

Oracle® Cloud

Using the Oracle WebLogic Migration (OWM) Tool



G57462-01
June 2026



Oracle Cloud Using the Oracle WebLogic Migration (OWM) Tool,

G57462-01

Copyright © 2019, 2026, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

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.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <https://www.oracle.com/corporate/accessibility/>.

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <https://support.oracle.com/portal/> or visit [Oracle Accessibility Learning and Support](#) if you are hearing impaired.

Contents

Preface

Documentation Accessibility	i
Diversity and Inclusion	i

1 About the WebLogic Migration (OWM) Tool

Limitations	2
Terminology	2

2 Installing and Using the WebLogic Migration Tool

Requirements	1
Installing the WebLogic Migration (OWM) Tool	2
Using the Migration Tool	3

3 Migrate WLS Domain to OCI

Resource Manager Provisioning Behavior	2
Required IAM Policies	2

4 Restore Process after Stack Apply

Check Cloud-init Status	1
Post-Restore Validation	1
Destroying the Migrated Stack	2

5 Known Issues

6 Logs and Error Handling

Preface

Using the Oracle WebLogic Migration (OWM) Tool guide explains how to lift-and-shift migration of single or multi-node WebLogic domains from on-premises environments to Oracle Cloud Infrastructure (OCI).

Topics:

- [Documentation Accessibility](#)
- [Diversity and Inclusion](#)

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <https://www.oracle.com/corporate/accessibility/>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <https://support.oracle.com/portal/> or visit [Oracle Accessibility Learning and Support](#) if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1

About the WebLogic Migration (OWM) Tool

The **OCI WebLogic Migration (OWM) Tool** enables lift-and-shift migration of single or multi-node WebLogic domains from on-premises environments to **Oracle Cloud Infrastructure (OCI)**, optionally fronted by a load balancer.

Use this tool if you need to migrate an existing WebLogic domain to OCI and want help with:

- Discovering the current WebLogic domain configuration.
- Discovering infrastructure referenced by the domain, including machines, clusters, managed servers, applications, and resources.
- Discovering JDBC datasources and generating OCI Resource Manager inputs.
- Building an OCI Resource Manager stack bundle.
- Archiving WebLogic, Java, domain, and custom directory assets. Optionally uploading generated artifacts to OCI Object Storage.

Key features of the tool:

- The tool inspects the existing WebLogic Domain to discover domain configuration, managed servers, clusters, applications, and resources.
- Based on this discovery, it generates a tailored Resource Manager stack that leverages OCI Resource Manager to securely provision and configure:
 - **OCI Networking resources** (VCN, subnets, gateways, route tables, and security lists)
 - **Compute Instances** to host the discovered WebLogic domain
 - Optional **load balancers** for distributing traffic across WebLogic managed servers
- The tool updates **JDBC datasource configuration files** in the domain so that WebLogic servers in OCI can connect to target databases.

After a successful run, the tool can generate:

- A WebLogic discovery JSON file, such as ``out/Discovered_<timestamp>.json``.
- An infrastructure discovery JSON file, such as ``out/infra_output_<timestamp>.json``.
- Datasource-related Terraform and schema files under ``oci/generated/``.
- An OCI Resource Manager stack ZIP under ``oci/stack/``, such as ``owm_rm_<timestamp>.zip``.
- Migration archives under ``out/``. - Run status and output metadata in ``logs/migration_data.json``.
- Optional OCI Object Storage uploads and a Pre-Authenticated Request (PAR) URL for the generated stack ZIP.

! Important

Important: This tool does **not** migrate on-premises databases to OCI. Database migration must be completed separately, and the resulting OCI database connection details must be supplied during the migration or Resource Manager workflow.

Limitations

- Only WebLogic domains created with version 12.2.1.4 or later are supported for migration with this tool.
- For domains with multiple managed servers using different listen ports, the load balancer configuration must be performed manually, and the backend targets must be adjusted accordingly.

Terminology

Learn some of the key terms used in the WebLogic Migration Tool.

Term	Description
OCI	Oracle Cloud Infrastructure.
On-Premise	Refers to the source environment where the WebLogic domain currently resides.
OWM	OCI WebLogic Migration Tool; the tool used to migrate on-premises WebLogic domains to OCI.
AdminServer host	The VM hosting the AdminServer of the on-premises WebLogic domain.
Custom directories	File system paths referenced by the WebLogic domain configuration that are outside the standard three categories (Domain Home, Middleware Home, Java Home). Example: External trust stores or keystores located outside the Domain or Middleware directories.
OS	Operating System installed on a host, e.g., Oracle Linux.
Archives	Compressed tarball files created from the WebLogic domain's Domain Home , Middleware Home , and Java Home directories for migration to OCI.

2

Installing and Using the WebLogic Migration Tool

Requirements

Before using this tool, ensure the following prerequisites are met.

On Premises Requirements:

- **Operating System:** Supports Linux environments, including Oracle Linux (OL) and RHEL-compatible distributions.
- **File system permissions:** The tool must be installed on the **AdminServer host**. The user must have **read/write permissions** on WebLogic Domain, Oracle Middleware, and Java Home directories.
- **SSH Connectivity:** Passwordless SSH must be established from the AdminServer host to all Managed Server hosts.
 - **For SSH Passwords:** Configure passwordless SSH between the AdminServer and Managed Servers.
 - **For SSH Key Authentication:** Use SSH agent to store your key and passphrase for seamless connectivity between the AdminServer and Managed Servers.

