Oracle® Cloud

Administering Oracle Java Cloud Service

Release 22.1.1

F52816-02

February 2022

Oracle Cloud Administering Oracle Java Cloud Service, Release 22.1.1

F52816-02

Contents

Preface

Audience xvi

Documentation Accessibility xvi

Related Resources xvi

Conventions xvii

Diversity and Inclusion xvii

1 Get Started with Oracle Java Cloud Service

About Oracle Java Cloud Service 1-1

- About Oracle Java Cloud Service Offerings and Oracle WebLogic Server Software Releases 1-2
- About Oracle WebLogic Server Editions Available for Oracle Java Cloud Service 1-5
- About Certified Oracle Fusion Middleware Products on Oracle Java Cloud Service 1-6
- About the Compute Infrastructure for Oracle Java Cloud Service 1-7
- About Application and Network Security in Oracle Java Cloud Service 1-7

About the Components of Oracle Java Cloud Service 1-8

About the Interfaces to Oracle Java Cloud Service 1-12

Before You Begin with Oracle Java Cloud Service 1-15

- Prerequisites for Instances in Oracle Cloud Infrastructure 1-15
- Create an SSH Key Pair 1-15
- Create an Object Storage Container 1-15
- Select an IP Network for a Service Instance with a Managed Load Balancer 1-16
- Create a Database 1-17
  - Create an Oracle Autonomous Database 1-18
  - Create an Oracle Cloud Infrastructure Database 1-19
  - Create an Oracle Database Cloud Service Database Deployment 1-20
  - Create a Custom Pluggable Database (PDB) 1-21
  - Use an Oracle Cloud Infrastructure Database on a Different Virtual Cloud Network 1-27

About Oracle Java Cloud Service Subscriptions and Licenses 1-31

How to Begin with Oracle Java Cloud Service Subscriptions 1-32

Access Oracle Java Cloud Service 1-32

- Access Oracle Java Cloud Service from the Infrastructure Console 1-33

Access Oracle Java Cloud Service from the Infrastructure Classic Console 1-33
Access Oracle Java Cloud Service from Oracle Cloud at Customer 1-34
Typical Workflow for Using Oracle Java Cloud Service 1-34
About Oracle Java Cloud Service Roles and User Accounts 1-36
Java Administrator 1-36
Related Service Administrators 1-37
Service Instance Users 1-37
Oracle Cloud Infrastructure Policies 1-38
About Java Cloud Service Instances in Oracle Cloud Infrastructure 1-39
Compare Oracle Cloud Services for Deploying Java Applications 1-43
2 Create an Oracle Java Cloud Service Instance
About Creating an Oracle Java Cloud Service Instance 2-1
About Life Cycle Management of Oracle Java Cloud Service Instances 2-2
Design Considerations for an Oracle Java Cloud Service Instance 2-5
Service Level 2-7
Software Release 2-7
Edition 2-8
License 2-8
Region 2-9
Compute Shape 2-9
WebLogic Cluster 2-10
Availability Domain 2-10
Subnet 2-11
IP Network 2-11
Public IP Address 2-12
Reserved IP Address 2-12
Domain Partition 2-13
User Authentication 2-13
Administrator Access 2-14
Client Access 2-14
Coherence Data Tier 2-14
Database 2-15
Load Balancer 2-17
Backup Location 2-19
Create an Oracle Java Cloud Service Instance by Using a QuickStart Template 2-19
Create a QuickStart Instance 2-20
Simple Java Web App 2-21
Multi-Tier Java EE App with High Availability 2-22
Highly Available Java EE App with Caching 2-23

Compare QuickStart Templates 2-24

Create an Oracle Java Cloud Service Instance Attached to a Public Subnet on Oracle Cloud Infrastructure 2-25

Start the Create New Instance Wizard 2-26

Specify Basic Service Instance Information 2-26

Specify the Service Instance Details 2-29

Specify WebLogic Configuration 2-29

Configure WebLogic Server Access 2-31

Configure the Coherence Data Tier 2-33

Configure the Databases 2-33

Configure Backup and Recovery 2-36

Configure the Load Balancer 2-37

Confirm Your Oracle Java Cloud Service Instance Creation 2-39

Create an Oracle Java Cloud Service Instance Attached to a Private Subnet on Oracle Cloud Infrastructure 2-41

Create the Required Resources in Oracle Cloud Infrastructure 2-41

Create an Oracle Cloud Infrastructure Database System Attached to a Private Subnet 2-44

Create an Oracle Java Cloud Service Instance Attached to a Private Subnet on Oracle Cloud Infrastructure Using the Wizard 2-45

