

Oracle® Cloud

Migrating Oracle Java Cloud Service

Instances to Oracle WebLogic Server for OCI

Using tar



F56557-11
March 2024



Oracle Cloud Migrating Oracle Java Cloud Service Instances to Oracle WebLogic Server for OCI Using tar, F56557-11

Copyright © 2022, 2023, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

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

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

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

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

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

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

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

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

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

Contents

Preface

Documentation Accessibility	v
Diversity and Inclusion	v

1 About Migrating

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

2 Prepare to Migrate

About Downtime Requirements	2-1
Select Oracle WebLogic Server for OCI Shapes	2-1
Configure Security Rules for the Network	2-2
Recommended simplified network configurations	2-2
Prepare for Migration	2-6

3 Migrate an Instance

Create an Oracle WebLogic Server for OCI Instance	3-1
Stop all WebLogic Server Processes	3-2
Replace Domain and Binaries	3-3
Update the Domain Content	3-4
Start the WebLogic Server Processes	3-5
IDCS Migration	3-6

4 Post-Migration

Upgrade a Domain	4-2
Stop all WebLogic Server Processes	4-2

Perform Prerequisite Tasks	4-2
Set up VNC Server	4-3
Perform Readiness Check	4-5
Upgrade Infrastructure Schemas	4-6
Reconfigure the Domain	4-8
Upgrade the Domain	4-8
Restart Servers	4-9
Post Upgrade	4-9
Roll Back an Upgrade	4-10

Preface

Migrating Oracle Java Cloud Service Instances to Oracle WebLogic Server for OCI Using tar explains how to migrate your existing Oracle Java Cloud Service instances to Oracle WebLogic Server for Oracle Cloud Infrastructure (Oracle WebLogic Server for OCI), if your Java Cloud Service instance uses an Oracle Cloud Infrastructure Database (not classic DBCS) or Oracle Autonomous Database.

Topics:

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

Documentation Accessibility

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

Access to Oracle Support

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

Diversity and Inclusion

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

1

About Migrating

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

Note:

Your Oracle Java Cloud Service instances may be running on Oracle Cloud Infrastructure (OCI) or Oracle Cloud Infrastructure Classic (OCI-C). To determine if your instances are on OCI or OCI-C, see the Oracle Support note [2984700.1](#).

If all the following prerequisites are met, then proceed with the migration process described in this guide:

- The Oracle Java Cloud Service uses Oracle Cloud Infrastructure Database (not classic DBCS) or Autonomous database.
- The version of the Oracle Java Cloud Service is 12.2.1.x or later.
- If you plan to use the same VCN for Oracle Java Cloud Service and Oracle WebLogic Server for OCI.

Note:

- If you plan to migrate Oracle Java Cloud Service to Oracle WebLogic Server for OCI, where Oracle Java Cloud Service uses a classic DBCS, then follow the instructions in the Migrating JCS on OCI-C to Oracle WebLogic Server for OCI Using tar guide. Although your instance is on OCI, the instructions for performing the DBCS migration and corresponding data source updates are applicable to this case.
- If you plan to migrate Oracle Java Cloud Service in OCI Classic to Oracle WebLogic Server for OCI, follow the instructions in the Migrating JCS on OCI-C to Oracle WebLogic Server for OCI Using tar guide.

Topics:

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

Why Migrate to Oracle WebLogic Server for OCI

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

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

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

- Supports WebLogic Server major release versions after 12.2.1.4.0.
- Enables better efficient control of Oracle Cloud Infrastructure resources.
- Offers flexibility to add or remove patches.
- Allows managing the operating system or domain after provisioning, without any restrictions.
- Allows choosing any supported method to backup, scale, or patch a domain.
- Allows selecting the compartment where you want to create any of your resources.
- Allows full customization of the Oracle-managed load balancer.
- Supports cloning instances.
- Includes optional autoscaling feature.
- Supports a documented disaster recovery process. See Oracle WebLogic Server for Oracle Cloud Infrastructure Disaster Recovery.
- Allows WebLogic Server domains without Java Required Files (JRF).
- Integrates with multiple Oracle Cloud Infrastructure services to monitor, trace, deploy, scale and autoscale your WebLogic domain, and other Oracle Cloud Infrastructure services to manage and patch your WebLogic domain resources.
- Offers the option to pay per OCPU/Hour for Oracle WebLogic Server for OCI images with the entitlement to install any Oracle WebLogic Server version.

Migration Scope

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

Oracle does *not* currently support the migration of Oracle Java Cloud Service instances running Oracle WebLogic Server 11g or 12.1.3 releases.

If all the following prerequisites are met, then proceed with the migration process described in this guide:

- The Oracle Java Cloud Service uses Oracle Cloud Infrastructure Database (not classic DBCS) or Autonomous database.

- The version of the Oracle Java Cloud Service is 12.2.1.x or later.
- If you plan to use the same VCN for Oracle Java Cloud Service and Oracle WebLogic Server for OCI.



Note:

If you plan to migrate Oracle Java Cloud Service to Oracle WebLogic Server for OCI, where Oracle Java Cloud Service uses a classic DBCS, then follow the instructions in the Migrating JCS on OCI-C to Oracle WebLogic Server for OCI Using tar guide. Although your instance is on OCI, the instructions for performing the DBCS migration and corresponding data source updates are applicable to this case.

