

# Oracle® Cloud

## Migrating Oracle Java Cloud Service

## Instances to Oracle WebLogic Server for OCI

## Using WDT



F39761-05  
April 2023



Oracle Cloud Migrating Oracle Java Cloud Service Instances to Oracle WebLogic Server for OCI Using WDT,  
F39761-05

Copyright © 2021, 2023, 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, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

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

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

# Contents

## Preface

---

Documentation Accessibility	v
Diversity and Inclusion	v

## 1 Learn About Migrating to Oracle WebLogic Server for OCI

---

Why Migrate to Oracle WebLogic Server for OCI	1-1
Migration Scope	1-2
About Oracle WebLogic Server for OCI	1-2
Compare Oracle Java Cloud Service to Oracle WebLogic Server for OCI	1-3
Migrate to Oracle WebLogic Server for OCI Using WDT	1-3
About the Oracle WebLogic Deploy Tool	1-4

## 2 Prepare to Migrate Oracle Java Cloud Service to Oracle WebLogic Server for OCI

---

About Downtime Requirements	2-1
Select Oracle WebLogic Server for OCI Shapes	2-1
Configure Security Rules for the Network	2-2
Get Information About the Target Databases	2-2

## 3 Migrate an Instance

---

Perform Oracle Cloud Infrastructure Prerequisites	3-1
Create the Target Domain Using Oracle WebLogic Server for OCI	3-3
Get Information About the Application Databases	3-5
Get Information About the Service Instances	3-6
Migrate Oracle Identity Cloud Service Roles and Policies	3-9
Stop All Oracle WebLogic Server Processes on the Target	3-11
Install the Oracle WebLogic Deploy Tool	3-12
Discover the Oracle WebLogic Server Domain on the Source Instance	3-12
Copy Supporting Files to the Target Instance	3-14
Edit the Domain Model and Copy It to the Target Instance	3-19

Update the Oracle WebLogic Server Domain on the Target Instance	3-30
Configure Node Manager SSL on the Target Instance	3-31
Start All Oracle WebLogic Server Processes on the Target	3-34
Recreate Oracle Fusion Middleware Security Resources	3-35
Troubleshoot Migration Problems	3-38

## 4 Complete the Post-Migration Tasks

---

Test the Target	4-1
Start the SMTP Service on the Target	4-1
Migrate FastConnect and VPN Connections to Oracle Cloud Infrastructure	4-2
Clean Up Resources in Oracle Cloud Infrastructure	4-3

# Preface

*Migrating Oracle Java Cloud Service Instances to Oracle WebLogic Server for OCI Using WDT* explains how to migrate your existing Oracle Java Cloud Service instances to Oracle WebLogic Server for Oracle Cloud Infrastructure (Oracle WebLogic Server for OCI), if your Java Cloud Service instance uses a different database or classic DBCS.

## Topics:

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

## 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.

## 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

## Learn About Migrating to Oracle WebLogic Server for OCI

These topics help you learn about the benefits to migrating your existing Oracle Java Cloud Service instances to Oracle WebLogic Server for Oracle Cloud Infrastructure (Oracle WebLogic Server for OCI), and also provide an overview of the migration process and tools.

### Topics:

- [Why Migrate to Oracle WebLogic Server for OCI](#)
- [Migration Scope](#)
- [About Oracle WebLogic Server for OCI](#)
- [Compare Oracle Java Cloud Service to Oracle WebLogic Server for OCI](#)
- [Migrate to Oracle WebLogic Server for OCI Using WDT](#)
- [About the Oracle WebLogic Deploy Tool](#)

## Why Migrate to Oracle WebLogic Server for OCI

Oracle encourages you to migrate your existing cloud resources to Oracle WebLogic Server for OCI.

Oracle Cloud Infrastructure is Oracle's modern cloud platform that's based on the latest cloud technologies and standards. It provides more consistent performance and better features at lower costs. Oracle continues to invest in Oracle Cloud Infrastructure, including the addition of new regions, services, and features. See [Data Regions for Platform and Infrastructure Services](#).

Oracle WebLogic Server for OCI allows you to quickly provision a WebLogic domain in Oracle Cloud Infrastructure and at the same time provides you the flexibility to customize your WebLogic domain. Following are some of the advantages to migrate to Oracle WebLogic Server for OCI:

- Supports WebLogic Server major release versions after 12.2.1.4.0.
- Enables better efficient control of Oracle Cloud Infrastructure resources.
- Offers flexibility to add or remove patches.
- Allows managing the operating system or domain after provisioning, without any restrictions.
- Allows choosing any supported method to backup, scale, or patch a domain.
- Allows selecting the compartment where you want to create any of your resources.
- Allows full customization of the Oracle-managed load balancer.
- Supports cloning instances.
- Includes optional autoscaling feature.

- Supports a documented disaster recovery process. See Oracle WebLogic Server for Oracle Cloud Infrastructure Disaster Recovery.
- Allows WebLogic Server domains without Java Required Files (JRF).
- Integrates with multiple Oracle Cloud Infrastructure services to monitor, trace, deploy, scale and autoscale your WebLogic domain, and other Oracle Cloud Infrastructure services to manage and patch your WebLogic domain resources.
- Offers the option to pay per OCPU/Hour for Oracle WebLogic Server for OCI images with the entitlement to install any Oracle WebLogic Server version.

## Migration Scope

Before you migrate your existing Oracle Java Cloud Service instances to Oracle WebLogic Server for OCI, ensure that the service instance meets the prerequisites for the migration.

Oracle does *not* currently support the migration of Oracle Java Cloud Service instances that meet any of the following conditions:

- The service instance includes multiple domain partitions.
- The service instance is running Oracle WebLogic Server 11g and includes Java Message Service (JMS) migratable targets.

This guide does not include detailed procedures on the configuration of basic Oracle WebLogic Server for OCI security, network and storage resources that might be required to support your new WebLogic Server domain. Instead, this guide provides references to the Oracle WebLogic Server for OCI documentation as appropriate.

## About Oracle WebLogic Server for OCI

Oracle WebLogic Server for OCI is available as a set of applications in the Oracle Cloud Infrastructure Marketplace. After launching one of these applications, you use a simple wizard interface to configure and provision an Oracle WebLogic Server domain along with any supporting cloud resources like compute instances, networks and load balancers.

After launching a domain using the Marketplace applications, you track and monitor its progress as a stack using Resource Manager in Oracle WebLogic Server for OCI. A stack also provides a convenient method of deleting the cloud resources for a domain when you no longer require them.

Like Oracle Java Cloud Service, you can administer the domain and deploy Java EE applications to it just like on-premises domains. Use standard Oracle WebLogic Server tools like the administration console, WebLogic Deploy Tool (WDT), and WebLogic Scripting Tool (WLST). You can also administer the operating system on the compute instances using a secure shell (SSH) client and standard Linux tools.

See *About the Components of Oracle WebLogic Server for Oracle Cloud Infrastructure in Using Oracle WebLogic Server for OCI*.

# Compare Oracle Java Cloud Service to Oracle WebLogic Server for OCI

This topic helps you get familiar with basic Oracle WebLogic Server for OCI security, network, and storage concepts, and compare them to their equivalent concepts in Oracle Java Cloud Service.

The following table compares the functionality of Oracle Java Cloud Service to Oracle WebLogic Server for OCI

Oracle Java Cloud Service	Oracle WebLogic Server for OCI
Supports Oracle WebLogic Server 11g, 12.2.1.3, and 12.2.1.4 Also, supports Oracle WebLogic Server 12.1.3	Supports Oracle WebLogic Server 11g, 12.2.1.3, 12.2.1.4, and 14.1.1.
Will not support major version new releases of Oracle WebLogic Server	Will support major version new releases of Oracle WebLogic Server
All domains include the Java Required Files (JRF) components and require a database	Create basic and JRF-enabled WebLogic Server 12c domains All WebLogic Server 11g domains are JRF-enabled and require a database
Must use Oracle Java Cloud Service to backup, scale, or patch a domain Certain changes to the operation system and domain are not supported (see Administration Best Practices)	Can choose any supported method to backup, scale, or patch a domain; the documentation provides recommendations and best practices No restrictions on managing the operating system or domain after provisioning
Can provision an Oracle-managed load balancer in Oracle Cloud Infrastructure, or a user-managed load balancer running Oracle Traffic Director	Can provision an Oracle-managed load balancer in Oracle Cloud Infrastructure
Limited customization of the Oracle-managed load balancer	Full customization of the Oracle-managed load balancer
Can use Oracle Identity Cloud Service for authentication A security application is created in Oracle Identity Cloud Service for each domain	Can use Oracle Identity Cloud Service for authentication Must create a confidential application in Oracle Identity Cloud Service prior to creating a domain Confidential application, enterprise application, and App Gateway are created in Oracle Identity Cloud Service for each domain
Supports domain name as <code>&lt;domain&gt;.&lt;tenancy&gt;.jcs.ocp.oraclecloud.com</code> and a certificate for <code>*.jcs.ocp.oraclecloud.com</code>	Does not get domain name and only a self-sign certificate
Provides Root certificates and DNS domain names for the Load Balancer	Not supported

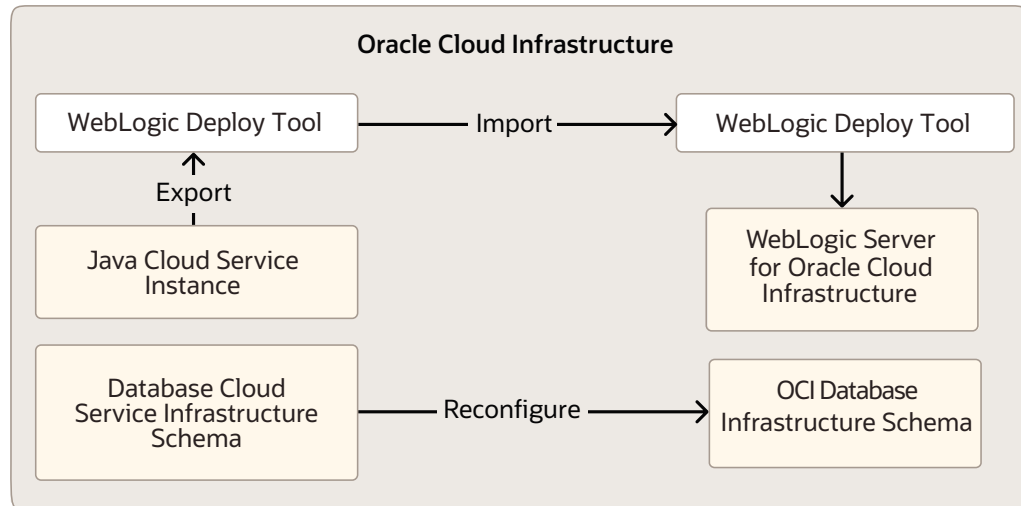
## Migrate to Oracle WebLogic Server for OCI Using WDT

Use the procedures in this guide to migrate service instances to Oracle WebLogic Server for OCI by using WebLogic Deploy Tooling (WDT) .



The following diagram shows the migration topology for a Oracle Java Cloud Service instance. The migration target is a domain created with Oracle WebLogic Server for OCI.

