirement	Description	Comments

Customize the migration job	You can customize the Zero Downtime Migration workflow by registering action scripts or plug-ins as pre-actions or post-actions to be performed as part of the operational phases involved in your migration job.
	The following topics describe how to customize a migration job.
	Registering Action Plug-ins
	Custom plug-ins must be registered to the Zero Downtime Migration service host to be plugged in as customizations for a particular operational phase.
	Determine the operational phase the given plug-in has to be associated with, and run the `ZDMCLI` command `add useraction`, specifying `-optype MIGRATE_DATABASE` and the respective phase of the operation, whether the plug-in is run -pre or -post relative to that phase, and any on-error requirements. You can register custom plug-ins for operational phases after `ZDM_SETUP_TGT` in the migration job workflow.
	What happens at runtime if the user action encounters an error can be specified with the `-onerror` option, which you can set to either `ABORT`, to end the process, or `CONTINUE`, to continue the migration job even if the custom plug-in exits with an error. See the example command usage below.
	Use the Zero Downtime Migration software installed user (for example, `zmduser`) to add useractions to a database migration job. Adding user actions `zdmvaltgt` and `zdmvalsrc` with the `add useraction` command would look like the following.
	`zdmuser>$ZDM_HOME/bin/zdmcli add useraction -useraction zdmvaltgt -optype MIGRATE_DATABASE -phase ZDM_VALIDATE_TGT -pre -onerror ABORT -actionscript /home/zdmuser/useract.sh`
	`zdmuser>$ZDM_HOME/bin/zdmcli add useraction -useraction zdmvalsrc -optype MIGRATE_DATABASE -phase ZDM_VALIDATE_SRC -pre -onerror`
		none

Requirement	Description	Comments

	`CONTINUE -actionscript /home/zdmuser/useract1.sh`
	In the above command, the scripts `/home/zdmuser/useract.sh` and `/home/zdmuser/useract1.sh` are copied to the Zero Downtime Migration service host repository, and they are run if they are associated with any migration job run using an action template.
	Creating an Action Template
	After the user action plug-ins are registered, you create an action template that combines a set of action plug-ins which can be associated with a migration job.
	An action template is created using the `ZDMCLI` command `add imagetype`, where the image type, `imagetype`, is a bundle of all of the user actions required for a specific type of database migration.
	Create an image type that associates all of the user action plug-ins needed for the migration of the database. Once created, the image type can be reused for all migration operations for which the same set of plug-ins are needed.
	The base type for the image type created here must be `CUSTOM_PLUGIN`, as shown in the example below.
	For example, you can create an image type `ACTION_ZDM` that bundles both of the user actions created in the previous example, `zdmvalsrc` and `zdmvaltgt`.
	`zdmuser>$ZDM_HOME/bin/zdmcli add imagetype -imagetype ACTION_ZDM -basetype CUSTOM_PLUGIN -useractions zdmvalsrc,zdmvaltgt`
	Updating Action Plug-ins
	You can update action plug-ins registered with the Zero Downtime Migration service host.
	The following example shows you how to modify the user action `zdmvalsrc` to be a -post action, instead of a -pre action.
	`zdmuser>$ZDM_HOME/bin/zdmcli modify useraction -useraction zdmvalsrc -phase`	

Requirement	Description	Comments

	`ZDM_VALIDATE_SRC -optype MIGRATE_DATABASE -post`
	This change is propagated to all of the associated action templates, so you do not need to update the action templates.
	Associating an Action Template with a Migration Job
	When you run a migration job you can specify the image type that specifies the plug-ins to be run as part of your migration job.
	As an example, run the migration command specifying the action template `ACTION_ZDM` created in previous examples, `-imagetype ACTION_ZDM`, including the image type results in running the `useract.sh` and `useract1.sh` scripts as part of the migration job workflow.
	By default, the action plug-ins are run for the specified operational phase on all nodes of the cluster. If the access credential specified in the migration command option `-tgtarg2` is unique for a specified target node, then an additional auth argument should be included to specify the auth credentials required to access the other cluster nodes. For example, specify `-tgtarg2 nataddrfile:auth_file_with_node_and_identity_file_mapping`.
	A typical `nataddrfile` for a 2 node cluster with node1 and node2 is shown here.
	`node1:node1:identity_file_path_available_on_zdmservice_node`
	`node2:node2:identity_file_path_available_on_zdmservice_node`	

4

Migrate the Database

The following topics instruct you on running the migration job.