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

About Oracle WebLogic Server for OCI

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

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

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

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

Compare Oracle Java Cloud Service to Oracle WebLogic Server for OCI

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

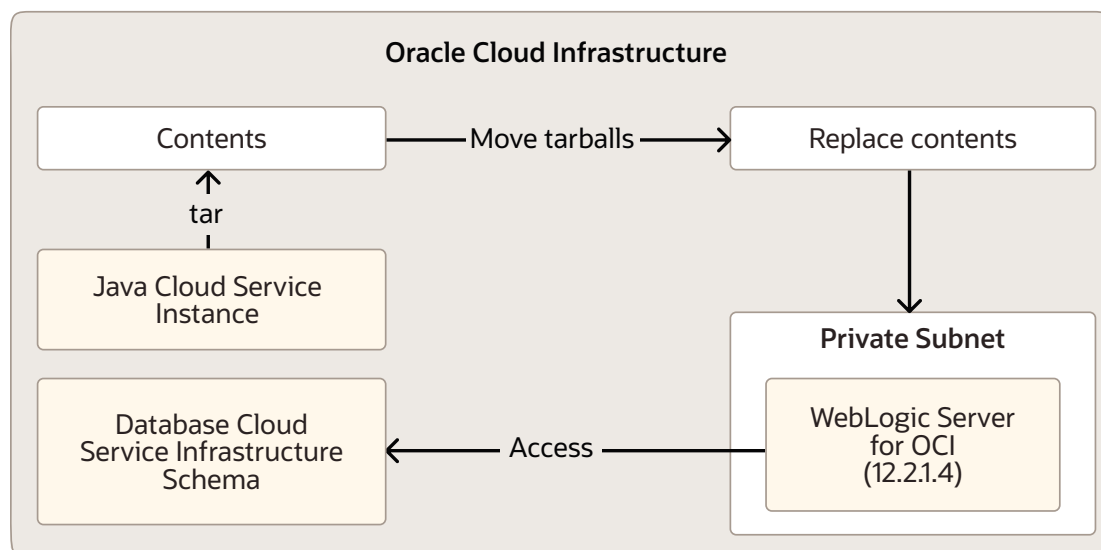
The following table compares the functionality of Oracle Java Cloud Service to Oracle WebLogic Server for OCI

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

Migrate Using tar

Use the procedures in this guide to migrate service instances to Oracle WebLogic Server for OCI by using tar.

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

Figure 1-1 Migration Topology

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

1. Create a Oracle WebLogic Server for OCI 12.2.1.4 instance in a private subnet.
2. Verify that the Oracle WebLogic Server for OCI compute instances can access the port to the OCI database used by Oracle Java Cloud Service.
3. Verify that the Oracle WebLogic Server for OCI compute instances can access the ssh port to the Oracle Java Cloud Service instances.
4. Tar the following contents from the Oracle Java Cloud Service instance:
 - /u01/app/oracle/middleware
 - /u01/data/domains
 - /u01/jdk
5. Move the tarballs to the Oracle WebLogic Server for OCI compute instances.
6. Replace the required contents on the Oracle WebLogic Server for OCI compute instances.
7. Replace the host names and database connection strings in the domain with values that point to the Oracle WebLogic Server for OCI host names and OCI database, respectively.
8. If you use Oracle Identity Cloud Service (IDCS):
 - Replace the client id and secret in the IDCS Authentication Provider.
 - Set up the protected paths in the IDCS application gateway.

2

Prepare to Migrate

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

Topics:

- [About Downtime Requirements](#)
- [Select Oracle WebLogic Server for OCI Shapes](#)
- [Configure Security Rules for the Network](#)
- [Prepare for Migration](#)

About Downtime Requirements

Two JRF instances must not be running at the same time. You can limit the downtime by not stopping the WebLogic servers on your Oracle Java Cloud Service instance until you are ready to start the WebLogic servers on the Oracle WebLogic Server for OCI instance.

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

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

Select Oracle WebLogic Server for OCI Shapes

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

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

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

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

Configure Security Rules for the Network

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

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

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

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

See Security Lists in the Oracle Cloud Infrastructure documentation.

Oracle recommends that you create a private subnet for Oracle WebLogic Server for OCI in the VCN used by Oracle Java Cloud Service.

Set up a private subnet that can be used for creating your Oracle WebLogic Server for OCI instance in the same VCN as the Oracle Java Cloud Service instance. This allows you to quickly set up a connection to the same database that Oracle Java Cloud Service uses and allow ssh connectivity between the Oracle WebLogic Server for OCI instance VMs and the Oracle Java Cloud Service VMs, so files can be transferred between them.

Recommended simplified network configurations

The following configurations show a Oracle Java Cloud Service set up with a public subnet since this is the most common configuration.



Note:

Public subnets are allowed for Oracle WebLogic Server for OCI instances, but not recommended.

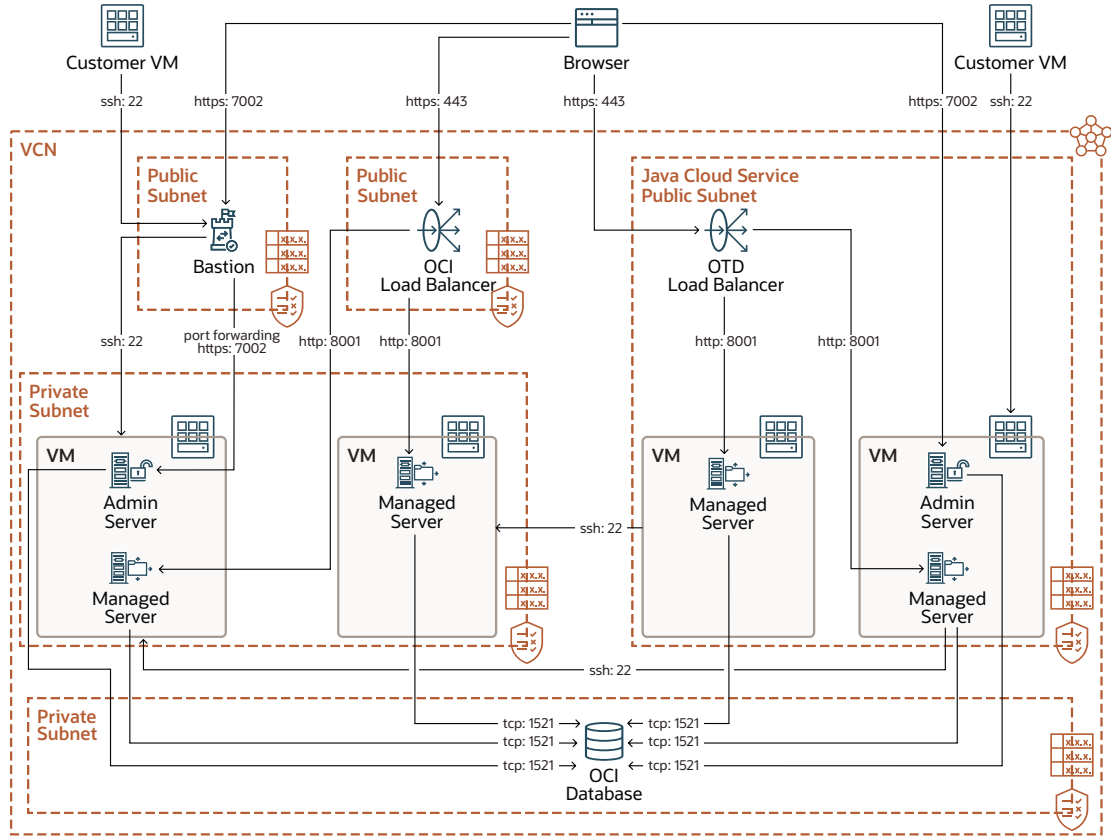
- [Oracle Java Cloud Service with OTD](#)
- [Oracle Java Cloud Service with IDCS](#)

Oracle Java Cloud Service with OTD

There is no Oracle Traffic Director (OTD) option in Oracle WebLogic Server for OCI. Instead, the OCI load balancer must be used to route traffic to the WebLogic managed servers.

