# Oracle® Cloud

# Migrating Oracle Java Cloud Service Instances to Oracle Cloud Infrastructure Using Migration Tools

Release 20.2.3
F20594-09
July 2020

Oracle Cloud Migrating Oracle Java Cloud Service Instances to Oracle Cloud Infrastructure Using Migration Tools, Release 20.2.3

F20594-09



Primary Authors: TJ Palazzolo, Poh Lee Tan

# Contents

## 7 Complete the Post-Migration Tasks

1

# Learn About Migrating to Oracle Cloud Infrastructure

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

**Topics:**



## Why Migrate to Oracle Cloud Infrastructure

Oracle encourages you to migrate your existing cloud resources to Oracle Cloud Infrastructure regions. You can gain several advantages by doing so.

In Oracle Cloud, you provision resources in specific regions, which are localized to geographic locations. Certain regions support the Oracle Cloud Infrastructure platform.

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.

You can benefit from these additional administrative features when you migrate your cloud resources to Oracle Cloud Infrastructure:

- Organize cloud resources into a hierarchy of logical compartments.
- Create fine-grained access policies for each compartment.

To learn more, see Upgrade Your Classic Services to Oracle Cloud Infrastructure.

## About the Migration Scope

Before you migrate your existing Oracle Java Cloud Service instances to Oracle Cloud Infrastructure, 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 these 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 Cloud Infrastructure security, network and storage resources that might be required to support your new WebLogic Server domain. Instead, this guide provides references to the Oracle Cloud Infrastructure documentation as appropriate.

Most service instances connect to one or more databases in order to access your application schemas. This guide does not include the detailed procedure for migrating these application databases from Oracle Cloud Infrastructure Classic to Oracle Cloud Infrastructure. One option is the Oracle Cloud Infrastructure Classic Database Backup Migration Tool, which uses Recovery Manager (RMAN). Another option is Oracle Data Guard. See Select a Method to Migrate Database Instances in *Migrating Infrastructure Classic Workloads to Oracle Cloud Infrastructure*.

# About Oracle WebLogic Server for Oracle Cloud Infrastructure

Oracle WebLogic Server for Oracle Cloud Infrastructure 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.

> **Note:**
>
> Oracle recommends migrating your existing domains in Oracle Java Cloud Service to Oracle WebLogic Server for Oracle Cloud Infrastructure.

After launching a domain using the Marketplace applications, you track and monitor its progress as a stack using Resource Manager in Oracle Cloud Infrastructure. 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 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.

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

| Oracle Java Cloud Service | Oracle WebLogic Server for Oracle Cloud Infrastructure |
|---|---|
| Supports Oracle WebLogic Server 11g, 12.2.1.3, and 12.2.1.4<br>Also supports Oracle WebLogic Server 12.1.3 | Supports Oracle WebLogic Server 11g, 12.2.1.3, and 12.2.1.4 |
| 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<br>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<br>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<br>No restrictions on changing 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<br>A security application is created in Oracle Identity Cloud Service for each domain | Can use Oracle Identity Cloud Service for authentication<br>Must create a confidential application in Oracle Identity Cloud Service prior to creating a domain<br>Confidential application, enterprise application, and App Gateway are created in Oracle Identity Cloud Service for each domain |

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

# Compare Oracle Cloud Infrastructure to Classic

This topic helps you get familiar with basic Oracle Cloud Infrastructure security, network, and storage concepts, and compare them to their equivalent concepts in Oracle Cloud Infrastructure Classic.

Cloud resources in Oracle Cloud Infrastructure are created in logical compartments. You also create fine-grained policies to control access to the resources within a compartment.

You create instances within an Oracle Cloud Infrastructure region. You also specify an availability domain (AD), if supported in the selected region. Oracle Cloud Infrastructure Classic does not use availability domains.

A virtual cloud network (VCN) is comprised of one or more subnets, and an instance is assigned to a specific subnet. In Oracle Cloud Infrastructure Classic, you assign instances to IP networks or the shared network. Typically, you create one subnet for the shared network, and create a separate subnet for each IP network in Oracle Cloud

Infrastructure Classic. Note that unlike Oracle Cloud Infrastructure Classic, Oracle Cloud Infrastructure does not allow you to reserve IP addresses for platform services.

A subnet's security lists permit and block traffic to and from specific IP addresses and ports. In Oracle Cloud Infrastructure Classic, an instance's access rules provide similar capabilities, although security lists are configured at the subnet level.

Instances can communicate with resources outside of Oracle Cloud by using Oracle Cloud Infrastructure FastConnect, which provides a fast, dedicated connection to your on-premises network. This service is equivalent to Oracle Cloud Infrastructure FastConnect Classic. Alternatively, use IPSec VPN in Oracle Cloud Infrastructure as a replacement for VPN as a Service (VPNaaS) or Corente in Oracle Cloud Infrastructure Classic.

A bucket in Oracle Cloud Infrastructure Object Storage can be used to store files and share them with multiple instances. A user's generated authentication token (auth token) is required to access the bucket. Oracle Cloud Infrastructure Object Storage Classic provides the same service in Oracle Cloud Infrastructure Classic, but does not use auth tokens.

To learn more, see Key Concepts and Terminology in the Oracle Cloud Infrastructure documentation.

You can create rules that automatically scale an Oracle Java Cloud Service instance that's running in Oracle Cloud Infrastructure Classic. You must scale instances in Oracle Cloud Infrastructure manually.

# About Oracle Cloud Infrastructure Users and Groups

Use the Identity and Access Management (IAM) system in Oracle Cloud Infrastructure to manage users, groups, and policies.

For example, the following Oracle Cloud Infrastructure policy grants members of the group `MyGroup` all privileges to all resources in the compartment `MyCompartment`:

```
Allow group MyGroup to manage all-resources in compartment MyCompartment
```

By default, this system is also configured to use Oracle Identity Cloud Service as a federated identity provider. Therefore, when you define policies in Oracle Cloud Infrastructure, you can reuse existing users and groups in Oracle Identity Cloud Service. You can either add users to a new group in Oracle Cloud Infrastructure, or map an existing Oracle Identity Cloud Service group to an Oracle Cloud Infrastructure group.

While policies control access to resources and services in Oracle Cloud Infrastructure, administrator roles control access to platform services. Assign Oracle Identity Cloud Service users and groups to administrator roles in order to grant them access to platform services.

- Common Policies in the Oracle Cloud Infrastructure documentation
- Federating with Oracle Identity Cloud Service in the Oracle Cloud Infrastructure documentation
- Add Users, Assign Policies and Roles in *Getting Started with Oracle Cloud*

# About the Migration Task Flow

Get an overview of the process that you use to migrate your existing Oracle Java Cloud Service instances to Oracle Cloud Infrastructure.

The process varies depending on the target service and the selected tools.

- Migrate to Oracle WebLogic Server for Oracle Cloud Infrastructure Using Application Migration Service
- Migrate to Oracle WebLogic Server for Oracle Cloud Infrastructure Using Classic Tools
- Migrate to Java Cloud Service Using Application Migration Service
- Migrate to Java Cloud Service Using Classic Tools

## Migrate to Oracle WebLogic Server for Oracle Cloud Infrastructure Using Application Migration Service

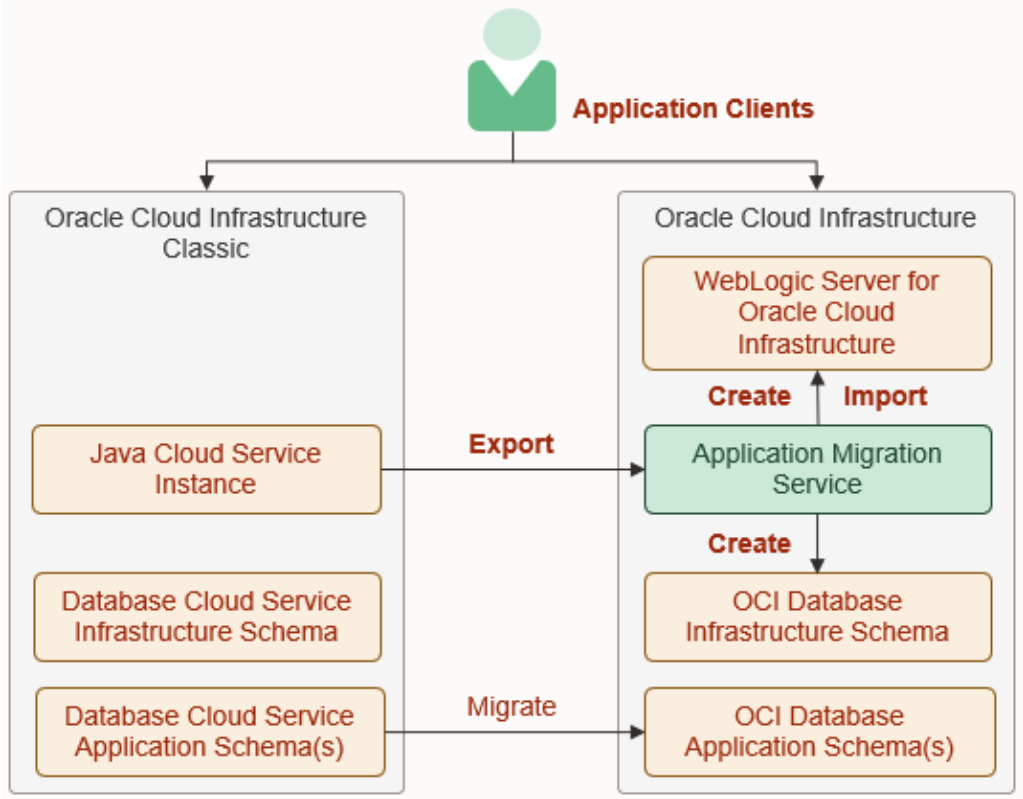
Use Application Migration in Oracle Cloud Infrastructure to migrate service instances to Oracle WebLogic Server for Oracle Cloud Infrastructure.

Application Migration does not support the migration of WebLogic Server domains that include these types of resources:

- Custom Identity or Trust Keystore
- Foreign JNDI Provider
- Foreign JMS Server
- JMS Bridge Destination
- Storage-and-Forward (SAF) Context
- JavaMail Session
- WebLogic Diagnostic Framework (WLDF) REST Notification Endpoint

If your source Oracle Java Cloud Service instance uses these resource types, then Oracle recommends using the Oracle Cloud Infrastructure Classic Java Migration Tool instead of Application Migration. See Migrate to Oracle WebLogic Server for Oracle Cloud Infrastructure Using Classic Tools.

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




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

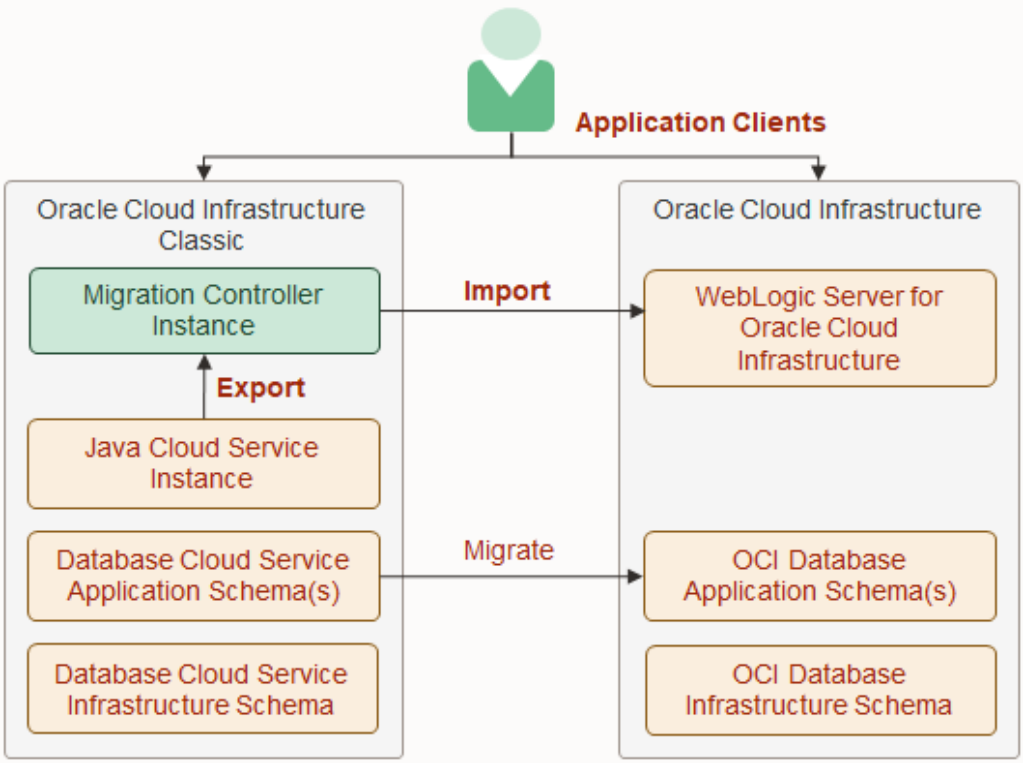
1. Prepare for the migration and perform any prerequisite tasks in Oracle Cloud Infrastructure.
2. Migrate any application databases in Oracle Cloud Infrastructure Classic regions to Oracle Cloud Infrastructure Database.
3. Use Application Migration to identify the source Oracle Java Cloud Service instance.
4. Use Application Migration to create and start a migration for the source Oracle Java Cloud Service instance.
   Application Migration exports the domain configuration and applications from your source instance, creates a database, creates a new domain with Oracle WebLogic Server for Oracle Cloud Infrastructure, and imports the domain configuration and applications to the target domain.
5. Test your applications on the target domain, and perform any other post-migration tasks.

See Migrate an Instance to Oracle WebLogic Server for Oracle Cloud Infrastructure Using Application Migration Service.

## Migrate to Oracle WebLogic Server for Oracle Cloud Infrastructure Using Classic Tools

Use the Oracle Cloud Infrastructure Classic Java Migration Tool to migrate service instances to Oracle WebLogic Server for Oracle Cloud Infrastructure.

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 Oracle Cloud Infrastructure.




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

1. Prepare for the migration and perform any prerequisite tasks in Oracle Cloud Infrastructure.
2. Create an Oracle Cloud Infrastructure Database for the required infrastructure schemas, or use an existing Oracle Cloud Infrastructure Database.
3. Migrate any application databases in Oracle Cloud Infrastructure Classic regions to Oracle Cloud Infrastructure Database.
4. Create a migration controller instance, Control-S, in your Oracle Cloud Infrastructure Classic account. The Oracle Cloud Infrastructure Classic Java Migration Tool is installed on this compute instance.
5. Use the Oracle Cloud Infrastructure Classic Java Migration Tool to export the domain configuration, applications and other supporting files from your source Oracle Java Cloud Service instance.
6. Create the target domain using Oracle WebLogic Server for Oracle Cloud Infrastructure.
7. Use the Oracle Cloud Infrastructure Classic Java Migration Tool to import the domain configuration and applications to your target in Oracle Cloud Infrastructure.
8. Test your applications on the target instance, and perform any other post-migration tasks.

See Migrate an Instance to Oracle WebLogic Server for Oracle Cloud Infrastructure Using Classic Tools.