- Evaluate the Database Migration Process
- Query the Migration Job Status
- Migrate the Database
- List Migration Job Phases
- Pause and Resume a Migration Job
- Rerun a Migration Job
- Post-Migration Tasks

4.1 Evaluate the Database Migration Process

Requirement	Description	Comments

Verify prechecks and setup	To evaluate the database migration process, use the `-eval` option with the `ZDMCLI migrate database` command.
	Have the completed response file available, and have the necessary credentials ready, such as the source database SYS password, Object Store (Bucket) swift authentication token, and, if the source server is accessed through the root user, then root user password.
	If connectivity to the source database server users root credentials, then the command is
	`zdmuser>$ZDM_HOME/bin/zdmcli migrate database -sourcedb source_db_unique_name_value -sourcenode source_database_server_name -srcroot -targetnode target_database_server_name -targethome target_database_ORACLE_HOME -backupuser Object_store_login_user_name -rsp response_file_location -tgtauth zdmauth -tgtarg1 user:target_database_server_login_user_name -tgtarg2 identity_file:ZDM_installed_user_private_key_file_location -tgtarg3 sudo_location:/usr/bin/sudo -eval`
	If connectivity to the source database server uses an SSH key, then the command is:
	`zdmuser>$ZDM_HOME/bin/zdmcli migrate database -sourcedb source_db_unique_name_value -sourcenode source_database_server_name -srcauth zdmauth -srcarg1 user:source_database_server_login_user_name -srcarg2 identity_file:ZDM_installed_user_private_key_file_location -srcarg3 sudo_location:/usr/bin/sudo -targetnode target_database_server_name -targethome target_database_ORACLE_HOME_value -backupuser`
		It is highly recommended that for each migration you run `migrate database` in evaluation mode first. The evaluation allows you to correct any potential problems in the setup and configuration before performing the actual migration. In evaluation mode, the migration process runs without effecting the changes to source and target databases. It is safe to run the command with the `-eval` option as many times as needed before running the actual migration.
		Zero Downtime Migration performs the following pre-checks.
		• Discovers information about the source database
		• Discovers information about the target database
		• Sets up Zero Downtime Migration helper modules on the source database server
		• Sets up Zero Downtime Migration helper modules on the target database server
		• Generates random password for encrypting RMAN backup
		This password is required to take a backup of the source database onto the Object Store.
		• Validates the source database
		– Archive log mode check
		– TDE setup
		– SYS password
		– Patch level
		• Validates the target database
		– Patch level
		– Data files storage locations (ASM disk group or ACFS file system)
		• Validates Object Store credentials
		Example
		For the prompts, specify the source database SYS password. If the backup destination is Object Store (Bucket), then specify the your swift authentication token. If the backup destination is Storage Classic (Container), then specify your tenancy login password. If the source server is accessed using the root user, then specify the root user password.
		If connectivity to the source database server uses the root credentials then command would be

Requirement	Description	Comments
	Object_store_login_user_name -rsp *response_file_location* -tgtauth zdmauth -tgtarg1 user:*target_database_server_login_user_name* -tgtarg2 *identity_file*:*ZDM_installed_user_private_key_file_location* -tgtarg3 *sudo_location*:/usr/bin/sudo -eval Note that if a source single instance database is deployed without a Grid Infrastructure home, then in the above command use -sourcesid in place of -sourcedb. Also, if a source database is configured for a PASSWORD based wallet, then add the -tdekeystorepasswd option to the command above, and for the prompt, specify the source database TDE keystore password value. The migrate database command checks for connectivity to the source and target database servers. The migrate database -eval command checks for patch compatibility between the source and target home patch level, and expects the target home patch level to be equal to or higher than the source. If the target home patch level is not as expected, then the migration job is stopped and missing patches are reported. You can either patch the target home with the necessary patches, or you can force continue the migration by appending the –ignore PATCH_CHECK or -ignore ALL option to the migration command.	zdmuser>$ZDM_HOME/bin/zdmcli migrate database -sourcedb ZDM12201_phx1sp -sourcenode zdm122011 -srcroot -targetnode exacstarget1 -targethome /u02/app/oracle/product/12.2.0/dbhome_3 -backupuser backup_user@example.com -rsp /u01/app/zdmhome/rhp/zdm/template/zdm_template_ZDM12201.rsp -tgtauth zdmauth -tgtarg1 user:opc -tgtarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk -tgtarg3 sudo_location:/usr/bin/sudo -ignore ALL -eval Enter source database ZDM12201_phx1sp SYS password: Enter source user "root" password: Enter user "backup_user@example.com" password: Then the following message is displayed with the job ID number. Make note of the job ID number for later use. Operation "zdmcli migrate database" scheduled with the job ID "5". If connectivity to the source database server uses an SSH key, then command is zdmuser>$ZDM_HOME/bin/zdmcli migrate database -sourcedb ZDM12201_phx1sp -sourcenode zdm122011 -srcauth zdmauth -srcarg1 user:opc -srcarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk -srcarg3 sudo_location:/usr/bin/sudo -targetnode exacstarget1 -targethome /u02/app/oracle/product/12.2.0/dbhome_3 -backupuser backup_user@example.com -

Requirement	Description	Comments

		`rsp /u01/app/zdmhome/rhp/zdm/template/zdm_template_ZDM12201.rsp -tgtauth zdmauth -tgtarg1 user:opc -tgtarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk -tgtarg3 sudo_location:/usr/bin/sudo -ignore ALL -eval`
		`Enter source database ZDM12201_phx1sp SYS password:`
		`Enter user "backup_user@example.com" password:`
		Then the following message is displayed with the job ID number. Make note of the job ID number for later use.
		`Operation "zdmcli migrate database" scheduled with the job ID "5".`
		See Prechecks Verification for sample command output.