The following diagram illustrates a Oracle Java Cloud Service instance using OTD and the Oracle WebLogic Server for OCI instance recommended to replace it. This illustration represents the state of both instances during the migration process, prior to deleting the Oracle Java Cloud Service instance.

Figure 2-1 Oracle Java Cloud Service with OTD



The private subnet for the OCI database and the public subnet for JCS must already exist. The Oracle WebLogic Server for OCI private subnet, OCI load balancer public subnet, and bastion host public subnet must be created.

Use the following tables to configure the security lists for the these three subnets:

Table 2-1 Oracle WebLogic Server for OCI Private Subnet

Rule Type	Source CIDR and Protocol	Default Destination Port	Description
Stateful Ingress	Bastion public subnet, TCP	22	SSH access
Stateful Ingress	Bastion public subnet, TCP	7002	Administration server SSL port
Stateful Ingress	OCI load balancer public subnet, TCP	8001	Managed server port

Table 2-1 (Cont.) Oracle WebLogic Server for OCI Private Subnet

Rule Type	Source CIDR and Protocol	Default Destination Port	Description
Stateful Ingress	Oracle Java Cloud Service public subnet, TCP	22	SSH access, so domain and binaries can be copied over
Stateful Ingress	Oracle WebLogic Server for OCI private subnet (self), TCP	22	SSH access between the compute instances of Oracle WebLogic Server for OCI
Stateful Ingress	Oracle WebLogic Server for OCI private subnet (self), TCP	9071	Used for provisioning and scaling
Stateful Ingress	Oracle WebLogic Server for OCI private subnet (self), TCP	9073	Used for WebLogic health check
Stateful Ingress	Oracle WebLogic Server for OCI private subnet (self), TCP	5556	Used for accessing node manager

Table 2-2 OCI Load Balancer Public Subnet

Rule Type	Source CIDR and Protocol	Default Destination Port	Description
Stateful Ingress	Internet (0.0.0.0/0), TCP	443	SSL access

Table 2-3 Bastion Public Subnet

Rule Type	Source CIDR and Protocol	Default Destination Port	Description
Stateful Ingress	Hosts with access to bastion, TCP	22	SSH access, port forwarding

Use the following table to add access for the Oracle WebLogic Server for OCI private subnet to the OCI database public subnet:

Table 2-4 Access to OCI Database Public Subnet

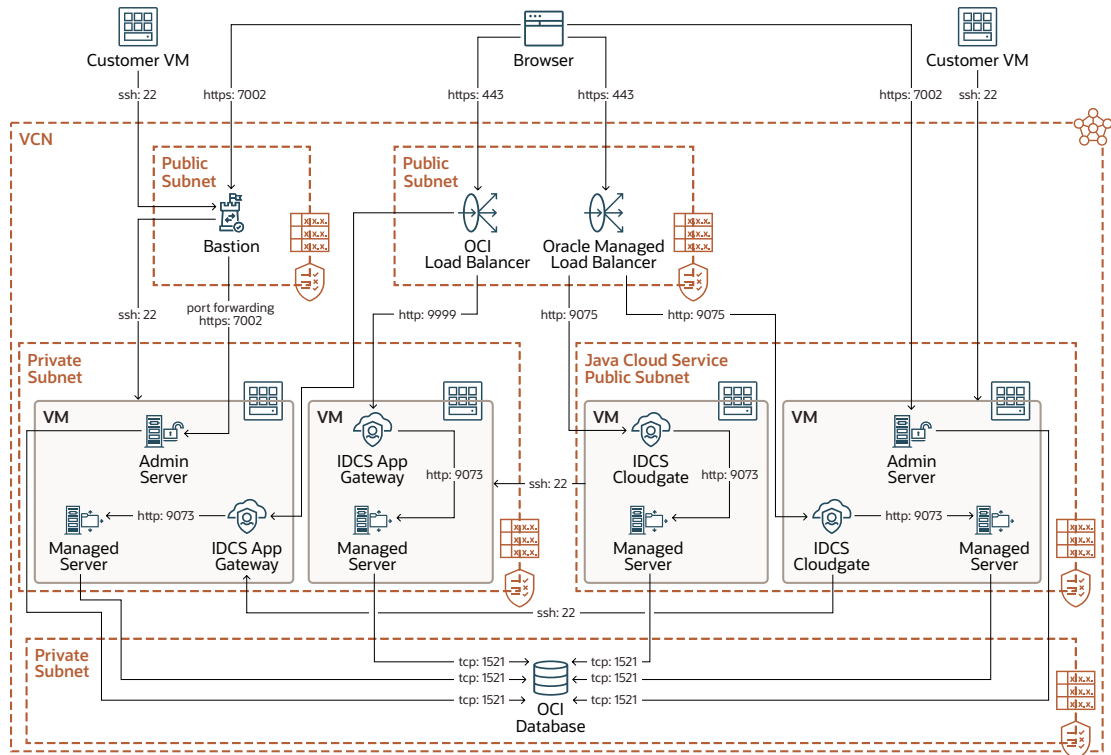
Rule Type	Source CIDR and Protocol	Default Destination Port	Description
Stateful Ingress	WLS for OCI private subnet, TCP	1521	DB access

Oracle Java Cloud Service with IDCS

The following diagram illustrates a Oracle Java Cloud Service instance using IDCS and the Oracle WebLogic Server for OCI instance recommended to replace it. This

illustration represents the state of both instances during the migration process, prior to deleting the Oracle Java Cloud Service instance.

Figure 2-2 Oracle Java Cloud Service with IDCS



The private subnet for the OCI database, the public subnet for the OCI load balancer, and the public subnet for JCS must already exist. The Oracle WebLogic Server for OCI private subnet and bastion host public subnet must be created.