**Figure 1-1 Migration Topology**



At a high level, the migration process is comprised of these tasks:

1. Prepare for the migration and perform any prerequisite tasks in Oracle Java Cloud Service.
2. Use WDT to export the domain configuration, applications and other supporting files from your source Oracle Java Cloud Service instance.
3. Create the target domain using Oracle WebLogic Server for OCI.
4. Use WDT to import the domain configuration and applications to your target in Oracle Cloud Infrastructure.
5. Test your applications on the target instance, and perform any other post-migration tasks.

## About the Oracle WebLogic Deploy Tool

You can use various tools to automate many of the tasks involved in migrating an Oracle Java Cloud Service instance to Oracle WebLogic Server for OCI.

Oracle WebLogic Deploy Tool (WDT) is an open-source project. It provides scripts that enable you to discover and export the configuration and application files from one Oracle WebLogic Server domain, and then import the configuration and applications into another existing domain.

WDT exports a domain configuration as a metadata file, and automatically excludes sensitive information like passwords. When updating a domain, you also provide a metadata file. This file needs to describe only the resources that you want to add or update. If an application is already deployed, the tool compares the binaries and determines whether the application needs to be redeployed.

See [Oracle WebLogic Deploy Tool](#) project on GitHub.

# 2

## Prepare to Migrate Oracle Java Cloud Service to Oracle WebLogic Server for OCI

Before you migrate your service instances to Oracle WebLogic Server for OCI, understand how the migration affects your existing instances, identify the necessary compute shapes, and create the network and databases to support your migrated service instances.

### Topics:

- [About Downtime Requirements](#)
- [Select Oracle WebLogic Server for OCI Shapes](#)
- [Configure Security Rules for the Network](#)
- [Get Information About the Target Databases](#)

### About Downtime Requirements

The migration process does not affect the availability of your existing Oracle Java Cloud Service instance in Oracle Cloud Infrastructure. This instance continues to run and can serve client requests during this process.

You can discover and export the configuration from your source Oracle WebLogic Server domain while it is running. The WebLogic Deploy tool does not modify your domain or significantly affect its performance.

After a service instance is migrated successfully, clients can be rerouted to the new instance in Oracle WebLogic Server for OCI.

### Select Oracle WebLogic Server for OCI Shapes

Identify the compute shapes that provide similar IaaS resources in Oracle WebLogic Server for OCI to the shapes that you're currently using for your service instances in Oracle Java Cloud Service.

A compute shape defines the IaaS resources, such as OCPUs and memory, that are available to a specific node in a service instance.

To ensure that a migrated service instance has the same performance characteristics as the original instance, and can support an equivalent workload, choose Oracle WebLogic Server for OCI shapes that most closely map to the Oracle Java Cloud Service shapes that you specified when you created the instance.

You must also confirm that the chosen shapes are available in your Oracle Cloud tenancy. Oracle configures shape limits for an Oracle WebLogic Server for OCI region, or for a specific availability domain within a region. You can use the console to view the current shape limits for your tenancy, and to request a limit increase if necessary. See *Service Limits* in the Oracle Cloud Infrastructure documentation.

## Configure Security Rules for the Network

If your instance communicates with other resources in Oracle Cloud or on the Internet, create or update the security lists for your target Oracle Cloud Infrastructure network.

A security list is assigned to subnets within your virtual cloud network (VCN). It includes ingress and egress rules that specify the types of traffic allowed in and out of the instances within the subnet. You can update an existing security list, or create a new one and assign it to a subnet.

You might need to create security rules if your Oracle WebLogic Server for OCI instance communicates with external resources, including these Oracle WebLogic Server resources:

- JavaMail Session
- Foreign Java Naming and Directory Interface (JNDI) Provider
- Foreign Java Message Service (JMS) Server
- Messaging Bridge
- Store-and-Forward
- WebLogic Diagnostic Framework (WLDF) REST Action

See Security Lists in the Oracle Cloud Infrastructure documentation.

## Get Information About the Target Databases

Gather information about the Oracle Cloud Infrastructure Database instances that your target WebLogic Server domain will use. You will use this information to perform the migration.

- If you plan not to reuse an existing database, create a new database. See [Create a Database in \*Using Oracle WebLogic Server for OCI\*](#).
- If using **Autonomous Database**, get the following:
  - The compartment in which you've created the application database.
  - The Autonomous database where you want to create the schemas for the application database.
  - The name of the Autonomous database.
  - The OCID of the secret that contains the password for the ADMIN user to access the selected autonomous database.
- If using **Database System**, get the following:
  - The compartment in which you've created the database.
  - The compartment in which the database's VCN is found.
  - The VCN on which you've created the database. If this VCN is different than the WebLogic Server VCN, they cannot have overlapping CIDRs. For example, you cannot create a domain on VCN 10.0.0.0/16 that uses a database on VCN 10.0.0.1/24.
  - The DB system to use for this WebLogic domain.

- The database home within the selected DB system.
- The database home version.
- The database within the selected DB system where you want to create the JRF schemas for this domain.
- The Pluggable database (PDB) name, only if the selected database is running Oracle Database 12c or later.
- The name of a database user with database administrator (DBA) privileges.
- The OCID of the secret that contains the password for the database administrator.
- The database listen port (1521 by default)

# 3

## Migrate an Instance

Use Oracle WebLogic Deploy Tool (WDT) in Oracle Cloud Infrastructure to migrate your Oracle WebLogic Server domain resources and applications from your existing Oracle Java Cloud Service instance in Oracle Cloud Infrastructure to a new domain in Oracle WebLogic Server for OCI.



### Note:

Migration is not supported from 12.1.3 release to any of the existing versions of Oracle WebLogic Server for OCI.

In this procedure,

- Oracle Java Cloud Service is the source instance that you would be migrating from.
- Oracle WebLogic Server for OCI is the target instance that you would be migrating to.

### Topics:

- [Perform Oracle Cloud Infrastructure Prerequisites](#)
- [Create the Target Domain Using Oracle WebLogic Server for OCI](#)
- [Get Information About the Application Databases](#)
- [Get Information About the Service Instances](#)
- [Recreate Oracle Fusion Middleware Security Resources](#)
- [Migrate Oracle Identity Cloud Service Roles and Policies](#)
- [Stop All Oracle WebLogic Server Processes on the Target](#)
- [Install the Oracle WebLogic Deploy Tool](#)
- [Discover the Oracle WebLogic Server Domain on the Source Instance](#)
- [Copy Supporting Files to the Target Instance](#)
- [Edit the Domain Model and Copy It to the Target Instance](#)
- [Update the Oracle WebLogic Server Domain on the Target Instance](#)
- [Configure Node Manager SSL on the Target Instance](#)
- [Start All Oracle WebLogic Server Processes on the Target](#)
- [Troubleshoot Migration Problems](#)
- [Clean Up Resources in Oracle Cloud Infrastructure](#)

## Perform Oracle Cloud Infrastructure Prerequisites

Before you create an WebLogic Server domain with Oracle WebLogic Server for OCI, you must create the required infrastructure and database resources.

1. Create the following Oracle Cloud Infrastructure resources, if they don't already exist:
  - A compartment
  - A virtual cloud network (VCN) and at least one subnet. See [Create a Virtual Cloud Network](#).

 **Note:**

Oracle recommends to create Oracle WebLogic Server for OCI in private subnets. However, Oracle Java Cloud Service does reveal Admin console to the public internet, which is not a best practice. If you want to continue and use public subnets or reveal Admin console to the public internet you can create a cloud network as specified. See [Create a Virtual Cloud Network](#).

- A vault and encryption key

 **Note:**

Before you provision an instance, you can estimate the cost of the resources and services to use in your instance. See [Oracle Cloud Cost Estimator](#).

2. Create a database in Oracle Cloud Infrastructure Database, if one doesn't already exist. The database must allow the target domain to access the database listen port (1521 by default). See [Create a Database](#).

Oracle WebLogic Server for OCI will provision the Java Required Files (JRF) schema to this database.

 **Note:**

Do not use Oracle Java Cloud Service database schema

3. If your source instance uses Oracle Identity Cloud Service for authentication, then create a new confidential application in Oracle Identity Cloud Service for the target domain. See [Create a Confidential Application](#).

Identify the client ID and secret of the confidential application.

4. Use Oracle Cloud Infrastructure Vault to create secrets for the passwords that you need for the target domain.
  - WebLogic Server administrator password.  
You can use the WebLogic Server administrator password of the existing Oracle Java Cloud Service instance.
  - Database administrator password for the database you created in step 2
  - Client secret, if using Oracle Identity Cloud Service for the Oracle Identity Cloud Service confidential application you created in step 3

See [Create Secrets for Passwords](#).

See Before You Begin with Oracle WebLogic Server for Oracle Cloud Infrastructure in *Using Oracle WebLogic Server for OCI*.

## Create the Target Domain Using Oracle WebLogic Server for OCI

Launch the Oracle WebLogic Server for OCI application in the Oracle Cloud Infrastructure Marketplace to create a new domain. This domain must have the same topology and configuration as the source Oracle Java Cloud Service instance.

Before creating a domain, copy the OCIDs for the secrets that contain your Oracle WebLogic Server administrator password and your database password. Use the same credentials as your source instance.

1. Sign in to the Oracle Cloud Infrastructure Console.
2. Click the navigation menu, and then select **Marketplace**.
3. Select the corresponding Oracle WebLogic Server edition as your source instance.

 **Note:**

If you own a valid WebLogic License and a support contract, select the BYOL edition, else, select the corresponding UCM edition.

**Table 3-1 Software Editions**

Oracle Java Cloud Service	Oracle WebLogic Server for OCI
Standard Edition	<ul style="list-style-type: none"> <li>• Oracle WebLogic Server Standard Edition BYOL</li> <li>• Oracle WebLogic Server Enterprise Edition UCM</li> </ul>
Enterprise Edition	<ul style="list-style-type: none"> <li>• Oracle WebLogic Server Enterprise Edition BYOL</li> <li>• Oracle WebLogic Server Enterprise Edition UCM</li> </ul>
High Performance	<ul style="list-style-type: none"> <li>• Oracle WebLogic Suite BYOL</li> <li>• Oracle WebLogic Suite UCM</li> </ul>

4. For **Version**, select the same major version (x.y) as the source instance.  
 For example, 12.2.1.3 and 12.2.1.4 are the same major versions of Oracle WebLogic Server.
5. Select the compartment in which you want to create the stack.
6. Click **Launch Stack**.
7. Enter a name for your stack.
8. Click **Next**.
9. Enter a resource name prefix.

10. Select an Oracle Cloud Infrastructure shape that most closely matches the number of Oracle Compute Units (OCPU) and the amount of memory that are available in the Oracle Java Cloud Service shape in your source instance.
11. Enter the SSH public key.
12. Select the same number of managed servers as the source instance.
13. Enter the WebLogic Server user name, and paste the OCID for the secret that contains the WebLogic Server password.
14. For **Network Compartment**, select the same compartment you selected earlier upon launching the stack.
15. For **Virtual Cloud Network Strategy**, select **Use Existing VCN** and then select the virtual cloud network (VCN) where you want to create the domain.
16. For **Subnet Strategy**, select **Use Existing Subnet** or **Create New Subnet**.
17. If you're using an existing subnet, ensure that it has all the required ports. See *Create a Subnet for the Oracle WebLogic Server Nodes*.  
  