Note

Note: Test SSH connectivity from the AdminServer to each Managed Server using `ssh <managed-server-host>` to ensure authentication works without prompting for a password or passphrase.

- **Storage space:** Each host must have sufficient disk space to archive **Oracle Home**, **JDK Home**, **Domain Home**, and any **custom directories**.

Oracle Cloud Requirements

- **Oracle Cloud Account (Tenancy):** Resources discovered on-premises will be recreated under an OCI Tenancy.
- **OCI Compartment:** Required to logically group all resources created by the OWM tool.
- **OCI Resource Manager:** Permissions to create, plan, and apply stacks.
- **OCI Permissions:** Ability to create and manage the following resources:
 - Virtual Cloud Networks (VCNs)
 - Compute Instances
 - Block Storage
 - Load Balancers (Optional)

Note

Refer to **Required IAM Policies for Non-Admin Users** for detailed IAM policy requirements.

If stack users need to create IAM policies in the **Default Identity Domain** under the **root compartment**, additional policy management permissions are required. Refer to the detailed **Required IAM Policies** section later in this document.

Installing the WebLogic Migration (OWM) Tool

Follow these procedures to install the WebLogic Migration tool.

At a high level, the steps to install and configure the WebLogic Migration Tool include the following:

1. Install dependencies.
2. Validate prerequisites on the source Linux host.
3. Discover the WebLogic domain.
4. Discover the infrastructure referenced by that domain.
5. Discover datasources and generate OCI-related files.
6. Build an OCI Resource Manager stack ZIP.
7. Archive the WebLogic environment for transfer.
8. Optionally upload generated artifacts to OCI Object Storage.
1. Initiate a Secure Shell (SSH) connection to the **AdminServer host**, utilizing a user account with read and write file system permissions. This step enables secure remote access and interaction with the server.
2. Obtain a local copy of the OWM Tool repository in a folder where the user has read and write permissions.

Note

If downloading manually, extract the repository contents to this folder.

3. If you plan to clone the repository using Git, ensure that the latest version of Git is installed on the host.
 - a. Download the most recent release of GIT from [Git](https://git-scm.com/)
 - b. Clone the repository with the commands in a folder where the user has read and write permissions:

```
git clone https://github.com/oracle-quickstart/oci-weblogic-migration
cd oci-weblogic-migration
ls
```

Example output:

```
$ git clone https://github.com/oracle-quickstart/oci-weblogic-migration
Cloning into 'oci-weblogic-migration'...
Username for 'https://github.com': <your_username>
Password for 'https://user@github.com': <your_password>
remote: Enumerating objects: 4391, done.
remote: Counting objects: 100% (962/962), done.
remote: Compressing objects: 100% (333/333), done.
remote: Total 4391 (delta 819), reused 629 (delta 629), pack-reused
3429 (from 2)
Receiving objects: 100% (4391/4391), 9.01 MiB | 40.84 MiB/s, done.
Resolving deltas: 100% (2822/2822), done.
```

```
ls -lrth oci-weblogic-migration
total 40K
-rw-rw-r--. 1 oracle oracle 1.9K Sep 10 08:18 LICENSE
-rw-rw-r--. 1 oracle oracle 18K Sep 10 08:19 README.md
drwxrwxr-x. 2 oracle oracle 41 Sep 10 08:19 config
drwxrwxr-x. 4 oracle oracle 37 Sep 10 08:19 lib
drwxrwxr-x. 6 oracle oracle 58 Sep 10 08:19 oci
drwxrwxr-x. 3 oracle oracle 17 Sep 10 08:19 wdtconfig
drwxrwxr-x. 4 oracle oracle 28 Sep 10 08:19 deps
drwxrwxr-x. 2 oracle oracle 4.0K Sep 11 08:21 bin
drwxrwxr-x. 3 oracle oracle 113 Sep 11 14:48 out
drwxrwxr-x. 2 oracle oracle 4.0K Sep 11 18:02 logs
```

Using the Migration Tool

This repository contains a consolidated script `migration_script.sh` that automates the complete migration of an on-premise WebLogic domain to Oracle Cloud Infrastructure (OCI).

The script combines all required tasks into a single workflow:

1. Install dependencies
2. Discover WebLogic domain and infrastructure
3. Discover datasources
4. Generate OCI Resource Manager stack
5. Upload stack to OCI Object Storage Bucket(Optional)
6. Archive WebLogic domain
7. Upload archives to OCI Object Storage Bucket(Optional)