Use the following tables to configure the security lists for the these three subnets:

Table 2-5 Oracle WebLogic Server for OCI Private Subnet

Rule Type	Source CIDR and Protocol	Default Destination Port	Description
Stateful Ingress	Bastion public subnet, TCP	22	SSH access
Stateful Ingress	Bastion public subnet, TCP	7002	Administration server SSL port
Stateful Ingress	OCI load balancer public subnet, TCP	9999	IDCS app gateway port
Stateful Ingress	Oracle Java Cloud Service public subnet, TCP	22	SSH access, so domain and binaries can be copied over

Table 2-5 (Cont.) Oracle WebLogic Server for OCI Private Subnet

Rule Type	Source CIDR and Protocol	Default Destination Port	Description
Stateful Ingress	Oracle WebLogic Server for OCI private subnet (self), TCP	22	SSH access between the compute instances of Oracle WebLogic Server for OCI
Stateful Ingress	Oracle WebLogic Server for OCI private subnet (self), TCP	9071	Used for provisioning and scaling
Stateful Ingress	Oracle WebLogic Server for OCI private subnet (self), TCP	9073	IDCS app gateway for accessing the managed server
Stateful Ingress	Oracle WebLogic Server for OCI private subnet (self), TCP	5556	Used for accessing node manager

Table 2-6 Bastion Public Subnet

Rule Type	Source CIDR and Protocol	Default Destination Port	Description
Stateful Ingress	Hosts with access to bastion, TCP	22	SSH access, port forwarding

Use the following table to add access for the Oracle WebLogic Server for OCI private subnet to the OCI database public subnet:

Table 2-7 Access to OCI Database Public Subnet

Rule Type	Source CIDR and Protocol	Default Destination Port	Description
Stateful Ingress	WLS for OCI private subnet, TCP	1521	DB access

Prepare for Migration

Before you start the migration process, you need to complete the prerequisites steps in *Before You Begin with Oracle WebLogic Server for OCI*.

When performing the pre-requisites steps, ensure to complete the following recommendation: Set Oracle Java Cloud Service WebLogic administration user name and password as the OCI secret.

This ensures that your password with the WebLogic Server global "Admin" role in the Oracle Java Cloud Service instance is stored as an OCI secret to support running the required start or stop scripts. See *Default Global Roles*.

1. Make a note of the user name and password for a user with the WebLogic Server administration role in the Oracle Java Cloud Service.

2. When you get to the prerequisite step Create Secrets for Passwords, add the password to the OCI secrets.

3

Migrate an Instance

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



Note:

Migration is supported from Oracle Java Cloud Service 12.2.1.x release to any of the existing versions of Oracle WebLogic Server for OCI.

In this procedure,

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

Topics:

- [Create an Oracle WebLogic Server for OCI Instance](#)
- [Stop all WebLogic Server Processes](#)
- [Replace Domain and Binaries](#)
- [Update the Domain Content](#)
- [Start the WebLogic Server Processes](#)
- [IDCS Migration](#)

Create an Oracle WebLogic Server for OCI Instance

From OCI Marketplace create either a **Oracle WebLogic Suite UCM** or **Oracle WebLogic Server Enterprise Edition UCM** instance corresponding to the WebLogic edition you used in your Oracle Java Cloud Service. Irrespective of the WebLogic version used in your Oracle Java Cloud Service instance, create a Oracle WebLogic Server for OCI 12.2.1.4 instance.

Create an Oracle WebLogic Server for OCI instance by following the steps documented in *Create a Domain in Using Oracle WebLogic Server for OCI*, along with the following customizations:

- Select the **WLS Instance Advanced Configuration** option and then change the **WebLogic Managed Server External Port** and **WebLogic Managed Server External SSL Port** to match those used in Oracle Java Cloud Service. By default, Oracle Java Cloud Service uses ports 8001 and 8002, respectively.
- If your Oracle Java Cloud Service instance used OTD, then select **Provision Load Balancer**.

- If your Oracle Java Cloud Service instance used Oracle Identity Cloud Service (IDCS), then select **Provision Load Balancer** and **Enable Authentication Using Identity Cloud Service**.
- Although all Oracle Java Cloud Services use JRF, you *do not* need to select the **Provision with JRF** option, since the domain migrated from the Oracle Java Cloud Service instance points to the existing JRF database schemas, not the new schemas created with the JRF option.
- Oracle WebLogic Server for OCI creates all block volumes at 50 GB. If you have block volumes for the Oracle Java Cloud Service domain or common volumes with larger sizes, set the same sizes for the corresponding Oracle WebLogic Server for OCI data and middleware volumes prior to migration.
- Oracle WebLogic Server for OCI creates only a data and block volume. If you created additional volumes for your Oracle Java Cloud Service instance, add those volumes using the OCI console after completing the provisioning process. See Adding a Block Volume in the Oracle Cloud Infrastructure documentation.

Stop all WebLogic Server Processes

Stop all the WebLogic server processes on both the Oracle WebLogic Server for OCI and Oracle Java Cloud Service instances.

In **Oracle WebLogic Server for OCI**:

1. As an opc user, log in to the each non-administration VM in the instance and run the following command:

```
sudo su - oracle
/opt/scripts/restart_domain.sh -o stop
# run jps to confirm that there are no more processes are running.
If there are run kill -9 on each.
jps
```

2. As an opc user, log in to the Administration VM in the instance and run the following command:

```
sudo su - oracle
/opt/scripts/restart_domain.sh -o stop
# run jps to confirm that there are no more processes are running.
If there are run kill -9 on each.
jps
```

In **Oracle Java Cloud Service**:

Using the Oracle WebLogic server administration console stop all the WebLogic servers on the Oracle Java Cloud Service instance.

Oracle recommends that you stop the WebLogic servers in your Oracle Java Cloud Service prior to beginning the migration process. Use the Oracle WebLogic Server Administration console to stop all the WebLogic servers on the Oracle Java Cloud Service instance. However, if you have minimal downtime requirements you can wait to stop the WebLogic servers in your Oracle Java Cloud Service at the point in which you start the WebLogic Server processes for the migrated Oracle WebLogic Server for OCI.

Replace Domain and Binaries

Replace the domain and binaries in the Oracle WebLogic Server for OCI instance.