If you're creating a new subnet, specify a CIDR for the new subnet.  
  
The new subnet's CIDR should not overlap with any other subnet CIDRs in the existing VCN.
18. For **Subnet Compartment**, select the compartment to be used for the existing subnet.
19. If your source instance includes a load balancer, then provision a load balancer for the domain.
  - a. Select **Provision Load Balancer**.
  - b. Select an existing subnet where you want to create the load balancer.
20. If your source instance uses Oracle Identity Cloud Service for authentication, then configure Oracle Identity Cloud Service for the target domain.  
  
This configuration is supported only for WebLogic Server 12c, and also requires a load balancer.
  - a. Select **Enable Authentication Using Identity Cloud Service**.
  - b. Enter your Oracle Identity Cloud Service (IDCS) tenant name, which is also referred to as the instance ID.
  - c. Enter the client ID and encrypted client secret of an existing confidential application in this Oracle Identity Cloud Service instance.  
  
The client secret must be encrypted.
21. In Oracle Java Cloud Service all WebLogic domains are Java Required Files (JRF) enabled domains. To enable JRF in Oracle WebLogic Server for OCI a database is required to create a domain that includes the JRF components.  
  
Select the **Provision with JRF** checkbox to create a JRF-enabled domain
22. For **Database Strategy**, select **Database System**.
23. Select the required database attributes.
24. If your domain and database are on different VCNs, then you must configure local VCN peering. See *Set Local VCN Peering*.



Oracle WebLogic Server for OCI creates a public subnet in each VCN, and then creates a compute instance in each subnet. These compute instances run software to forward DNS requests across the VCNs.

25. Configure Application Database Strategy. See [Configure a Data Source for an Application Database](#).
26. If your domain and application database are on different VCNs, then you must configure local VCN peering. See [Set Local VCN Peering for an Application Database](#)  
  
Oracle WebLogic Server for OCI creates a public subnet in each VCN, and then creates a compute instance in each subnet. These compute instances run software to forward DNS requests across the VCNs.
27. Click **Next**, and then click **Create**.

## Get Information About the Application Databases

Gather information about the Oracle Cloud Infrastructure Database instances that your target Oracle Java Cloud Service instance will use to access your application schemas. You will use this information to perform the migration.

If the Application Database is running Oracle Cloud Infrastructure, then complete the following steps:

1. Access the Oracle Cloud Infrastructure console.
2. Click the menu icon, and under **Database**, select **Bare Metal, VM, and Exadata**
3. Select the **Region** and **Compartment** where your database resides.
4. Click the name of your database.
5. From the DB System Details page, record these values.
  - The public IP address of the first database node
  - The host name prefix for the database (for example, `myappdb`)
  - The domain name for the database (for example, `mydbsubnet.myvcn.oraclevcn.com`)
  - The database port number
  - The database name and unique name (for example, `ORCL` and `ORCL_iad1zj`)
6. If your database is running Oracle Database 12c or later, then identify the pluggable database (PDB) that contains your application schemas.
  - a. Use a Secure Shell (SSH) client to connect to the database node as the `opc` user.

```
ssh -i <privatekey> opc@<database_IP>
```

- b. Switch to the `oracle` user.

```
sudo su - oracle
```

- c. Locate the `ORACLE_HOME` directory for the database on the file system.

Example:

```
/u01/app/oracle/product/12.1.0.2/dbhome_1
```

- d. If you are accessing this database node for the first time, run the `oraenv` command to configure the environment.

```
source oraenv
```

When prompted, enter the database name (SID) and the `ORACLE_HOME` directory.

Example:

```
ORACLE_SID = ORCL
ORACLE_HOME = /u01/app/oracle/product/12.1.0.2/dbhome_1
```

- e. Start `sqlplus` as the `sysdba` role.

```
sqlplus / as sysdba
```

- f. Print the list of PDBs in this database.

```
SELECT PDB, NETWORK_NAME, CON_ID FROM CDB_SERVICES;
```

- g. In the command output, identify the PDB name.

Example:

```
MYPDB mypdb.mydbsubnet.myvcn.oraclevcn.com
```

7. Repeat [step 1](#) through [step 6](#) for any other application databases to which the target instance will connect.

## Get Information About the Service Instances

Gather information about your source and target instances. You will use this information when you perform the migration.

1. To get information of the source instance:
  - a. Access the Oracle Java Cloud Service console.
  - b. Click the name of your *source* instance.
  - c. From the Overview page, record these values.
    - The public IP address of the first node that is running the Administration Server
    - The host names of all Managed Server nodes (for example, `myinstance-wls-2`)
    - The names of the Administration Server and all Managed Servers (for example, `MyInstan_server_1`)
  - d. Access the Oracle WebLogic Server administration console on the source instance.

```
https://<source_admin_IP>:7002/console
```

If you did not enable console access when you created the source instance, see *Enable Console Access for a Service Instance* in *Administering Oracle Java Cloud Service*.

- e. After you sign in to the console, record the domain name (for example, `MyInstan_domain`).
  - f. From the **Domain Structure** panel, expand **Environment**, and then click **Clusters**.
  - g. Record the names of the clusters (for example, `MyInstan_cluster`).
  - h. From the **Domain Structure** panel, expand **Environment**, and then click **Machines**.
  - i. Record the names of the machines (for example, `MyInstan_machine_1`).
  - j. Return to the Instances page of the Oracle Java Cloud Service console.
2. To get information of the target instance:
- a. Access the Oracle Cloud Infrastructure Console.
  - b. From the navigation menu, click **Compute**, then click the name of your *target* instance.
  - c. From the **Compartment** dropdown, select the compartment in which your domain is created.
  - d. Click the name of the domain instance that has the Administration Server node.  
For example: `myinstance-wls-0`
  - e. Based on whether the Oracle WebLogic Server compute instances are assigned to a public subnet or private subnet, follow the steps:

- For public subnet:

 **Note:**

Oracle recommends to use a private subnet.

- i. Copy the public IP address value.
- ii. Access the Oracle WebLogic Server administration console on the *target* instance using the public IP address.

```
https://<IP-address>:7002/console
```

The default SSL port is 7002, unless it was changed during stack creation.

- iii. From the **Domain Structure** panel, expand **Environment**, and then click **Clusters**.
  - iv. Record the names of the clusters (for example, `MyInstan_cluster`).
  - v. From the **Domain Structure** panel, expand **Environment**, and then click **Machines**.
  - vi. Record the names of the machines (for example, `MyInstan_machine_1`).
- For private subnet:
    - i. Copy the private IP address value.

- ii. Return to the Compute Instances page.
- iii. Click the name of the bastion instance that's associated with the domain.  
For example: `myinstance-bastion-instance`
- iv. Copy the public IP address value.
- v. In the terminal window, run the following SSH command to access the bastion host:

```
ssh -C -D 1088 -i <path_to_private_key>  
opc@<bastion_public_ip>
```

where, `privateKeyPath` is the full path to the private SSH key that corresponds to the public SSH key that you specified when you created the domain and `bastionPublicIP` is the public IP address of the bastion host.

- vi. In your browser, set up the SOCKS (version 5) proxy configuration. Specify your local computer and the same SOCKS port that you used in your SSH command.
- vii. Access the Oracle WebLogic Server administration console on the target instance using the private IP address:

```
https://<private_ip_address>:7002/console
```

The default SSL port is 7002, unless it was changed during stack creation.

- viii. From the **Domain Structure** panel, expand **Environment**, and then click **Clusters**.
- ix. Record the names of the clusters (for example, `MyInstan_cluster`).
- x. From the **Domain Structure** panel, expand **Environment**, and then click **Machines**.
- xi. Record the names of the machines (for example, `MyInstan_machine_1`).
- xii. Access the Oracle WebLogic Server hosts on the target instance:

```
ssh -i <path_to_privatekey> -o ProxyCommand="ssh -W %h:%p  
-i <path_to_privatekey> opc@<Public_IP>"  
opc@<target_admin_IP>
```

- xiii. To copy files to the target instance:

```
scp -i <path_to_privatekey> -o ProxyCommand="ssh -W %h:%p  
-i <path_to_privatekey> opc@<Public_IP>" <source_domain>  
opc@<target_admin_IP>:</destination>
```

# Migrate Oracle Identity Cloud Service Roles and Policies

If your source Oracle Java Cloud Service instance uses Oracle Identity Cloud Service for authentication, then you must migrate the administrator roles and web tier policy to the target domain.


The source and target are each associated with a security application in Oracle Identity Cloud Service. The security application grants administrative rights for the WebLogic Server domain to specific users and groups in Oracle Identity Cloud Service.


1. Access the source Oracle Identity Cloud Service console.
2. Go to the Oracle Java Cloud Service instance.
3. In the **Instance Overview** page, click on the link against **IDCS Application**.
4. In the application details page, go to the **Web Tier Policy** tab, and then select **Export**.
5. Open the exported policy, and locate the following web tier policy in the response.

```
...
"webtierPolicy": [
  {
    "policyName": "jcs_cg_policy",
    "resourceFilters": [
      ...
    ]
  }
]
```

See [Get an App](#) in *REST API for Oracle Identity Cloud Service*.

6. Return to the Oracle Identity Cloud Service console.
7. From the application details page, click **Application Roles**.
8. Click **Export**, and then select **Export All**.
9. When prompted for confirmation, click **Export Application Roles**, and then click **Close**.
10. Click the job ID.


If a job ID link is not displayed, click the navigation drawer , select **Jobs**, and then click the job.

11. After the export job has finished, click **Download**. Save the file `AppRoleExport_<id>.csv`.
12. If your source and target are in different identity domains, then you must access the Oracle Identity Cloud Service console for the target identity domain.
13. In the target domain, click the navigation drawer , and then select **Applications**.
14. Click the security application for your target domain, `<stack>_enterprise_idcs_app_<timestamp>`.
15. Click **SSO Configuration**.
16. From the web tier policy that you exported with the REST API, identify the first entry in the `resourceFilters` block.

Example:

```
{
  "cloudgatePolicy": {
    "disableAuthorize": false,
    "allowCors": false,
    "requireSecureCookies": true,
    "webtierPolicy": [
      {
        "policyName": "jcs_cg_policy",
        "resourceFilters": [
          {
            "type": "regex",
            "filter": "/myapp/.*",
            "method": "oauth",
            "authorize": false
          },
          ...
        ]
      }
    ]
  }
}
```

Copy the value of the `filter` property.

17. Expand **Resources**.
18. Within the Resources section, click **Add**.
19. Enter a **Resource Name**.  
For example, `myapp`
20. For **Resource URL**, paste the value of the `filter` property.
21. If the filter's `type` property is `regex`, then select **Regex**.
22. Click **OK**.
23. Expand **Authentication Policy**. Under Managed Resources, click **Add**.
24. For **Resource**, select your new resource.
25. For **Authentication Method**, choose an option based on the filter's `method` property.
  - `oauth` - Select **Form or Access Token**
  - `public` - Select **Public**
  - `unsupported` - Select **Unsupported**
26. Click **Add**.
27. Repeat from step 17 to step 26 for each custom filter in the exported web tier policy.
28. Click the navigation drawer , and then select **Groups**.
29. Create these groups for the target domain.

 **Note:**

Ensure you have the required permissions to create groups.