Step 1. Declare WebLogic source domain details in `$toolHome/config/on-prem.env`

Example:

```
#####
# ssh_user: An Operating System user with read and write permissions
#           on WebLogic Domain, Oracle Middleware and Java Home.
# Example oracle
#####
ssh_user=oracle
```

```
#####
# CLEAR VALUE (leave blank) if ssh_password_file is set.
# ssh_private_key_file: The file name of a file that contains the user's
private key
#
to use when authenticating with a public/private key
pair.
# Example /home/oracle/.ssh/id_rsa
#####
ssh_private_key_file=/home/oracle/.ssh/id_rsa

#####
# CLEAR VALUE (leave blank) if ssh_private_key_file is not passphrase
protected.
# ssh_private_key_pass_file: Path to file containing passphrase for private
key
#
(required only if your private key is passphrase
protected).
#####
ssh_private_key_pass_file=

#####
# CLEAR VALUE (leave blank) if ssh_private_key_file is set.
# ssh_password_file: The file name of a file that contains the password
string for the
#
user's password to use when authenticating with user
name and password.
#####
ssh_password_file=

#####
# domain_home: Weblogic Domain's path. Domain to migrate to OCI.
# oracle_home: Home directory of the Oracle WebLogic installation.
#
Set to ORACLE_HOME in local Linux Server.
# java_home : The location of the JDK that WebLogic Server uses for
execution
and was used during its installation.
#####
domain_home=/u01/data/domains/test_domain
oracle_home=/u01/app/oracle/middleware
java_home=/u01/jdk

#####
# skip_transfer: Defaults to false.
# - false: Archives will be transferred to OCI Object Storage.
#
bucket_name, compartment_ocid, and tenancy_namespace must be set.
# - true : Archive transfer is skipped. The below OCI storage parameters
are not required.
#####
skip_transfer=false

#####
# The following values are required ONLY if skip_transfer=false
# bucket_name: Name of the OCI Object Storage Bucket where archives
will be uploaded.
# compartment_ocid: The compartment where the OCI Object Storage bucket
exists(in case bucket
```

```
#           is pre-existing) or to be created(in case bucket is not
pre-existing).
# tenancy_namespace: Namespace of the tenancy.
#####
bucket_name=test_bucket
compartment_ocid=ocid1.compartment_tenancy_namespace=<name>
```

Step 2. Pre-requisite Steps for executing the Migration Script

Before executing the migration script `migration_script.sh`, ensure the following pre-requisites are completed:

1. **Access to GitHub** The WebLogic Migration Tool requires access to GitHub to download required libraries (WebLogic Deployment Tool) to discover the source WebLogic environment.

① Note

Alternatively, specific releases can be manually downloaded and placed in `$toolHome/deps/wdt`

2. **All Source domain Servers Must Be Up and Running (Required)** In the source environment, all WebLogic Server hosts (AdminServer and Managed Servers) are expected to be **up and running**. This is mandatory to ensure the migration script can successfully connect, discover, and archive domain configurations.
3. **Manual Archive Transfer** If you plan to perform step 5 (upload stack) and step 7 (upload archives) manually, set `skip_transfer=true` in `on-prem.env`. Transfer the archives manually by following the instructions in `$toolHome/logs/migration_script.log`. The archives are available inside `$toolHome/out` on the **AdminServer host**, provided the **AdminServer host** has enough space to accommodate the archives from all the hosts.
4. **OCI-CLI Installation (Optional)**
 - If you want the script to handle uploads automatically, install the OCI-CLI on the **AdminServer host**.
 - Installation and configuration details: [OCI CLI Documentation](#) **Installation Steps:**

```
# Enable the Oracle Linux Developer repository
sudo dnf -y install oraclelinux-developer-release-el8
```

```
# Install OCI CLI package
sudo dnf install python36-oci-cli
```

```
# Upgrade pip and required dependencies
pip3 install --user --upgrade pip
pip3 install --user six==1.15.0 --upgrade
pip3 install --user click==8.0.4
pip3 install --user oci==2.157.0 --upgrade
```

- To verify installation, run:

```
oci iam region list
```

IAM Policy Requirement: The user running this command must belong to a group with the following policy at the tenancy level:

Allow group <YourGroupName> to inspect regions in tenancy

- **Required User Policies for Non-Admin Users** (if automatic upload is needed):
 - a. Allow group Non-Admin to manage object-family in the compartment specified in on-prem.env for storage bucket details.
 - b. Allow group Non-Admin to read buckets in the tenancy.

5. **SSH Authentication** The **AdminServer host** of the source domain must have **passwordless SSH authentication** established to all **WebLogic Managed Server** Linux hosts **and to itself**. This ensures seamless remote operations during migration.

Example: On the AdminServer host, verify SSH connectivity to itself and to each Managed Server host without a password prompt.

```
# Get the full hostname of the AdminServer
hostname -f
```

```
# Test SSH connection to the AdminServer itself
[oracle@hostname-wls-0 ~]$ ssh <hostname>
[oracle@hostname-wls-0 ~]$ exit
```

```
# Test SSH connection to each Managed Server host
[oracle@hostname-wls-0 ~]$ ssh <hostname>
[oracle@hostname-wls-1 ~]$ exit
```

Repeat this process from the **AdminServer host** to all hosts in the WebLogic domain to confirm passwordless SSH access is properly configured.

Step 3. Run the migration script from the \$toolHome/bin directory.

```
$ bash migration_script.sh
```

The script is idempotent: If a step has already completed successfully in a previous run, it will be skipped automatically.

3

Migrate WLS Domain to OCI

OCI Resource Manager requires the compressed stack file created in the **Generate OCI Resource Manager Stack** step by `migration_script.sh`.