Complete the following steps:

1. Access the VM running the WebLogic administration server in the Oracle Java Cloud Service instance and run the following commands to preserve the WebLogic domain and binaries:

```
sudo su - oracle
cd /u01/data/domains
tar cvf /tmp/domain.tar .
cd /u01/app/
tar cvf /tmp/mw.tar .
cd /u01/jdk/
tar cvf /tmp/jdk.tar .
```

2. Log in as an `oracle` user and ssh to each additional VM in the Oracle Java Cloud Service instance, and preserve the WebLogic domain contents. You do not have to save the binaries from each VM, as the binaries are the same on each VM.

```
cd /u01/data/domains
tar cvf /tmp/domain.tar .
```

3. Set up a ssh access from the Oracle Java Cloud Service instance to the Oracle WebLogic Server for OCI instance:

- a. On the source instance generate a public key from the private key that is in the `oracle` user's home:

```
# Must be run as oracle user. If you are the opc user, run 'sudo su -
oracle' first.
ssh-keygen -y -f ~/.ssh/id_rsa
```

- b. Save the output.
- c. Access the Oracle WebLogic Server for OCI instance as the `opc` user.
- d. On each VM in the Oracle WebLogic Server for OCI instance append the output to the `authorized_keys` for the `oracle` user:

```
sudo su - oracle
vi ~/.ssh/authorized_keys
# add the public key to this file.
```

- e. On each VM in the Oracle WebLogic Server for OCI instance run the following command to move the contents out of the current domain, middleware home, and JDK home:

```
sudo su - oracle
mkdir /tmp/domain_bak
mv /u01/data/domains/* /tmp/domain_bak/
```

```
mkdir /tmp/mw_bak
mv /u01/app/* /tmp/mw_bak/
```

4. Copy the `domain.tar`, `mw.tar`, and `jdk.tar` files to the Oracle WebLogic Server for OCI instance:
 - a. Access the Oracle Java Cloud Service instance as the `opc` user and run `scp` to copy the tar files from each VM to the corresponding VM on the Oracle WebLogic Server for OCI instance. In the following example there is a single VM on the Oracle WebLogic Server for OCI instance with an IP address `10.1.1.1`:

```
sudo su - oracle
scp -o UserKnownHostsFile=/dev/null -o
StrictHostKeyChecking=no /tmp/domain.tar 10.1.1.1:/tmp/
scp -o UserKnownHostsFile=/dev/null -o
StrictHostKeyChecking=no /tmp/mw.tar 10.1.1.1:/tmp/
scp -o UserKnownHostsFile=/dev/null -o
StrictHostKeyChecking=no /tmp/jdk.tar 10.1.1.1:/tmp/
```

- b. Ssh to the Oracle WebLogic Server for OCI instance as the `opc` user and on each VM run the following to `untar` the source instance domain:

```
sudo su - oracle
tar xvf /tmp/domain.tar -C /u01/data/domains/
tar xvf /tmp/mw.tar -C /u01/app/
mkdir /u01/app/oracle/jdk
tar xvf /tmp/jdk.tar -C /u01/app/oracle/jdk/
```



Note:

Oracle WebLogic Server for OCI creates only a data and block volume. If you created additional volumes for your Oracle Java Cloud Service instance, then add a new volume and transfer the data using the above steps in this topic.

Update the Domain Content

To update the domain content by using the cloning scripts, complete the following steps:

1. As a `oracle` user, run the following commands on each VM in the Oracle WebLogic Server for OCI instance:

```
sudo su - oracle
python3 /opt/scripts/cloning/update_service_config.py metadata 1
```

This ensures that the metadata used by restart and scaling scripts on Oracle WebLogic Server for OCI has the correct information.

 **Note:**

Oracle Java Cloud Service limits the domain objects to 8 characters. However, Oracle WebLogic Server for OCI does not have this limitation.

2. Change the host names in the domain configuration files to the host name of the VMs in the Oracle WebLogic Server for OCI instance.
As a `oracle` user, run the following command on each VM in the Oracle WebLogic Server for OCI instance:

```
sudo su - oracle
python3 /opt/scripts/cloning/update_service_config.py hostname 1
```

3. If the Demo Identity certificates have the incorrect host name, then, as a `oracle` user, run the following command on each VM in the Oracle WebLogic Server for OCI instance

```
sudo su - oracle
/opt/scripts/cloning/generate_demo_certs.sh
```

4. Ensure that there is a marker file present so Compute instance reboot restarts the servers:

```
touch /u01/data/domains/<jcs_domain_name>/provCompletedMarker
```

where `<jcs domain name>` can be found by running:

```
ls /u01/data/domains/ | grep _domain
```

Start the WebLogic Server Processes

If you did not stop the WebLogic servers on the Oracle Java Cloud Service instance, then stop the WebLogic servers on the Oracle Java Cloud Service instance now. Oracle does not support running two WebLogic domains using JRF simultaneously. Use the WebLogic Server Administrative console to stop all the servers.

Complete the following steps on the migrated Oracle WebLogic Server for OCI instance:

1. As an `opc` user, log in to the Administration VM in the instance and run the following command:

```
sudo su - oracle
/opt/scripts/restart_domain.sh -o start
# run jps to confirm that the processes are running.
jps
```

2. As an `opc` user, log in to the each non-administration VM in the instance and run the following command:

```
sudo su - oracle
/opt/scripts/restart_domain.sh -o start
```

```
# run jps to confirm that the processes are running.  
jps
```

IDCS Migration

The Oracle Identity Cloud Service (IDCS) confidential application created by Oracle WebLogic Server for OCI must replace the IDCS application created for Oracle Java Cloud Service.

Complete the following steps:

1. Update the client and secret in the IDCS authentication provider.
 - a. From the IDCS console, expand the **Navigation Drawer**, and then click **Applications**.
 - b. Click the enterprise application associated with your domain.
The name of the application is:
`<stack>_confidential_idcs_app_<timestamp>`

Example:
`myweblogic_confidential_idcs_app_2019-08-01T01:02:01.123456.`
 - c. Click the **Configuration** tab.
 - d. Retrieve the `Client ID` and `Client Secret` values.
 - e. Use the WebLogic Server Administration Console to update the client and secret in the IDCS authentication provider:
 - i. Log in to WebLogic Server Administration Console.
See [Access the WebLogic Console in Using Oracle WebLogic Server for OCI](#).
 - ii. Navigate to **security realms**.
 - iii. Select the realm. By default this is `myrealm`.
 - iv. Click on the **Providers** tab.
 - v. Select **IDCSIntegrator**.
 - vi. Click on the **Provider Specific** tab.
 - vii. Click **Lock and Edit**.
 - viii. Change the **Client Id** and **Client Secret** to the values retrieved from the IDCS confidential application.
2. Add Cloudgate role.