- `<wls_domain_name>_Administrators`
- `<wls_domain_name>_Deployers`
- `<wls_domain_name>_Operators`
- `<wls_domain_name>_Monitors`

For example:

- `MyWLS_Domain_Administrators`
  - `MyWLS_Domain_Deployers`
  - `MyWLS_Domain_Operators`
  - `MyWLS_Domain_Monitors`
30. Open `AppRoleExport_<id>.csv`, and identify the users and groups assigned to the `Administrators` role in the source instance.
  31. Edit the `<wls_domain_name>_Administrators` group, and add the same users and groups as the `Administrators` role in the source instance.
  32. Repeat the previous step for the remaining roles in `AppRoleExport_<id>.csv`:
    - Add the members of the `Deployers` role to the `<wls_domain_name>_Deployers` group.
    - Add the members of the `Operators` role to the `<wls_domain_name>_Operators` group.
    - Add the members of the `Monitors` role to the `<wls_domain_name>_Monitors` group.
  33. Sign in to the WebLogic Server Administration Console for the target domain.  
`https://<target_admin_ip>:7002/console`
  34. Click **Security Realms**.
  35. Click the default realm.
  36. Click the **Roles and Policies** tab.
  37. From the Roles table, expand **Global Roles**, and then expand **Roles**.
  38. Click **View Role Conditions** for the `Admin` role.
  39. Click the group name assigned to this role. The default is **Administrators**.
  40. Enter `<wls_domain_name>_Administrators`.
  41. Click **OK**, and then click **Save**.
  42. From the breadcrumb links at the top of the page, click **Realm Roles**.
  43. Repeat from step 37 for the remaining administrator roles:
    - Map `Deployer` to `<wls_domain_name>_Deployers`
    - Map `Operator` to `<wls_domain_name>_Operators`
    - Map `Monitor` to `<wls_domain_name>_Monitors`

## Stop All Oracle WebLogic Server Processes on the Target

Before you perform the migration on the target domain, you must stop all Oracle WebLogic Server and Node Manager processes.

For information about how to stop a domain, see [Start and Stop a Domain](#).

## Install the Oracle WebLogic Deploy Tool

Download and install the Oracle WebLogic Server Deploy Tool (WDT) to your source and target Oracle Java Cloud Service instances.

WDT is an open-source project. It provides scripts that enable you to discover and export the configuration and application files from one Oracle WebLogic Server domain, and then import the configuration and applications into another domain.

1. Download the latest `weblogic-deploy.zip` file from the Oracle WebLogic Deploy Tool project on [GitHub](#).

Download version 0.22 or later.

2. Use a Secure Copy (SCP) client to upload the file to the Administration Server node in your *source* instance.

```
scp -i <privatekey> weblogic-deploy.zip opc@<source_admin_IP>:/tmp
```

3. Use a Secure Shell (SSH) client to connect to the node.

```
ssh -i <privatekey> opc@<source_admin_IP>
```

4. Change the owner of the `weblogic-deploy.zip` file to the `oracle` user.

```
sudo chown oracle:oracle /tmp/weblogic-deploy.zip
```

5. Switch to the `oracle` user.

```
sudo su - oracle
```

6. Extract `weblogic-deploy.zip` to `/u01`.

```
unzip -d /u01 /tmp/weblogic-deploy.zip
```

7. Disconnect from the node.
8. Repeat [step 2](#) through [step 7](#) for your *target* instance.

## Discover the Oracle WebLogic Server Domain on the Source Instance

Run the Oracle WebLogic Deploy Tool (WDT) on your source Oracle Java Cloud Service instance to capture its domain configuration, applications and other supporting files.

1. Use a Secure Shell (SSH) client to connect to the Administration Server node on the *source* instance as the `opc` user.

```
ssh -i <privatekey> opc@<source_admin_IP>
```



2. Switch to the `oracle` user.

```
sudo su - oracle
```

3. Determine the locations of the `DOMAIN_HOME` and `MIDDLEWARE_HOME` directories on the file system.

```
echo $DOMAIN_HOME
echo $MIDDLEWARE_HOME
```

4. Navigate to the `/u01/weblogic-deploy` directory.

```
cd /u01/weblogic-deploy
```

5. Run the `discoverDomain.sh` command and specify the following parameters:

- The locations of your `DOMAIN_HOME` and `MIDDLEWARE_HOME` directories
- The names of the two output files (model and archive)
- The `JRF` domain type

#### Caution:

You must specify the `JRF` domain type, so that the tool ignores standard resources and applications that are found in all service instances.

#### Format:

```
/u01/weblogic-deploy/bin/discoverDomain.sh -domain_home $DOMAIN_HOME -
oracle_home $MIDDLEWARE_HOME -model_file <source_domain>.yaml -
archive_file <source_domain>.zip -domain_type JRF
```

#### Example:

```
/u01/weblogic-deploy/bin/discoverDomain.sh -domain_home $DOMAIN_HOME -
oracle_home $MIDDLEWARE_HOME -model_file MyInstan_domain.yaml -
archive_file MyInstan_domain.zip -domain_type JRF
```

6. Verify that the `discoverDomain.sh` command completed successfully with no errors.

```
Total: WARNING: 1 SEVERE: 0
```

Ignore any warnings related to these resources, which you will address later:

- Trust Service Identity Asserter
- Oracle Identity Cloud Service (IDCS) Integrator Authentication Provider
- WebLogic Diagnostic Framework (WLDF) script actions

7. Copy the output files to `/tmp`.

```
cp <source_domain>* /tmp
```

8. Change the owner of the output files to the `opc` user.

```
exit
sudo chown opc:opc /tmp/<source_domain>*
```

9. Disconnect from the node.

## Copy Supporting Files to the Target Instance

Identify and copy any files to your target Oracle Java Cloud Service instance that are not managed by Oracle WebLogic Deploy Tool (WDT).

WDT automatically finds and archives the following types of files in your source instance's domain configuration. It also adds these files to your target instance's domain configuration:

- Application deployments
- Library deployments
- Custom keystores

Other files that your applications or domain resources require are not automatically managed by WDT, including files that are located outside the `DOMAIN_HOME` directory. You must manually copy these files to the target instance.

1. If your source instance includes custom Java Database Connectivity (JDBC) data sources for Autonomous databases, then you need to copy the Autonomous database wallet from the source instance or get it with the target instance helper scripts. And, depending on the data source target, copy into every node in the cluster or into the individual targeted nodes.

- If you are using a new Autonomous database, complete the instructions listed in *Create a Data Source for an Oracle Autonomous Database in Using Oracle WebLogic Server for OCI*.
- If you are reusing the same Autonomous database, complete the following steps:
  - a. Use SSH to connect to the Administration Server node in your *source* instance.

```
ssh -i <privatekey> opc@<source_admin_IP>
```

- b. CD to `/tmp` directory.
- c. Switch to the `oracle` user.

```
sudo su - oracle
```

- d. Zip the contents of target wallet directory:  
Format:

```
zip <MyAutonomousDBWallet.zip> /u01/data/domains/  
<sourceDomain>/config/<MyAutonomousDBWallet>/*
```

- e. Exit Oracle user.

- f. Change ownership of `MyAutonomousDBWallet.zip` to `opc`:

```
sudo chown opc:opc MyAutonomousDBWallet.zip
```

- g. Disconnect from the source instance.
- h. Use a Secure Copy (SCP) client to download the wallet from your Administration server node in your source instance to your local computer.

```
scp -i <privatekey> opc@<source_admin_IP>:/tmp/  
MyAutonomousDBWallet.zip .
```

- i. For each node where the data source is targeted to, upload the wallet:

 **Note:**

Following is sample of the target nodes:

```
resources:  
  JDBCSystemResource:  
    mydatasource:  
      Target: <cluster | adminServer |  
managedServer1 | managedServer2 | ...>  
      JdbcResource:  
        ...
```

- i. Use SSH to connect to the node.

```
ssh -i <privatekey> opc@<target_Node_IP>
```

If you are using a private subnet, use the following command to connect to the Administration Server node in your *target* instance:

```
scp -i <path_to_privatekey> -o ProxyCommand="ssh -W %h:%p -i  
<path_to_privatekey> opc@<Public_IP>" MyAutonomousDBWallet.zip  
opc@<target_Node_IP>:/<destination>
```

- ii. Change the owner of the `MyAutonomousDBWallet.zip` file to Oracle user:

```
sudo chown oracle:oracle /tmp/MyAutonomousDBWallet.zip
```

- iii. Switch to the `oracle` user.

```
sudo su - oracle
```

- iv. Create `MyAutonomousDBWallet` folder with the domain config folder

**Format:**

```
mkdir /u01/data/domains/<MyTarget_domain>/config/  
MyAutonomousDBWallet
```

- v. Extract wallet file into `MyAutonomousDBWallet` directory.

```
unzip MyAutonomousDBWallet.zip -d /u01/data/domains/  
<MyTarget_domain>/config/MyAutonomousDBWallet
```

Ensure that for this DB the definition in the domain model file corresponds to the following values:

- `oracle.net.tns_admin`: Full path to the location of the unzipped autonomous db wallet on target instance.  
Example: `/u01/data/domains/MyTarget_domain/config/MyAutonomousDBWallet`
  - `javax.net.ssl.trustStore`: Full path to the location of the `truststore.jks` on target instance.  
Example: `/u01/data/domains/MyTarget_domain/config/MyAutonomousDBWallet/truststore.jks`
  - `javax.net.ssl.keyStore`: Full path to the location of the `keystore.jks` on target instance.  
Example: `/u01/data/domains/MyTarget_domain/config/MyAutonomousDBWallet/keystore.jks`
2. If the Managed Servers in your source instance are configured to use custom identity and trust keystore files, then copy the keystore files from the Administration Server node to the Managed Server nodes.

Oracle WebLogic Server automatically stages application files to target Managed Server nodes, but does not do the same for keystore files.

- a. Use a Secure Shell (SSH) client to connect to the Administration Server node in your *target* instance.

```
ssh -i <privatekey> opc@<target_admin_IP>
```

If you are using a private subnet, use the following command to connect to the Administration Server node in your *target* instance:

```
ssh -i <path_to_privatekey> -o ProxyCommand="ssh -W %h:%p -i  
<path_to_privatekey> opc@<Public_IP>" opc@<target_admin_IP>
```

- b. Switch to the `oracle` user.

```
sudo su - oracle
```

- c. Use SSH to connect from the Administration Server node to the host name of the Managed Server node.

Example:

```
ssh myinstance-wls-2
```

- d. Navigate to the `DOMAIN_HOME` directory.

```
cd /u01/data/domains/<target_domain>
```

- e. Use a SCP client to download the archive file from the Administration Server node.

Format:

```
scp <target_admin_hostname>:/u01/weblogic-deploy/<source_domain>.zip .
```

Example:

```
scp myinstance-wls-1:/u01/weblogic-deploy/MyInstan_domain.zip .
```

- f. Extract the archive file to the current directory.

```
unzip <source_domain>.zip
```

- g. Disconnect from the Managed Server node.
  - h. Repeat Step 1 for any other Managed Servers that use custom keystores.
3. Use SSH to connect to the Administration Server node in your *source* instance.

```
ssh -i <privatekey> opc@<source_admin_IP>
```

4. Switch to the `oracle` user.

```
sudo su - oracle
```

5. Identify any supporting files that need to be copied to the target instance.
6. Copy the files to the `/tmp` directory.