4.2 Query the Migration Job Status

Requirement	Description	Comments
Query the migration job status	You can query the migration job status while the job is running. Query the status of a database migration job using the `ZDMCLI query job` command, specifying the job ID. The job ID is shown in the command output when the database migration job is submitted. `zdmuser>$ZDM_HOME/bin/zdmcli query job -jobid` *`job-id`* You can find the migration job console output in the file (Result file path:), given in the `query job` command output. You can see migration progress messages in the specified file, as shown in this sample output. `Result file path: "/u01/app/zdmbase/chkbase/scheduled/job-5-2019-12-16-06:37:04.log"`	For example `zdmuser>$ZDM_HOME/bin/zdmcli query job -jobid 5`

4.3 Migrate the Database

Before submitting the database migration job, ensure that the evaluation migration job is successful to determine how the process may fare with your configuration and settings. If the evaluation migration job is not sucessful, then this migration job will not be successful.

Determine if the migration process needs to be paused and resumed before you start the database migration. Once the migration job is started, the job system runs the job as configured. If the migration job needs to pause and resume at a particular point, then see List Migration Job Phases, Pause and Resume a Migration Job, and Rerun a Migration Job for instructions.

Requirement	Description	Comments
Migrate the database	To migrate the database, have the completed response file, and keep the necessary credentials ready, such as the source database SYS password, Object Store (Bucket) swift authentication token, and if the source database server is accessed using the root user, then have the root user password. If connectivity to the source database server uses the root credentials then the command is: zdmuser>$ZDM_HOME/bin/zdmcli migrate database -sourcedb *source_db_unique_name_value* -sourcenode *source_database_server_name* -srcroot -targetnode *target_database_server_name* -targethome *target_database_ORACLE_HOME* -backupuser *Object_store_login_user_name* -rsp *response_file_location* -tgtauth zdmauth -tgtarg1 user:*target_database_server_login_user_name* -tgtarg2 identity_file:*ZDM_installed_user_private_key_file_location* -tgtarg3 sudo_location:/usr/bin/sudo If connectivity to the source database server uses an SSH key, then the command is: zdmuser>$ZDM_HOME/bin/zdmcli migrate database -sourcedb *source_db_unique_name_value* -sourcenode *source_database_server_name* -srcauth zdmauth -srcarg1 user:*source_database_server_login_user_name* -srcarg2 identity_file:*ZDM_installed_user_private_key_file_location* -srcarg3 sudo_location:/usr/bin/sudo -targetnode *target_database_server_name* -targethome *target_database_ORACLE_HOME_value* -backupuser *Object_store_login_user_name* -rsp *response_file_location* -	**Example** For the prompts, specify the source database SYS password. If the backup destination is Object Store (Bucket), then specify your swift authentication token. If the backup destination is Storage Classic (Container), then specify your tenancy login password. If source server is accessed using the root user, then specify the root user password. If connectivity to the source database server uses root credentials then command is zdmuser>$ZDM_HOME/bin/zdmcli migrate database -sourcedb ZDM12201_phx1sp -sourcenode zdm122011 -srcroot -targetnode exacstarget1 -targethome /u02/app/oracle/product/12.2.0/dbhome_3 -backupuser backup_user@example.com -rsp /u01/app/zdmhome/rhp/zdm/template/zdm_template_ZDM12201.rsp -tgtauth zdmauth -tgtarg1 user:opc -tgtarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk -tgtarg3 sudo_location:/usr/bin/sudo -ignore ALL Enter source database ZDM12201_phx1sp SYS password: Enter source user "root" password: Enter user "backup_user@example.com" password: Then the following message is displayed with the job ID number. Make note of the job ID number for later use. Operation "zdmcli migrate database" scheduled with the job ID "6". If connectivity to the source database server uses an SSH key, then command is zdmuser>$ZDM_HOME/bin/zdmcli migrate database -sourcedb

Requirement	Description	Comments

	tgtauth zdmauth -tgtarg1 user:*target_database_server_login_user_name* -tgtarg2 identity_file:*ZDM_installed_user_private_key_file_location* -tgtarg3 sudo_location:/usr/bin/sudo
	Note that if a source single instance database is deployed without a Grid Infrastructure home, then in the above command use `-sourcesid` in place of `-sourcedb`.
	Also, if a source database is configured for a `PASSWORD` based wallet, then add the `-tdekeystorepasswd` option to the command above, and for the prompt, specify the source database TDE keystore password value.
	The `migrate database` command checks for patch compatibility between the source and target home patch level, and expects the target home patch level to be equal to or higher than the source. If the target home patch level is not as expected, then the migration job is stopped and missing patches are reported. You can either patch the target home with the necessary patches or you can force continue the migration by appending the `–ignore PATCH_CHECK` or `-ignore ALL` option to the `migrate database` command.
		`ZDM12201_phx1sp -sourcenode zdm122011 -srcauth zdmauth -srcarg1 user:opc -srcarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk -srcarg3 sudo_location:/usr/bin/sudo -targetnode exacstarget1 -targethome /u02/app/oracle/product/12.2.0/dbhome_3 -backupuser backup_user@example.com -rsp /u01/app/zdmhome/rhp/zdm/template/zdm_template_ZDM12201.rsp -tgtauth zdmauth -tgtarg1 user:opc -tgtarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk -tgtarg3 sudo_location:/usr/bin/sudo -ignore ALL`
		`Enter source database ZDM12201_phx1sp SYS password:`
		`Enter user "backup_user@example.com" password:`
		Then the following message is displayed with the job ID number. Make note of the job ID number for later use.
		`Operation "zdmcli migrate database" scheduled with the job ID "6".`
		Query the migration job status using the following command.
		`$ZDM_HOME/bin/zdmcli query job -jobid 6`
		See Migration Job Output for sample command output.