The IDCS confidential application created for the Oracle WebLogic Server for OCI instance sets up a more restrictive default role, *Authenticator Client*, than is set up in Oracle Java Cloud Service. To keep the same level of permissions so that the Enterprise Manager Fusion Middleware Control console, update the client application to add the *Cloud Gate* role. See [Integrate OPSS User and Group APIs with Identity Cloud Service](#).

3. Restart the WebLogic server processes on the Oracle WebLogic Server for OCI instance:

- a. As an opc user, log in to the Administration VM in the instance and run the following command:

```
sudo su - oracle
/opt/scripts/restart_domain.sh
# run jps to confirm that the processes are running.
jps
```

- b. As an opc user, log in to the each non-administration VM in the instance and run the following command:

```
sudo su - oracle
/opt/scripts/restart_domain.sh
# run jps to confirm that the processes are running.
jps
```

4. Move any protected paths.

If you set up any server context paths to be protected by IDCS, then you must migrate them. See [Migrate Oracle Identity Cloud Service Roles and Policies](#).

4

Post-Migration

After successfully migrating your Oracle Java Cloud Service instances to Oracle WebLogic Server for OCI, perform the following post-migration tasks:

- If you were using custom SSL certificates on Oracle Traffic Director (OTD), then set up SSL certificates on the OCI Load Balancer.
- If you use a hosting provider to manage DNS make sure to reset the `CNAME` records at your hosting provider to point to the new IP addresses of the OCI load balancer and WebLogic VMs.
- Verify that all the deployed applications are functioning as expected. If you have enabled Oracle WebLogic Server Administration Console communication, verify that you can access the WebLogic Server Administration Console. If your instance includes custom data sources that access your application databases, you can test database connectivity directly from the WebLogic Server Administration Console. Select a data source, click the **Monitoring** tab, and then click the **Testing** tab.
- It is recommended that you stop your Oracle Java Cloud Service instance using the Oracle Java Cloud Service console. With the instance stopped you will not be charged for the Oracle Java Cloud Service instance. If you discover any issues with the Oracle WebLogic Server for OCI set up you can always switch back to the Oracle Java Cloud Service by starting it up again. See [Before You Begin with Oracle WebLogic Server for OCI](#).
- After monitoring the migrated Oracle WebLogic Server for OCI for a few days or weeks, if your applications are all running as expected, you should delete your Oracle Java Cloud Service instance. See [Delete an Oracle Java Cloud Service Instance](#).

! Important:

When deleting the Oracle Java Cloud Service instance, do not delete the JRF schemas since they are now being used in the Oracle WebLogic Server for OCI instance. To prevent deletion, enter an invalid password for **Database Administrator User Password** and select **Force Delete**. This causes the termination process to fail to delete the schemas, but proceed with removing the Oracle Java Cloud Service instance.

- If you migrated your Oracle Java Cloud Service instance to Oracle WebLogic Server for OCI, and the WebLogic Server version that you migrated is 12.2.1.2 or 12.2.1.3, you must upgrade the WebLogic Server versions post-migration.
 - If the WebLogic Server version is 12.2.1.3, see [Upgrade a Domain in Using Oracle WebLogic Server for OCI](#).
 - If the WebLogic Server version is 12.2.1.2, see [Upgrade a Domain](#).

Upgrade a Domain

If you migrated your Oracle Java Cloud Service instance to WebLogic Server version 12.2.1.2 for Oracle WebLogic Server for OCI, you must perform an in-place upgrade to upgrade the WebLogic Server version.

Topics:

- [Stop all WebLogic Server Processes](#)
- [Perform Prerequisite Tasks](#)
- [Set up VNC Server](#)
- [Perform Readiness Check](#)
- [Upgrade Infrastructure Schemas](#)
- [Reconfigure the Domain](#)
- [Upgrade the Domain](#)
- [Restart Servers](#)
- [Post Upgrade](#)
- [Roll Back an Upgrade](#)

Stop all WebLogic Server Processes

Stop all the WebLogic server processes on the Oracle WebLogic Server for OCI instance.

1. As an `opc` user, log in to the each non-administration VM in the instance and run the following command:

```
sudo su - oracle
/opt/scripts/restart_domain.sh -o stop
# run jps to confirm that there are no more processes are running.
If there are run kill -9 on each.
jps
```

2. As an `opc` user, log in to the Administration VM in the instance and run the following command:

```
sudo su - oracle
/opt/scripts/restart_domain.sh -o stop
# run jps to confirm that there are no more processes are running.
If there are run kill -9 on each.
jps
```

Perform Prerequisite Tasks

Before upgrading your service instance, back up the database and the domain directories.

To perform prerequisite tasks:

 **Note:**

You need to perform steps 1 to 3 in case the upgrade fails and you want to roll back the service instance to its original state.

1. Back up the Oracle Cloud Infrastructure database to object storage. See [Create an On-Demand Full Backup of a Database](#) in Oracle Cloud Infrastructure documentation.
2. On each VM in the Oracle WebLogic Server for OCI instance, back up the domain directory on your service node.

```
sudo su - oracle
mkdir /u01/data/old_domain_version
cp -r /u01/data/domains/* /u01/data/old_domain_version/
```

3. On each VM in your Oracle WebLogic Server for OCI, move the middleware home contents to a temporary location to replace the older binaries with the WebLogic Server 12.2.1.4 binaries.

```
sudo su - oracle
mkdir /tmp/old_mw_version
mv /u01/app/* /tmp/old_mw_version/
```

4. Copy the 12.2.1.4.0 binaries located in `/tmp/mw_bak/` to `/u01/app/`.

 **Note:**

When migrating your Oracle Java Cloud Service instances to Oracle WebLogic Server for OCI, you would have moved the 12.2.1.4 binaries to `/tmp/mw_bak/` location. See step 3e in [Replace Domain and Binaries](#).

```
sudo su - oracle
cp -r /tmp/mw_bak/* /u01/app/
```

Set up VNC Server

To use the Reconfiguration Wizard and Upgrade Assistant (Fusion Middleware tools) during the upgrade process, you need a graphical user interface (GUI) environment. The instructions in this step explain how to set up a VNC server and use port forwarding through a bastion host. If you are familiar with using X11, then X11 forwarding can be used to forward the GUI to your local desktop and you can skip this step.

 **Note:**

This step is *not applicable* for a non-JRF instance.