Example:

```
cp /u01/myfiles/app.properties /tmp
```

 **Note:**

If you have multiple files to transfer, then consider adding them to a single archive file.

7. Change the owner of the files to the `opc` user.

Example:

```
exit  
sudo chown opc:opc /tmp/app.properties
```

8. Disconnect from the node.
9. Use SCP to download the files from the Administration Server node in your source instance to your local computer.

Example:

```
scp -i <privatekey> opc@<source_admin_IP>:/tmp/app.properties .
```

10. Use SCP to upload the files to the Administration Server node in your *target* instance.

Example:

```
scp -i <privatekey> app.properties opc@<target_admin_IP>:/<destination>
```

If you are using a private subnet, use the following command to upload the files to the Administration Server node in your *target* instance:

```
scp -i <path_to_privatekey> -o ProxyCommand="ssh -W %h:%p -i <path_to_privatekey> opc@<Public_IP>" <source_file>
opc@<target_admin_IP>:/<destination>
```

11. Use SSH to connect to the Administration Server node in your target instance.

```
ssh -i <privatekey> opc@<target_admin_IP>
```

If you are using a private subnet, use the following command to connect to the Administration Server node in your *target* instance:

```
ssh -i <path_to_privatekey> -o ProxyCommand="ssh -W %h:%p -i <path_to_privatekey> opc@<Public_IP>" opc@<target_admin_IP>
```

12. Change the owner of the files to the `oracle` user.

Example:

```
sudo chown oracle:oracle /tmp/app.properties
```

13. Switch to the `oracle` user.

```
sudo su - oracle
```

14. Move the files to the same location that they were found on the source instance.

Example:

```
mkdir /u01/myfiles
mv /tmp/app.properties /u01/myfiles
```

15. Disconnect from the node.

## Edit the Domain Model and Copy It to the Target Instance

Oracle WebLogic Deploy Tool (WDT) exports a domain as a YAML file, which is referred to as the metadata model. Modify the YAML file so that it matches the configuration of your target Oracle Java Cloud Service instance.

For security purposes, WDT excludes the values of all password configuration attributes in the model file.

The domain model syntax allows you to externalize variables in a separate properties file. Oracle recommends that you use a separate file to configure the passwords that are required in your domain configuration, including data source and keystore passwords. To refer to a variable in the model file, use the `@@PROP:<property_name>@@` format .

1. Using a Secure Copy (SCP) client, download the model file and archive file from the Administration Server node in your *source* instance to your local computer.

```
scp -i <privatekey> opc@<source_admin_IP>:/tmp/<source_domain>.* .
```

2. Create a backup copy of the model file.

```
cp <source_domain>.yaml <source_domain>.yaml.bak
```

3. Open the `<source_domain>.yaml` model file in a text editor.
4. If necessary, find the names of all servers, clusters and machines in the model file, and replace them with the corresponding server, cluster and machine names of your target instance.

### Note:

If the first eight characters of the source instance name are the same as the first eight characters in the target instance name, then this step is not required.

Example:

```
...
Cluster:
  MyTarget_cluster:
    ...
Server:
  MyTarget_adminserver:
    ...
    Machine: MyTarget_machine_1
  MyTarget_server_1:
    ...
    Machine: MyTarget_machine_1
    Cluster: MyTarget_cluster
    ServerTemplate: MyTarget_cluster_Template
    JTMigratableTarget:
      Cluster: MyTarget_cluster
      UserPreferredServer: MyTarget_server_1
```

```

...
MigratableTarget:
  MyTarget_server_1 (migratable):
    Cluster: MyTarget_cluster
    UserPreferredServer: MyTarget_server_1
...
ServerTemplate:
  MyTarget_cluster_Template:
    Cluster: MyTarget_cluster
    JTAMigratableTarget:
      Cluster: MyTarget_cluster
...
UnixMachine:
  MyTarget_machine_1:
...
JDBCSystemResource:
  'MyDataSource':
    Target: MyTarget_cluster
...
CoherenceClusterSystemResource:
  DataGridConfig:
    Target: MyTarget_cluster
...
Application:
  MyApp:
    Target: MyTarget_cluster

```

5. Find and remove the following applications from the model file, if they exist:

- OraJaaSmon
- sample-app
- \_\_auth-mgmt-app

In the following example, remove the highlighted lines.

```

Application:
  OraJaaSmon:
    SourcePath: wlsdeploy/applications/OraJaaSmon.war
    ModuleType: war
    StagingMode: nostage
    Target: MyTarget_adminserver
  'sample-app':
    SourcePath: 'wlsdeploy/applications/sample-app.war'
    ModuleType: war
    StagingMode: stage
    Target: MyTarget_cluster
  '__auth-mgmt-app':
    SourcePath: 'wlsdeploy/applications/__auth-mgmt-app.war'
    ModuleType: war
    StagingMode: stage
    Target: MyTarget_adminserver

```

6. Find and remove all occurrences of the following attributes from the model file:

- ListenAddress



- NodeManagerPasswordEncrypted
- CredentialEncrypted
- FrontendHost

 **Note:**

If you want to reuse the source instance (Oracle Java Cloud Service) ports, then you need to consider Load Balancer configuration and security rules in target instance (Oracle WebLogic Server for OCI).

7. Find and remove the `NMProperties` node from the model file.
8. For each server in the model file, find and remove the `PublicAddress` attribute from the following default `NetworkAccessPoint` nodes:
  - `channel-dep`
  - `SecuredExternAdmin`
  - `ExternAdmin`
  - `SecuredExternContent`
  - `ExternContent`

In the following example, the highlighted line should be removed.

```
Server:
  MyInstan_adminserver:
    ...
    NetworkAccessPoint:
      'channel-dep':
        ...
        PublicAddress: 203.0.113.10
```

9. Find the `PublicAddress` attribute of any custom `NetworkAccessPoint` nodes in the model file (not in the previous list, in step 7), and replace the current value with the corresponding public IP address that is assigned to your target instance.

Example:

```
Server:
  MyInstan_adminserver:
    ...
    NetworkAccessPoint:
      MyChannel:
        ...
        PublicAddress: <target_IP>
```

10. Find and remove all occurrences of the following attributes, under `domaininfo`:
  - `AdminUserName`
  - `AdminPassword`
11. Within the `SecurityConfiguration` node in your model file, remove the `Realm` node and any child nodes, if they exist.

In the following example, remove the highlighted lines.

```
SecurityConfiguration:
  ...
  Realm:
    myrealm:
      ...
```

12. Within the `SecurityConfiguration` node in your model file, remove the `NodeManagerPasswordEncrypted` attribute.

In the following example, remove the highlighted lines.

```
SecurityConfiguration:
  ...
  NodeManagerPasswordEncrypted:
  ...
```

13. For each server in the model file, find the `Arguments` attribute within the `ServerStart` node:

- If you configured any custom startup arguments for a server in your source instance, then replace the current value of `Arguments` with the custom arguments only.
- If you did not configure any custom startup arguments for a server, then remove the entire `Arguments` line and the `ServerStart` node.

In the following example, the server has custom startup arguments:

```
MyInstan_server_1:
  ...
  ServerStart:
    Arguments: '-Dmy.custom.arg=true'
```

14. Create a file named `wdt.properties`.
15. If the servers in your source instance are configured to use custom identity and trust keystore files, then update the model file with the keystore passwords.

If you are enabling SSL on Oracle WebLogic Server for OCI, then complete the instructions at [Configure SSL for a Domain](#).

- a. Enter the required passwords for your keystores and private keys as properties in the `wdt.properties` file.

Example:

```
keystore1.password=<your_password>
trustkeystore1.password=<your_password>
privatekey1.password=<your_password>
```

- b. For each server in your model file, find the following attributes, and replace the current placeholder values with references to the corresponding properties:
- `CustomIdentityKeyStorePassPhraseEncrypted`
  - `CustomTrustKeyStorePassPhraseEncrypted`

## Example:

```

Server:
  MyInstan_server_1:
    ...
    CustomIdentityKeyStorePassPhraseEncrypted:
'@@PROP:keystore1.password@@'
    CustomTrustKeyStorePassPhraseEncrypted:
'@@PROP:trustkeystore1.password@@'

```

- c. For each server in your model file, find the `ServerPrivateKeyPassPhraseEncrypted` attribute in the `SSL` node, and then replace the current placeholder values with a reference to the corresponding property.

## Example:

```

Server:
  MyInstan_server_1:
    ...
    SSL:
      ServerPrivateKeyPassPhraseEncrypted:
'@@PROP:privatekey1.password@@'

```

- d. For each server in your model file, if the `CustomIdentityKeyStoreType` or `CustomTrustKeyStoreType` attribute is set to the value `KSS`, then set the location of your KSS keystores.

Add the following attributes to the server, if not already present:

- `CustomIdentityKeyStoreFileName`: `<keystore_url>`
- `CustomTrustKeyStoreFileName`: `<trust_keystore_url>`

## Example:

```

Server:
  MyInstan_server_1:
    ...
    CustomIdentityKeyStoreType: KSS
    CustomTrustKeyStoreType: KSS
    CustomIdentityKeyStoreFileName: 'kss://system/mykeystore'
    CustomTrustKeyStoreFileName: 'kss://system/trust'

```

- e. Add the following attributes to the `SSL` node for your administration server, if they are not already present:

- `Enabled`: `true`
- `ListenPort`: `9072`

## Example:

```

Server:
  MyInstan_adminserver:
    ...
    SSL:
      Enabled: true
      ListenPort: 9072

```

```

ServerPrivateKeyPassPhraseEncrypted:
'@@PROP:privatekey1.password@@'

```

- f. For each managed server in your model file, add the following attributes to the SSL node, if they are not already present:

- Enabled: true
- ListenPort: 9074

Example:

```

Server:
  MyInstan_server_1:
    ...
    SSL:
      Enabled: true
      ListenPort: 9074
      ServerPrivateKeyPassPhraseEncrypted:
'@@PROP:privatekey1.password@@'

```

 **Note:**

If the application database is an Oracle Cloud Infrastructure database and it is in a different VCN than the target instance, then to enable communication between WebLogic servers and the database, you might have to complete the VNC pairing process. See .

16. If your source instance includes custom Java Database Connectivity (JDBC) data sources, then provide the location and password of the application databases in Oracle Cloud Infrastructure.
- a. Identify the OCI DB data sources found within the `JDBCSystemResource` node in your model file.
  - b. Enter the required passwords for your data sources as properties in the `wdt.properties` file.

Example:

```

datasource1.password=<your_password>
datasource2.password=<your_password>

```

- c. For each data source in your model file, find the `PasswordEncrypted` attribute and replace the current placeholder value with a reference to the corresponding property.

Example:

```

JDBCSystemResource:
  MyDataSource1:
    ...
    JdbcResource:
      ...
      JDBCDriverParams:

```

URL: ...

**PasswordEncrypted:** '@@PROP:datasource1.password@'

- d. For each data source in your model file, find the `URL` attribute and replace the current value with the URL to the corresponding Oracle Cloud Infrastructure Database.