4.4 List Migration Job Phases

Requirement	Description	Comments
List migration job phases	To list the operation phases involved in the migration job, add the `-listphases` option in the `ZDMCLI migrate database` command. For example `zdmuser> $ZDM_HOME/bin/zdmcli migrate database -sourcedb ZDM12201_phx1sp -sourcenode zdm122011 -srcauth zdmauth -srcarg1 user:opc -srcarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk -srcarg3 sudo_location:/usr/bin/sudo -targetnode exacstarget1 -targethome /u02/app/oracle/product/12.2.0/dbhome_3 -backupuser backup_user@example.com -rsp /u01/app/zdmhome/rhp/zdm/template/zdm_template_ZDM12201.rsp -tgtauth zdmauth -tgtarg1 user:opc -tgtarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk -tgtarg3 sudo_location:/usr/bin/sudo -listphases`	See List Migration Job Phases for sample command output.

4.5 Pause and Resume a Migration Job

Requirement	Description	Comments
Pause and resume a migration job	You can pause a migration job at any point after the `ZDM_SETUP_TGT` phase, and resume the job at any time. To pause a migration job, specify the –`pauseafter` option in the `ZDMCLI migrate database` command with a valid phase at which to pause. Choose a valid phase that is listed in the `-listphases` command output. Pausing after the `ZDM_SETUP_TGT` phase is recommended. For example, if you specify `-pauseafter ZDM_CONFIGURE_DG_SRC`, the migration job will pause after completing the `ZDM_CONFIGURE_DG_SRC` phase. `zdmuser> $ZDM_HOME/bin/zdmcli migrate database -sourcedb ZDM12201_phx1sp -sourcenode zdm122011 -srcauth zdmauth -srcarg1 user:opc -srcarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk -srcarg3 sudo_location:/usr/bin/sudo -targetnode exacstarget1 -targethome /u02/app/oracle/product/12.2.0/dbhome_3 -backupuser backup_user@example.com -rsp /u01/app/zdmhome/rhp/zdm/template/zdm_template_ZDM12201.rsp -tgtauth zdmauth -tgtarg1 user:opc -tgtarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk -tgtarg3 sudo_location:/usr/bin/sudo` **`-pauseafter ZDM_CONFIGURE_DG_SRC`** A paused job can be resumed any time by running the `ZDMCLI resume job` command, and specifying the job ID. To schedule another pause, specify the `–pauseafter` option in the `resume job` command with a valid phase at which to pause. Choose a valid phase later than phase the job is currently paused at, and that is listed in the `-listphases` command output.	**Note:** After the phase `ZDM_CONFIGURE_DG_SRC` completes, a standby is created on the target and sync will happen from source to target. You can monitor and decide to resume the operation for the role change. For example, to pause at another place when resuming the migration job: `zdmuser> $ZDM_HOME/bin/zdmcli resume job -jobid 6 -pauseafter ZDM_SWITCHOVER_TGT`

Requirement	Description	Comments
	The `resume job` command sytax is `zdmuser> $ZDM_HOME/bin/zdmcli resume job -jobid Job_ID [-pauseafter valid-phase]` Note that the `-pauseafter` option allows only one phase to be specified.	

4.6 Rerun a Migration Job

Requirement	Description	Comments
Rerun a migration job	If there are unexpected errors in the migration workflow, you can correct them and rerun the migration job. The errors are recorded in the job output, which can be queried using the `ZDMCLI query job` command. When the error is resolved, the failed job can be continued from the point of failure. Rerun the migration job by running the `ZDMCLI resume job` command, specifying the job ID of the job to be rerun, as shown here. `zdmuser> $ZDM_HOME/bin/zdmcli resume job -jobid Job_ID`	For example: `$ZDM_HOME/bin/zdmcli resume job -jobid 6`

4.7 Post-Migration Tasks

Requirement	Description	Comments

Run the datapatch utility on the target database	After you complete the database migration job, if the target database environment is at a higher patch level than the source database, you must run the datapatch utility on the target database.
	For example, if your source database is at Oct 2018 PSU/BP and the target is at Jan 2019 PSU/BP), you must run the datapatch utility.
	Before running datapatch on the target, ensure you apply the target patch level to the binaries at the source (standby) database.
	If you are running a multitenant architecture, open the PDBs.
	`SQL> alter pluggable database all open;`
	It is recommended that you run datapatch on all of the PDBs; however, if you only want to open a subset of the PDBs in the CDB, you can use the following command instead. Datapatch only runs on the CDB and opened PDBs.
	`SQL> alter pluggable database` *`PDB_NAME`* `open;`
	To run datapatch on a PDB later (previously skipped or newly plugged in), open the database using the `AlTER PLUGGABLE DATABASE` command and rerun the datapatch utility.
	Go to the OPatch directory in `ORACLE_HOME` and run the datapatch utility.
	`% cd $ORACLE_HOME/OPatch`
	`% ./datapatch -verbose`
	The datapatch utility runs the necessary apply scripts to load the modified SQL files into the database. An entry is added to the `dba_registry_sqlpatch` view indicating the patch application.
	Check for errors. Error logs are located in the `$ORACLE_BASE/cfgtoollogs/sqlpatch/patch#/`*`unique patch ID`* directory in the following format:
	`patch#_apply_database_SID_CDB_name_timestamp``.log`
	where `database_SID` is the database SID, `CDB_name` is the name of the multitenant container database, and
		none