1. Log in to administration compute instance as an `opc` user.

2. Run the following command to install the GUI packages on the target 12.2.1.4 instance:

```
sudo bash
yum group install "Server with GUI"
# enter 'y' when prompted.
```

3. As an `opc` user, run the following command to set up the VNC server for the `oracle` user on the administration compute instance:

```
sudo bash
yum install tigervnc-server -y
exit
sudo su - oracle
vncpasswd
# enter the password and confirmation password.
# respond with "n" when prompted if this should be a view only
password
exit
sudo bash
cp /lib/systemd/system/vncserver@.service /etc/systemd/system/
vncserver@\:1.service
vi /etc/systemd/system/vncserver@\:1.service
  Replace <USER> with oracle
systemctl daemon-reload
systemctl enable vncserver@\:1.service
systemctl start vncserver@\:1.service
systemctl status vncserver@\:1.service
# Confirm it is running

# Set up the firewall to allow the VNC server port to be accessed:
iptables -I INPUT -m state --state NEW -p tcp --destination-port
5901 -j ACCEPT
```

4. Sign in to the Oracle Cloud Infrastructure Console and update the subnet to have a security list that enables inbound access to the VNC server port 5901.

 **Note:**

If you have not already set up your bastion VM to be able to access VMs as the `opc` user, then place your private key pem for the `opc` user on the disk. This is used for port forwarding.

5. Create the tunnel by accessing the bastion host as the `opc` user:

```
ssh -i <privatekey.ppk> -L <vnc_port_on_wlsm-private-ip>:<wlsvm-
private-ip>:<port_on_bastion> <wlsvm-private-ip>
```

Following is an example, where the IP address of the administration compute instance is 10.1.1.1 and the private key for the `opc` user is in `~/.ssh/id_rsa`

```
ssh -i ~/.ssh/id_rsa -L 5901:10.1.1.1:5901 10.1.1.1
```

Windows instructions for launching GUI

1. Install and launch PuTTY.
2. For **Host Name**, type the bastion IP address.
3. For **Saved Sessions**, type `bastion`.
4. Under **Category**, go to **Connection > Data**.
5. For **Auto-logout username**, type `opc`.
6. Under **Category**, go to **Connection > Data > SSH > Tunnels**.
7. Type the following values for the respective fields:
 - Source port: `5901`
 - Destination: `localhost:5901`
8. Click **Add**.
9. Under **Category**, go to **Connection > Data > SSH > Auth**.
10. For **Private key file for authentication**, browse and select the xperiment private key that you have created.
11. Under the **Category**, select **Session**.
12. Select **Save** and then select **Open** to establish the connection
13. Verify that you connected successfully to the putty session.
14. Install a VNC Viewer and set up a new connection to use `localhost:5901` to verify that you can connect correctly.

 **Note:**

Ensure that you have set up the `vncserver` as the `oracle` user, as this creates a session with the `oracle` user even though you have port forwarding via the `opc` user ssh keys.

Perform Readiness Check

Perform a readiness check to determine if your service instance is ready for upgrade.

 **Note:**

This step is *not applicable* for a non-JRF instance.

1. By using the VNC viewer session, start the Upgrade Assistant.

```
export USER_MEM_ARGS=-Djava.security.egd=file:/dev/urandom  
/u01/app/oracle/middleware/oracle_common/upgrade/bin/ua -readiness
```

Setting `USER_MEM_ARGS` to use the `/dev/urandom` device reduces the time it takes to run the Oracle Fusion Middleware upgrade tools.

2. Use the Upgrade Assistant to perform a readiness check. See [Upgrade from a previous 12c release to 12.2.1.4](#) in *Upgrading to the Oracle Fusion Middleware Infrastructure*

3. On the **Readiness Check Type** screen, select the domain-based readiness check.

The domain-based readiness check enables the Upgrade Assistant to discover and select all upgrade-eligible schemas or component configurations in the domain specified in the Domain Directory field.

4. On the **End of Readiness** screen in the Upgrade Assistant, review the results of the readiness check (Readiness Success OR Readiness Failure).
 - If the readiness check is successful, click **View Readiness Report** to review the complete report. Oracle recommends that you review the Readiness Report before you perform the upgrade even when the readiness check is successful. Use the **Find** option to search for a particular word or phrase within the report. The report also indicates where the completed Readiness Check Report file is located.
 - If the readiness check encounters an issue or error, click **View Log** to review the log file, identify and correct the issues, and then restart the readiness check.

Upgrade Infrastructure Schemas

This step helps to identify if you have an earlier version of infrastructure database schemas or have installed other Oracle products.



Note:

This step is *not applicable* for a non-JRF instance.

1. Start the Upgrade Assistant if you have not already done so. For example:

```
export USER_MEM_ARGS=-Djava.security.egd=file:/dev/./urandom
/u01/app/oracle/middleware/oracle_common/upgrade/bin/ua
```

2. Upgrade the schemas. See *Upgrading Schemas Using the Upgrade Assistant* in *Upgrading to the Oracle Fusion Middleware Infrastructure*:
3. On the Selected Schemas screen, select **All Schemas Used by a Domain**, and then enter a domain directory name in the **Domain Directory** field.

The **All Schemas Used by a Domain** selection allows the Upgrade Assistant to discover and select all components that have a schema available to upgrade in the domain specified in the **Domain Directory** field. This is also known as a domain-assisted schema upgrade. In addition, the Upgrade Assistant prepopulates connection information on the schema input screens.

4. On the Upgrade Progress screen in the Upgrade Assistance, monitor the schema upgrade progress.
5. Finish the schema upgrade process.

- If the schema upgrade succeeds, click **Close** to complete the upgrade and close the wizard.
- If the upgrade fails, click **View Log** to view and troubleshoot the errors. The logs are available in the following directory:

```
/u01/app/oracle/middleware/oracle_common/upgrade/logs
```

6. Remedy the database connection failure if one occurs. See *Problems with Database Connectivity When Upgrading the Infrastructure Schema Database* in *Administering Oracle Java Cloud Service*.
7. Verify the schema upgrade was successful by checking that the schemas in `schema_version_registry` have been properly updated. See *Problems with Database Connectivity When Upgrading the Infrastructure Schema Database* in *Upgrading to the Oracle Fusion Middleware Infrastructure*.

One way to verify the schema upgrade is to use SQL*Plus commands to obtain data from the `SCHEMA_VERSION_REGISTRY`.