Start the Create New Instance Wizard 2-46

Specify Basic Service Instance Information 2-46

Specify the Service Instance Details 2-48

Confirm Your Oracle Java Cloud Service Instance Creation 2-59

Create an Oracle Java Cloud Service Instance Attached to a Private Subnet Using REST API 2-61

Create a Custom Oracle Java Cloud Service Instance on Oracle Cloud Infrastructure Classic 2-65

Start the Create New Instance Wizard 2-67

Specify Basic Service Instance Information 2-67

Specify the Service Instance Details 2-70

Specify WebLogic Configuration 2-70

Assign Reserved IP Addresses for a Service Instance in a Region 2-72

Assign Reserved IP Addresses for an Oracle Database Exadata Cloud Service Database 2-73

Configure WebLogic Server Access 2-73

Configure the Coherence Data Tier 2-75

Configure the Databases 2-75

Configure Backup and Recovery 2-77

Configure the Load Balancer 2-78

Confirm Your Oracle Java Cloud Service Instance Creation 2-80

Create an Oracle Java Cloud Service Instance with Cloud Stack 2-82

Get Started with Cloud Stack 2-82

Template Parameters 2-83

Create a Stack with the CLI 2-83
Customize the Template 2-83
About the Sample Application Deployed to an Oracle Java Cloud Service Instance 2-85

3 Manage the Life Cycle of Oracle Java Cloud Service Instances

Typical Workflow for Managing the Life Cycle of Oracle Java Cloud Service Instances 3-1
View All Oracle Java Cloud Service Instances 3-2
Monitor Activity 3-3
Reserve IP Addresses 3-3
View Detailed Information About an Oracle Java Cloud Service Instance 3-4
Change the License Type for an Oracle Java Cloud Service Instance 3-5
View the Service Metrics for an Oracle Java Cloud Service Instance 3-5
Suspend an Oracle Java Cloud Service Instance 3-6
Stop, Start, and Restart an Oracle Java Cloud Service Instance and Individual Nodes 3-6
About Stopping, Starting, and Restarting an Oracle Java Cloud Service Instance and Individual Nodes 3-6
Stop or Start an Instance 3-9
Stop or Start a Node 3-10
Delete an Oracle Java Cloud Service Instance 3-10
Manage Tags for a Service Instance 3-12
Create, Assign, and Unassign Tags 3-12
List Tags 3-13
Find Tags and Instances Using Search Expressions 3-13
Delete Tags 3-16
Identify the Cloud Infrastructure Used by a Service Instance 3-16
Explore the Oracle Java Cloud Service Console 3-17
Explore the Oracle Java Cloud Service Welcome Page 3-21
Explore the Oracle Java Cloud Service Instance Overview Page 3-22

4 Administer Oracle Java Cloud Service Software

Set Up Fast and Secure Connections to Oracle Cloud 4-1
Oracle Cloud Infrastructure FastConnect 4-2
VPN 4-2
Access the Administration Consoles for Oracle Java Cloud Service 4-3
About the Security Checkup Tool 4-3
Configure the Wildcard Host Name Verifier 4-5
Update Administration Server Startup Properties 4-5
Restart Managed Server Using Node Manager 4-6
Set the Java Properties 4-6
Disable Remote Anonymous RMI T3 and IIOP Requests 4-7

Disable Tunneling on Server Channel 4-7

Access an Administration Console for a Service Instance 4-7

Access the Console of a Related Oracle Cloud Service 4-8

Access the Administration Console for a Service Instance Attached to a Private Subnet 4-9

Access a Node with a Secure Shell (SSH) 4-9

Generate a Key Pair with OpenSSH 4-10

Connect to a Node with OpenSSH 4-11

Create an SSH Tunnel to a Node with OpenSSH 4-11

Connect to a Private Node with OpenSSH 4-12

Generate a Key Pair with PuTTY 4-13

Convert a Private Key with PuTTY 4-14

Connect to a Node with PuTTY 4-15

Create an SSH Tunnel to a Node with PuTTY 4-15

Connect to a Node with VNC 4-17

Switch Users on a Node 4-18

Add an SSH Public Key 4-19

Add an SSH User 4-20

Use WLST to Administer a Service Instance 4-21

About WLST Online and Offline 4-22

Run WLST Commands on a Node 4-22

Run WLST Commands from a Different Host 4-23

Shut Down and Start Server Processes 4-24

Use the WebLogic Server Administration Console to Shut Down Servers 4-24

Use WLST Commands to Start the Administration Server 4-25

Use the WebLogic Server Administration Console to Start Managed Servers 4-27

About JVM Heap Settings 4-27

Default Heap Sizes 4-27

Custom Heap Sizes 4-28

About Data Sources 4-28

Predefined Data Sources 4-29

Data Source Types 4-29

Custom Data Sources 4-30

Data Source Network Connectivity 4-30

Manage Associations for a Service Instance 4-30

View Association Details 4-31

Associate an Oracle Java Cloud Service Instance with a Different Database 4-31

Change the Database Schema Password for an Oracle Java Cloud Service Instance 4-33