Requirement	Description	Comments
	timestamp is in the format `YYYYMMMDD_HH_MM_SS`	

Requirement	Description	Comments

Register the Exadata Cloud Service database	Register the Exadata Cloud Service database, and make sure its meets all of the requirements.
	Run the following commands on the Exadata Cloud Service database server as the root user.
	`/root>dbaascli registerdb prereqs --dbname` *`db_name`* `--db_unique_name` *`db_unique_name`*
	`/root>dbaascli registerdb begin --dbname` *`db_name`* `--db_unique_name` *`db_unique_name`*
		For example
		`/root>dbaascli registerdb prereqs --dbname ZDM12201 --db_unique_name ZDM12201_phx1xx`
		`DBAAS CLI version 18.2.3.2.0`
		`Executing command registerdb prereqs --db_unique_name ZDM12201_phx1xx`
		`INFO: Logfile Location: /var/opt/oracle/log/ZDM12201/registerdb/registerdb_2019-08-14_05:35:31.157978280334.log`
		`INFO: Prereqs completed successfully`
		`/root>`
		`/root>dbaascli registerdb begin --dbname ZDM12201 --db_unique_name ZDM12201_phx1xx`
		`DBAAS CLI version 18.2.3.2.0`
		`Executing command registerdb begin --db_unique_name ZDM12201_phx1xx`
		`Logfile Location: /var/opt/oracle/log/ZDM12201/registerdb/registerdb_2019-08-14_05:45:27.264851309165.log`
		`Running prereqs`
		`DBAAS CLI version 18.2.3.2.0`
		`Executing command registerdb prereqs --db_unique_name ZDM12201_phx1xx`
		`INFO: Logfile Location: /var/opt/oracle/log/ZDM12201/registerdb/registerdb_2019-08-14_05:45:29.000432309894.log`
		`INFO: Prereqs completed successfully`
		`Prereqs completed`
		`Running OCDE .. will take time ..`
		`OCDE Completed successfully.`
		`INFO: Database ZDM12201 registered as Cloud database`

Requirement	Description	Comments
		`/root>`
Apply patches	Make sure that you can apply the patches to migrated database, using either the console or commands and APIs provided by the target platform.	none
Create backups	Make sure that you can back up the migrated database, using either the console or commands and APIs provided by the target platform.	none

5

Troubleshooting and Known Issues

Requirement	Description	Comments

Troubleshooting and known issues	Check the Zero Downtime Migration service host log at `$ZDM_BASE/crsdata/zdm_service_node/rhp/rhpserver.log.0`
	On the source or target database server check the log associated with the operational phase in `/tmp/zdm-unique-id/zdm/log`.
	For example:
	`ls -ltrd /tmp/zdm*`
	`zdm1748010429`
	`cd /tmp/zdm1748010429/zdm/log`
	`default.log mZDM_obc_install_19852.log`
	`mZDM_backup_20603.log mZDM_oss_standby_setup_src_34039.log`
	`mZDM_backup_29362.log mZDM_oss_standby_validate_src_18814.log`
	`mZDM_backup_43124.log mZDM_resource_36058.log`
	`Abort migration job : $ZDM_HOME/bin/zdmcli abort job -jobid Job_ID`
		See Oracle Zero Downtime Migration Release Notes for more information about knowing issues and troubleshooting information.

6 Additional Information

Requirement	Description	Comments
Find additional information about Zero Downtime Migration	• Oracle Zero Downtime Migration Product Page • Oracle Zero Downtime Migration Documentation • Oracle Zero Downtime Migration MOS Note • Oracle Zero Downtime Migration Download • Introducing MOVE to the Oracle Cloud • Administering Oracle Database Exadata Cloud Service • (ADB) MV2ADB: move data to Autonomous Database in "one-click" (Doc ID 2463574.1) • (OCI) MV2OCI: move data to Oracle Cloud Database in "one-click" (Doc ID 2514026.1) • Oracle Cloud Migration Documentation • Exadata Cloud Service in OCI Best Practices (Doc ID 2570952.1) • Oracle Database Tablespace Encryption Behavior in Oracle Cloud (Doc ID 2359020.1)	none
Learn about considerations for setting up applications before database migration	Oracle Cloud Infrastructure, "Network Setup for DB Systems" PeopleSoft application migration using Cloud Manager	For other application migrations, see their respective documentation sets.

7

Appendices

The following appendices contain information that you may need to complete the requirements in the previous topics.