- a. Find the Oracle Java Cloud Service instance's schema prefix in the Upgrade Assistant log file at `/u01/app/oracle/middleware/oracle_common/upgrade/logs`.
- b. Connect to the database as a user having Oracle DBA privileges and run the following commands from SQL*Plus to get the current version numbers.

```
sqlplus / as sysdba
SQL> connect <user_name>/<password>@<host_name>:<port>/<service_name>
as sysdba
SQL> SELECT MRC_NAME,COMP_ID,OWNER,VERSION,STATUS,UPGRADED FROM
SCHEMA_VERSION_REGISTRY WHERE MRC_NAME like 'SP1556690734';
```

Example output for the `SCHEMA_VERSION_REGISTRY`:

MRC_NAME	COMP_ID	OWNER	VERSION	STATUS	UPGRADED
SP1556690734	IAU	SP1556690734_ IAU	12.2.1.2.0	VALID	Y
SP1556690734	IAU_APPEND	SP1556690734_ IAU_APPEND	12.2.1.2.0	VALID	N
SP1556690734	IAU_VIEWER	SP1556690734_ IAU_VIEWER	12.2.1.2.0	VALID	Y
SP1556690734	MDS	SP1556690734_ MDS	12.2.1.3.0	VALID	Y
SP1556690734	OPSS	SP1556690734_ OPSS	12.2.1.0.0	VALID	Y
SP1556690734	STB	SP1556690734_ STB	12.2.1.3.0	VALID	Y
SP1556690734	UCSUMS	SP1556690734_ UMS	12.2.1.0.0	VALID	N
SP1556690734	WLS	SP1556690734_ WLS	12.2.1.0.0	VALID	N

Reconfigure the Domain

 **Note:**

- This step is *not applicable* for a non-JRF instance.
- Running the reconfiguration wizard is not required for an upgrade from 12.2.1.3. However, if it is not run and you do not replace 12.2.1.3 with 12.2.1.4 in config.xml, then you will encounter the Incorrect Version Numbers After a Reduced Downtime Upgrade issue.

1. Start the Reconfiguration Wizard as user `oracle` with the following logging options, with `log_file` as the absolute path of the log file you'd like to create for the domain reconfiguration session. This can be helpful if you need to troubleshoot the reconfiguration process.

For example:

```
/u01/app/oracle/middleware/oracle_common/common/bin/reconfig.sh -  
log_priority=all -log="/u01/reconfig0212.log"
```

2. Perform the reconfiguration tasks as described in *Upgrading to the Oracle Fusion Middleware Infrastructure*. See [Reconfiguring the Domain with the Reconfiguration Wizard](#).
3. On the Advanced Configuration screen of the Reconfiguration Wizard, select **Deployment and Services**.
4. Target the `wsm-pm` app to the cluster containing the managed servers.
5. Click **Reconfig**.
6. Check the End of Configuration screen to learn whether the reconfiguration process completed successfully or failed.
 - If the reconfiguration is successful, **Oracle WebLogic Server Reconfiguration Succeeded** is displayed. The location of the domain that was reconfigured as well as the Administration Server URL (including the listen port) are displayed as well.
 - If the reconfiguration process did not complete successfully, an error message is displayed which indicates the reason. Take appropriate action to resolve the error.

Upgrade the Domain

 **Note:**

This step is *not applicable* for a non-JRF instance.

1. Start the Upgrade Assistant, for example:

```
export USER_MEM_ARGS=-Djava.security.egd=file:/dev/./urandom
/u01/app/oracle/middleware/oracle_common/upgrade/bin/ua
```

2. Use the Upgrade Assistant to upgrade the domain configurations. See [Upgrading the Domain Configurations with the Upgrade Assistant](#) in *Upgrading to the Oracle Fusion Middleware Infrastructure*.
3. On the All Configurations screen, select **All Configurations Used by a Domain** and specify your domain location in the **Domain Directory** field. Enter the domain directory directly or click **Browse** to select a valid domain directory.
4. On the Upgrade Summary page, review the summary of the options you have selected for the component configuration upgrade, and then click **Upgrade** to start the upgrade process.
5. View the Upgrade Progress page to monitor the upgrade.
6. View the results and finish the upgrade.
 - If the upgrade succeeds, the Upgrade Success page is displayed. Click **Close** to complete the upgrade and close the wizard.
 - If the upgrade fails, the Upgrade Failure screen is displayed. Click **View Log** to view and troubleshoot the errors. The logs are available at

```
ORACLE_HOME/oracle_common/upgrade/logs
```

Restart Servers

Access the 12.2.1.4 instance as the `opc` user, and on each VM run the following command:

```
sudo su - oracle
/opt/scripts/restart_domain.sh
```

Post Upgrade

If the upgrade was successful, complete following tasks if they apply to your instance:

- If you use a Hosting Provider to manage DNS, then reset the `CNAME` records at your Hosting Provider to point to the new IP addresses of the load balancer and WebLogic VMs.
- Destroy the source 12.2.1.2 instance. See [Destroy Stack Resources](#) in *Using Oracle WebLogic Server for OCI*.

WARNING:

Do not Delete a JRF Database Schema or Delete the Identity Cloud Service Resources in *Using Oracle WebLogic Server for OCI* as these resources are required in the upgraded instance.

- Run the following commands to remove the UI libraries. This stops the VNC server, removes the VNC server package, and removes the Server with GUI group packages.

```
# Remove VNC server
sudo su - oracle
vncserver -list
# Locate the X Display value and kill this - typically this is :1
vncserver -kill :1
# Remove the Linux service
exit
sudo bash
systemctl stop vncserver@\:1.service
systemctl disable vncserver@\:1.service
# Uninstall package
yum remove tigervnc-server -y
# Remove GUI
yum group remove "Server with GUI"
```

Roll Back an Upgrade

If the upgrade fails, you can roll back the upgrade to restore the WebLogic Server to its previous version.

1. Shutdown all WebLogic Server processes, including node manager, on each node. See [Stop all WebLogic Server Processes](#).
2. Restore the backup of the Oracle Cloud Infrastructure database instance from object storage.
See [Restore a Database Using a Specific Backup from Object Storage](#) in Oracle Cloud Infrastructure documentation.
3. Replace the domain contents in domain directory (`/u01/data/domains/*`) on each node with those you saved in `/u01/data/old_domain_version/`. See [Perform Prerequisite Tasks](#).
4. Replace the binaries in `/u01/app/*` with those saved in `/tmp/old_mw_version`. See [Perform Prerequisite Tasks](#).
5. Restart the node manager and the WebLogic Server processes. See [Start the WebLogic Server Processes](#).