# Migrate to Java Cloud Service Using Application Migration Service

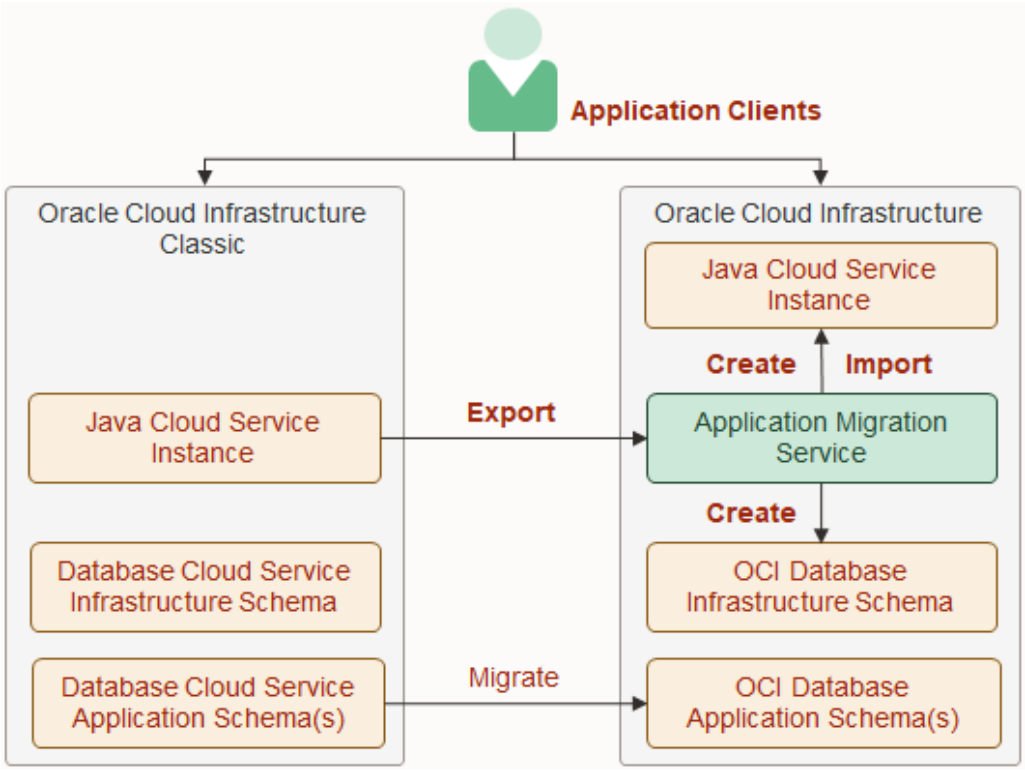
Use Application Migration in Oracle Cloud Infrastructure to migrate service instances to Oracle Java Cloud Service on Oracle Cloud Infrastructure.

Application Migration does not support the migration of WebLogic Server domains that include these types of resources:

- Custom Identity or Trust Keystore
- Foreign JNDI Provider
- Foreign JMS Server
- JMS Bridge Destination
- Storage-and-Forward (SAF) Context
- JavaMail Session
- WebLogic Diagnostic Framework (WLDF) REST Notification Endpoint

If your source Oracle Java Cloud Service instance uses these resource types, then Oracle recommends using the Oracle Cloud Infrastructure Classic Java Migration Tool instead of Application Migration. See Migrate to Java Cloud Service Using Classic Tools.

The following diagram shows the migration topology for a Oracle Java Cloud Service instance using Application Migration. The migration target is a new Oracle Java Cloud Service instance.




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

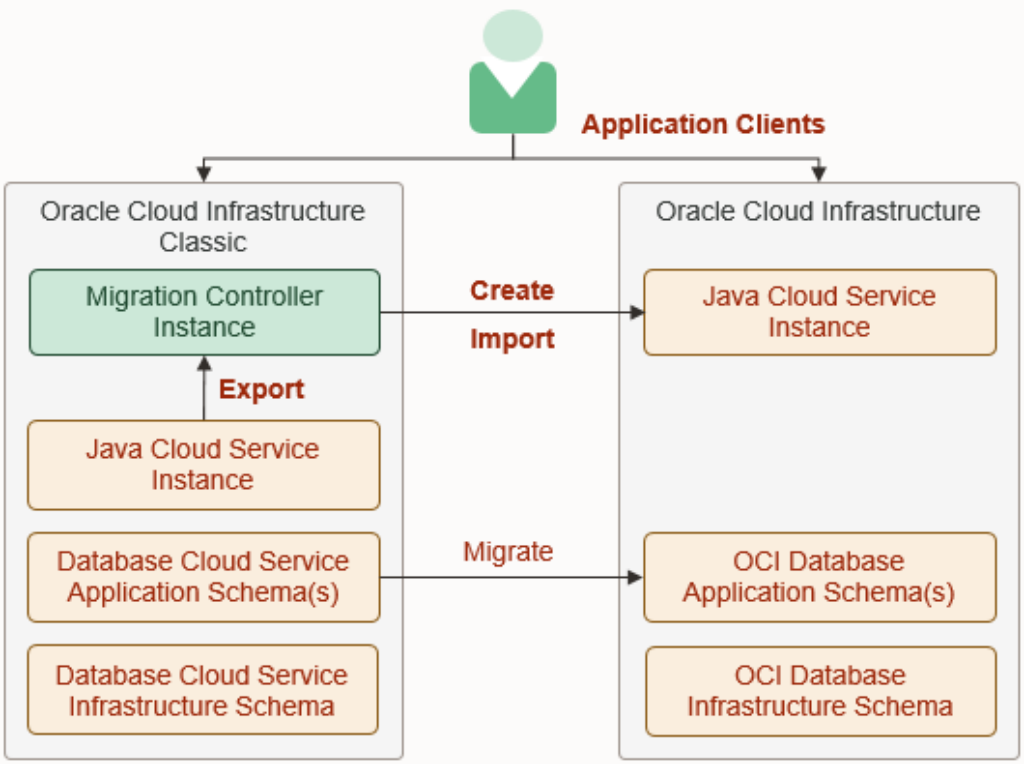
1. Prepare for the migration and perform any prerequisite tasks in Oracle Cloud Infrastructure.
2. Migrate any application databases in Oracle Cloud Infrastructure Classic regions to Oracle Cloud Infrastructure Database.
3. Use Application Migration to identify the source Oracle Java Cloud Service instance.
4. Use Application Migration to create and start a migration for the source Oracle Java Cloud Service instance.
   Application Migration exports the domain configuration and applications from your source instance, creates a database, creates the target Oracle Java Cloud Service instance, and imports the domain configuration and applications to the target instance.
5. Test your applications on the target instance, and perform any other post-migration tasks.

See Migrate an Instance to Oracle Java Cloud Service Using Application Migration Service.

## Migrate to Java Cloud Service Using Classic Tools

Use the Oracle Cloud Infrastructure Classic Java Migration Tool to migrate service instances to Oracle Java Cloud Service on Oracle Cloud Infrastructure.

The following diagram shows the migration topology for a Oracle Java Cloud Service instance. The migration target is a new Oracle Java Cloud Service instance.




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

1. Prepare for the migration and perform any prerequisite tasks in Oracle Cloud Infrastructure.

2. Create an Oracle Cloud Infrastructure Database for the required infrastructure schemas, or use an existing Oracle Cloud Infrastructure Database.
3. Migrate any application databases in Oracle Cloud Infrastructure Classic regions to Oracle Cloud Infrastructure Database.
4. Create a migration controller instance, Control-S, in your Oracle Cloud Infrastructure Classic account. The Oracle Cloud Infrastructure Classic Java Migration Tool is installed on this compute instance.
5. Use the Oracle Cloud Infrastructure Classic Java Migration Tool to export the domain configuration, applications and other supporting files from your source Oracle Java Cloud Service instance.
6. Use the Oracle Cloud Infrastructure Classic Java Migration Tool and Terraform to create the target Oracle Java Cloud Service instance.
7. Use the Oracle Cloud Infrastructure Classic Java Migration Tool to import the domain configuration and applications to your target Oracle Java Cloud Service instance.
8. Test your applications on the target instance, and perform any other post-migration tasks.

See Migrate an Instance to Oracle Java Cloud Service Using Classic Tools.

# About the Migration Tooling

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

Oracle WebLogic Server Deploy Tooling 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.

Oracle WebLogic Server Deploy Tooling 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.

Application Migration Service provides a simple, graphical interface for migrating Oracle Java Cloud Service instances. It is available from the Oracle Cloud Infrastructure console. It automates the discovery of Oracle Cloud Infrastructure Classic resources in your account and the recreation of these resources in Oracle Cloud Infrastructure. Application Migration Service creates the target Oracle Java Cloud Service instance or Oracle WebLogic Server for Oracle Cloud Infrastructure domain, and creates the required database. It also automates the installation and execution of the Oracle WebLogic Server Deploy Tooling on the source and target domains.

The Oracle Cloud Infrastructure Classic Java Migration Tool is included in a custom compute image named Oracle Cloud Infrastructure Classic Migration Tools. It is a collection of command line tools that have similar capabilities to Application Migration Service.

> **Note:**
>
> Unlike Application Migration Service, the Oracle Cloud Infrastructure Classic Java Migration Tool cannot create your target domain with Oracle WebLogic Server for Oracle Cloud Infrastructure. You must manually create the target domain using Oracle Cloud Infrastructure Marketplace.

See:

- Oracle WebLogic Server Deploy Tooling project on GitHub
- Overview of Application Migration in the Oracle Cloud Infrastructure documentation
- Review the List of Available Migration Tools in *Migrating Infrastructure Classic Workloads to Oracle Cloud Infrastructure*

2

# Prepare to Migrate Oracle Java Cloud Service to Oracle Cloud Infrastructure

Before you migrate your service instances from Oracle Cloud Infrastructure Classic to Oracle Cloud Infrastructure, 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

The migration process does not affect the availability of your existing Oracle Java Cloud Service instance in Oracle Cloud Infrastructure Classic. 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 migration 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 Cloud Infrastructure.

## Select Oracle Cloud Infrastructure Shapes

Identify the compute shapes that provide similar IaaS resources in Oracle Cloud Infrastructure to the shapes that you're currently using for your service instances in Oracle Cloud Infrastructure Classic.

A compute shape defines the IaaS resources, such as OCPUs and memory, that are available to a specific node in a service instance. Oracle Cloud Infrastructure and Oracle Cloud Infrastructure Classic each has its own set of standard compute shapes. See:

- About Shapes in *Using Oracle Cloud Infrastructure Compute Classic*
- Compute Shapes in the Oracle Cloud Infrastructure documentation

Application Migration Service automatically selects a shape in Oracle Cloud Infrastructure for your new instance that most closely matches the shape in the source instance.

The Oracle Cloud Infrastructure Classic Java Migration Tool can generate a Terraform configuration to help you provision your new instance, and it automatically selects an Oracle Cloud Infrastructure shape for this configuration.

To ensure that a migrated service instance has the same performance characteristics as the original instance, and can support an equivalent workload, choose Oracle Cloud Infrastructure shapes that most closely map to the Oracle Cloud Infrastructure Classic 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 Cloud Infrastructure 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.

# Design the Oracle Cloud Infrastructure Network

Before you migrate your service instances from Oracle Cloud Infrastructure Classic to Oracle Cloud Infrastructure, you must design and implement a virtual cloud network (VCN) to support your migrated service instances.

You can create new Oracle Cloud Infrastructure compartments, VCNs, and subnets for your service instances, or you can use existing ones. See these topics in the Oracle Cloud Infrastructure documentation:

- Managing Compartments
- VCNs and Subnets
- Security Lists

Consider the following guidelines when you create or select a network for your service instances:

- If instances communicate using the default shared network in Oracle Cloud Infrastructure Classic, then use a single subnet for these instances.
- If instances are on separate IP networks in Oracle Cloud Infrastructure Classic, then use separate subnets for these instances.
- A VCN should have an address range that includes all of the IP networks in Oracle Cloud Infrastructure Classic that need to communicate. Alternatively, configure peering between multiple VCNs.
- A subnet should have at least the same number of addresses as the corresponding IP network in Oracle Cloud Infrastructure Classic.
- If an instance was created in Oracle Cloud Infrastructure Classic without public IP addresses, then use a private subnet for this instance.
- If custom access rules were created for an instance in Oracle Cloud Infrastructure Classic to control communication to or from the instance, then create a security list in Oracle Cloud Infrastructure and assign the security list to the appropriate subnets. To use custom security lists, you must assign the instance to a custom subnet, and not the default subnet.

Oracle Cloud Infrastructure Classic Java Migration Tool creates a bastion compute instance in Oracle Cloud Infrastructure in order for the migration controller (Control-S) compute instance to access your target service instance. After updating the target service instance, the tool deletes the temporary bastion compute instance.

Before you create service instances in Oracle Cloud Infrastructure that use your new network resources, you must create policies that grant your service access to these resources. See Prerequisites for Oracle Platform Services 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 Java Cloud Service 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.

# Migrate the Application Databases

If the applications in your Oracle Java Cloud Service instance use database instances that were created in an Oracle Cloud Infrastructure Classic region, migrate these application databases to Oracle Cloud Infrastructure Database.

There are multiple methods of migrating your database. The Oracle Cloud Infrastructure Classic Database Backup Migration Tool uses Recovery Manager (RMAN). Another option is Oracle Data Guard. See Select a Method to Migrate Database Instances in *Migrating Infrastructure Classic Workloads to Oracle Cloud Infrastructure*.

1. Create the Oracle Cloud Infrastructure Database instances in the same region and virtual cloud network (VCN) that you plan to create your target WebLogic Server domain.
2. If the databases and target WebLogic Server domain will be on different subnets, then configure security rules that allow the target's subnet to communicate with the database ports.

# 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.

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, click **Nodes**, and then record these values.
   - The public IP address of the first database node
   - The host name prefix for the database (for example, `myappdb`)
   - The Scan DNS name
   - The domain name for the database (for example, `mydbsubnet.myvcn.oraclevcn.com`)
   - The database port number
6. Click **Databases**. Record the database name and unique name.