Step 1: Pre-requisites for Resource Manager

Before launching the stack, ensure the following are completed:

1. **Database Migration (for JRF WebLogic-enabled domains)** Complete the on-premise database migration to OCI-DB or ATP-DB as appropriate before running the Resource Manager **Apply** action.
2. **JDBC Connection Strings (for JRF WebLogic-enabled domains)** JDBC connection strings of the OCI Database must be known before running the Resource Manager **Apply** action.
3. **IAM Policies for Non-Admin Users** Ensure you have the correct IAM permissions. For Non-Admin users, see **Required IAM Policies for Non-Admin Users**.
4. **Dynamic Group Policies** If you unselect the "Create Policies" checkbox during stack creation, ensure appropriate dynamic group policies are already in place. See **Dynamic Group Policies** for details.
5. **Marketplace Terms (if using UCM images)** If you plan to use an Oracle WebLogic for OCI UCM image from the Marketplace, you must **accept the terms** for that listing in advance.
 - If you do not accept the terms and select a UCM image during stack creation, the Resource Manager **Apply job will fail**.

Step 2: Collect the Stack File

1. **If `migration_script.sh` was executed with `skip_transfer=true`:** Transfer the stack file to a workstation that has access to Oracle Cloud via a browser.
2. **If `migration_script.sh` was executed with `skip_transfer=false`:**
 - **Windows:** Use the PAR URL generated by the script to access the stack directly.
 - **Linux:** Download the `stack.zip` locally using the PAR URL.

Step 3: Launch the Stack

1. Open a browser and log in to your OCI Tenancy.
2. From the top-left corner, select **Developer Services**.
3. In the dropdown, click **Stacks**, then click **Create Stack**.
4. In the **Create Stack** wizard, under **Stack Configuration**, select **.Zip file** and upload the collected stack file.
5. Click **Next**.
6. Customize any stack variables required for your environment (see **Inputs to Resource Manager**), or leave them as default.
7. Click **Create**.

8. From the **Stack Details** page, click **Apply** to launch the stack and provision the OCI resources.

Resource Manager Provisioning Behavior

Based on the values selected in the ORM Stack variables, Resource Manager will:

- Provision the required number of OCI Compute Instances for WebLogic Servers, along with all associated networking resources (VCN, subnets, gateways, Network Security Gateways, and load balancer).
- Allow the user to either:
 - Create a new VCN.
 - Use a pre-existing VCN.
- Allow the user to provision optional Resources:
 - Bastion Host – Can be provisioned for secure SSH access to private WebLogic compute instances.
 - Public Load Balancer (LB) – Can be added to distribute traffic across managed servers.
 - IAM Policies – Can be created automatically for Object Storage and database access, or the user may use pre-existing policies.
- Allow the user to select from multiple images for Compute instances:
 - Oracle WebLogic Server Enterprise Edition UCM Image
 - Oracle Weblogic Suite UCM Image
 - Oracle WebLogic Server Enterprise Edition BYOL Image
 - Oracle Weblogic Suite BYOL Image
- Provides options to recreate the datasource connection strings in OCI for WebLogic domains with JDBC Datasources, using one of the following:
 - Autonomous Database (ADB)
 - OCI Database (DB System)
 - Edit JDBC string

Note

Note: For Multi Data Source (MDS) configurations for RAC DB, only **Edit JDBC string** option is supported.

Required IAM Policies

Non-Admin User Group Policies

If the user applying the Resource Manager stack is **not an OCI administrator**, your OCI administrator must first grant the following **user group policies** to allow proper provisioning and access:

Policy Statement	Purpose	Policy Location
------------------	---------	-----------------

Allow group Non-Admin to read buckets in tenancy	To check if Object Storage bucket is pre-existing	Root Compartment
Allow group Non-Admin to inspect tenancies in tenancy	To locate the home region for the tenancy	Root Compartment
Allow group Non-Admin to inspect limits in tenancy	To determine if resources are available in various compartments	Root Compartment
Allow group Non-Admin to manage object-family in compartment MyCompartment	To create Object Storage bucket and upload the archives	Bucket Compartment
Allow group Non-Admin to manage instance-family in compartment MyCompartment	To create Compute Instances	Stack Compartment
Allow group Non-Admin to manage volume-family in compartment MyCompartment	To create Block Volumes	Stack Compartment
Allow group Non-Admin to manage orm-family in compartment MyCompartment	To create Resource Manager stacks	Stack Compartment
Allow group Non-Admin to manage virtual-network-family in compartment MyNetworkCompartment	To create networking resources (VCNs, Subnets, Gateways)	WebLogic Network Compartment
Allow group Non-Admin to manage dns-family in compartment MyNetworkCompartment	To manage DNS resources (Zones and Private views)	WebLogic Network Compartment

Optional User Group Policies

The following policies are required **only if specific features are enabled**:

Policy Statement	Purpose	Policy Location
Allow group Non-Admin to manage load-balancers in compartment MyNetworkCompartment	To create and manage Load Balancers (required if "Provision Load Balancer" checkbox is selected)	WebLogic Network Compartment
Allow group Non-Admin to manage orm-private-endpoints in compartment MyNetworkCompartment	To create Private Endpoints (required only if "Provision Bastion Instance" checkbox is not selected)	WebLogic Network Compartment
Allow group Non-Admin to read orm-work-requests in compartment MyNetworkCompartment	Required for Private Endpoint creation when "Provision Bastion Instance" checkbox is not selected	WebLogic Network Compartment

Note

Note: These policies are conditional and should only be applied when the corresponding feature checkboxes are selected during Stack Apply.

Dynamic Group Policies (for users who unselect "Create Policies" checkbox)

When Compute instances are started, certain scripts make OCI API calls. These instances gain permissions through **dynamic groups** and associated **policies**.