The following table shows the URL format to use, depending on the Oracle Database version, and whether you created a Virtual Machine (VM) or Bare Metal database type.

Database Version	Database Type	URL Format
12c	VM	<code>jdbc:oracle:thin:@//&lt;db_hostname&gt;-scan.&lt;db_domain&gt;:&lt;db_port&gt;/&lt;pdb_name&gt;.&lt;db_domain&gt;</code>
12c	Bare Metal	<code>jdbc:oracle:thin:@//&lt;db_hostname&gt;.&lt;db_domain&gt;:&lt;db_port&gt;/&lt;pdb_name&gt;.&lt;db_domain&gt;</code>
11g	VM	<code>jdbc:oracle:thin:@//&lt;db_hostname&gt;-scan.&lt;db_domain&gt;:&lt;db_port&gt;/&lt;db_unique_name&gt;.&lt;db_domain&gt;</code>
11g	Bare Metal	<code>jdbc:oracle:thin:@//&lt;db_hostname&gt;.&lt;db_domain&gt;:&lt;db_port&gt;/&lt;db_unique_name&gt;.&lt;db_domain&gt;</code>

If you did not specify a PDB name when you created an Oracle Cloud Infrastructure Database that is running Oracle Database 12c, the default name is `<db_name>_pdb1`.

The following example shows a Virtual Machine database named `myappdb`, that is running Oracle Database 12c, and contains a PDB named `pdb1`:

JDBCDriverParams:

**URL:** `jdbc:oracle:thin:@//myappdb-scan.mydbsubnet.myvcn.oraclevcn.com:1521/pdb1.mydbsubnet.myvcn.oraclevcn.com`

17. If your source instance includes custom Java Database Connectivity (JDBC) data sources for Autonomous databases, then provide the password of the new or existing autonomous databases in Oracle Cloud Infrastructure.
- Identify the Autonomous database data sources found within the `JDBCSystemResource` node in your model file.
  - Enter the required passwords for your data sources as properties in the `wdt.properties` file.

Example:

`atpdataource1.password=<your_password>`  
`atpdataource2.password=<your_password>`

- For each data source in your model file, find the `PasswordEncrypted` attribute and replace the current placeholder value with a reference to the corresponding property.

**Example:**

```
JDBCSystemResource:
  MyDataSource1:
    ...
    JdbcResource:
      ...
      JDBCDriverParams:
        URL: ...
        PasswordEncrypted:
'@@PROP:atpdatasource1.password@@'
```

**d. Ensure the following properties match with your target instance directory hierarchy.**

- `oracle.net.tns_admin`: Full path to the location of the unzipped autonomous db wallet on target instance.
- `javax.net.ssl.trustStore`: Full path to the location of the `truststore.jks` on target instance.
- `javax.net.ssl.keyStore`: Full path to the location of the `keystore.jks` on target instance.

**Example:**

```
JDBCSystemResource:
  MyDataSource1:
    ...
    JdbcResource:
      ...
      JDBCDriverParams:
        ...
        Properties:
          oracle.net.tns_admin:
            Value: /u01/data/domains/
MyTarget_domain/config/MyAutonomousDBWallet
          javax.net.ssl.trustStore:
            Value: /u01/data/domains/
MyTarget_domain/config/MyAutonomousDBWallet/truststore.jks
          javax.net.ssl.keyStore:
            Value: /u01/data/domains/
MyTarget_domain/config/MyAutonomousDBWallet/keystore.jks
```

 **Note:**

In this example, the folder `/u01/data/domains/MyTarget_domain/config/` must already exist in the target instance. However, `MyAutonomousDBWallet` is a new directory where the wallet was unzipped.

18. If your source instance includes any Foreign JNDI Providers, Foreign JMS Servers, JMS Bridge Destinations, or Store-and-Forward (SAF) Contexts, then provide the locations and passwords for these external resources.

- a. Identify the `ForeignJNDIProvider` nodes in your model file.
- b. Enter the required passwords for your Foreign JNDI Providers as properties in the `wdt.properties` file.

Example:

```
foreignjndi1.password=<your_password>
```

- c. For each `ForeignJNDIProvider` node in your model file, find the `PasswordEncrypted` attribute and replace the current placeholder value with a reference to the corresponding property.

Also update the `ProviderUrl` attribute if the location of this JNDI server is different than the JNDI server in the source environment.

Example:

```
ForeignJNDIProvider:
  MyJNDIProvider1:
    PasswordEncrypted: '@@PROP:foreignjndi1.password@@'
    ProviderUrl: 't3://myjndiserver.example.com:9073'
```

- d. Identify the `ForeignServer` nodes in your model file.
- e. Enter the required passwords for your Foreign JMS Servers as properties in the `wdt.properties` file.

Example:

```
foreignjms1.password=<your_password>
```

- f. For each `ForeignServer` node in your model file, find the `PasswordEncrypted` and `JNDIPropertiesCredentialEncrypted` attributes, and replace the current placeholder value with a reference to the corresponding properties.

Also update the `ConnectionURL` attribute if the location of this JMS server is different than the JMS server in the source environment.

Example:

```
ForeignServer:
  MyForeignJMS1:
    ConnectionURL: 't3://myjms.example.com:9073'
    JNDIPropertiesCredentialEncrypted:
      '@@PROP:foreignjms1.password@@'
    ForeignConnectionFactory:
      MyForeignJMS1Factory:
        PasswordEncrypted: '@@PROP:foreignjms1.password@@'
```

- g. Identify the `JMSBridgeDestination` nodes in your model file.
- h. Enter the required passwords for your JMS Bridge Destinations as properties in the `wdt.properties` file.

Example:

```
jmsbridge1.password=<your_password>
```

- i. For each `JMSBridgeDestination` node in your model file, find the `UserPasswordEncrypted` attribute and replace the current placeholder value with a reference to the corresponding property.

Also update the `ConnectionURL` attribute if the location of this bridge destination is different than the bridge destination in the source environment.  
Example:

```
JMSBridgeDestination:
  MyBridgeDest1:
    ConnectionURL: 't3://myjms.example.com:9073'
    UserPasswordEncrypted: '@@PROP:jmsbridge1.password@@'
```

- j. Identify the `SAFLoginContext` nodes in your model file.
- k. Enter the required passwords for your Store-and-Forward Contexts as properties in the `wdt.properties` file.

Example:

```
saf1.password=<your_password>
```

- l. For each `SAFLoginContext` node in your model file, find the `PasswordEncrypted` attribute and replace the current placeholder value with a reference to the corresponding property.

Also update the `LoginURL` attribute if the Store-and-Forward destination server is different than the server in the source environment.  
Example:

```
SAFLoginContext:
  MySAF1:
    PasswordEncrypted: '@@PROP:saf1.password@@'
    LoginURL: 't3://myjms.example.com:9073'
```

19. If your source instance includes any JavaMail sessions, then provide the passwords for these mail sessions.

- a. Identify the `MailSession` nodes in your model file.
- b. Enter the required passwords for your mail sessions as properties in the `wdt.properties` file.

Example:

```
mailsession1.password=<your_password>
```

- c. For each `MailSession` node in your model file, find the `SessionPasswordEncrypted` attribute and replace the current placeholder value with a reference to the corresponding property.

Example:

```
MailSession:
  MyMailSession1:
    SessionPasswordEncrypted:
      '@@PROP:mailsession1.password@@'
```



- d. For each `MailSession` node in your model file, find the `Properties` attribute and replace any password placeholder values with references to the corresponding properties.

Example:

```
MailSession:
  MyMailSession1:
    Properties:
      ...
      mail.smtp.password: '@@PROP:mailsession1.password@@'
      mail.imap.password: '@@PROP:mailsession1.password@@'
```

20. If your source instance includes any custom WebLogic Diagnostic Framework (WLDF) REST notification endpoints, then provide the locations and passwords for these endpoints.

- a. Identify the `RestNotification` nodes in your model file.
- b. Enter the required passwords for your notification endpoints as properties in the `wdt.properties` file.

For example:

```
restnotification1.password=<your_password>
```

- c. For each `RestNotification` node in your model file, find the `HttpAuthenticationPasswordEncrypted` attribute and replace the current placeholder value with a reference to the corresponding property.

Also update the `EndpointUrl` attribute if the destination server is different than the server in the source environment.

Example:

```
RestNotification:
  MyNotification1:
    HttpAuthenticationPasswordEncrypted:
      '@@PROP:restnotification1.password@@'
    EndpointUrl: 'http://myserver.example.com:9073/notify'
```

21. Use a Secure Copy (SCP) client to upload files to the Administration Server node in your *target* instance.

```
scp -i <privatekey> <source_files> opc@<target_admin_IP>:/<destination>
scp -i <privatekey> wdt.properties opc@<target_admin_IP>:/<destination>
```

If you are using a private subnet, use the following command to upload the files to the Administration Server node in your *target* instance:

```
scp -i <path_to_privatekey> -o ProxyCommand="ssh -W %h:%p -i
<path_to_privatekey> opc@<Public_IP>" <source_files>
opc@<target_admin_IP>:/<destination>
```

## Update the Oracle WebLogic Server Domain on the Target Instance

Run the Oracle WebLogic Deploy Tool (WDT) on your target Oracle Java Cloud Service instance to update its domain configuration and to deploy your applications.

1. Use a Secure Shell (SSH) client to connect to the Administration Server node on the *target* instance as the `opc` user.

```
ssh -i <privatekey> opc@<target_admin_IP>
```

If you are using a private subnet, use the following command to connect to the Administration Server node in your *target* instance:

```
ssh -i <path_to_privatekey> -o ProxyCommand="ssh -W %h:%p -i <path_to_privatekey> opc@<Public_IP>" opc@<target_admin_IP>
```

2. Change the owner of the archive, model, and properties files to the `oracle` user.

```
sudo chown oracle:oracle /tmp/<source_domain>.*
sudo chown oracle:oracle /tmp/wdt.properties
```

3. Switch to the `oracle` user.

```
sudo su - oracle
```

4. Navigate to the `/u01/weblogic-deploy` directory.

```
cd /u01/weblogic-deploy
```

5. Copy the input files to the current directory.

```
cp /tmp/<source_domain>.* .
cp /tmp/wdt.properties .
```

6. Run the `validateModel.sh` command and specify the following parameters:

- The location of your `MIDDLEWARE_HOME` directory
- The names of the model, archive and properties files
- The `JRF` domain type

Format:

```
/u01/weblogic-deploy/bin/validateModel.sh -
oracle_home $MIDDLEWARE_HOME -model_file <source_domain>.yaml -
archive_file <source_domain>.zip -variable_file wdt.properties -
domain_type JRF
```

**Example:**

```
/u01/weblogic-deploy/bin/validateModel.sh -oracle_home $MIDDLEWARE_HOME -
model_file MyInstan_domain.yaml -archive_file MyInstan_domain.zip -
variable_file wdt.properties -domain_type JRF
```

7. Verify that the `validateModel.sh` command completed successfully. Correct any errors.

```
###<timestamp> <INFO> <validate> <__perform_model_file_validation>
<WLSPLY-05403>
<Validation of /u01/weblogic-deploy/<source_domain>.yaml completed with 0
error(s), 0 warning(s) and 0 info(s) items>
validateModel.sh completed successfully (exit code = 0)
```

8. Run the `updateDomain.sh` command and specify the following parameters:

- The locations of your `DOMAIN_HOME` and `MIDDLEWARE_HOME` directories
- The names of the model, archive, and properties files
- The `JRF` domain type

**Format:**

```
/u01/weblogic-deploy/bin/updateDomain.sh -domain_home /u01/data/domains/
<target_domain> -oracle_home /u01/app/oracle/middleware/ -model_file
<source_domain>.yaml -archive_file <source_domain>.zip -variable_file
wdt.properties -domain_type JRF
```

**Example:**

```
/u01/weblogic-deploy/bin/updateDomain.sh -domain_home /u01/data/domains/
MyInstan_domain -oracle_home /u01/app/oracle/middleware/ -model_file
MyInstan_domain.yaml -archive_file MyInstan_domain.zip -variable_file
wdt.properties -domain_type JRF
```

9. Verify that the `updateDomain.sh` command completed successfully with no errors.

```
updateDomain.sh completed successfully (exit code = 0)
```

Log files are in the `/u01/weblogic-deploy/logs` directory.