   For example, `ORCL` and `ORCL_iad1zj`.
7. Click the name of the database.
8. Click **DB Connection**, and then record the Easy Connect string.
9. 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
```

10. Repeat Steps 1 to 9 for any other databases to which the target instance will connect.

3

# Migrate an Instance to Oracle WebLogic Server for Oracle Cloud Infrastructure Using Application Migration Service

Use Application Migration 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 Classic to a new domain in Oracle WebLogic Server for Oracle Cloud Infrastructure.

> **Note:**
>
> Oracle recommends migrating your existing domains in Oracle Java Cloud Service to Oracle WebLogic Server for Oracle Cloud Infrastructure.

Application Migration is available only in specific Oracle Cloud Infrastructure regions. See Overview of Application Migration in the Oracle Cloud Infrastructure documentation.

Application Migration does not support the migration of WebLogic Server domains that include these types of resources:

- Custom Identity or Trust Keystore
- Foreign JNDI Provider
- Foreign JMS Server
- JMS Bridge Destination
- Storage-and-Forward (SAF) Context
- JavaMail Session
- WebLogic Diagnostic Framework (WLDF) REST Notification Endpoint

If your source Oracle Java Cloud Service instance uses these resource types, then Oracle recommends using the Oracle Cloud Infrastructure Classic Java Migration Tool instead of Application Migration. See Migrate an Instance to Oracle WebLogic Server for Oracle Cloud Infrastructure Using Classic Tools.

Before you begin the migration process, see Prepare to Migrate Oracle Java Cloud Service to Oracle Cloud Infrastructure.

When you migrate an Oracle Java Cloud Service instance, the following terms are used:

- *Source*: The connection to your Oracle Cloud Infrastructure Classic account in Application Migration.
- *Source Instance*: The Oracle Java Cloud Service instance in Oracle Cloud Infrastructure Classic.

- *Target*: The domain and related cloud resources in Oracle WebLogic Server for Oracle Cloud Infrastructure.

**Topics:**



# Perform Prerequisite Tasks for Oracle WebLogic Server for Oracle Cloud Infrastructure

Before you use Application Migration Service to create a domain using Oracle WebLogic Server for Oracle Cloud Infrastructure, you must create the required infrastructure 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.
   - A vault and encryption key
2. 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.

   Identify the client ID and secret of the confidential application.
3. Use Oracle Cloud Infrastructure Vault to create secrets for the passwords that you need for the target domain.
   - WebLogic Server administrator password
   - Database administrator password
   - Client secret, if using Oracle Identity Cloud Service

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

Application Migration Service automatically creates an Oracle Cloud Infrastructure Database before it creates the target domain. Oracle WebLogic Server for Oracle Cloud Infrastructure provisions the required infrastructure schema to this database.

# Create a Source

Use Application Migration to connect to your Oracle Cloud Infrastructure Classic account and region.

1. From the Oracle Cloud Infrastructure console, navigate to **Application Migration**.
2. Select the **Compartment** in which to create the source.
3. Click **Sources**.
4. Click **Create Source**.
5. Enter a **Name** and **Description** for the source.
6. For **Source Type**, select **Oracle Cloud Infrastructure - Classic**.
7. For **Account**, enter the name of your Oracle Cloud Infrastructure Classic account.
8. Select the Oracle Cloud Infrastructure Classic **Region** in which you created your source Oracle Java Cloud Service instance.
9. Enter credentials for this Oracle Cloud Infrastructure Classic account that have access to Oracle Java Cloud Service.
10. Click **Create**.

For more information, see Manage Sources in the Oracle Cloud Infrastructure documentation.

# Create a Migration

Use Application Migration to connect to the WebLogic Server domain for the Oracle Java Cloud Service instance within your source.

1. From the Oracle Cloud Infrastructure console, navigate to **Application Migration**.
2. Select the **Compartment** that contains your source.
3. Click **Sources**, and then select your source.
4. Click **Actions** ⋮ for the Oracle Java Cloud Service instance that you want to migrate, and then click **Create Migration**.
5. Enter a **Name** and **Description** for the migration.
6. Enter the WebLogic Server administrator credentials for the Oracle Java Cloud Service instance.
7. Set the **Target Instance Type** to Oracle WebLogic Server for Oracle Cloud Infrastructure.
8. Click **Create**.

For more information, see Manage Migrations in the Oracle Cloud Infrastructure documentation.

# Configure and Run a Migration

Use Application Migration to create the target domain in Oracle WebLogic Server for Oracle Cloud Infrastructure. Specify a network, credentials, databases, and other details.

1. From the Oracle Cloud Infrastructure console, navigate to **Application Migration**.
2. Select the **Compartment** that contains your migration.
3. Click **Migrations**, and then select your migration.
4. Click **Configure**.
5. In the Configure Service section, click **Configure**.
6. Select the **Availability Domain** in which you want to create the target instance.
7. Select the **Virtual Cloud Network** and **Subnet** in which you want to create the target instance.
8. For **Secrets OCID for Database Administrator Password**, paste the OCID of the secret that contains the password for the new Oracle Cloud Infrastructure Database.
9. Enter the same password in **System Database Administrator Password**.
10. Upload or paste the public **SSH Key** to use for the target instance and database.
11. Enter the WebLogic Server administrator credentials for the new domain.
    a. Enter the **WebLogic Server Admin User Name**.
    b. For **Secrets OCID for WebLogic Server Admin Password**, paste the OCID of the secret that contains the password.
    c. Enter the same password in **WebLogic Server Admin Password**.
12. If your source instance uses Oracle Identity Cloud Service (IDCS) for authentication, then provide details about the confidential application that you created for the target domain.
    a. For **IDCS Tenant**, enter your Oracle Identity Cloud Service (IDCS) tenant name, which is also referred to as the instance ID.

       This ID is typically found in the URL that you use to access Oracle Identity Cloud Service, and has the format `idcs-<GUID>`.
    b. Enter the **Client ID** of an existing confidential application in this Oracle Identity Cloud Service instance.
    c. For **Secrets OCID for IDCS Client Secret**, paste the OCID of the secret that contains the client secret of the confidential application.
13. Click **Configure** to return to the Configure Migration page.
14. If your source instance includes custom Java Database Connectivity (JDBC) data sources, then provide the location and password of the new application databases in Oracle Cloud Infrastructure.
    a. In the Configure Application section, click **Configure**.
    b. For each data source, enter the **Connection String** 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 | `jdbc:oracle:thin:@//<db_hostname>-scan.<db_domain>:<db_port>/<pdb_name>.<db_domain>` |
| 12c | Bare Metal | `jdbc:oracle:thin:@//<db_hostname>.<db_domain>:<db_port>/<pdb_name>.<db_domain>` |
| 11g | VM | `jdbc:oracle:thin:@//<db_hostname>-scan.<db_domain>:<db_port>/<db_unique_name>.<db_domain>` |
| 11g | Bare Metal | `jdbc:oracle:thin:@//<db_hostname>.<db_domain>:<db_port>/<db_unique_name>.<db_domain>` |

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

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

c. For each data source, set the **Data Source Password**.

d. Click **Configure** to return to the Configure Migration page.

15. Click **Save and Run**.

16. When prompted for confirmation, click **Start**.

Use Application Migration to monitor the progress of your work request. The target domain is provisioned as a Terraform stack using Resource Manager. To access the new domain, see these topics in *Using Oracle WebLogic Server for Oracle Cloud Infrastructure*:

- View the Cloud Resources for a Domain
- Access the WebLogic Console

If the work request indicates that the stack creation failed, use Resource Manager to view the log files. See Stack Creation Failed in *Using Oracle WebLogic Server for Oracle Cloud Infrastructure*.

If the work request indicates that the import step of the migration failed, you can get additional information by connecting to the first node in the target domain. Access the log files found at `/u01/weblogic-deploy` and `/u01/jcsmig`.

After correcting any problems, you can run the migration again.

For more information, see Manage Migrations in the Oracle Cloud Infrastructure documentation.

# Copy Supporting Files to the Target Instance

Identify and copy any files to your target Oracle WebLogic Server for Oracle Cloud Infrastructure domain that are not automatically managed by Application Migration.

Application Migration migrates the following types of files from your source instance's domain configuration to your target domain's configuration:

- Application deployments
- Library deployments
- Custom keystores

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

1. Use SSH to connect to the Administration Server node in your *source* instance.

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

2. Switch to the `oracle` user.

   ```
   sudo su - oracle
   ```

3. Identify any supporting files that need to be copied to the target instance.
4. 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.

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

   Example:

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

6. Disconnect from the node.
7. 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 .
   ```

8. 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>:/tmp
   ```

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

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

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

    Example:

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

11. Switch to the `oracle` user.

    ```
    sudo su - oracle
    ```

12. 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
    ```

13. Disconnect from the node.

# 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 Oracle Cloud Infrastructure 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 Oracle Cloud Infrastructure*.

4. Recreate users and groups.
   a. From both consoles, click **WebLogic Domain**, select **Security**, and then select **Security Realms**.

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. For **Name**, select **Includes**, and then enter the text `common/wsm-agent-core`.

h. Click **Search System Security Grants**  .

i. 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` |

| Keystore | Aliases |
|---|---|
| system/castore | democa |
| system/publiccacerts | <*name*> [jdk], idcs_root_ca |
| opss/trustservice_ts | trustservice, cloudca |
| opss/trustservice_ks | trustservice |
| owsm/keystore | oauth_<*identity_domain*>_trust_sign, 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.
    a. From both consoles, click **WebLogic Domain**, select **Web Services**, and then select **WSM Policies**.
    b. 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)

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

# 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 in Oracle WebLogic Server for Oracle Cloud Infrastructure.

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 Oracle Identity Cloud Service console.
2. Click the navigation drawer  , and then select **Applications**.
3. Click the security application for your source instance, `JaaS_<source_instance_name>`.
4. Copy the following information for the security application:
   - Application ID
   - Client ID
   - Client secret
5. Encode the following string in base64 format.

   `<client_id>:<client_secret>`

6. Use the Oracle Identity Cloud Service REST API to request an access token for the source instance's security application.

   ```
   curl --location --request POST 'https://<idcs_host>/oauth2/v1/
   token' \
   --header 'Authorization: Basic <base64_encoded_clientid:secret>' \
   --header 'Content-Type: application/x-www-form-urlencoded' \
   --data-urlencode 'grant_type=password' \
   --data-urlencode 'scope=urn:opc:idm:__myscopes__' \
   --data-urlencode 'username=<idcs_user_name>' \
   --data-urlencode 'password=<idcs_password>'
   ```

   Copy the access token from the response.

   See Generate Access Token and Other OAuth Runtime Tokens to Access the Resource in *REST API for Oracle Identity Cloud Service*.

7. Use the Oracle Identity Cloud Service REST API to export the web tier policy for the security application.

   ```
   curl -X GET 'https://<idcs_host>/admin/v1/Apps/
   <application_ID>&attributes=displayName,urn:ietf:params:scim:schemas
   :oracle:idcs:extension:webTierPolicy:App:webTierPolicyJson' \
   -H 'Authorization:Bearer <access_token>'
   ```

   Locate the web tier policy in the response:

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

   See Get an App in *REST API for Oracle Identity Cloud Service*.

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

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

13. After the export job has finished, click **Download**. Save the file `AppRoleExport_<id>.csv`.
14. Click the navigation drawer  , and then select **Applications**.

15. Click the security application for your target domain, `<stack>_enterprise_idcs_app_<timestamp>`.

    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.

16. Click **SSO Configuration**.

17. 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.

18. Expand **Resources**.

19. Within the Resources section, click **Add**.

20. Enter a **Resource Name**.

    For example, `myapp`

21. For **Resource URL**, paste the value of the `filter` property.

22. If the filter's `type` property is `regex`, then select **Regex**.

23. Click **OK**.

24. Expand **Authentication Policy**. Under Managed Resources, click **Add**.

25. For **Resource**, select your new resource.

26. 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**

27. Click **Add**.

28. *Repeat from step 18* for each additional filter in the exported web tier policy.

29. Click the navigation drawer  , and then select **Groups**.

30. Create these groups for the target domain.
    - `<domain>_Administrators`
    - `<domain>_Deployers`
    - `<domain>_Operators`
    - `<domain>_Monitors`

    For example:
    - `MyDomain_Administrators`
    - `MyDomain_Deployers`
    - `MyDomain_Operators`
    - `MyDomain_Monitors`
31. Open `AppRoleExport_<id>.csv`, and identify the users and groups assigned to the `Administrators` role in the source instance.
32. Edit the `<domain>_Administrators` group, and add the same users and groups as the `Administrators` role in the source instance.
33. Repeat the previous step for the remaining roles in `AppRoleExport_<id>.csv`:
    - Add the members of the `Deployers` role to the `<domain>_Deployers` group.
    - Add the members of the `Operators` role to the `<domain>_Operators` group.
    - Add the members of the `Monitors` role to the `<domain>_Monitors` group.
34. Sign in to the WebLogic Server Administration Console for the target domain.

    `https://<target_admin_ip>:7002/console`
35. Click **Security Realms**.
36. Click the default realm.
37. Click the **Roles and Policies** tab.
38. From the Roles table, expand **Global Roles**, and then expand **Roles**.
39. Click **View Role Conditions** for the `Admin` role.
40. Click the group name assigned to this role. The default is **Administrators**.
41. Enter `<domain>_Administrators`.
42. Click **OK**, and then click **Save**.
43. From the breadcrumb links at the top of the page, click **Realm Roles**.
44. *Repeat from step 38* for the remaining administrator roles:
    - Map `Deployer` to `<domain>_Deployers`
    - Map `Operator` to `<domain>_Operators`
    - Map `Monitor` to `<domain>_Monitors`

# Integrate Fusion Middleware Components with Oracle Identity Cloud Service

If your source Oracle Java Cloud Service instance uses Oracle Identity Cloud Service for authentication, then you can integrate certain Oracle Fusion Middleware components in the target domain with Oracle Identity Cloud Service.

If your source instance uses Oracle Web Services Manager to protect web service applications and clients, then see Secure Web Services Using Identity Cloud Service in *Using Oracle WebLogic Server for Oracle Cloud Infrastructure*.

If applications on your source instance use Oracle Platform Security Services APIs to look up user and group information, then see Integrate OPSS User and Group APIs with Identity Cloud Service in *Using Oracle WebLogic Server for Oracle Cloud Infrastructure*.

4

# Migrate an Instance to Oracle WebLogic Server for Oracle Cloud Infrastructure Using Classic Tools

Use the Oracle Cloud Infrastructure Classic Java Migration Tool to migrate your Oracle WebLogic Server domain resources and applications from your existing Oracle Java Cloud Service instance in Oracle Cloud Infrastructure Classic to a new domain in Oracle WebLogic Server for Oracle Cloud Infrastructure.

> **Note:**
>
> Oracle recommends migrating your existing domains in Oracle Java Cloud Service to Oracle WebLogic Server for Oracle Cloud Infrastructure.

Before you begin the migration process, see Prepare to Migrate Oracle Java Cloud Service to Oracle Cloud Infrastructure.

When you migrate an Oracle Java Cloud Service instance, the following terms are used:

- *Source*: The Oracle Java Cloud Service instance in Oracle Cloud Infrastructure Classic.
- *Target*: The domain and related cloud resources in Oracle WebLogic Server for Oracle Cloud Infrastructure.

**Topics:**

- Integrate Fusion Middleware Components with Oracle Identity Cloud Service
- Edit the Domain Configuration File
- Copy Supporting Files to the Target
- Update the Target Domain

# Get Information About the Target Environment

Collect the information required for the migration tools to connect to the target Oracle Cloud Infrastructure environment.

1. Access the Oracle Cloud Infrastructure Console.
2. From the menu, choose **Administration** and then choose **Tenancy Details**.
3. Record the tenancy's OCID and Home Region.
4. From the menu, choose **Identity** and then **Users**.
5. Click your user name.
6. Record the user's OCID. Under API Keys, record the Fingerprint.

   You will also need the corresponding PEM key file.
7. From the menu, choose **Identity** and then **Compartments**.
8. Record the OCID of the compartment where you want to create the instance.
9. From the menu, choose **Identity** and then **Federation**.
10. From the Oracle Identity Cloud Service Console URL, identify and record the identity domain ID, which has the format `idcs-<guid>`.
11. From the menu, choose **Networking** and then **Virtual Cloud Networks** (VCN).
12. Select the **Compartment** where you want to create the instance.
13. Click the VCN in which you want to create this instance.
14. Under Subnets, click the subnet in which you want to create this instance.
15. Record the subnet's OCID. If it is not a regional subnet, then also record the subnet's Availability Domain.
16. If you selected a regional subnet, then choose an availability domain for the target instance.
    a. Access the Oracle Java Cloud Service Console.
    b. Click **Create Instance**.
    c. Select your **Region**.
    d. From **Availability Domain**, record the name of the availability domain in which you want to create this instance.
    e. Click **Cancel**.

The following table shows sample values for each input.

| Name | Sample Value |
|---|---|
| Tenancy OCID | `ocid1.tenancy.oc1..aaaaaaaaju6k54i7...` |
| User OCID | `ocid1.user.oc1..aaaaaaaahvtv5qo...` |
| User API Key Fingerprint | `81:45:aa:...` |
| Compartment OCID | `ocid1.compartment.oc1..aaaaaaaaz...` |
| Region | `us-ashburn-1` |
| Availability Domain | `kWVD:US-ASHBURN-AD-3` |
| Subnet OCID | `ocid1.subnet.oc1.iad.aaaaaaaarz7...` |
| Identity Domain ID | `idcs-9bd53...` |

# Launch the Migration Controller Instance in the Source Environment

In your Oracle Cloud Infrastructure Compute Classic account, create the source controller (Control-S) instance, which includes Oracle Cloud Infrastructure Classic Java Migration Tool.

The Control-S compute instance must be created in the same identity domain and site as the source Oracle Java Cloud Service instance that you want to migrate.

The Control-S compute instance and associated storage volumes are by default billed at the applicable rates for your account. However, you can rename these resources so that the name includes `/oraclemigration` as a container. Resources created in this `/oraclemigration` container aren't billed to your account.

1. Access the Oracle Cloud Infrastructure Compute Classic Console.
2. Click **Create Instance**.

3. Click **Show All Images**.
4. Select the image `OL_7.5_UEKR4_x86_64_MIGRATION`, which is found under **Oracle Images**.
5. Click **Next**.
6. Select a **Shape** with a sufficient number of OCPUs for the migration task.
7. Click **Next**.
8. Enter a **Name**, or use the default instance name.
9. Select an existing public **SSH Key** or add a new one. You'll use the corresponding private key to connect to the Control-S instance.
10. Click **Next**.
11. Verify that **Shared Network** is selected.
12. For **Public IP Address**, select **Persistent Public IP Reservation**.
13. For **Security Lists**, verify that the `default` security list is selected, which allows SSH inbound traffic.

    Also ensure that security rules are in place to allow SSH outbound, SMB inbound, and HTTPS outbound traffic.
14. If you want to migrate instances that have an interface on an IP network, then configure the network interfaces of the Control-S instance on the relevant IP networks as well, so that the Control-S instance can access the source instances that you want to migrate.
15. Complete the creation of the compute instance.

    Wait until its status is **Running**.
16. Optional: Move the Control-S instance and storage volumes into the `/oraclemigration` container.

    Alternatively, if you create the Control-S instance using the API, CLI, or Terraform, you can specify `/oraclemigration` in the resource names as part of the instance parameters.

    a. Click the **Orchestrations** tab.

    b. Locate the relevant orchestration for your compute instance, and from the menu, select **Suspend**.

    c. After the orchestration status changes to **Suspended,** from the menu, select **Update**.

    d. From the Instance section, click the menu and select **Edit JSON**.

    e. In the Edit Orchestration Object JSON window, locate the instance name. This is usually displayed within the `template` section, after `networking`.

    ```
    "name": "/Compute-Identity_Domain/User/Instance",
    ```

Modify the instance name to include the `/oraclemigration` container. For example:

```
"name": "/Compute-ExampleDomain/user@example.com/oraclemigration/
MyControlS",
```

f. Click **Update**.

g. From the Orchestrations page, go to the relevant orchestration and from the menu, select **Terminate**.

h. After the orchestration status changes to **Stopped**, from the menu, select **Update**.

i. From the Storage Volume section, go to the relevant storage volume, click the menu and select **Edit JSON**.

j. In the Edit Orchestration Object JSON window, locate the storage volume name in the `template` section:

```
"name": "/Identity_Domain/User/Volume",
```

Modify the instance name to include the `/oraclemigration` container. For example:

```
"name": "/Compute-ExampleDomain/user@example.com/oraclemigration/
MyControlS_storage",
```

k. Click **Update**.

l. Repeat these steps for any other storage volume in this orchestration that you want to move to the `/oraclemigration` container.

m. From the Orchestrations page, go to the relevant orchestration and from the menu, select **Start**.

17. Use an SSH client and your private key to log into the Control-S compute instance as the `opc` user.

# Update the Secret File

All of the tools required for the migration are already installed on the Control-S instance, but additional configuration is required to provide details about the source and target environments.

A single Control-S instance can migrate resources only from a single Oracle Cloud Infrastructure Classic account and site, and only to a single Oracle Cloud Infrastructure tenancy, region, and availability domain.

1. From the Control-S compute instance, copy `/home/opc/ansible/secret.yml.sample` to `/home/opc/ansible/secret.yml`.
2. Edit `/home/opc/ansible/secret.yml`.
3. Update the following Oracle Cloud Infrastructure parameters.
   - `compartment_id` is the OCID of the compartment where you want to create the target instance.

- `user_id` is the OCID of the Oracle Cloud Infrastructure user.
- `fingerprint` is the API key fingerprint of the user.
- `tenancy_id` is the OCID of the Oracle Cloud Infrastructure tenancy.
- `region` is the Oracle Cloud Infrastructure region where you want to create the target instance.
- `availability_domain` is the availability domain where you want to create the target instance.
- `subnet_id` is the OCID of the subnet where you want to create the instance.

For example:

```
# OCI info
compartment_id: ocid1.compartment.oc1..aaaaaaaa...
user_id: ocid1.user.oc1..aaaaaaaa...
fingerprint: a0:a0:a0:a0:a0...
tenancy_id: ocid1.tenancy.oc1..aaaaaaaa...
region: us-ashburn-1
availability_domain: kWVD:US-ASHBURN-AD-3
...
subnet_id: ocid1.subnet.oc1.iad.aaaaaaaa...
```

4. Modify permissions on this file to restrict access.

```
chmod 600 /home/opc/ansible/secret.yml
```

5. Apply the configuration to the system.

```
opcmigrate migrate instance service setup
```

This command creates the required files `/home/opc/.opc/profiles/default` and `/home/opc/.oci/config`.

6. Copy the Oracle Cloud Infrastructure user's PEM key file to the Control-S instance. Name the file `/home/opc/.oci/oci_api_key.pem`.
7. Modify permissions on the Oracle Cloud Infrastructure key file to restrict access.

```
chmod 600 /home/opc/.oci/oci_api_key.pem
```

8. Copy the public and private SSH key files required for accessing your source Oracle Java Cloud Service instance to the Control-S instance.
9. Modify permissions on the Oracle Java Cloud Service key files to restrict access.

For example:

```
chmod 600 /home/opc/jcskey.pub
chmod 600 /home/opc/jcskey
```

# Update the Default Profile File

The Oracle Cloud Infrastructure Classic Java Migration Tool connects to your source environment using information that you provide in a profile file.

The information you provide in the profile file includes the user name or identity for each service in the source environment, as well as the service end point and region. If you want to run the tool in multiple regions or tenancies, you can create separate profile files for each region and tenancy.

You also provide connectivity details for each Oracle Java Cloud Service instance that you want to migrate. If you include the WebLogic Server administrator credentials for a service instance, Oracle Cloud Infrastructure Classic Java Migration Tool also migrates any Oracle Fusion Middleware security resources (custom users, groups, roles, policies, or credential maps) to the target domain.

1. Access the Oracle Cloud Infrastructure Compute Classic Console.
2. Click the **Site** select box.
3. Record the REST Endpoint.
4. Identify and record your Oracle Java Cloud Service REST Endpoint.

   See Send Requests in *REST API for Oracle Java Cloud Service*.
5. Access the Oracle Java Cloud Service Console.
6. Click the source instance.
7. Click **Instance Details**  , and then record the Region in which the source instance was created.
8. From the Control-S compute instance, create a properties file with the WebLogic Server administrator user name and password of your source instance.

   ```
   admin_user=your_username
   admin_password=your_password
   ```

   This step is required only if the source domain includes custom users, groups, roles, policies or credential maps.
9. From the Control-S compute instance, edit the file `/home/opc/.opc/profiles/default`.
10. In the `compute` section, update the `endpoint` and `user` parameters. Enter the name of a user with access to Oracle Cloud Infrastructure Compute Classic.

    ```
    "compute": {
      "endpoint": "Compute_Endpoint",
      "user": "/Compute-Identity_Domain/User_Name"
      ...
    ```

For example:

```
"compute": {
  "endpoint": "compute.uscom-central-1.oraclecloud.com"
  "user": "/Compute-ExampleDomain/user@example.com",
  ...
```

11. **Optional:** Enter the location of a file that contains your Oracle Cloud Infrastructure Compute Classic password.

    For example:

```
"compute": {
  "endpoint": "compute.uscom-central-1.oraclecloud.com"
  "user": "/Compute-ExampleDomain/user@example.com",
  "password-file": "/home/opc/.opc/password-file",
  ...
```

    If you don't specify a password file for a service, you'll be prompted to provide the password when you run the tool.

12. If not already present, add the `paas` section to the file.

```
{
  ...
  "compute": {
  ...
  },
  "paas": {
    "user": "User_Name",
    "identity_id": "Identity_Domain_Id",
    "endpoint": "PaaS_Endpoint",
    "region": "Source_Region"
  }
}
```

    For example:

```
{
  ...
  "compute": {
  ...
  },
  "paas": {
    "user": "user@example.com",
    "identity_id": "idcs-0000abcd0000defg0000hijk0000lmno",
    "endpoint": "psm.us.oraclecloud.com",
    "region": "uscom-central-1"
  }
}
```

13. Add the `jcs` section to the file. Specify the locations of the public and private SSH key files for your source Oracle Java Cloud Service instance.

```
{
  ...
  "paas": {
  ...
  },
  "jcs": {
    "Instance_Name": {
      "ssh_private_key": "Private_Key_File",
      "ssh_public_key": "Public_Key_File"
    }
  }
}
```

For example:

```
{
  ...
  "paas": {
  ...
  },
  "jcs": {
    "MyJavaInstance": {
      "ssh_private_key": "/home/opc/jcskey",
      "ssh_public_key": "/home/opc/jcskey.pub"
    }
  }
}
```

14. In the `jcs` section, specify the location of the properties file that contains the WebLogic Server credentials for the source instance.

For example:

```
...
  "jcs": {
    "MyJavaInstance": {
      "ssh_private_key": "/home/opc/jcskey",
      "ssh_public_key": "/home/opc/jcskey.pub",
      "wls_admin_properties": "/home/opc/wls_admin_properties"
    }
```

This step is required only if the source domain includes custom users, groups, roles, policies or credential maps.

# Discover Resources in Your Source Environment

To discover all Oracle Cloud Infrastructure Classic resources in the services for which you've provided credentials, log in to the Control-S instance and run the following command.

```
opcmigrate discover
```

When prompted, enter the passwords for the user names that you specified in the default profile.

For example:

```
opcmigrate discover
Compute Classic Password [/Compute-ExampleDomain/user@example.com]:
INFO Authenticating with OCI Classic Compute API
INFO Compute Endpoint: https://compute.uscom-central-1.oraclecloud.com
INFO Discovering resources for "ExampleDomain".
WARNING Load Balancer Classic credentials not configured in profile
PaaS Services Password [user@example.com]:
WARNING Object Storage Classic credentials not configured in profile
INFO Discovering containers: ['/Compute-ExampleDomain']
INFO Getting Account Resources for /Compute-ExampleDomain
INFO Getting Network Resources for /Compute-ExampleDomain
INFO Getting Network Resources for /oracle/public
INFO Getting Instance Resources for /Compute-ExampleDomain
INFO Getting Instance Resources for /oracle/public
INFO Getting Instances for /Compute-ExampleDomain
INFO Getting PaaS Resources for uscom-central-1
INFO Storing discovered resources to 'resources-default.json'
```

# List Your Oracle Java Cloud Service Instances

To list the Oracle Java Cloud Service instances in the source environment, log in to the Control-S instance and run the following command.

```
opcmigrate migrate jcs list
```

This command uses the output generated by the `discover` command to identify and list the available Oracle Java Cloud Service instances.

For example:

```
opcmigrate migrate jcs list
INFO Loaded resources from 'resources-default.json' ...
Java Cloud Service Instances

Name                 Version              State
Description
-------------------- -------------------- --------------------
------------------------------------------------------------
```

```
MyJavaInstance      11gR1               READY               My first
instance
AnotherInstance     12cR3               READY               My
second instance
```

# Export Your Source Instance Configuration

To create an archive of the source Oracle Java Cloud Service instance using the WebLogic Server Deploy Tooling, log in to the Control-S instance and run the following command.

```
opcmigrate migrate jcs export -s <instance_name>
```

This command creates the following files:

- `<instance_name>-<date>-<timestamp>.tgz`: An archive of the source instance, which includes the applications that are on the source instance as well as the domain configuration metadata. This archive is uploaded to Oracle Cloud Infrastructure Object Storage.
- `<instance_name>-<date>-<timestamp>.json`: You edit this file to specify the required passwords for the target domain, as well as to specify any configuration parameters that will be different on the target instance.

For example:

```
opcmigrate migrate jcs export -s MyJavaInstance
INFO Loaded resources from 'resources-default.json' ...
INFO Exporting JCS service 'MyJavaInstance'
INFO Installing Oracle WebLogic Server Deploy Tooling on 203.0.113.13
INFO Create temporary directory on controller
INFO Download WebLogic Deploy Tooling to controller
INFO Upload and Extract WebLogic Deploy Tooling archive to remote host
INFO Remove temporary directory from controller
INFO Exporting WebLogic Domain at 203.0.113.13
INFO Create temporary directory on remote host
INFO Run WebLogic Deploy Tooling discoverDomain.sh command
INFO Download discovered domain files to controller
INFO Remove temp directory from remote
INFO Generating WebLogic config template
'MyJavaInstance-20190722-18:50:35.json'
INFO Creating instance archive 'MyJavaInstance-20190722-18:50:35.tgz'
INFO Uploading artifacts to object storage
INFO JCS service 'MyJavaInstance' export complete
```

By default, this command uses the `resources-default.json` file in the local directory. You can use the `--file` option to specify a resources file with a different name or in a different directory.

# Perform Prerequisite Tasks for Oracle WebLogic Server for Oracle Cloud Infrastructure

Before you create a WebLogic Server domain using Oracle WebLogic Server for Oracle Cloud Infrastructure, 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.
   - A vault and encryption key
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).

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

   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
   - Database administrator password
   - Client secret, if using Oracle Identity Cloud Service

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

# Create the Target Domain Using Oracle WebLogic Server for Oracle Cloud Infrastructure

Launch the Oracle WebLogic Server for Oracle Cloud Infrastructure 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.

> **Note:**
>
> The migration import tooling uses the SSH keys specified for the source instance in your `~/.opc/profiles/default` file (or `~/.opc/profiles/<profile>` file) on the Control-S instance. Be sure to use the same SSH key pair to create your target domain with Oracle WebLogic Server for Oracle Cloud Infrastructure.

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 same Oracle WebLogic Server edition as your source instance.
4. For **Version**, select the same major version (`X.Y`) as the source instance.

   For example, `12.2.1.2` and `12.2.1.3` are the same major version 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 (OCPUs) and the amount of memory that are available in the Oracle Cloud Infrastructure Classic shape in your source instance.

    See Select Oracle Cloud Infrastructure Shapes.
11. Enter the SSH public key.
12. Select the availability domain where you want to create the domain.
13. Select the same number of managed servers as the source instance.
14. Enter the WebLogic Server user name, and paste the OCID for the secret that contains the WebLogic Server password.
15. For **Network Compartment**, select the same compartment you selected earlier upon launching the stack.
16. For **Virtual Cloud Network Strategy**, select **Use Existing VCN** and then select the virtual cloud network (VCN) where you want to create the domain.
17. For **Subnet Strategy**, select **Use Existing Subnet** or **Create New Subnet**.
18. 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.

19. If your source instance includes an Oracle Traffic Director 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. For **Database Strategy**, select **Database System**.
22. Select the compartment and VCN in which you created the database.
23. Select your DB System, database home, database version, and database.
24. Enter the pluggable database (PDB) name if the selected database is running Oracle Database 12c or later.
25. Enter the name of a database user with database administrator (DBA) privileges, and paste the OCID of the secret that contains the database password.
26. Enter the database listen port (1521 by default).
27. If your domain and database are on different VCNs, then you must configure local VCN peering.

    Oracle WebLogic Server for Oracle Cloud Infrastructure 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.
    a. Specify a CIDR for the new subnet in the WebLogic VCN.
    b. Specify a CIDR for the new subnet in the database VCN.
    c. Select a shape for the new DNS Forwarder compute instance in each VCN.
28. Click **Next**, and then click **Create**.

See Create a JRF-Enabled Domain in *Using Oracle WebLogic Server for Oracle Cloud Infrastructure*.

# Migrate Oracle Fusion Middleware Security Resources

If you customized the Oracle WebLogic Server security providers in your source Oracle Java Cloud Service instance, then you must apply the same changes in the target domain.

If you specified the WebLogic Server administrator credentials for your source instance in the default profile, the Oracle Cloud Infrastructure Classic Java Migration Tool automatically migrates the following Oracle Fusion Middleware security resources from the source domain to the target domain:

- Users
- Groups
- Roles
- Policies
- Keystores
- Credential maps
- Audit policies
- Web Services Manager (WSM) policies

The tool does not automatically update the security providers in the target domain.

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 Oracle Cloud Infrastructure*.
4. From both consoles, click **WebLogic Domain**, select **Security**, and then select **Security Provider Configuration**.
5. 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.

# 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 Oracle Identity Cloud Service console.
2. Click the navigation drawer  , and then select **Applications**.
3. Click the security application for your source instance, `JaaS_<source_instance_name>`.
4. Copy the following information for the security application:
   - Application ID
   - Client ID
   - Client secret

5. Encode the following string in base64 format.

   ```
   <client_id>:<client_secret>
   ```

6. Use the Oracle Identity Cloud Service REST API to request an access token for the source instance's security application.

   ```
   curl --location --request POST 'https://<idcs_host>/oauth2/v1/
   token' \
   --header 'Authorization: Basic <base64_encoded_clientid:secret>' \
   --header 'Content-Type: application/x-www-form-urlencoded' \
   --data-urlencode 'grant_type=password' \
   --data-urlencode 'scope=urn:opc:idm:__myscopes__' \
   --data-urlencode 'username=<idcs_user_name>' \
   --data-urlencode 'password=<idcs_password>'
   ```

   Copy the access token from the response.

   See Generate Access Token and Other OAuth Runtime Tokens to Access the Resource in *REST API for Oracle Identity Cloud Service*.

7. Use the Oracle Identity Cloud Service REST API to export the web tier policy for the security application.

   ```
   curl -X GET 'https://<idcs_host>/admin/v1/Apps/
   <application_ID>&attributes=displayName,urn:ietf:params:scim:schemas
   :oracle:idcs:extension:webTierPolicy:App:webTierPolicyJson' \
   -H 'Authorization:Bearer <access_token>'
   ```

   Locate the web tier policy in the response:

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

   See Get an App in *REST API for Oracle Identity Cloud Service*.

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

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

13. After the export job has finished, click **Download**. Save the file `AppRoleExport_<id>.csv`.

14. Click the navigation drawer  , and then select **Applications**.
15. Click the security application for your target domain, `<stack>_enterprise_idcs_app_<timestamp>`.

    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.
16. Click **SSO Configuration**.
17. 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.
18. Expand **Resources**.
19. Within the Resources section, click **Add**.
20. Enter a **Resource Name**.

    For example, `myapp`
21. For **Resource URL**, paste the value of the `filter` property.
22. If the filter's `type` property is `regex`, then select **Regex**.
23. Click **OK**.
24. Expand **Authentication Policy**. Under Managed Resources, click **Add**.
25. For **Resource**, select your new resource.
26. 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**
27. Click **Add**.
28. *Repeat from step 18* for each additional filter in the exported web tier policy.

29. Click the navigation drawer  , and then select **Groups**.
30. Create these groups for the target domain.
    - `<domain>_Administrators`
    - `<domain>_Deployers`
    - `<domain>_Operators`
    - `<domain>_Monitors`

    For example:
    - `MyDomain_Administrators`
    - `MyDomain_Deployers`
    - `MyDomain_Operators`
    - `MyDomain_Monitors`
31. Open `AppRoleExport_<id>.csv`, and identify the users and groups assigned to the `Administrators` role in the source instance.
32. Edit the `<domain>_Administrators` group, and add the same users and groups as the `Administrators` role in the source instance.
33. Repeat the previous step for the remaining roles in `AppRoleExport_<id>.csv`:
    - Add the members of the `Deployers` role to the `<domain>_Deployers` group.
    - Add the members of the `Operators` role to the `<domain>_Operators` group.
    - Add the members of the `Monitors` role to the `<domain>_Monitors` group.
34. Sign in to the WebLogic Server Administration Console for the target domain.

    `https://<target_admin_ip>:7002/console`
35. Click **Security Realms**.
36. Click the default realm.
37. Click the **Roles and Policies** tab.
38. From the Roles table, expand **Global Roles**, and then expand **Roles**.
39. Click **View Role Conditions** for the `Admin` role.
40. Click the group name assigned to this role. The default is **Administrators**.
41. Enter `<domain>_Administrators`.
42. Click **OK**, and then click **Save**.
43. From the breadcrumb links at the top of the page, click **Realm Roles**.
44. *Repeat from step 38* for the remaining administrator roles:
    - Map `Deployer` to `<domain>_Deployers`
    - Map `Operator` to `<domain>_Operators`
    - Map `Monitor` to `<domain>_Monitors`

# Integrate Fusion Middleware Components with Oracle Identity Cloud Service

If your source Oracle Java Cloud Service instance uses Oracle Identity Cloud Service for authentication, then you can integrate certain Oracle Fusion Middleware components in the target domain with Oracle Identity Cloud Service.

If your source instance uses Oracle Web Services Manager to protect web service applications and clients, then see Secure Web Services Using Identity Cloud Service in *Using Oracle WebLogic Server for Oracle Cloud Infrastructure*.

If applications on your source instance use Oracle Platform Security Services APIs to look up user and group information, then see Integrate OPSS User and Group APIs with Identity Cloud Service in *Using Oracle WebLogic Server for Oracle Cloud Infrastructure*.

# Edit the Domain Configuration File

The `export` command creates a file that contains parameters for updating the target WebLogic Server domain. Specify JDBC URLs and passwords, SSL keystore passwords, and other details for the target instance.

For security purposes, Oracle WebLogic Server Deploy Tooling excludes the values of all passwords during domain discovery.

1. On the Control-S instance, edit the file `<instance_name>-<date-time-stamp>.json`.

   Refer to the output from the export command to determine the specific file name.

2. Update the following attributes.

   - `JCSServiceName` - The name of the target domain that you created with Oracle WebLogic Server for Oracle Cloud Infrastructure (including the resource name prefix you provided when you created the domain)
   - `JCSAdminIPAddress` - The IP address of the first node in the target instance (running the Administration Server)
   - `WeblogicAdminUser` - The user name for the WebLogic Server domain administrator on the target instance
   - `WeblogicAdminPassword` - The password for the WebLogic Server domain administrator on the target instance

3. If your source instance includes a load balancer, then update the `FrontendHost` attribute for each cluster in the `Cluster` node.

   Enter the public IP address of the load balancer in your target instance.

   Example:

```
"topology": {
  "Cluster": {
    "cluster": {
      "FrontendHost": "<target_LB_IP>"
    }
```

```
  ...
}
```

4. If you configured any custom startup arguments for a server in your source instance, then update the `AdditionalServerStartArguments` attribute for each server in the `Server` node.

   Set the value of `AdditionalServerStartArguments` to the custom arguments only.

   Example:

```
"topology": {
  ...
  "Server": {
    ...
    "server_1": {
      ...
      "AdditionalServerStartArguments": "-Dmy.custom.arg=true"
    }
  ...
}
```

5. If the servers in your source instance are configured to use custom identity and trust keystore files, then update the file with the keystore passwords.

   - `CustomIdentityKeyStorePassPhrase`
   - `CustomTrustKeyStorePassPhrase`
   - `ServerPrivateKeyPassPhrase`

   Example:

```
"topology": {
  ...
  "Server": {
    "server_1": {
      ...
      "CustomIdentityKeyStorePassPhrase": "<your_password>",
      "CustomTrustKeyStorePassPhrase": "<your_password>",
      "ServerPrivateKeyPassPhrase": "<your_password>"
    }
  ...
}
```

6. If your source instance includes custom Java Database Connectivity (JDBC) data sources, then provide the location and password of the new application databases in Oracle Cloud Infrastructure.

   a. For each data source in the `JDBCSystemResource` node, update the `password` attribute.

      Example:

```
"resources": {
  "JDBCSystemResource": {
    "MyDataSource1": {
      ...
      "password": "<your_password>"
```

```
    }
  }
  ...
```

b. For each data source, find the `url` attribute and specify 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 | `jdbc:oracle:thin:@//<db_hostname>-scan.<db_domain>:<db_port>/<pdb_name>.<db_domain>` |
| 12c | Bare Metal | `jdbc:oracle:thin:@//<db_hostname>.<db_domain>:<db_port>/<pdb_name>.<db_domain>` |
| 11g | VM | `jdbc:oracle:thin:@//<db_hostname>-scan.<db_domain>:<db_port>/<db_unique_name>.<db_domain>` |
| 11g | Bare Metal | `jdbc:oracle:thin:@//<db_hostname>.<db_domain>:<db_port>/<db_unique_name>.<db_domain>` |

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

```
"resources": {
  "JDBCSystemResource": {
    "MyDataSource1": {
      "url": "jdbc:oracle:thin:@//
myappdb-scan.mydbsubnet.myvcn.oraclevcn.com:1521/
pdb1.mydbsubnet.myvcn.oraclevcn.com",
      ...
    }
  }
  ...
```

7. 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. For each provider in the `ForeignJNDIProvider` node, update the `password` attribute.

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

Example:

```
"resources": {
  ...
  "ForeignJNDIProvider": {
    "MyJNDIProvider1": {
```

```
      "url": "t3://myjndiserver.example.com:9073",
      "password": "<your_password>"
    }
  }
  ...
```

b. For each foreign server in the ForeignJMSServer node, update the password attributes.

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

   Example:

```
"resources": {
  ...
  "ForeignJMSServer": {
    "MyForeignJMS1": {
      "url": "t3://myjms.example.com:9073",
      "ForeignConnectionFactory": {
        "MyForeignJMS1Factory": {
          "password": "<your_password>"
        }
      }
    }
  }
  ...
```

c. For each bridge destination in the `JMSBridgeDestination` node, update the `password` attribute.

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

   Example:

```
"resources": {
  ...
  "JMSBridgeDestination": {
    "MyBridgeDest1": {
      "url": "t3://myjms.example.com:9073",
      "password": "<your_password>"
    }
  }
  ...
```

d. For each SAF context in the `SAFLoginContext` node, update the `password` attribute.

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

   Example:

```
"resources": {
  ...
  "SAFLoginContext": {
```

```
        "MySAF1": {
          "url": "t3://myjms.example.com:9073",
          "password": "<your_password>"
        }
      }
      ...
```

8. If your source instance includes any JavaMail sessions, then update the passwords for each mail session in the `MailSession` node.

   Example:

```
"resources": {
  ...
  "MailSession": {
    "MyMailSession1": {
      "password": "<your_password>",
      "properties": {
        "mail.smtp.password": "<your_password>",
        "mail.imap.password": "<your_password>"
      }
    }
  }
  ...
```

9. If your source instance includes any custom WebLogic Diagnostic Framework (WLDF) REST notification endpoints, then provide the passwords for each WLDF resource in the `WLDFSystemResource` node.

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

   Example:

```
"resources": {
  ...
  "WLDFSystemResource": {
    "MyModule": {
      "RestNotification": {
        "MyNotification1": {
          "url": "http://myserver.example.com:9073/notify",
          "password": "<your_password>"
        }
      }
    }
  }
  ...
```

# Copy Supporting Files to the Target

Identify and copy any files to your target domain that are not managed by Oracle WebLogic Server Deploy Tooling.

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

- Application deployments
- Library deployments
- Custom keystores

Other files that your applications or domain resources require are not automatically managed by Oracle WebLogic Server Deploy Tooling, including files that are located outside the `DOMAIN_HOME` directory. You must manually copy these files to the target nodes.

1. Use SSH to connect to the Administration Server node in your *source* Oracle Java Cloud Service instance.

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

2. Switch to the `oracle` user.

   ```
   sudo su - oracle
   ```

3. Identify any supporting files that need to be copied to the target.
4. 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.

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

   Example:

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

6. Disconnect from the node.
7. 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 .
```

8. Use SCP to upload the files to the Administration Server node in your *target* domain.

   Example:

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

9. Use SSH to connect to the Administration Server node in your target domain.

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

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

    Example:

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

11. Switch to the `oracle` user.

    ```
    sudo su - oracle
    ```

12. 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
    ```

13. Disconnect from the node.

# Update the Target Domain

After the target Oracle WebLogic Server for Oracle Cloud Infrastructure domain is running, use the Oracle Cloud Infrastructure Classic Java Migration Tool to import the WebLogic Server domain configuration that you previously exported from the source instance.

The tool downloads the archive from Oracle Cloud Infrastructure Object Storage, updates your target instance's domain configuration with the values provided in the JSON file, deploys your applications, and then restarts the domain.

The Oracle Cloud Infrastructure Classic Java Migration Tool creates a bastion compute instance in Oracle Cloud Infrastructure in order for the Control-S instance to access your target service instance. After updating the target service instance, the tool deletes the temporary bastion compute instance.

1. Edit your `~/.oci/config` file.
   a. Specify the required target compartment using the parameter `target_compartment_name`.
   b. Specify an optional `shape` for the bastion instance that's created during the import procedure. If not specified, the default shape is `VM.Standard2.1`.

Example:

```
[DEFAULT]
user=ocid1.user.oc1..aaaaaaaa6yasw6w6jwn2ddkrynekd
fingerprint=1e:c0:d0:07:45:fc:eb:b6:04:a1:fa:82:39:52:19:ac
key_file=~/.oci/oci_api_key.pem
region=us-ashburn-1
tenancy=ocid1.tenancy.oc1..aaaaaaaae5rd2r3ry4qrvylabwdydhj3usa72tb4s
znokyk
shape=VM.Standard2.2
target_compartment_name=MyCompartment
```

2. Log in to the Control-S instance and run the following command.

```
opcmigrate migrate jcs import -c <instance_name>-<date>-
<timestamp>.json -a <instance_name>-<date>-<timestamp>.tgz -p
<Control-S-instance-public-ip-address>
```

Example:

```
opcmigrate migrate jcs import -c mysource-20191105-22:04:37.json -a
mysource-20191105-22:04:37.tgz -p 192.0.2.1

2019-11-15T16:32:36 INFO Importing JCS service 'mysource'
2019-11-15T16:32:36 INFO Using local archive file
mysource-20191105-22:04:37.tgz
2019-11-15T16:32:36 INFO Inflating instance archive
'mysource-20191105-22:04:37.tgz'
2019-11-15T16:32:36 INFO Weblogic model file processing complete
2019-11-15T16:32:48 INFO Using migration tools instance NAT IP:
192.0.2.147
2019-11-15T16:32:48 INFO Provisioning bastion host
2019-11-15T16:32:49 INFO Initializing Terraform
2019-11-15T16:32:52 INFO Applying Terraform
2019-11-15T16:34:22 INFO Bastion host provisioned: 203.0.113.50
2019-11-15T16:34:22 INFO Waiting for SSH to become available
2019-11-15T16:34:23 WARNING Retry#1: Waiting for bastion SSH to
become reachable
2019-11-15T16:35:03 INFO Installing Oracle WebLogic Server Deploy
Tooling on 203.0.113.14
2019-11-15T16:35:08 INFO Create temporary directory on controller
2019-11-15T16:35:09 INFO Download WebLogic Deploy Tooling to
controller
2019-11-15T16:35:11 INFO Upload and Extract WebLogic Deploy Tooling
archive to remote host
2019-11-15T16:35:17 INFO Remove temporary directory from controller
2019-11-15T16:35:17 INFO Uploading WebLogic Domain related files to
203.0.113.14
2019-11-15T16:35:21 INFO Identify destination files
2019-11-15T16:35:21 INFO Copy WebLogic domain files to destination
2019-11-15T16:35:32 INFO Clean up script directory
2019-11-15T16:35:34 INFO Unarchive update domain script
2019-11-15T16:35:39 INFO Upload scripts to remote hosts
2019-11-15T16:36:02 INFO Validating uploaded WebLogic model file
```

```
2019-11-15T16:36:05 INFO Identify uploaded files
2019-11-15T16:36:06 INFO Run WebLogic Deploy Tooling
validateModel.sh command
2019-11-15T16:36:22 INFO Stopping WebLogic domain
2019-11-15T16:36:26 INFO Stop WebLogic Domain
2019-11-15T16:41:26 INFO Updating WebLogic domain
2019-11-15T16:41:30 INFO Set target host full path names for
uploaded files
2019-11-15T16:41:30 INFO Update WebLogic Domain
2019-11-15T16:43:04 INFO Starting WebLogic domain
2019-11-15T16:43:08 INFO Start WebLogic domain
2019-11-15T16:58:49 INFO Tearing down bastion host...
2019-11-15T16:58:49 INFO Tearing down Terraform
```

5

# Migrate an Instance to Oracle Java Cloud Service Using Application Migration Service

Use Application Migration 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 Classic to a new instance in Oracle Cloud Infrastructure.

Application Migration is available only in specific Oracle Cloud Infrastructure regions. See Overview of Application Migration in the Oracle Cloud Infrastructure documentation.

Application Migration does not support the migration of WebLogic Server domains that include these types of resources:

- Custom Identity or Trust Keystore
- Foreign JNDI Provider
- Foreign JMS Server
- JMS Bridge Destination
- Storage-and-Forward (SAF) Context
- JavaMail Session
- WebLogic Diagnostic Framework (WLDF) REST Notification Endpoint

If your source Oracle Java Cloud Service instance uses these resource types, then Oracle recommends using the Oracle Cloud Infrastructure Classic Java Migration Tool instead of Application Migration. See Migrate an Instance to Oracle Java Cloud Service Using Classic Tools.