You have two options:

- **Pre-create the policies** before stack creation.
- **Let the Terraform scripts** create them automatically by selecting the **OCI Policies** option.

If you want Terraform to create the dynamic groups and policies, and you are **not an OCI administrator**, your OCI administrator must first grant the following **user group policies**:

Policy Statement	Purpose	Policy Location
Allow group MyGroup to manage dynamic-groups in tenancy	To create dynamic groups	Root Compartment
Allow group MyGroup to manage policies in tenancy	To create policies in the root compartment	Root Compartment

Dynamic Group Policies Created When "Create Policies" Checkbox Is Selected

The following **dynamic group and network policies** are **automatically created** if the **"Create Policies"** checkbox is selected.

Note

Important If the checkbox is **not selected**, these policies must be **added manually** by your OCI administrator.

Policy Statement	Purpose	Policy Location
Allow dynamic-group <dynamic-group> to read buckets in tenancy	To find the compartment where Object Storage buckets exist	Root Compartment
Allow dynamic-group <dynamic-group> to read objects in compartment <BucketCompartment>	To upload archives or overwrite existing objects in Object Storage	Bucket Compartment
Allow dynamic-group <dynamic-group> to use instance-family in compartment <ComputeCompartment>	Required for instance-level operations such as starting or stopping compute instances during migration	Stack Compartment
Allow dynamic-group <dynamic-group> to manage volume-family in compartment <ComputeCompartment>	Required to attach or detach block volumes and manage	Stack Compartment

	storage used by migrated servers	
Allow dynamic-group <dynamic-group> to inspect virtual-network-family in compartment <NetworkCompartment>	Required to inspect VCN and subnet information for network configuration validation	Weblogic Network Compartment

ATP-DB Dynamic Group Policy (Optional)

If the migrated domain uses an Autonomous Database (ATP/ADW), The following policy is required to download the ATP or ADW database wallet:

Policy Statement	Policy Location
Allow dynamic-group <dynamic-group> to use autonomous-transaction-processing-family in compartment MyCompartment	Database Compartment

VCN Peering Dynamic Group Policies (Optional)

If VCN Peering is required (i.e., when the WebLogic VCN is different from the DB VCN), the following policies are needed:

Policy Statement	Policy Location
Allow dynamic-group <dynamic-group> to manage virtual-network-family in compartment MyDBNetworkCompartment	DB Network Compartment
Allow dynamic-group <dynamic-group> to manage virtual-network-family in compartment MyNetworkCompartment	WebLogic Network Compartment

WLS to DB Access Dynamic Group Policy (Optional)

The following policy is required only if the "Add Rule for WLS to Access DB" checkbox is selected:

Policy Statement	Policy Location
Allow dynamic-group <dynamic-group> to manage virtual-network-family in compartment MyDBNetworkCompartment	DB Network Compartment

Note**Note:**

- Replace MyGroup, MyCompartment, MyDBNetworkCompartment, and <dynamic-group> with your actual group names, compartment OCIDs, and dynamic group definitions.
- These policies ensure the user has sufficient permissions to provision networking, compute, storage, and WebLogic resources required by the migration stack.

4

Restore Process after Stack Apply

Once the ORM stack is applied, the restore process ensures that the cloud environment mirrors the on-premise WebLogic domain. During the OCI Compute Instances boot process, cloud-init is initiated which completes the migration of the on-premises domain to OCI.

Important: No WebLogic Server domain servers or Node Manager processes are started automatically after the restore. You should first review the migrated contents before starting services. See [Start OCI WebLogic Domain and Verify Services](#) for the next steps.

Check Cloud-init Status

To verify if the restore process is complete:

```
cloud-init status
status: done
```

Check Logs

All restore logs are available under /var/log/owm/ in the compute instances.

```
ls -lrth /var/log/owm
total 16K
-rw-r--r--. 1 root    root    2.7K Jul 31 09:42 network-ports.log
-rw-rw-rw-. 1 oracle  oracle    0 Jul 31 09:42 provisioning.log
-rw-r--r--. 1 root    root    1.4K Jul 31 09:42 os-users.log
-rw-r--r--. 1 root    root    1.8K Jul 31 09:43 wls-restore.log
-rw-r--r--. 1 root    root    3.4K Jul 31 09:44 datasource_update.log
```

Post-Restore Validation

After the ORM stack apply completes and all compute instances have initialized, **SSH to each instance** and perform the following validation steps before starting any WebLogic processes. All OCI Instances (Servers hosting AdminServers and Managed Servers) will be created in a Private Subnet in OCI Virtual Cloud Network. To access them, an SSH session should be established via a Bastion Host. To find the commands to ssh, click on Stack - Application Details Tab and copy the SSH command example given. The complete ssh command to access the servers should follow this format:

Example:

```
ssh -i <private ssh key file> -o 'UserKnownHostsFile /dev/null' -o
'StrictHostKeyChecking no' -o 'ProxyCommand ssh -W %h:%p -i <private ssh key
file> -l opc <Bastion Public IP>' -l opc <AdminServer Private IP>
```

```
$admin-server> sudo su - <same username as on-premises>
```

1. Verify Restore Completion

To confirm that the restore process has finished successfully, run the following command:

```
cloud-init status  
status: done
```

Proceed to the next step only if the status is reported as done.

2. Check Restore Logs

Ensure that **no errors** are reported in any of the log files located under:

```
/var/log/owm/
```

Review the contents of these logs to confirm that all migration steps have completed successfully.