10. Disconnect from the Administration Server node.

## Configure Node Manager SSL on the Target Instance

If you configured your source Oracle Java Cloud Service instance to use custom identity or trust keystores, then you must manually configure the Node Manager on each node in the target instance to use the custom keystores.

1. Use a Secure Shell (SSH) client to connect to the Administration Server node on the *target* instance as the `opc` user.

```
ssh -i <privatekey> opc@<target_admin_IP>
```

If you are using a private subnet, use the following command to connect to the Administration Server node in your *target* instance:

```
ssh -i <path_to_privatekey> -o ProxyCommand="ssh -W %h:%p -i
<path_to_privatekey> opc@<Public_IP>" opc@<target_admin_IP>
```

**2. Switch to the oracle user.**

```
sudo su - oracle
```

**3. Using the model file and properties file, identify the SSL configuration for the servers on this node:**

- The identity keystore file, path, and password
- The trust keystore file, path, and password
- The key alias and password

Example:

```
Server:
...
MyInstan_server_1:
...
    CustomIdentityKeyStoreFileName: wlsdeploy/servers/
MyInstan_server_1/identity.jks
    CustomTrustKeyStoreFileName: wlsdeploy/servers/
MyInstan_server_1/trust.jks
    CustomIdentityKeyStorePassPhraseEncrypted:
'@@PROP:keystore1.password@@'
    CustomTrustKeyStorePassPhraseEncrypted:
'@@PROP:trustkeystore1.password@@'
...
SSL:
    ServerPrivateKeyAlias: server_cert
    ServerPrivateKeyPassPhraseEncrypted:
'@@PROP:privatekey1.password@@'
```

**4. Edit the nodemanager.properties file located under the DOMAIN\_HOME directory.**

```
vi $DOMAIN_HOME/nodemanager/nodemanager.properties
```

**5. Add the following lines to the end of the file. Specify the full path to the keystore files.**

```
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityKeystoreType=jks
CustomIdentityKeyStoreFileName=/u01/data/domains/<target_domain>/
wlsdeploy/servers/<target_server_name>/<identity_keystore_file>
CustomIdentityKeyStorePassPhrase=<identity_keystore_password>
CustomIdentityPrivateKeyPassPhrase=<key_password>
CustomIdentityAlias=<key_alias>
CustomTrustKeystoreType=jks
CustomTrustKeyStoreFileName=/u01/data/domains/<target_domain>/
```

```
wlsdeploy/servers/<target_server_name>/<trust_keystore_file>
CustomTrustKeyStorePassPhrase=<trust_keystore_password>
```

**Example:**

```
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityKeystoreType=jks
CustomIdentityKeyStoreFileName=/u01/data/domains/MyInstan/wlsdeploy/
servers/MyInstan_adminserver/myidentity.jks
CustomIdentityKeyStorePassPhrase=<identity_keystore_password>
CustomIdentityPrivateKeyPassPhrase=<key_password>
CustomIdentityAlias=server_cert
CustomTrustKeystoreType=jks
CustomTrustKeyStoreFileName=/u01/data/domains/MyInstan/wlsdeploy/servers/
MyInstan_adminserver/mytrust.jks
CustomTrustKeyStorePassPhrase=<trust_keystore_password>
```

6. Edit the `setDomainEnv.sh` file located under the `DOMAIN_HOME` directory.

```
vi $DOMAIN_HOME/bin/setDomainEnv.sh
```

7. Add the following line to the end of the file.

```
export WLST_PROPERTIES="${WLST_PROPERTIES} -
Dweblogic.ssl.JSSEEnabled=true -Dweblogic.security.SSL.enableJSSE=true -
Dweblogic.security.SSL.ignoreHostnameVerification=true -
Dweblogic.security.TrustKeyStore=CustomTrust -
Dweblogic.security.CustomTrustKeyStoreFileName=/u01/data/domains/
<target_domain>/wlsdeploy/servers/<target_server_name>/
<trust_keystore_file> -Dweblogic.security.CustomTrustKeyStoreType=JKS"
```

**Example:**

```
export WLST_PROPERTIES="${WLST_PROPERTIES} -
Dweblogic.ssl.JSSEEnabled=true -Dweblogic.security.SSL.enableJSSE=true -
Dweblogic.security.SSL.ignoreHostnameVerification=true -
Dweblogic.security.TrustKeyStore=CustomTrust -
Dweblogic.security.CustomTrustKeyStoreFileName=/u01/data/domains/MyInstan/
wlsdeploy/servers/MyInstan_adminserver/mytrust.jks -
Dweblogic.security.CustomTrustKeyStoreType=JKS"
```

8. Connect to all Managed Server nodes in the target instance that use custom keystores, and then repeat Steps 4 to 7.

**Example:**

```
ssh myinstance-wls-2
vi $DOMAIN_HOME/nodemanager/nodemanager.properties
vi $DOMAIN_HOME/bin/setDomainEnv.sh
exit
```

9. Disconnect from the Administration Server node.

# Start All Oracle WebLogic Server Processes on the Target

After you update the target domain, you must restart all Oracle WebLogic Server and Node Manager processes.

The Administration Server must be running before you start any Managed Servers.

- Start the Administration Server on the first node and verify that it started successfully.
- Start the Managed Servers on all nodes.

If you previously shut down the server processes by using the `kill` command, then Node Manager restarts them for you automatically. Otherwise, you must start the server processes manually.

1. Identify the IP address of the node in your domain.

The name of the node is `servicename-wls-n`, where `servicename` is the resource name prefix you provided during stack creation. The Administration Server runs on the first node, `servicename-wls-0`

- If your domain is on a public subnet, then use the public IP address of the compute instance.
- If your domain is on a private subnet, then use the public IP address of the bastion and the private IP address of the compute instance.

2. Open an SSH connection to the node as the `opc` user.

```
ssh -i <path_to_private_key> opc@<node_public_ip>
```

Or,

```
ssh -i <path_to_private_key> -o ProxyCommand="ssh -W %h:%p -i  
<path_to_private_key> opc@<bastion_public_ip>" opc@<node_private_ip>
```

3. Change to the `oracle` user.

```
sudo su - oracle
```

4. Execute the `restart_domain.sh` script.

```
/opt/scripts/restart_domain.sh -o [restart]
```

- `restart` - stop and then start the Node Manager and all servers on this node

If you modified certain settings after creating the domain, like the administrator password or port number, then you must provide additional parameters to `restart_domain.sh`:

- `-u` - User name for the domain administrator
- `-p` - Password for the domain administrator
- `--adminhost` - Hostname of the administration server

- `--port` - Port number of the administration server
- `--domain-name` - Name of the domain
- `--domain-home` - Home directory for the domain
- `--admin-servername` - Name of the administration server

If your domain is running Oracle WebLogic Server 11g and you configured the servers to use a custom trust keystore, then set the location of the file in the `JAVA_OPTIONS` environment variable before running `restart_domain.sh`. For example:

```
export JAVA_OPTIONS="${JAVA_OPTIONS} -
Dweblogic.security.SSL.trustedCAKeyStore=/u01/data/mytrust.jks"
```

5. Use SSH to connect from the Administration Server node to the host name of each Managed Server node in this domain, and then repeat [step 1](#) through [step 4](#).

Example:

```
ssh mydomain-wls-1
```

6. Disconnect from the Administration Server node.
7. Verify that you can sign in to the Oracle WebLogic Server Administration Console.

```
https://<target_admin_IP>:7002/console
```

See [Access the WebLogic Console](#) in *Using Oracle WebLogic Server for OCI*.

## Recreate Oracle Fusion Middleware Security Resources

If you created any custom users, groups, roles or policies in your source Oracle Java Cloud Service instance, then you must recreate them in the target Oracle WebLogic Server for OCI domain.

Application Migration does not automatically migrate any Oracle Fusion Middleware security resources that you created to support your applications, including users, roles and policies. Perform this task if your source domain includes applications that use Oracle Fusion Middleware (FMW), Oracle Platform Security Services (OPSS), Oracle Application Development Framework (ADF) or Oracle Web Services Manager (WSM).

1. Access the Fusion Middleware Control Console for your *source* instance.

```
https://<source_admin_ip>:7002/em
```

2. Sign in to the console as your Oracle WebLogic Server system administrator.

3. From a different browser window or tab, sign in to the Fusion Middleware Control Console for your *target* domain.

```
https://<target_admin_ip>:7002/em
```

See [Access the Fusion Middleware Control Console](#) in *Using Oracle WebLogic Server for OCI*.


4. Recreate users and groups.

- a. From both consoles, click **WebLogic Domain**, select **Security**, select **Security Realms**, and then click **myrealm** (default WebLogic domain).

- b. From both consoles, click the realm, and then click **Users and Groups**.
  - c. Identify any custom users in the source instance, and then recreate these users in the target instance.
  - d. From both consoles, click **Groups**.
  - e. Identify any custom groups in the source instance, and then recreate these groups in the target instance.
5. Recreate roles and policies.
- a. From both consoles, click **WebLogic Domain**, select **Security**, and then select **Application Roles**.
  - b. Identify any roles in the source instance, and then recreate these roles in the target instance.

For more information, see these topics in *Securing Applications with Oracle Platform Security Services*:

- [Managing Policies with Fusion Middleware Control \(12.2.1.3\)](#)
- [Managing Policies with Fusion Middleware Control \(12.2.1.2\)](#)
- [Managing Policies with Fusion Middleware Control \(12.1.3\)](#)
- [Managing Policies with Fusion Middleware Control \(11.1.1.7\)](#)

- c. From both consoles, click **WebLogic Domain**, select **Security**, and then select **Application Policies**.
- d. Identify any policies in the source instance, and then recreate these policies in the target instance.
- e. From both consoles, click **WebLogic Domain**, select **Security**, and then select **System Policies**.
- f. Identify any system policies in the source instance, and then recreate these system policies in the target instance.
- g. Click **Search System Security Grants** .
- h. Identify any custom permissions that you created for this system library in the source instance, and then recreate these permissions in the target instance.

Repeat this process if you created custom permissions for other system libraries.

6. Recreate keystores.
- a. From both consoles, click **WebLogic Domain**, select **Security**, and then select **Keystore**.
  - b. Identify any custom keystores in the source instance, and then recreate these keystores in the target instance.