- Setting Up the Transparent Data Encryption Wallet
  For Oracle Database 12c Release 2 and later, if the source database does not have TDE enabled, then it is mandatory that you configure the TDE wallet before migration begins. Enabling TDE on Oracle Database 11g Release 2 (11.2.0.4) and Oracle Database 12c Release 1 is not required.
- Generating a Private SSH Key Without a Passphrase
  If, on the Zero Downtime Migration service host, source database server, or target database server, the authentication key pairs are not available without a passphrase for the Zero Downtime Migration software installed user, you can generate a new SSH key using the following procedure.
- Prechecks Verification
- List Migration Job Phases
- Migration Job Output

7.1 Setting Up the Transparent Data Encryption Wallet

For Oracle Database 12c Release 2 and later, if the source database does not have TDE enabled, then it is mandatory that you configure the TDE wallet before migration begins. Enabling TDE on Oracle Database 11g Release 2 (11.2.0.4) and Oracle Database 12c Release 1 is not required.

If Transparent Data Encryption (TDE) is not already configured as required on the source and target databases, use the following instructions to set up the (TDE) wallet. TDE should be enabled, the `WALLET` status on both source and target databases must be set to `OPEN`, and the `WALLET_TYPE` must be set to `AUTOLOGIN`.

1. Set `ENCRYPTION_WALLET_LOCATION` in $ORACLE_HOME/network/admin/sqlnet.ora file.

   ```
   $ cat /u01/app/oracle/product/12.2.0.1/dbhome_2/network/admin/
   sqlnet.ora

   ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=FILE)
     (METHOD_DATA=(DIRECTORY=/u01/app/oracle/product/12.2.0.1/dbhome_2/
   network/admin/)))
   ```

2. Connect to the database and configure the keystore.

   ```
   $ sqlplus "/as sysdba"
   SQL> ADMINISTER KEY MANAGEMENT CREATE KEYSTORE '/u01/app/oracle/product/
   12.2.0.1/dbhome_2/network/admin'
   ```

```
 identified by **********;
keystore altered.
```

For a non-CDB environment, run the following command.

```
SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY
**********;
keystore altered.
```

For a CDB environment, run the following command.

```
SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY
********** container = ALL;
```

For a non-CDB environment, run the following command.

```
SQL> ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY ********** with
backup;
keystore altered.
```

For a CDB environment, run the following command.

```
SQL> ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY ********** with
backup container = ALL;
```

Then run,

```
SQL> select * FROM v$encryption_keys;
```

3. Set up autologin.

```
SQL> SELECT * FROM v$encryption_wallet;

WRL_TYPE    WRL_PARAMETER
--------------------
------------------------------------------------------------------------
--------
STATUS                         WALLET_TYPE          WALLET_OR
FULLY_BAC    CON_ID
------------------------------ -------------------- ---------
---------            ----------
FILE        /u01/app/oracle/product/12.2.0.1/dbhome_2/network/admin/
OPEN                           PASSWORD             SINGLE
NO         0

SQL> ADMINISTER KEY MANAGEMENT CREATE AUTO_LOGIN KEYSTORE FROM KEYSTORE
 '/u01/app/oracle/product/12.2.0.1/dbhome_2/network/admin/' IDENTIFIED
BY **********;
keystore altered.
```

If you are using an Oracle RAC database, copy the files below to the same location on each cluster node, or to a shared file system.

```
/u01/app/oracle/product/12.2.0.1/dbhome_2/network/admin/ew*
/u01/app/oracle/product/12.2.0.1/dbhome_2/network/admin/cw*

SQL> SELECT * FROM v$encryption_wallet;
WRL_TYPE    WRL_PARAMETER
--------------------
------------------------------------------------------------------------
--------
STATUS                         WALLET_TYPE          WALLET_OR
FULLY_BAC        CON_ID
------------------------------ -------------------- ---------
---------            ----------
FILE        /u01/app/oracle/product/12.2.0.1/dbhome_2/network/admin/
OPEN                           PASSWORD             SINGLE
NO                 0
```

At this stage, the `PASSWORD` based wallet is enabled. To enable an `AUTOLOGIN` based wallet, complete the remaining steps in this procedure.

Close the password wallet.

```
SQL> administer key management set keystore close identified by
**********;
keystore altered.
```

Then verify that autologin is configured. Set `TDE WALLET` status to `OPEN` and `WALLET_TYPE` to `AUTOLOGIN`, otherwise the wallet configuration is not correctly set up.

```
$ sqlplus "/as sysdba"
SQL> SELECT * FROM v$encryption_wallet;
WRL_TYPE WRL_PARAMETER
--------------------
------------------------------------------------------------------------
--------
STATUS WALLET_TYPE WALLET_OR FULLY_BAC CON_ID
------------------------------ -------------------- --------- ---------
---------
FILE /u01/app/oracle/product/12.2.0.1/dbhome_2/network/admin/
OPEN AUTOLOGIN SINGLE NO
```

Upon migration of your Oracle Database to the Oracle Cloud, bear in mind that Oracle databases in the Oracle Cloud are TDE enabled by default. Zero Downtime Migration will take care of the encryption of your target database, even if your source Oracle Database is not TDE enabled by default. However, once the switchover phase of the migration has taken place, the redo logs that the new primary database in the Oracle Cloud sends to the new standby database on your premises will be encrypted. Therefore, if you decide to switch back and role swap again making the on-premises database the primary again and the database in the Oracle Cloud the standby, the on-premises database will not be able to read the newly encrypted changed blocks applied by the redo logs unless TDE is enabled on-premises.