① Note

For **JRF domains**, if VCN peering fails and `/var/log/owm/datasource_update.log` indicates the failure, VCN peering must be performed manually. Follow the instructions in the Oracle documentation: [Manual VCN Peering](#)

3. Proceed with Startup

Only after confirming that the restore process is complete and no errors are present in the logs should you proceed to start the **WebLogic Server** and **Node Manager** processes.

After completing all post-restore verification steps, connect to the new AdminServer instance using SSH. Navigate to the WebLogic Domain Home directory and start the AdminServer, followed by all Managed Servers. Verify and test the WebLogic domain to ensure that the migration has completed successfully.

Destroying the Migrated Stack

If you need to remove the migrated environment, you can destroy the ORM stack. However, before running the destroy operation, specific cleanup steps are required depending on your domain type: JRF or non JRF.

For JRF Domains

If the migrated stack includes VCN peering and the Add Rule for WLS to Access DB option was selected during stack creation, you must first execute the cleanup script to remove some networking configurations before destroying the stack.

These resources are created dynamically through API calls during migration and are not managed by Terraform, so Terraform cannot automatically identify or remove them. Running the cleanup script ensures these configurations are deleted before executing terraform destroy otherwise, residual networking configurations may remain in your tenancy and terraform destroy may fail.

Example:

```
cd /opt/scripts  
python3 cleanup_resources.py
```

Sample output:

```
Cleaning up security list...
Updated subnet ocid1.subnet.oc1.<tenancy>: set security_list_ids = [...]
Removed security list: wls-to-db-seclist-0 from subnet: subnet
Deleted Security List: wls-to-db-seclist-0
Cleaning up Route Rules...
Removed route to <IP> via ocid1.localpeeringgateway.oc1.<gateway>
Removed route to <IP> via ocid1.localpeeringgateway.oc1.<gateway>
Cleanup complete. You can now safely run 'terraform destroy'.
```

The cleanup script performs operations such as deleting security lists and modifying route tables and subnets. To allow these actions, ensure the dynamic group used for migration has the following IAM policies in place:

Policy Statement	Policy Location
Allow dynamic-group <dynamic-group> to manage virtual-network-family in compartment MyDBNetworkCompartment	DB Network Compartment
Allow dynamic-group <dynamic-group> to manage virtual-network-family in compartment MyNetworkCompartment	WebLogic Network Compartment

These permissions enable the cleanup script to manage network resources that were created outside of Terraform before running the terraform destroy operation.

Caution

If you run `terraform destroy` without first executing the cleanup script, the destroy operation may fail with errors such as:

```
Error: 409-IncorrectState, Local Peering Gateway
ocid1.localpeeringgateway.oc1... is associated with one or more
entities that are in use
```

In that case, the destroy process will not remove certain dynamically created networking resources (for example, route rules or security lists). If this issue occurs, run the cleanup script in the Cloud Shell before retrying the destroy operation.

This script requires proper execution permissions in Cloud Shell and specific IAM policies to be granted to your user group.

The cleanup script is located in the tool repository at:

```
$toolHome/bin/cloud_shell_cleanup.py
```

Copy the script and save the file in Cloud Shell, for example, as "cloud_shell_cleanup.py".

Before executing the script, ensure it has execution permission: `chmod +x cloud_shell_cleanup.py`

Run the following command in OCI Cloud Shell:

```
python3 cloud_shell_cleanup.py --stack-id <your_stack_ocid>
```

The following IAM policies must be granted at the compartment level:

Policy Statement	Purpose	Policy Location
Allow group Non-Admin to read orm-family in compartment MyCompartment	To read stack, job, and Terraform state data.	Stack Compartment
Allow group Non-Admin to manage virtual-network-family in compartment MyNetworkCompartment	To delete networking resources (Route rules).	WebLogic Network Compartment
Allow group Non-Admin to manage virtual-network-family in compartment MyDatabaseNetworkCompartment	To delete networking resources (Security Lists, Route rules).	Database Network Compartment
Allow group Non-Admin to read database-family in compartment MyDBNetworkCompartment	Allows the script to fetch details for DB Systems.	Database Compartment
Allow group Non-Admin to read autonomous-database-family in compartment MyDBNetworkCompartment	Allows the script to fetch details for Autonomous Databases.	Autonomous Database Compartment

Note

These permissions enables the script to successfully read and modify networking and database-related resources during cleanup.

For Non-JRF Domains

If your stack is non-JRF, you can directly run the terraform destroy command to remove all resources — no additional cleanup is required.

5

Known Issues

1. Migration script fails during infrastructure discovery

When running the migration script, you may encounter the following error:

```
bash migration_script.sh
2025-09-24 06:03:46 [info] Installing dependencies...
2025-09-24 06:03:47 [info] Checking prerequisites...
2025-09-24 06:03:57 [info] Discovering WebLogic domain...
2025-09-24 06:04:20 [info] Discovering infrastructure...
2025-09-24 06:04:30 [error] "Discovering infrastructure" failed. Check /
home/domain/mig/oci-weblogic-migration/logs/migration_script.log for
details. Run migration_script.sh again after resolving the issue.
2025-09-24 06:04:30 [error] Migration failed.
```

In the log file (/home/domain/mig/oci-weblogic-migration/logs/migration_script.log), you may see an error similar to:

```
<SEVERE> <discover_infra> <WLSDPY-20035> <verifySSH encountered an
unexpected runtime exception.
Please file an issue on GitHub and attach the log file and stdout.
Exception: exceptions.IndexError>
```

Cause

This is an intermittent issue triggered during the infrastructure discovery phase by the verifySSH step.