If any of the following aliases are present in the system keystores, do not modify them:

Keystore	Aliases
system/trust	democa, idcs_root_ca
system/demoidentity	DemoIdentity
system/castore	democa



Keystore	Aliases
system/publiccacerts	<name> [jdk], idcs_root_ca
opss/trustservice_ts	trustservice, cloudca
opss/trustservice_ks	trustservice
owsm/keystore	oauth_<identity_domain>_trust_si gn, cloudca, orakey

For more information, see these topics in *Securing Applications with Oracle Platform Security Services*:

- [Common Keystore Operations \(12.2.1.3\)](#)
- [Common Keystore Operations \(12.2.1.2\)](#)
- [Common Keystore Operations \(12.1.3\)](#)
- [Common Keystore Operations \(11.1.1.7\)](#)

7. Recreate credential maps.

- a. From both consoles, click **WebLogic Domain**, select **Security**, and then select **Credentials**.
- b. Identify any custom credential maps in the source instance, and then recreate these credential maps in the target instance.

Do not modify the default credential maps, including `oracle.wsm.security`.

For more information, see these topics in *Securing Applications with Oracle Platform Security Services*:

- [Managing Credentials \(12.2.1.3\)](#)
- [Managing Credentials \(12.2.1.2\)](#)
- [Managing the Credential Store \(12.1.3\)](#)
- [Managing the Credential Store \(11.1.1.7\)](#)

8. Reconfigure security providers.

- a. From both consoles, click **WebLogic Domain**, select **Security**, and then select **Security Provider Configuration**.
- b. Compare the security provider configuration of the source and target instances, and then update the configuration of the target instance as necessary.

Do not modify the Security Store.

9. Reconfigure the audit service.

- a. From both consoles, click **WebLogic Domain**, select **Security**, and then select **Audit Registration and Policy**.
- b. Compare the audit policy settings of the source and target instances, and then update the settings of the target instance as necessary.

For more information, see these topics in *Securing Applications with Oracle Platform Security Services*:

- [Managing Audit Policies with Fusion Middleware Control \(12.2.1.3\)](#)
- [Managing Audit Policies with Fusion Middleware Control \(12.2.1.2\)](#)
- [Managing Audit Policies \(12.1.3\)](#)

- [Managing Audit Policies \(11.1.1.7\)](#)

10. Recreate Web Services Manager (WSM) policies.

- From both consoles, click **WebLogic Domain**, select **Web Services**, and then select **WSM Policies**.
- Identify any custom policies in the source instance, and then recreate these policies in the target instance.

The default policies are read-only and identified with a lock icon.

For more information, see these topics in *Securing Web Services and Managing Policies with Oracle Web Services Manager*:

- [Managing Web Service Policies with Fusion Middleware Control \(12.2.1.3\)](#)
- [Managing Web Service Policies with Fusion Middleware Control \(12.2.1.2\)](#)
- [Managing Web Service Policies with Fusion Middleware Control \(12.1.3\)](#)
- [Managing Web Services Policies \(11.1.1.7\)](#)

- From both consoles, click **WebLogic Domain**, select **Web Services**, and then select **WSM Policy Sets**.
- Identify any policy sets in the source instance, and then recreate these policy sets in the target instance.

If you made significant changes to the target instance using the Fusion Middleware Control Console, Oracle recommends taking another backup of your target instance and its infrastructure database schemas.

## Troubleshoot Migration Problems

If you encounter problems migrating your Oracle Java Cloud Service instance to Oracle Cloud Infrastructure, inspect the log files for the migration tools and servers. After correcting the problems, you can restore the target instance to its initial state, and then try the migration again.

- Use a Secure Shell (SSH) client to connect to the Administration Server node on the *target* instance as the `opc` user.

```
ssh -i <privatekey> opc@<target_admin_IP>
```



If you are using a private subnet, use the following command to connect to the Administration Server node in your *target* instance:

```
ssh -i <path_to_privatekey> -o ProxyCommand="ssh -W %h:%p -i <path_to_privatekey> opc@<Public_IP>" opc@<target_admin_IP>
```

- Switch to the `oracle` user.

```
sudo su - oracle
```

- Check for warnings or errors in the Oracle WebLogic Deploy Tool (WDT) log files, which are located in the `/u01/weblogic-deploy/logs` directory.

4. Check for warnings or errors in the Oracle WebLogic Server domain log file, which is located at `/u01/data/domains/<target_domain>/servers/target_server/logs/<target_domain>.log`.
5. Fix any problems that you identify.  
For example, edit the `<source_domain>.yaml` model file.
6. Access the Oracle Java Cloud Service console.
7. Click the name of the service instance that you want to restore.
8. On the Overview page, click the **Administration** tile.
9. Click the **Backup** tab.
10. Under **Available Backups**, beside the backup that you want to restore, click **Menu** , and then select **Restore**.
11. For **Notes**, enter any free-form text to provide additional information about the restoration. For example, describe why you are restoring the service instance.
12. Click **Restore**.
13. When prompted for confirmation, perform one of the following steps:
  - If the selected backup has an associated database backup, select the check box to confirm that you have already restored the database, and then click **Continue with Restore**.
  - Click **Yes, Restore Service**.
14. To check the status of the restore operation, periodically click **Refresh** .
15. Perform these tasks again.
  - [Stop All Oracle WebLogic Server Processes on the Target](#)
  - [Install the Oracle WebLogic Deploy Tool](#) (target instance only)
  - [Update the Oracle WebLogic Server Domain on the Target Instance](#)
  - [Start All Oracle WebLogic Server Processes on the Target](#)

# 4

## Complete the Post-Migration Tasks

After successfully migrating your Oracle Java Cloud Service instances to Oracle WebLogic Server for OCI, test your applications thoroughly, and then perform cleanup and other optional configuration tasks.

### Topics:

- [Test the Target](#)
- [Start the SMTP Service on the Target](#)
- [Migrate FastConnect and VPN Connections to Oracle Cloud Infrastructure](#)
- [Clean Up Resources in Oracle Cloud Infrastructure](#)

### Test the Target

Verify that your Java applications and other Oracle WebLogic Server resources are accessible and function correctly on the target Oracle WebLogic Server domain.

Ensure to thoroughly run all application test cases.

If you have enabled Oracle WebLogic Server Administration Console communication, verify that you can access the WebLogic Server Administration Console. If your instance includes custom data sources that access your application databases, you can test database connectivity directly from the WebLogic Server Administration Console. Select a data source, click the **Monitoring** tab, and then click the **Testing** tab.

### Start the SMTP Service on the Target

If your applications use JavaMail and require access to the local Simple Mail Transfer Protocol (SMTP) server on the operating system, then you must start the SMTP server.



#### Note:

For the Oracle Java Cloud Service nodes in Oracle Cloud Infrastructure Classic, the SMTP server is not configured to run by default for nodes in Oracle Cloud Infrastructure.

Unlike Oracle Java Cloud Service nodes in Oracle Cloud Infrastructure Classic, the SMTP server is not configured to run by default for nodes in Oracle Cloud Infrastructure.

Alternatively, you can configure your JavaMail sessions to use Oracle Cloud Infrastructure Email Delivery. See [Overview of the Email Delivery Service](#) in the Oracle Cloud Infrastructure documentation.

1. Use a Secure Shell (SSH) client to connect to the Administration Server node on the target instance as the `opc` user.

2. Configure and start the SMTP server on the node.
3. Connect to all Managed Server nodes in the target instance that require access to the local SMTP server, and then repeat the previous step.

## Migrate FastConnect and VPN Connections to Oracle Cloud Infrastructure

Use Oracle Cloud Infrastructure to create a connection between your private, on-premises network and a network in Oracle Cloud.

### Note:

This topic is not applicable, if you are migrating Oracle Java Cloud Service on Oracle Cloud Infrastructure to Oracle WebLogic Server for OCI. As the instance can be created in the existing VCN, where FastConnect or IPSec is already configured.

A Virtual Private Network (VPN) uses a public network to create a secure connection between two private networks. Oracle supports two connectivity solutions for a Virtual Cloud Network (VCN) in Oracle Cloud Infrastructure:

- Oracle Cloud Infrastructure FastConnect - Create dedicated, high-speed, virtual circuits for production systems that communicate with your on-premises network using the Border Gateway Protocol (BGP). This service is equivalent to Oracle Cloud Infrastructure FastConnect Classic.
- IPSec VPN - Create secure connections with your on-premises network using the IPSec protocol. This solution replaces VPN as a Service (VPNaaS) and CoreNet in Oracle Cloud Infrastructure Classic.

When migrating from Oracle Cloud Infrastructure Classic, update the existing BGP or VPN configuration in your on-premises network to use either Oracle Cloud Infrastructure FastConnect or IPSec VPN. Alternatively, if you require connectivity to instances in both Oracle Cloud Infrastructure and Oracle Cloud Infrastructure Classic during the migration process, create a separate BGP or VPN configuration in your on-premises network.

In Oracle Cloud Infrastructure, creating a connection to your on-premises network includes these tasks:

- Create a Dynamic Routing Gateway (DRG) in the VCN.
- Create a route table in the VCN that directs external traffic to the DRG.
- Assign the route table to a subnet in the VCN.






Refer to these topics in the Oracle Cloud Infrastructure documentation:

- FastConnect
- IPSec VPN

## Clean Up Resources in Oracle Cloud Infrastructure

After testing your target Oracle WebLogic Server domain in Oracle Cloud Infrastructure, you can delete the source Oracle Java Cloud Service instance and supporting cloud resources in Oracle WebLogic Server for OCI.

Delete these Oracle Cloud Infrastructure resources to avoid costs for services that you no longer use.

1. Access the Oracle Java Cloud Service console.
2. Delete the source Oracle Java Cloud Service instances that you created in Oracle Cloud Infrastructure.
  - a. Click **Manage this instance**  for the service instance, and then select **Delete**.
  - b. Enter the **Database Administrator User Name** and **Database Administrator User Password** for the infrastructure schema database.  
Alternatively, select **Force Delete** if you plan to delete this database as well.
  - c. Click **Delete**.
3. Click **IP Reservations**.
4. Delete any IP reservations that you created for your source Oracle Java Cloud Service instances.
  - a. Click **Delete**  for the IP reservation.
  - b. When prompted for confirmation, click **OK**.
5. Access the Oracle Database Classic Cloud Service console (Database Classic).
6. Delete the Oracle Database Classic Cloud Service instances that you created in Oracle WebLogic Server for OCI to support your source Oracle Java Cloud Service instances.  
Do not delete a database if it is still in use by other services.
  - a. Click **Manage this instance**  for the database instance, and then select **Delete**.
  - b. When prompted for confirmation, click **Delete**.
7. Click **IP Reservations**.
8. Delete any IP reservations that you created for your Oracle Database Classic Cloud Service instances.
  - a. Click **Delete**  for the IP reservation.
  - b. When prompted for confirmation, click **OK**.
9. Access the Oracle Cloud Infrastructure Object Storage Classic console (Storage Classic).
10. Delete the object storage containers that you created in Oracle Cloud Infrastructure to support your source Oracle Java Cloud Service instances.  
Do not delete a container if it is still in use by other services.
  - a. Click the delete icon  for the container.
  - b. When prompted for confirmation, click **OK**.