In order to avoid post migration conflict, prior to performing the original switchover as part of the migration process, the recommended best practice is to perform appropriate testing and validation. There are options outside of Zero Downtime Migration for testing with a snapshot standby database, and once you are ready to proceed, delete the snapshot standby database and instruct Zero Downtime Migration to perform the switchover and finalize the migration process.

7.2 Generating a Private SSH Key Without a Passphrase

If, on the Zero Downtime Migration service host, source database server, or target database server, the authentication key pairs are not available without a passphrase for the Zero Downtime Migration software installed user, you can generate a new SSH key using the following procedure.

SSH connectivity during Zero Downtime Migration operations requires direct, non-interactive access between the Zero Downtime Migration service host and the source and target database servers, and also between the source and target database servers, without the need to enter a passphrase.

> **Note:**
>
> The following steps show examples for generating a private SSH key for the software installed user. You can also use these steps for the `opc` user.

Run the following command as the Zero Downtime Migration software installed user on the Zero Downtime Migration service host.

```
zdmuser> ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/opc/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/opc/.ssh/id_rsa.
Your public key has been saved in /home/opc/.ssh/id_rsa.pub.
The key fingerprint is:
c7:ed:fa:2c:5b:bb:91:4b:73:93:c1:33:3f:23:3b:30 opc@rhost1
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|                 |
|         . . .   |
|        S o . =  |
|         . E . * |
|            X.+o.|
|          .= Bo.o|
|          o+*o.  |
+-----------------+
```

This command generates the id_rsa and id_rsa.pub files in the zdmuser home, for example, /home/zdmuser/.ssh.

You can add the public key (for example, /home/zdmuser/.ssh/id_rsa.pub) to the source and target database servers using the Oracle Cloud Infrastructure Console, or you can add it manually to the authorized_keys file on those servers, as shown below.

Add the contents of the Zero Downtime Migration service host /home/zdmuser/.ssh/id_rsa.pub file to the Oracle Cloud Infrastructure server opc user /home/opc/.ssh/authorized_keys file, as shown here.

```
[opc@rptest.ssh]$ export PS1='$PWD>'
/home/opc/.ssh>ls
authorized_keys  authorized_keys.bkp  id_rsa  id_rsa.pub  known_hosts
zdmkey
/home/opc/.ssh>cat id_rsa.pub >> authorized_keys
```

You should save the private key in a separate, secure file, and use it to connect to the source and target database servers. For example, create a zdm_service_node.ppk file with permissions set to 600, and put the private key file into it on the Zero Downtime Migration service host software installed user home/.ssh to connect source and target database servers.

7.3 Prechecks Verification

```
$ZDM_HOME/bin/zdmcli query job -jobid 5
rachu3dom.hu.oracle.com: Audit ID: 79
Job ID: 5
User: zdmuser
Client: rachu3dom
Scheduled job command: "zdmcli migrate database -sourcedb ZDM12201_phx1sp
 -sourcenode zdm122011 -srcauth zdmauth -srcarg1 user:opc
 -srcarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk
 -srcarg3 sudo_location:/usr/bin/sudo -targetnode exacstarget1
 -targethome /u02/app/oracle/product/12.2.0/dbhome_3
 -backupuser backup_user@example.com
 -rsp /u01/app/zdmhome/rhp/zdm/template/zdm_template_ZDM12201.rsp
 -tgtauth zdmauth -tgtarg1 user:opc
 -tgtarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk
 -tgtarg3 sudo_location:/usr/bin/sudo -ignore ALL -eval"
Scheduled job execution start time: 2019-07-31T13:21:39+02.
 Equivalent local time: 2019-07-31 13:21:39
Current status: SUCCEEDED
Result file path:
 "/u01/app/zdmbase/chkbase/scheduled/job-5-2019-07-31-13:21:39.log"
Job execution start time: 2019-07-31 13:21:39
Job execution end time: 2019-07-31 13:35:20
Job execution elapsed time: 13 minutes 40 seconds
ZDM_GET_SRC_INFO .......... COMPLETED
ZDM_GET_TGT_INFO .......... COMPLETED
ZDM_SETUP_SRC ............. COMPLETED
ZDM_SETUP_TGT ............. COMPLETED
ZDM_GEN_RMAN_PASSWD ....... COMPLETED
ZDM_PREUSERACTIONS ........ COMPLETED
ZDM_PREUSERACTIONS_TGT .... COMPLETED
```

```
ZDM_VALIDATE_SRC .......... COMPLETED
ZDM_VALIDATE_TGT .......... COMPLETED
```