Workaround

Re-run the migration script. Previous successful steps will be skipped automatically, and the script should proceed normally:

```
bash migration_script.sh
2025-09-24 06:05:49 [info] "Installing dependencies" already completed
successfully. Skipping.
2025-09-24 06:05:49 [info] "Checking prerequisites" already completed
successfully. Skipping.
2025-09-24 06:05:49 [info] "Discovering WebLogic domain" already
completed successfully. Skipping.
2025-09-24 06:05:49 [info] Discovering infrastructure...
...
2025-09-24 06:08:16 [info] Migration script completed successfully!
```

2. Migration script fails in Archive Domain Step

If the script fails during the Archive Domain step due to insufficient disk space on any of the nodes, the log file will display a TODO message. These TODO messages include the

exact commands that must be executed manually on the affected host to complete the archive creation process.

Example: If the migration_script.log shows:

WARNING Messages:

1. WLSPLY-05027: Not enough space on mbnjrf_machine_1 to create the archives. Please run the commands manually mentioned in the TODO to create the archive, scp to the admin host and upload to bucket.

2. WLSPLY-05027: Not enough space on mbnjrf_machine_2 to create the archives. Please run the commands manually mentioned in the TODO to create the archive, scp to the admin host and upload to bucket.

TODO Messages:

1. WLSPLY-06042: Please create the FILE_STORE in the archive file at /usr/bin/tar czf /tmp/mbnjrf_machine_1-mbnjrf_domain-java_home.tar.gz --exclude='.pid' --exclude='.state' --exclude='.core' --exclude='diag/ofm/*/lck/*.lck' --exclude='servers/*/logs/*.log' --exclude='*.log*[0-9]' --exclude='*.log' --exclude='*.out' --exclude='*.out*[0-9]' --exclude='servers/*/data/store/diagnostics/*' --exclude='oracle-dfw-*/sampling/jvm_threads*' /u01/app/oracle/jdk .

2. WLSPLY-06042: Please create the FILE_STORE in the archive file at /usr/bin/tar czf /tmp/mbnjrf_machine_1-mbnjrf_domain-domain_home.tar.gz --exclude='.pid' --exclude='.state' --exclude='.core' --exclude='diag/ofm/*/lck/*.lck' --exclude='servers/*/logs/*.log' --exclude='*.log*[0-9]' --exclude='*.log' --exclude='*.out' --exclude='*.out*[0-9]' --exclude='servers/*/data/store/diagnostics/*' --exclude='oracle-dfw-*/sampling/jvm_threads*' /u01/data/domains/mbnjrf_domain .

3. WLSPLY-06042: Please create the FILE_STORE in the archive file at /usr/bin/tar czf /tmp/mbnjrf_machine_1-mbnjrf_domain-weblogic_home.tar.gz --exclude='.pid' --exclude='.state' --exclude='.core' --exclude='diag/ofm/*/lck/*.lck' --exclude='servers/*/logs/*.log' --exclude='*.log*[0-9]' --exclude='*.log' --exclude='*.out' --exclude='*.out*[0-9]' --exclude='servers/*/data/store/diagnostics/*' --exclude='oracle-dfw-*/sampling/jvm_threads*' /u01/app/oracle/middleware .

4. WLSPLY-06042: Please create the FILE_STORE in the archive file at /usr/bin/tar czf /tmp/mbnjrf_machine_1-mbnjrf_domain-custom_dirs.tar.gz --exclude='.pid' --exclude='.state' --exclude='.core' --exclude='diag/ofm/*/lck/*.lck' --exclude='servers/*/logs/*.log' --exclude='*.log*[0-9]' --exclude='*.log' --exclude='*.out' --exclude='*.out*[0-9]' --exclude='servers/*/data/store/diagnostics/*' --exclude='oracle-dfw-*/sampling/jvm_threads*' /u01/data/domains/mbnjrf_domain/wlsdeploy/applications .

5. WLSPLY-06042: Please create the FILE_STORE in the archive file at /usr/bin/tar czf /tmp/mbnjrf_machine_2-mbnjrf_domain-java_home.tar.gz --exclude='.pid' --exclude='.state' --exclude='.core' --exclude='diag/ofm/*/lck/*.lck' --exclude='servers/*/logs/*.log' --exclude='*.log*[0-9]' --exclude='*.log' --exclude='*.out' --exclude='*.out*[0-9]' --exclude='servers/*/data/store/diagnostics/*' --exclude='oracle-dfw-*/sampling/jvm_threads*' /u01/app/oracle/jdk .