Before you begin the migration process, see Prepare to Migrate Oracle Java Cloud Service to Oracle Cloud Infrastructure.

When you migrate an Oracle Java Cloud Service instance, the following terms are used:

- *Source*: The connection to your Oracle Cloud Infrastructure Classic account in Application Migration.
- *Source Instance*: The Oracle Java Cloud Service instance in Oracle Cloud Infrastructure Classic.
- *Target Instance*: The Oracle Java Cloud Service instance in Oracle Cloud Infrastructure.

**Topics:**

- Configure and Run a Migration
- Copy Supporting Files to the Target Instance
- Recreate Oracle Fusion Middleware Security Resources
- Migrate Oracle Identity Cloud Service Roles and Policies

# Perform Prerequisite Tasks for Oracle Java Cloud Service

Before you use Application Migration Service to create an Oracle Java Cloud Service instance in an Oracle Cloud Infrastructure region, you must create the required infrastructure resources.

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
- A storage bucket and user authentication token for backups (optional)
- Policies that allow Oracle Java Cloud Service to access the resources in your compartment

See Prerequisites for Oracle Platform Services in the Oracle Cloud Infrastructure documentation.

Application Migration Service automatically creates an Oracle Cloud Infrastructure Database before it creates the target instance. Oracle Java Cloud Service provisions the required infrastructure schema to this database.

# Create a Source

Use Application Migration to connect to your Oracle Cloud Infrastructure Classic account and region.

1. From the Oracle Cloud Infrastructure console, navigate to **Application Migration**.
2. Select the **Compartment** in which to create the source.
3. Click **Sources**.
4. Click **Create Source**.
5. Enter a **Name** and **Description** for the source.
6. For **Source Type**, select **Oracle Cloud Infrastructure - Classic**.
7. For **Account**, enter the name of your Oracle Cloud Infrastructure Classic account.
8. Select the Oracle Cloud Infrastructure Classic **Region** in which you created your source Oracle Java Cloud Service instance.
9. Enter credentials for this Oracle Cloud Infrastructure Classic account that have access to Oracle Java Cloud Service.
10. Click **Create**.

For more information, see Manage Sources in the Oracle Cloud Infrastructure documentation.

# Create a Migration

Use Application Migration to connect to the WebLogic Server domain for the Oracle Java Cloud Service instance within your source.

1. From the Oracle Cloud Infrastructure console, navigate to **Application Migration**.
2. Select the **Compartment** that contains your source.
3. Click **Sources**, and then select your source.
4. Click **Actions** ⋮ for the Oracle Java Cloud Service instance that you want to migrate, and then click **Create Migration**.
5. Enter a **Name** and **Description** for the migration.
6. Enter the WebLogic Server administrator credentials for the Oracle Java Cloud Service instance.
7. Set the **Target Instance Type** to Oracle Java Cloud Service.
8. Click **Create**.

For more information, see Manage Migrations in the Oracle Cloud Infrastructure documentation.

# Configure and Run a Migration

Use Application Migration to create the target Oracle Java Cloud Service instance in Oracle Cloud Infrastructure. Specify a network, databases, and other details.

1. From the Oracle Cloud Infrastructure console, navigate to **Application Migration**.
2. Select the **Compartment** that contains your migration.
3. Click **Migrations**, and then select your migration.
4. Click **Configure**.
5. In the Configure Service section, click **Configure**.
6. Select the **Availability Domain** in which you want to create the target instance.
7. Select the **Virtual Cloud Network** and **Subnet** in which you want to create the target instance.
8. Enter the **System Database Administrator Password** for the new Oracle Cloud Infrastructure Database.
9. Upload or paste the public **SSH Key** to use for the target instance and database.
10. Enter the WebLogic Server administrator credentials for the target instance.
11. Click **Configure** to return to the Configure Migration page.
12. If your source instance includes custom Java Database Connectivity (JDBC) data sources, then provide the location and password of the new application databases in Oracle Cloud Infrastructure.
    a. In the Configure Application section, click **Configure**.
    b. For each data source, enter the **Connection String** 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 | `jdbc:oracle:thin:@//<db_hostname>-scan.<db_domain>:<db_port>/<pdb_name>.<db_domain>` |
| 12c | Bare Metal | `jdbc:oracle:thin:@//<db_hostname>.<db_domain>:<db_port>/<pdb_name>.<db_domain>` |
| 11g | VM | `jdbc:oracle:thin:@//<db_hostname>-scan.<db_domain>:<db_port>/<db_unique_name>.<db_domain>` |
| 11g | Bare Metal | `jdbc:oracle:thin:@//<db_hostname>.<db_domain>:<db_port>/<db_unique_name>.<db_domain>` |

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

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

c. For each data source, set the **Data Source Password**.

d. Click **Configure** to return to the Configure Migration page.

13. Click **Save and Run**.

14. When prompted for confirmation, click **Start**.

Use Application Migration to monitor the progress of your work request.

If the work request indicates that the import step of the migration failed, you can get additional information by connecting to the first node in the target instance. Access the log files found at `/u01/weblogic-deploy` and `/u01/jcsmig`. After correcting the problem, you can run the migration again.

For more information, see Manage Migrations in the Oracle Cloud Infrastructure documentation.

# Copy Supporting Files to the Target Instance

Identify and copy any files to your target Oracle Java Cloud Service instance that are not automatically managed by Application Migration.

Application Migration migrates the following types of files from your source instance's domain configuration 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 Application Migration, including files that are located outside the `DOMAIN_HOME` directory. You must manually copy these files to the target instance.

1. Use SSH to connect to the Administration Server node in your *source* instance.

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

2. Switch to the `oracle` user.

   ```
   sudo su - oracle
   ```

3. Identify any supporting files that need to be copied to the target instance.
4. 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.

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

   Example:

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

6. Disconnect from the node.
7. 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 .
   ```

8. 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>:/tmp
   ```

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

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

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

Example:

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

11. Switch to the `oracle` user.

```
sudo su - oracle
```

12. 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
```

13. Disconnect from the node.

# 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 instance.

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* instance.

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

4. Recreate users and groups.
   a. From both consoles, click **WebLogic Domain**, select **Security**, and then select **Security Realms**.
   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. For **Name**, select **Includes**, and then enter the text `common/wsm-agent-core`.

h. Click **Search System Security Grants**  .

i. 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` |
| `system/publiccacerts` | `<name> [jdk]`, `idcs_root_ca` |
| `opss/trustservice_ts` | `trustservice`, `cloudca` |
| `opss/trustservice_ks` | `trustservice` |
| `owsm/keystore` | `oauth_<identity_domain>_trust_sign`, `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.
    a. From both consoles, click **WebLogic Domain**, select **Web Services**, and then select **WSM Policies**.
    b. 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)

c. From both consoles, click **WebLogic Domain**, select **Web Services**, and then select **WSM Policy Sets**.

d. Identify any policy sets in the source instance, and then recreate these policy sets in the target instance.

# 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 instance.

The source and target instances 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.

Your source and target instances must be in the same identity domain.

1. Access the Oracle Identity Cloud Service console.
2. Click the navigation drawer  , and then select **Applications**.
3. Export the administrator roles for your *source* instance.
   a. Click the security application for your source instance, `JaaS_<source_instance_name>`.
   b. Click **Application Roles**.
   c. Click **Export**, and then select **Export All**.
   d. When prompted for confirmation, click **Export Application Roles**.
   e. Click the job ID.

      If a job ID link is not displayed, click the navigation drawer  , select **Jobs**, and then click the job.
   f. After the export job has finished, click **Download**. Save the file `AppRoleExport_<id>.csv` to your computer.
   g. Click the navigation drawer  , and then select **Applications**.
4. Export the web tier policy for your *source* instance.
   a. Click the security application for your source instance, `JaaS_<source_instance_name>`.
   b. Click **Web Tier Policy**.
   c. Click **Export**. Save the file `CloudGatePolicy.txt` to your computer.
   d. From the navigation links at the top of the page, click **Applications**.
5. Import the administrator roles to your *target* instance.
   a. Click the security application for your target instance, `JaaS_<target_instance_name>`.

b. Click **Application Roles**.

c. Click **Import**.

d. Select the file `AppRoleExport_<id>.csv` on your computer, and then click **Import**.

e. Click the job ID link.

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

f. Verify that the import job has finished.

   You can ignore any error in the job with these messages:

   - "Grant already exists" - the same user or group is already assigned to the role
   - "Missing required attribute(s): Grantee" - no users or groups are assigned to the role

g. Click the navigation drawer  , and then select **Applications**.

6. Import the web tier policy to your *target* instance.

   a. Click the security application for your target instance, `JaaS_<target_instance_name>`.

   b. Click **Web Tier Policy**.

   c. Click **Import**. Select the file `CloudGatePolicy.txt` on your computer.

      You might need to refresh your web browser in order to view the resources that you imported from the policy file.

   d. Click **Validate**.

   e. Verify that the web tier policy validation was successful.

   f. From the navigation links at the top of the page, click **Applications**.

7. Assign any custom applications to the roles for your target instance.

   a. Click the security application for your source instance, `JaaS_<source_instance_name>`.

   b. Click **Application Roles**.

   c. If there is an **Applications Assigned** link for the first role, click this link, and then record the application names for this role.

   d. Repeat the previous step for all remaining roles in this application.

   e. From the navigation links at the top of the page, click **Applications**.

   f. Click the security application for your target instance, `JaaS_<target_instance_name>`.

   g. Click **Application Roles**.

   h. Click **Menu**  for the first role, and then select **Assign Applications**.

   i. Select the same applications that are assigned to this role in the source instance, and then click **OK**.

   j. Repeat the previous step for all remaining roles in this application.

k. From the navigation links at the top of the page, click **Applications**.

8. Configure any custom security settings for your target instance.

   a. Click the security application for your source instance, `JaaS_<source_instance_name>`.

      The Details page is displayed.

   b. From a different browser window or tab, view the security application for your target instance, `JaaS_<target_instance_name>`.

   c. Compare the Details page of the source and target applications, and then update the target application as necessary. Click **Save**.

   d. From both browser windows or tabs, click **Configuration**.

   e. Expand **Resources** on the Configuration page.

   f. Compare the Resources of the source and target applications, and then update the target application as necessary. Click **Save**.

      Ignore the resources named `OCMSApp` and `LBAAS`. You do not need to create these resources in the target instance.

   g. Expand **Client Configuration** on the Configuration page.

   h. Compare the Client Configuration of the source and target applications, and then update the target application as necessary.

6

# Migrate an Instance to Oracle Java Cloud Service Using Classic Tools

Use the Oracle Cloud Infrastructure Classic Java Migration Tool to migrate your Oracle WebLogic Server domain resources and applications from your existing Oracle Java Cloud Service instance in Oracle Cloud Infrastructure Classic to a new Oracle Java Cloud Service instance in Oracle Cloud Infrastructure.

Before you begin the migration process, see Prepare to Migrate Oracle Java Cloud Service to Oracle Cloud Infrastructure.

When you migrate an Oracle Java Cloud Service instance, the following terms are used:

- *Source*: The Oracle Java Cloud Service instance in Oracle Cloud Infrastructure Classic.
- *Target*: The Oracle Java Cloud Service instance in Oracle Cloud Infrastructure.

**Topics:**



## Get Information About the Target Environment

Collect the information required for the migration tools to connect to the target Oracle Cloud Infrastructure environment.

1. Access the Oracle Cloud Infrastructure Console.
2. From the menu, choose **Administration** and then choose **Tenancy Details**.

3. Record the tenancy's OCID and Home Region.
4. From the menu, choose **Identity** and then **Users**.
5. Click your user name.
6. Record the user's OCID. Under API Keys, record the Fingerprint.

   You will also need the corresponding PEM key file.
7. From the menu, choose **Identity** and then **Compartments**.
8. Record the OCID of the compartment where you want to create the instance.
9. From the menu, choose **Identity** and then **Federation**.
10. From the Oracle Identity Cloud Service Console URL, identify and record the identity domain ID, which has the format `idcs-<guid>`.
11. From the menu, choose **Networking** and then **Virtual Cloud Networks** (VCN).
12. Select the **Compartment** where you want to create the instance.
13. Click the VCN in which you want to create this instance.
14. Under Subnets, click the subnet in which you want to create this instance.
15. Record the subnet's OCID. If it is not a regional subnet, then also record the subnet's Availability Domain.
16. If you selected a regional subnet, then choose an availability domain for the target instance.
    a. Access the Oracle Java Cloud Service Console.
    b. Click **Create Instance**.
    c. Select your **Region**.
    d. From **Availability Domain**, record the name of the availability domain in which you want to create this instance.
    e. Click **Cancel**.

The following table shows sample values for each input.

| Name | Sample Value |
|---|---|
| Tenancy OCID | `ocid1.tenancy.oc1..aaaaaaaaju6k54i7...` |
| User OCID | `ocid1.user.oc1..aaaaaaaahvtv5qo...` |
| User API Key Fingerprint | `81:45:aa:...` |
| Compartment OCID | `ocid1.compartment.oc1..aaaaaaaaz...` |

| Name | Sample Value |
| --- | --- |
| Region | `us-ashburn-1` |
| Availability Domain | `kWVD:US-ASHBURN-AD-3` |
| Subnet OCID | `ocid1.subnet.oc1.iad.aaaaaaaarz7..` `.` |
| Identity Domain ID | `idcs-9bd53...` |

# Launch the Migration Controller Instance in the Source Environment

In your Oracle Cloud Infrastructure Compute Classic account, create the source controller (Control-S) instance, which includes Oracle Cloud Infrastructure Classic Java Migration Tool.

The Control-S compute instance must be created in the same identity domain and site as the source Oracle Java Cloud Service instance that you want to migrate.

The Control-S compute instance and associated storage volumes are by default billed at the applicable rates for your account. However, you can rename these resources so that the name includes `/oraclemigration` as a container. Resources created in this `/oraclemigration` container aren't billed to your account.

1. Access the Oracle Cloud Infrastructure Compute Classic Console.
2. Click **Create Instance**.
3. Click **Show All Images**.
4. Select the image `OL_7.5_UEKR4_x86_64_MIGRATION`, which is found under **Oracle Images**.
5. Click **Next**.
6. Select a **Shape** with a sufficient number of OCPUs for the migration task.
7. Click **Next**.
8. Enter a **Name**, or use the default instance name.
9. Select an existing public **SSH Key** or add a new one. You'll use the corresponding private key to connect to the Control-S instance.
10. Click **Next**.
11. Verify that **Shared Network** is selected.
12. For **Public IP Address**, select **Persistent Public IP Reservation**.

13. For **Security Lists**, verify that the `default` security list is selected, which allows SSH inbound traffic.

    Also ensure that security rules are in place to allow SSH outbound, SMB inbound, and HTTPS outbound traffic.

14. If you want to migrate instances that have an interface on an IP network, then configure the network interfaces of the Control-S instance on the relevant IP networks as well, so that the Control-S instance can access the source instances that you want to migrate.

15. Complete the creation of the compute instance.

    Wait until its status is **Running**.

16. Optional: Move the Control-S instance and storage volumes into the `/oraclemigration` container.

    Alternatively, if you create the Control-S instance using the API, CLI, or Terraform, you can specify `/oraclemigration` in the resource names as part of the instance parameters.

    a. Click the **Orchestrations** tab.

    b. Locate the relevant orchestration for your compute instance, and from the menu, select **Suspend**.

    c. After the orchestration status changes to **Suspended,** from the menu, select **Update**.

    d. From the Instance section, click the menu and select **Edit JSON**.

    e. In the Edit Orchestration Object JSON window, locate the instance name. This is usually displayed within the `template` section, after `networking`.

    ```
    "name": "/Compute-Identity_Domain/User/Instance",
    ```

    Modify the instance name to include the `/oraclemigration` container. For example:

    ```
    "name": "/Compute-ExampleDomain/user@example.com/oraclemigration/
    MyControlS",
    ```

    f. Click **Update**.

    g. From the Orchestrations page, go to the relevant orchestration and from the menu, select **Terminate**.

    h. After the orchestration status changes to **Stopped**, from the menu, select **Update**.

    i. From the Storage Volume section, go to the relevant storage volume, click the menu and select **Edit JSON**.

    j. In the Edit Orchestration Object JSON window, locate the storage volume name in the `template` section:

    ```
    "name": "/Identity_Domain/User/Volume",
    ```

Modify the instance name to include the `/oraclemigration` container. For example:

```
"name": "/Compute-ExampleDomain/user@example.com/oraclemigration/
MyControlS_storage",
```

k. Click **Update**.

l. Repeat these steps for any other storage volume in this orchestration that you want to move to the `/oraclemigration` container.

m. From the Orchestrations page, go to the relevant orchestration and from the menu, select **Start**.

17. Use an SSH client and your private key to log into the Control-S compute instance as the `opc` user.

# Update the Secret File

All of the tools required for the migration are already installed on the Control-S instance, but additional configuration is required to provide details about the source and target environments.

A single Control-S instance can migrate resources only from a single Oracle Cloud Infrastructure Classic account and site, and only to a single Oracle Cloud Infrastructure tenancy, region, and availability domain.

1. From the Control-S compute instance, copy `/home/opc/ansible/secret.yml.sample` to `/home/opc/ansible/secret.yml`.
2. Edit `/home/opc/ansible/secret.yml`.
3. Update the following Oracle Cloud Infrastructure parameters.
   - `compartment_id` is the OCID of the compartment where you want to create the target instance.
   - `user_id` is the OCID of the Oracle Cloud Infrastructure user.
   - `fingerprint` is the API key fingerprint of the user.
   - `tenancy_id` is the OCID of the Oracle Cloud Infrastructure tenancy.
   - `region` is the Oracle Cloud Infrastructure region where you want to create the target instance.
   - `availability_domain` is the availability domain where you want to create the target instance.
   - `subnet_id` is the OCID of the subnet where you want to create the instance.

   For example:

```
# OCI info
compartment_id: ocid1.compartment.oc1..aaaaaaaa...
user_id: ocid1.user.oc1..aaaaaaaa...
fingerprint: a0:a0:a0:a0:a0...
tenancy_id: ocid1.tenancy.oc1..aaaaaaaa...
region: us-ashburn-1
availability_domain: kWVD:US-ASHBURN-AD-3
```

```
...
subnet_id: ocid1.subnet.oc1.iad.aaaaaaaa...
```

4. Modify permissions on this file to restrict access.

   ```
   chmod 600 /home/opc/ansible/secret.yml
   ```

5. Apply the configuration to the system.

   ```
   opcmigrate migrate instance service setup
   ```

   This command creates the required files `/home/opc/.opc/profiles/default` and `/home/opc/.oci/config`.

6. Copy the Oracle Cloud Infrastructure user's PEM key file to the Control-S instance. Name the file `/home/opc/.oci/oci_api_key.pem`.
7. Modify permissions on the Oracle Cloud Infrastructure key file to restrict access.

   ```
   chmod 600 /home/opc/.oci/oci_api_key.pem
   ```

8. Copy the public and private SSH key files required for accessing your source Oracle Java Cloud Service instance to the Control-S instance.
9. Modify permissions on the Oracle Java Cloud Service key files to restrict access.

   For example:

   ```
   chmod 600 /home/opc/jcskey.pub
   chmod 600 /home/opc/jcskey
   ```

# Update the Default Profile File

The Oracle Cloud Infrastructure Classic Java Migration Tool connects to your source environment using information that you provide in a profile file.

The information you provide in the profile file includes the user name or identity for each service in the source environment, as well as the service end point and region. If you want to run the tool in multiple regions or tenancies, you can create separate profile files for each region and tenancy.

You also provide connectivity details for each Oracle Java Cloud Service instance that you want to migrate. If you include the WebLogic Server administrator credentials for a service instance, Oracle Cloud Infrastructure Classic Java Migration Tool also migrates any Oracle Fusion Middleware security resources (custom users, groups, roles, policies, or credential maps) to the target domain.

1. Access the Oracle Cloud Infrastructure Compute Classic Console.
2. Click the **Site** select box.
3. Record the REST Endpoint.
4. Identify and record your Oracle Java Cloud Service REST Endpoint.

   See Send Requests in *REST API for Oracle Java Cloud Service*.

5. Access the Oracle Java Cloud Service Console.

6. Click the source instance.
7. Click **Instance Details**  , and then record the Region in which the source instance was created.
8. From the Control-S compute instance, create a properties file with the WebLogic Server administrator user name and password of your source instance.

   ```
   admin_user=your_username
   admin_password=your_password
   ```

   This step is required only if the source domain includes custom users, groups, roles, policies or credential maps.
9. From the Control-S compute instance, edit the file `/home/opc/.opc/profiles/default`.
10. In the `compute` section, update the `endpoint` and `user` parameters. Enter the name of a user with access to Oracle Cloud Infrastructure Compute Classic.

    ```
    "compute": {
      "endpoint": "Compute_Endpoint",
      "user": "/Compute-Identity_Domain/User_Name"
      ...
    ```

    For example:

    ```
    "compute": {
      "endpoint": "compute.uscom-central-1.oraclecloud.com"
      "user": "/Compute-ExampleDomain/user@example.com",
      ...
    ```

11. **Optional:** Enter the location of a file that contains your Oracle Cloud Infrastructure Compute Classic password.

    For example:

    ```
    "compute": {
      "endpoint": "compute.uscom-central-1.oraclecloud.com"
      "user": "/Compute-ExampleDomain/user@example.com",
      "password-file": "/home/opc/.opc/password-file",
      ...
    ```

    If you don't specify a password file for a service, you'll be prompted to provide the password when you run the tool.
12. If not already present, add the `paas` section to the file.

    ```
    {
      ...
      "compute": {
      ...
      },
      "paas": {
    ```

```
    "user": "User_Name",
    "identity_id": "Identity_Domain_Id",
    "endpoint": "PaaS_Endpoint",
    "region": "Source_Region"
  }
}
```

For example:

```
{
  ...
  "compute": {
  ...
  },
  "paas": {
    "user": "user@example.com",
    "identity_id": "idcs-0000abcd0000defg0000hijk0000lmno",
    "endpoint": "psm.us.oraclecloud.com",
    "region": "uscom-central-1"
  }
}
```

13. Add the `jcs` section to the file. Specify the locations of the public and private SSH key files for your source Oracle Java Cloud Service instance.

```
{
  ...
  "paas": {
  ...
  },
  "jcs": {
    "Instance_Name": {
      "ssh_private_key": "Private_Key_File",
      "ssh_public_key": "Public_Key_File"
    }
  }
}
```

For example:

```
{
  ...
  "paas": {
  ...
  },
  "jcs": {
    "MyJavaInstance": {
      "ssh_private_key": "/home/opc/jcskey",
      "ssh_public_key": "/home/opc/jcskey.pub"
    }
  }
}
```

14. In the `jcs` section, specify the location of the properties file that contains the WebLogic Server credentials for the source instance.

    For example:

```
...
  "jcs": {
    "MyJavaInstance": {
      "ssh_private_key": "/home/opc/jcskey",
      "ssh_public_key": "/home/opc/jcskey.pub",
      "wls_admin_properties": "/home/opc/wls_admin_properties"
    }
```

    This step is required only if the source domain includes custom users, groups, roles, policies or credential maps.

# Discover Resources in Your Source Environment

To discover all Oracle Cloud Infrastructure Classic resources in the services for which you've provided credentials, log in to the Control-S instance and run the following command.

```
opcmigrate discover
```

When prompted, enter the passwords for the user names that you specified in the default profile.

For example:

```
opcmigrate discover
Compute Classic Password [/Compute-ExampleDomain/user@example.com]:
INFO Authenticating with OCI Classic Compute API
INFO Compute Endpoint: https://compute.uscom-central-1.oraclecloud.com
INFO Discovering resources for "ExampleDomain".
WARNING Load Balancer Classic credentials not configured in profile
PaaS Services Password [user@example.com]:
WARNING Object Storage Classic credentials not configured in profile
INFO Discovering containers: ['/Compute-ExampleDomain']
INFO Getting Account Resources for /Compute-ExampleDomain
INFO Getting Network Resources for /Compute-ExampleDomain
INFO Getting Network Resources for /oracle/public
INFO Getting Instance Resources for /Compute-ExampleDomain
INFO Getting Instance Resources for /oracle/public
INFO Getting Instances for /Compute-ExampleDomain
INFO Getting PaaS Resources for uscom-central-1
INFO Storing discovered resources to 'resources-default.json'
```

# List Your Oracle Java Cloud Service Instances

To list the Oracle Java Cloud Service instances in the source environment, log in to the Control-S instance and run the following command.

```
opcmigrate migrate jcs list
```

This command uses the output generated by the `discover` command to identify and list the available Oracle Java Cloud Service instances.

For example:

```
opcmigrate migrate jcs list
INFO Loaded resources from 'resources-default.json' ...
Java Cloud Service Instances

Name                 Version              State
Description
-------------------- -------------------- --------------------
------------------------------------------------------------
MyJavaInstance       11gR1                READY               My first
instance
AnotherInstance      12cR3                READY               My
second instance
```

# Export Your Source Instance Configuration

To create an archive of the source Oracle Java Cloud Service instance using the WebLogic Server Deploy Tooling, log in to the Control-S instance and run the following command.

```
opcmigrate migrate jcs export -s <instance_name>
```

This command creates the following files:

- `<instance_name>-<date>-<timestamp>.tgz`: An archive of the source instance, which includes the applications that are on the source instance as well as the domain configuration metadata. This archive is uploaded to Oracle Cloud Infrastructure Object Storage.
- `<instance_name>-<date>-<timestamp>.json`: You edit this file to specify the required passwords for the target domain, as well as to specify any configuration parameters that will be different on the target instance.

For example:

```
opcmigrate migrate jcs export -s MyJavaInstance
INFO Loaded resources from 'resources-default.json' ...
INFO Exporting JCS service 'MyJavaInstance'
INFO Installing Oracle WebLogic Server Deploy Tooling on 203.0.113.13
INFO Create temporary directory on controller
INFO Download WebLogic Deploy Tooling to controller
```

```
INFO Upload and Extract WebLogic Deploy Tooling archive to remote host
INFO Remove temporary directory from controller
INFO Exporting WebLogic Domain at 203.0.113.13
INFO Create temporary directory on remote host
INFO Run WebLogic Deploy Tooling discoverDomain.sh command
INFO Download discovered domain files to controller
INFO Remove temp directory from remote
INFO Generating WebLogic config template
'MyJavaInstance-20190722-18:50:35.json'
INFO Creating instance archive 'MyJavaInstance-20190722-18:50:35.tgz'
INFO Uploading artifacts to object storage
INFO JCS service 'MyJavaInstance' export complete
```

By default, this command uses the `resources-default.json` file in the local directory. You can use the `--file` option to specify a resources file with a different name or in a different directory.

# Perform Prerequisite Tasks for Oracle Java Cloud Service

Before you create an Oracle Java Cloud Service instance in an Oracle Cloud Infrastructure region, 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
   - A storage bucket and user authentication token for backups (optional)
   - Policies that allow Oracle Java Cloud Service to access the resources in your compartment

   See Prerequisites for Oracle Platform Services in the Oracle Cloud Infrastructure documentation.
2. Create a database in Oracle Cloud Infrastructure Database if one doesn't already exist.

   Oracle Java Cloud Service will provision the required infrastructure schema to this database. See Managing Bare Metal and Virtual Machine DB Systems in the Oracle Cloud Infrastructure documentation.

# Create the Target Instance on Oracle Cloud Infrastructure

Create a new Oracle Java Cloud Service instance in an Oracle Cloud Infrastructure region. This instance must have the same topology and configuration as the source instance in Oracle Cloud Infrastructure Classic.

You can use a Terraform configuration generated by Oracle Cloud Infrastructure Classic Java Migration Tool to create the target instance. Alternatively, you can create the target instance by using the Oracle Java Cloud Service Console, CLI, API, or any other supported method.

Because your source and target instances are located in the same Oracle Cloud account (identity domain), they cannot have identical instance names.

The domain, server, and cluster names in a service instance are derived from the first eight characters of the instance name. For example, the following instance names are different, but result in identical domain, server, and cluster names in Oracle WebLogic Server:

- `MyJCSInstance`
- `MyJCSInstanceOCI`

## Create the Target Instance Using the Console

You can use the Oracle Java Cloud Service Console to create the service instance on Oracle Cloud Infrastructure.

1. From the Oracle Java Cloud Service Console, click **Create Instance**.
2. Enter an **Instance Name**.
3. Select an Oracle Cloud Infrastructure **Region**, **Availability Domain**, and **Subnet**.
4. For **Service Level** and **Software Edition**, select the same values as the source instance.
5. For **Software Release**, select the same major version (`X.Y`) as the source instance.

   For example, `12.2.1.2` and `12.2.1.3` are the same major version of Oracle WebLogic Server.
6. Click **Next**.
7. Click **Advanced**.
8. For **WebLogic Clusters**, create the same number of clusters as the source instance. Also set the cluster names and server counts to the same values as the source instance.

   For example, if the source instance has a single cluster named `cluster1` with a server count of `3`, then the target instance must have the same configuration.
9. For the **Compute Shape** of your WebLogic Cluster, select an Oracle Cloud Infrastructure shape that most closely matches the number of Oracle Compute Units (OCPUs) and the amount of memory that are available in the Oracle Cloud Infrastructure Classic shape in your source instance.

   See Select Oracle Cloud Infrastructure Shapes.
10. If your source instance uses Oracle Identity Cloud Service for authentication, then select **Enable Authentication Using Identity Cloud Service**.
11. For **SSH Public Key**, upload an existing key or generate a new one.
12. For **Local Administrative User Name** and **Password**, enter the same Oracle WebLogic Server administrator credentials as your source instance.
13. If your source instance includes an Oracle Traffic Director load balancer, then select a **Compute Shape** for the load balancer that most closely matches the Oracle Cloud Infrastructure Classic shape in the source instance.
14. If your source instance includes an Oracle Coherence data grid cluster, then select the same **Cluster Size** and **Managed Servers Per Node** as the data grid cluster

in the source instance. Also select a **Compute Shape** for the data grid cluster that most closely matches the Oracle Cloud Infrastructure Classic shape.

15. For **Database Type**, select **Oracle Cloud Infrastructure Database**.
16. Select the **Compartment Name** where your Oracle Cloud Infrastructure Database resides.
17. For **Database Instance Name**, select the Oracle Cloud Infrastructure Database that you created for the Oracle Java Cloud Service infrastructure schema.

    Also enter a value for **PDB Name** if applicable.
18. Enter the **Password** for your database system administrator.
19. Complete the instance creation wizard.

For more information about using the console, see Create an Oracle Java Cloud Service Instance Attached to a Public Subnet on Oracle Cloud in *Administering Oracle Java Cloud Service*.

## Create the Target Instance Using Terraform

You can use Oracle Cloud Infrastructure Classic Java Migration Tool and Terraform to create the target instance.

1. Log in to the Control-S instance and run the following command:

   ```
   opcmigrate migrate jcs create -s <instance_name> -o jcs_instance.tf
   ```

   In the Terraform configuration generated by this command, the appropriate compute shape is selected automatically. The tool also creates a unique name for the target instance based on the name of the source instance.
2. Create a file named `terraform.tfvars` file. Add the following parameters:
   - `user_ocid` is the OCID of the Oracle Cloud Infrastructure user.
   - `fingerprint` is the API key fingerprint of the user.
   - `private_key_path` is the path to the API PEM key file.
   - `region` is the Oracle Cloud Infrastructure region where you want to create the instance.
   - `tenancy_ocid` is the OCID of the Oracle Cloud Infrastructure tenancy.
   - `compartment_id` is the OCID of the compartment where you want to create the instance.
   - `subnet_id` is the OCID of the subnet where you want to create the instance.
   - `identity_domain` is the Identity Service Id.
   - `identity_user` is the name of the federated user that has the Java Administrator role.
   - `identity_password` is the password of the federated user.
   - `weblogic_admin_username` is the user name of the WebLogic Server administrator.
   - `weblogic_admin_password` is the password of the WebLogic Server administrator.

- `weblogic_database_username` is the user name of the database administrator.
- `weblogic_database_password` is the password of the database administrator.

For example:

```
user_ocid="ocid1.user.oc1..aaa..."
fingerprint="81:45..."
private_key_path="/home/opc/oci_api_key.pem"
region="us-ashburn-1"
tenancy_ocid="ocid1.tenancy.oc1..aaa..."
compartment_id="ocid1.compartment.oc1..aaa..."
subnet_id="ocid1.subnet.oc1.iad.aaa..."
identity_domain="idcs-9bd53..."
identity_user="user@example.com"
identity_password="Your_Password"
weblogic_admin_username="weblogic"
weblogic_admin_password="Your_Password"
weblogic_database_username="sys"
weblogic_database_password="Your_Password"
```

3. Add the `weblogic_database_connect_string` parameter to the file. Enter the database connection string used by the instance to connect to the database and to provision the infrastructure schema.

   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 | *//<db_hostname>*-scan.*<db_domain>*:*<db_port>*/*<pdb_name>*.*<db_domain>* |
| 12c | Bare Metal | *//<db_hostname>*.*<db_domain>*:*<db_port>*/*<pdb_name>*.*<db_domain>* |
| 11g | VM | *//<db_hostname>*-scan.*<db_domain>*:*<db_port>*/*<db_unique_name>*.*<db_domain>* |
| 11g | Bare Metal | *//<db_hostname>*.*<db_domain>*:*<db_port>*/*<db_unique_name>*.*<db_domain>* |

   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.

   For example:

```
weblogic_database_connect_string="//myinfradb-
scan.mydbsubnet.myvcn.oraclevcn.com:1521/
pdb1.mydbsubnet.myvcn.oraclevcn.com"
```

4. By default, the Terraform configuration creates your instance in the first availability domain in the specified region. If you want to change the availability domain where the instance is created, edit the Terraform configuration file.

a. Edit the file `jcs_instance.tf`.

b. Edit the local variable, `ad`.