7.4 List Migration Job Phases

```
$ZDM_HOME/bin/zdmcli migrate database -sourcedb ZDM12201_phx1sp
 -sourcenode zdm122011 -srcauth zdmauth -srcarg1 user:opc
 -srcarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk
 -srcarg3 sudo_location:/usr/bin/sudo -targetnode exacstarget1
 -targethome /u02/app/oracle/product/12.2.0/dbhome_3
 -backupuser backup_user@example.com
 -rsp /u01/app/zdmhome/rhp/zdm/template/zdm_template_ZDM12201.rsp
 -tgtauth zdmauth -tgtarg1 user:opc
 -tgtarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk
 -tgtarg3 sudo_location:/usr/bin/sudo -listphases
rachu3dom.hu.oracle.com: Processing response file ...
pause and resume capable phases for this operation: "
ZDM_GET_SRC_INFO
ZDM_GET_TGT_INFO
ZDM_SETUP_SRC
ZDM_SETUP_TGT
ZDM_GEN_RMAN_PASSWD
ZDM_PREUSERACTIONS
ZDM_PREUSERACTIONS_TGT
ZDM_VALIDATE_SRC
ZDM_VALIDATE_TGT
ZDM_OBC_INST_SRC
ZDM_OBC_INST_TGT
ZDM_BACKUP_FULL_SRC
ZDM_BACKUP_INCREMENTAL_SRC
ZDM_DISCOVER_SRC
ZDM_COPYFILES
ZDM_PREPARE_TGT
ZDM_SETUP_TDE_TGT
ZDM_CLONE_TGT
ZDM_FINALIZE_TGT
ZDM_CONFIGURE_DG_SRC
ZDM_SWITCHOVER_SRC
ZDM_SWITCHOVER_TGT
ZDM_MANIFEST_TO_CLOUD
ZDM_NONCDBTOPDB_PRECHECK
ZDM_NONCDBTOPDB_CONVERSION
ZDM_POSTUSERACTIONS
ZDM_POSTUSERACTIONS_TGT
ZDM_CLEANUP_SRC
ZDM_CLEANUP_TGT"
```

7.5 Migration Job Output

```
$ZDM_HOME/bin/zdmcli query job -jobid 6
rachu3dom.hu.oracle.com: Audit ID: 80
Job ID: 6
User: zdmuser
Client: rachu3dom
Scheduled job command: "zdmcli migrate database -sourcedb ZDM12201_phx1sp
 -sourcenode zdm122011 -srcauth zdmauth -srcarg1 user:opc
 -srcarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk
 -srcarg3 sudo_location:/usr/bin/sudo -targetnode exacstarget1
 -targethome /u02/app/oracle/product/12.2.0/dbhome_3
 -backupuser backup_user@example.com
 -rsp /u01/app/zdmhome/rhp/zdm/template/zdm_template_ZDM12201.rsp
 -tgtauth zdmauth -tgtarg1 user:opc
 -tgtarg2 identity_file:/home/zdmuser/.ssh/zdm_service_host.ppk
 -tgtarg3 sudo_location:/usr/bin/sudo -ignore ALL"
Scheduled job execution start time: 2019-07-31T13:47:27+02. Equivalent
 local time: 2019-07-31 13:47:27
Current status: SUCCEEDED
Result file path:
 "/u01/app/zdmbase/chkbase/scheduled/job-6-2019-07-31-13:47:39.log"
Job execution start time: 2019-07-31 21:37:13
Job execution end time: 2019-07-31 21:40:18
Job execution elapsed time: 23 minutes 5 seconds
ZDM_GET_SRC_INFO .............. COMPLETED
ZDM_GET_TGT_INFO .............. COMPLETED
ZDM_SETUP_SRC ................. COMPLETED
ZDM_SETUP_TGT ................. COMPLETED
ZDM_GEN_RMAN_PASSWD ........... COMPLETED
ZDM_PREUSERACTIONS ............ COMPLETED
ZDM_PREUSERACTIONS_TGT ........ COMPLETED
ZDM_VALIDATE_SRC .............. COMPLETED
ZDM_VALIDATE_TGT .............. COMPLETED
ZDM_OBC_INST_SRC .............. COMPLETED
ZDM_OBC_INST_TGT .............. COMPLETED
ZDM_BACKUP_FULL_SRC ........... COMPLETED
ZDM_BACKUP_INCREMENTAL_SRC .... COMPLETED
ZDM_DISCOVER_SRC .............. COMPLETED
ZDM_COPYFILES ................. COMPLETED
ZDM_PREPARE_TGT ............... COMPLETED
ZDM_SETUP_TDE_TGT ............. COMPLETED
ZDM_CLONE_TGT ................. COMPLETED
ZDM_FINALIZE_TGT .............. COMPLETED
ZDM_CONFIGURE_DG_SRC .......... COMPLETED
ZDM_SWITCHOVER_SRC ............ COMPLETED
ZDM_SWITCHOVER_TGT ............ COMPLETED
ZDM_MANIFEST_TO_CLOUD ......... COMPLETED
ZDM_NONCDBTOPDB_PRECHECK ...... COMPLETED
ZDM_NONCDBTOPDB_CONVERSION .... COMPLETED
ZDM_POSTUSERACTIONS ........... COMPLETED
ZDM_POSTUSERACTIONS_TGT ....... COMPLETED
```

```
ZDM_CLEANUP_SRC ............... COMPLETED
ZDM_CLEANUP_TGT ............... COMPLETED
```

Index