6. WLSPLY-06042: Please create the FILE_STORE in the archive file at /usr/bin/tar czf /tmp/mbnjrf_machine_2-mbnjrf_domain-domain_home.tar.gz --exclude='.pid' --exclude='.state' --exclude='.core' --

```
exclude='diag/ofm/*/lck/*.lck' --exclude='servers/*/logs/*.log' --
exclude='*.log*[0-9]' --exclude='*.log' --exclude='*.out' --
exclude='*.out*[0-9]' --exclude='servers/*/data/store/diagnostics/*' --
exclude='oracle-dfw-*/sampling/jvm_threads*' /u01/data/domains/
mbnjrf_domain .
```

```
7. WLSPLY-06042: Please create the FILE_STORE in the archive file
at /usr/bin/tar czf /tmp/mbnjrf_machine_2-mbnjrf_domain-
weblogic_home.tar.gz --exclude='.pid' --exclude='.state' --exclude='.core'
--exclude='diag/ofm/*/lck/*.lck' --exclude='servers/*/logs/*.log' --
exclude='*.log*[0-9]' --exclude='*.log' --exclude='*.out' --
exclude='*.out*[0-9]' --exclude='servers/*/data/store/diagnostics/*' --
exclude='oracle-dfw-*/sampling/jvm_threads*' /u01/app/oracle/middleware .
```

Workaround

Then, log in to that specific host and run the indicated command manually. After the archive is successfully created:

- a. Upload the generated archive file to the designated Object Storage bucket manually or
- b. Use the OCI CLI to upload the file if the OCI CLI is installed on the instance.

Example (OCI CLI upload):

```
oci os object put --bucket-name <bucket_name> --file <archive_file_path> --
force
```

Ensure that all required domain archives are uploaded before proceeding with stack creation.

3. Error when starting Managed Servers (Hostname Verification Failure)

You may encounter an error when attempting to start managed servers:

```
<Warning> <Security> <BEA-090504> <Certificate chain received from
xxxx.example.com - 10.0.2.229 failed hostname verification check.
Certificate contained CN=xxxx but check expected xxxx.example.com>
```

Cause

This occurs because the SSL certificate presented by the server does not match the expected hostname.

For example, the certificate contains the common name CN=xxxx, while WebLogic expects xxxx.example.com. This mismatch causes hostname verification to fail.

Workaround

If the source domain is configured with the following in config.xml:

```
weblogic.security.SSL.ignoreHostnameVerification=true
```

Then you must apply the same setting manually for each managed server:

- a. Start the Admin Server.
- b. Log in to the WebLogic Administration Console.
- c. Navigate to: Environment → Servers → [Managed Server Name] → Server Start tab.

- d. For each Managed Server, add the following JVM option in the Arguments field:

```
-Dweblogic.security.SSL.ignoreHostnameVerification=true
```

- e. Use the startManagedServer.sh script to start the Managed Servers.

4. Error when starting Managed Server in WebLogic 12.2.1.4 (secure mode)

When starting a Managed Server with WebLogic 12.2.1.4 in secure mode, you may see errors such as:

```
<BEA-003111> <No channel exists for replication calls for cluster  
domainCcluster>  
<BEA-000386> <Server subsystem failed. Reason: No replication server  
channel for managedserver1>  
Server state changed to FAILED
```

Cause

In secure mode, WebLogic expects replication channels for cluster communication. If no replication channel is configured, the Managed Server startup fails.

Workaround

Disable secure replication in the domain configuration (config.xml) before starting the servers.

```
<secure-replication-enabled>>false</secure-replication-enabled>
```

Note

This issue has been addressed in the latest code base, and the configuration is automatically handled by the migration tool. No manual action is required. However, if the `/var/log/owm/wls-restore.log` file indicates that this configuration step has failed, perform the above workaround manually before starting the servers, following the instructions provided in this section.

6

Logs and Error Handling

Detailed execution logs are available under `$toolHome/logs`. The main log for this script is `migration_script.log`. Log is always appended to, so all previous execution attempts will remain in the log. To get the log for only the most recent script execution, either move the existing log to another location or delete it before re-running the script.

If any step fails:

1. Check the error details in `migration_script.log`
2. Fix the issue (for example, missing permissions, incorrect environment variables, or insufficient disk space)
3. Rerun the script:

```
$ bash migration_script.sh
```

Since the script is idempotent, previously completed steps will be skipped, and execution will resume from the failed step onward.

Example Run:

```
bash migration_script.sh
2025-09-17 13:52:15 [info] Installing dependencies...
2025-09-17 13:52:15 [info] Checking prerequisites...
2025-09-17 13:52:17 [info] Discovering WebLogic domain...
2025-09-17 13:52:56 [info] Discovering infrastructure...
2025-09-17 13:53:14 [info] Discovering datasources...
2025-09-17 13:53:25 [info] Building OCI Resource Manager stack...
2025-09-17 13:53:25 [info] Stack file created: /home/oracle/mig/oci-weblogic-
migration/oci/stack/owm_rm_202509171353.zip
2025-09-17 13:53:25 [info] Uploading owm_rm_202509171353.zip to OCI Object
Storage bucket wls_mbimg...
2025-09-17 13:53:28 [info] owm_rm_202509171353.zip uploaded to bucket
wls_mbimg.
2025-09-17 13:53:28 [info] Generated PAR URL (valid 6 months): https://
ax8cfrmecktw.objectstorage.us-ashburn-1.oci.customer-oci.com/p/
ItgGDQVxon0cVpmqP4yp5E3fZkX2xMJ4LKydQLTjVrb8DAWgh-0n0XZwTb0A8kk/n/
ax8cfrmecktw/b/wls_mbimg/o/owm_rm_202509171353.zip
2025-09-17 13:53:28 [info] Archiving WebLogic domain...
2025-09-11 14:11:13 [info] Migration script completed successfully!
```