```
locals {
  ad                          = 0
  subnet_availability_domain = ...
}
```

The value `0` is used to specify AD-1. Change the value to `1` to specify AD-2, or `2` to specify AD-3.

5. To initialize Terraform, run this command.

```
terraform init
```

6. To create the Oracle Java Cloud Service instance, run this command.

```
terraform apply
```

Enter `yes` when prompted.

For example:

```
terraform apply
...
Terraform will perform the following actions:
  + oraclepaas_java_service_instance.MyJavaInstance_9de860
      id:                                          <computed>
      availability_domain:                         "QnsC:US-ASHBURN-
AD-1"
...
Plan: 1 to add, 0 to change, 0 to destroy.
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.
  Enter a value: yes
oraclepaas_java_service_instance.MyJavaInstance_9de860: Creating...
...
oraclepaas_java_service_instance.MyJavaInstance_9de860: Still
creating... (23m0s elapsed)
oraclepaas_java_service_instance.MyJavaInstance_9de860: Creation
complete after 23m8s (ID: MyJavaInstanceOCI)
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

From the output, identify the name of the new instance.

7. Use the Oracle Java Cloud Service Console to identify the IP address of the first node in the target instance.

For more information about the Oracle Cloud Platform provider for Terraform, see oraclepaas_java_service_instance in the HashiCorp Terraform documentation.

# Migrate Oracle Fusion Middleware Security Resources

If you customized the Oracle WebLogic Server security providers in your source Oracle Java Cloud Service instance, then you must apply the same changes in the target service instance.

If you specified the WebLogic Server administrator credentials for your source instance in the default profile, the Oracle Cloud Infrastructure Classic Java Migration Tool automatically migrates the following Oracle Fusion Middleware security resources from the source domain to the target domain:

- Users
- Groups
- Roles
- Policies
- Keystores
- Credential maps
- Audit policies
- Web Services Manager (WSM) policies

The tool does not automatically update the security providers in the target instance.

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* instance.

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

4. From both consoles, click **WebLogic Domain**, select **Security**, and then select **Security Provider Configuration**.
5. 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.

# 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 instance.

The source and target instances 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.

Your source and target instances must be in the same identity domain.

1. Access the Oracle Identity Cloud Service console.

2. Click the navigation drawer  , and then select **Applications**.
3. Export the administrator roles for your *source* instance.
   a. Click the security application for your source instance, `JaaS_<source_instance_name>`.
   b. Click **Application Roles**.
   c. Click **Export**, and then select **Export All**.
   d. When prompted for confirmation, click **Export Application Roles**.
   e. Click the job ID.

      If a job ID link is not displayed, click the navigation drawer  , select **Jobs**, and then click the job.
   f. After the export job has finished, click **Download**. Save the file `AppRoleExport_<id>.csv` to your computer.
   g. Click the navigation drawer  , and then select **Applications**.
4. Export the web tier policy for your *source* instance.
   a. Click the security application for your source instance, `JaaS_<source_instance_name>`.
   b. Click **Web Tier Policy**.
   c. Click **Export**. Save the file `CloudGatePolicy.txt` to your computer.
   d. From the navigation links at the top of the page, click **Applications**.
5. Import the administrator roles to your *target* instance.
   a. Click the security application for your target instance, `JaaS_<target_instance_name>`.
   b. Click **Application Roles**.
   c. Click **Import**.
   d. Select the file `AppRoleExport_<id>.csv` on your computer, and then click **Import**.
   e. Click the job ID link.

      If a job ID link is not displayed, click the navigation drawer  , select **Jobs**, and then click the job.
   f. Verify that the import job has finished.

      You can ignore any error in the job with these messages:

      - "Grant already exists" - the same user or group is already assigned to the role
      - "Missing required attribute(s): Grantee" - no users or groups are assigned to the role

   g. Click the navigation drawer  , and then select **Applications**.
6. Import the web tier policy to your *target* instance.
   a. Click the security application for your target instance, `JaaS_<target_instance_name>`.

b. Click **Web Tier Policy**.

c. Click **Import**. Select the file `CloudGatePolicy.txt` on your computer.

   You might need to refresh your web browser in order to view the resources that you imported from the policy file.

d. Click **Validate**.

e. Verify that the web tier policy validation was successful.

f. From the navigation links at the top of the page, click **Applications**.

7. Assign any custom applications to the roles for your target instance.

   a. Click the security application for your source instance, `JaaS_<source_instance_name>`.

   b. Click **Application Roles**.

   c. If there is an **Applications Assigned** link for the first role, click this link, and then record the application names for this role.

   d. Repeat the previous step for all remaining roles in this application.

   e. From the navigation links at the top of the page, click **Applications**.

   f. Click the security application for your target instance, `JaaS_<target_instance_name>`.

   g. Click **Application Roles**.

   h. Click **Menu**  for the first role, and then select **Assign Applications**.

   i. Select the same applications that are assigned to this role in the source instance, and then click **OK**.

   j. Repeat the previous step for all remaining roles in this application.

   k. From the navigation links at the top of the page, click **Applications**.

8. Configure any custom security settings for your target instance.

   a. Click the security application for your source instance, `JaaS_<source_instance_name>`.

      The Details page is displayed.

   b. From a different browser window or tab, view the security application for your target instance, `JaaS_<target_instance_name>`.

   c. Compare the Details page of the source and target applications, and then update the target application as necessary. Click **Save**.

   d. From both browser windows or tabs, click **Configuration**.

   e. Expand **Resources** on the Configuration page.

   f. Compare the Resources of the source and target applications, and then update the target application as necessary. Click **Save**.

      Ignore the resources named `OCMSApp` and `LBAAS`. You do not need to create these resources in the target instance.

   g. Expand **Client Configuration** on the Configuration page.

   h. Compare the Client Configuration of the source and target applications, and then update the target application as necessary.

# Edit the Target Configuration File

The `export` command creates a file that contains parameters for updating the target WebLogic Server domain. Specify JDBC URLs and passwords, SSL keystore passwords, and other details for the target instance.

For security purposes, Oracle WebLogic Server Deploy Tooling excludes the values of all passwords during domain discovery.

1. On the Control-S instance, edit the file <*instance_name*>-<*date-time-stamp*>.json.

   Refer to the output from the `export` command to determine the specific file name.

2. Update the following attributes.

   - `JCSServiceName` - The name of the target Oracle Java Cloud Service instance
   - `JCSAdminIPAddress` - The IP address of the first node in the target instance (running the Administration Server)
   - `WeblogicAdminUser` - The user name for the WebLogic Server domain administrator on the target instance
   - `WeblogicAdminPassword` - The password for the WebLogic Server domain administrator on the target instance

3. If your source instance includes a load balancer, then update the `FrontendHost` attribute for each cluster in the `Cluster` node. Enter the public IP address of the load balancer in your target instance.

   Example:

   ```
   "topology": {
     "Cluster": {
       "cluster": {
         "FrontendHost": "<target_LB_IP>"
       }
     ...
   }
   ```

4. If you configured any custom startup arguments for a server in your source instance, then update the `AdditionalServerStartArguments` attribute for each server in the `Server` node.

   Set the value of `AdditionalServerStartArguments` to the custom arguments only. Example:

   ```
   "topology": {
     ...
     "Server": {
       ...
       "server_1": {
         ...
         "AdditionalServerStartArguments": "-Dmy.custom.arg=true"
       }
   ```

```
  ...
}
```

5. If the servers in your source instance are configured to use custom identity and trust keystore files, then update the file with the keystore passwords.

   - `CustomIdentityKeyStorePassPhrase`
   - `CustomTrustKeyStorePassPhrase`
   - `ServerPrivateKeyPassPhrase`

   Example:

```
"topology": {
  ...
  "Server": {
    "server_1": {
      ...
      "CustomIdentityKeyStorePassPhrase": "<your_password>",
      "CustomTrustKeyStorePassPhrase": "<your_password>",
      "ServerPrivateKeyPassPhrase": "<your_password>"
    }
  ...
}
```

6. If your source instance includes custom Java Database Connectivity (JDBC) data sources, then provide the location and password of the new application databases in Oracle Cloud Infrastructure.

   a. For each data source in the `JDBCSystemResource` node, update the `password` attribute.

      Example:

```
"resources": {
  "JDBCSystemResource": {
    "MyDataSource1": {
      ...
      "password": "<your_password>"
    }
  }
  ...
```

   b. For each data source, find the `url` attribute and specify 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 | `jdbc:oracle:thin:@//<db_hostname>-scan.<db_domain>:<db_port>/<pdb_name>.<db_domain>` |

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

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

```
"resources": {
  "JDBCSystemResource": {
    "MyDataSource1": {
      "url": "jdbc:oracle:thin:@//
myappdb-scan.mydbsubnet.myvcn.oraclevcn.com:1521/
pdb1.mydbsubnet.myvcn.oraclevcn.com",
      ...
    }
  }
  ...
```

7. 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. For each provider in the `ForeignJNDIProvider` node, update the `password` attribute.

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

```
"resources": {
  ...
  "ForeignJNDIProvider": {
    "MyJNDIProvider1": {
      "url": "t3://myjndiserver.example.com:9073",
      "password": "<your_password>"
    }
  }
  ...
```

   b. For each foreign server in the `ForeignJMSServer` node, update the `password` attributes.

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

Example:

```
"resources": {
  ...
  "ForeignJMSServer": {
    "MyForeignJMS1": {
      "url": "t3://myjms.example.com:9073",
      "ForeignConnectionFactory": {
        "MyForeignJMS1Factory": {
          "password": "<your_password>"
        }
      }
    }
  }
  ...
```

c. For each bridge destination in the `JMSBridgeDestination` node, update the `password` attribute.

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

```
"resources": {
  ...
  "JMSBridgeDestination": {
    "MyBridgeDest1": {
      "url": "t3://myjms.example.com:9073",
      "password": "<your_password>"
    }
  }
  ...
```

d. For each SAF context in the `SAFLoginContext` node, update the `password` attribute.

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

```
"resources": {
  ...
  "SAFLoginContext": {
    "MySAF1": {
      "url": "t3://myjms.example.com:9073",
      "password": "<your_password>"
    }
  }
  ...
```

8. If your source instance includes any JavaMail sessions, then update the passwords for each mail session in the `MailSession` node.

Example:

```
"resources": {
  ...
  "MailSession": {
    "MyMailSession1": {
      "password": "<your_password>",
      "properties": {
        "mail.smtp.password": "<your_password>",
        "mail.imap.password": "<your_password>"
      }
    }
  }
  ...
```

9. If your source instance includes any custom WebLogic Diagnostic Framework (WLDF) REST notification endpoints, then provide the passwords for each WLDF resource in the `WLDFSystemResource` node.

   Also update the `url` attribute if the destination server is different than the server in the source environment.
   Example:

```
"resources": {
  ...
  "WLDFSystemResource": {
    "MyModule": {
      "RestNotification": {
        "MyNotification1": {
          "url": "http://myserver.example.com:9073/notify",
          "password": "<your_password>"
        }
      }
    }
  }
  ...
```

# 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 Server Deploy Tooling.

Oracle WebLogic Server Deploy Tooling 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 Oracle WebLogic Server Deploy Tooling, including files that are located outside the `DOMAIN_HOME` directory. You must manually copy these files to the target instance.

1. Use SSH to connect to the Administration Server node in your *source* instance.

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

2. Switch to the `oracle` user.

   ```
   sudo su - oracle
   ```

3. Identify any supporting files that need to be copied to the target instance.
4. 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.

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

   Example:

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

6. Disconnect from the node.
7. 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 .
   ```

8. 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>:/tmp
   ```

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

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

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

    Example:

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

11. Switch to the `oracle` user.

    ```
    sudo su - oracle
    ```

12. 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
    ```

13. Disconnect from the node.

# Import the Target Instance Configuration

After the target Oracle Java Cloud Service instance is running on Oracle Cloud Infrastructure, use Oracle Cloud Infrastructure Classic Java Migration Tool to import the WebLogic Server domain configuration that you previously exported from the source instance.

The tool downloads the archive from Oracle Cloud Infrastructure Object Storage, updates your target instance's domain configuration with the values provided in the JSON file, deploys your applications, and then restarts the domain.

Oracle Cloud Infrastructure Classic Java Migration Tool creates a bastion compute instance in Oracle Cloud Infrastructure in order for the Control-S instance to access your target service instance. After updating the target service instance, the tool deletes the temporary bastion compute instance.

Run the following command.

```
opcmigrate migrate jcs import -c <instance_name>-<date>-
<timestamp>.json -a <instance_name>-<date>-<timestamp>.tgz
```

For example:

```
opcmigrate migrate jcs import -c MyJavaInstance-20190813.json -a
MyJavaInstance-20190813.tgz
INFO Loaded resources from 'resources-default.json' ...
INFO Importing JCS service 'MyJavaInstance'
...
INFO Provisioning bastion host
INFO Initializing Terraform
INFO Applying Terraform
INFO Bastion host provisioned: 203.0.113.50
INFO Waiting for SSH to become available
INFO Installing Oracle WebLogic Server Deploy Tooling on 203.0.113.14
INFO Create temporary directory on controller
INFO Download WebLogic Deploy Tooling to controller
INFO Upload and Extract WebLogic Deploy Tooling archive to remote host
INFO Remove temporary directory from controller
INFO Uploading WebLogic Domain related files to 203.0.113.14
...
INFO Run WebLogic Deploy Tooling validateModel.sh command
INFO Stopping WebLogic domain
```

```
INFO Stop WebLogic Domain
INFO Updating WebLogic domain
INFO Set target host full path names for uploaded files
INFO Update WebLogic Domain
INFO Starting WebLogic domain
INFO Start WebLogic domain
INFO Tearing down bastion host...
INFO Tearing down Terraform
INFO JCS service 'MyJavaInstance' import complete
```

7

# Complete the Post-Migration Tasks

After successfully migrating your Oracle Java Cloud Service instances from Oracle Cloud Infrastructure Classic to Oracle Cloud Infrastructure, test your applications thoroughly, and then perform cleanup and other optional configuration tasks.

**Topics:**



## 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.

Be sure to thoroughly run all application test cases. Also 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.

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.

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 Corente 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 Classic

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 Cloud Infrastructure Classic.

Delete these Oracle Cloud Infrastructure Classic 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 Classic.
    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 Cloud Infrastructure Classic 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 Classic 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**.