Change the Schema Password with the Console 4-33

Change the Schema Password Manually 4-34

Connect an Oracle Java Cloud Service Instance to an Application Database 4-39

Connect to an Oracle Autonomous Database 4-39

Download the Oracle Autonomous Database Wallet 4-39
Copy and Unpack the Wallet 4-40
Create a Data Source in the WebLogic Server Console 4-41
Connect to an Oracle Cloud Infrastructure Database 4-43
Configure an Oracle Java Cloud Service Instance for an Oracle RAC Database 4-43
Configure a Vanity Domain Name for a Service Instance 4-47
Register a Custom Domain Name with a Service Provider 4-48
Add a Vanity URL to an Oracle-Managed Load Balancer 4-48
Delete a Vanity URL from an Oracle-Managed Load Balancer 4-49
Update Oracle Traffic Director to Use a Custom Domain Name 4-49
Configure a Custom URL for an Application Deployed to a Service Instance 4-50
Configure a Custom URL for the WebLogic Server Console 4-51
Configure a Custom URL for the Sample Application 4-51
Monitor Applications with Oracle Java Flight Recorder and Oracle Java Mission Control 4-52
Administration Best Practices 4-53

5 Deploy and Undeploy Applications for an Oracle Java Cloud Service Instance

Overview of Deploying Applications to Oracle Java Cloud Service Instances 5-1
Use Fusion Middleware Control to Deploy and Undeploy an Application 5-2
Use Fusion Middleware Control to Deploy an Application 5-2
Use Fusion Middleware Control to Undeploy an Application 5-3
Use the WebLogic Server Administration Console to Deploy and Manage Applications 5-3
Use the WebLogic Server Administration Console to Deploy an Application 5-4
Use the WebLogic Server Administration Console to Start an Application 5-5
Use the WebLogic Server Administration Console to Undeploy an Application 5-5
Use WLST Commands to Deploy and Undeploy an Application 5-6
Use an IDE to Deploy and Undeploy an Application 5-7
Prerequisites for Deploying Using an IDE 5-7
Connect the IDE to a Remote WebLogic Server 5-8
Use an IDE to Deploy an Application to a Cluster 5-9
Use an IDE to Undeploy an Application 5-10
Deploy an Application to an Oracle Java Cloud Service Instance with Multiple Clusters 5-10
Access an Application Deployed to an Oracle Java Cloud Service Instance 5-11
Enable the JVM Debug Port on an Oracle Java Cloud Service Instance 5-12
Set Up the Debug Port in WebLogic Server 5-12
Create an Access Rule for the Debug Port 5-14
Use Third-Party Frameworks with Oracle Java Cloud Service 5-14
Third-Party Application Development Frameworks Tested with Oracle Java Cloud Service 5-15

Information for Configuring Apache Axis/Java 5-17
Omit Checks for Updates to Quartz Job Scheduler 5-17

6 Scale an Oracle Java Cloud Service Instance

About Scaling an Oracle Java Cloud Service Instance 6-1
About Scaling an Oracle Java Cloud Service Cluster 6-2
About Scaling Out a Cluster 6-2
About Scaling In a Cluster 6-3
About Adding a New Cluster 6-3
About Scaling an Oracle Java Cloud Service Node 6-3
About Changing the Compute Shape of a Node 6-4
What Happens When a Node is Being Scaled 6-4
What Happens After a Node is Scaled 6-5
About Adding Block Storage to a Node 6-5
About Automatic Scaling 6-6
How It Works 6-6
Prerequisites 6-6
What Are the Rule Components 6-6
Overview of Scaling Tasks for an Oracle Java Cloud Service Instance 6-7
Scale Out an Oracle Java Cloud Service Cluster 6-7
Scale Out a Cluster 6-8
Add a New Cluster to an Instance 6-8
Scale In a Cluster 6-8
Scale an Oracle Java Cloud Service Node 6-9
Scale a Node 6-9
Add Storage to a Node 6-10
Scale Automatically 6-11
Create a Scaling Rule 6-11
Edit a Scaling Rule 6-13
Delete a Scaling Rule 6-13
View Scaling Requests 6-14

7 Back Up and Restore an Oracle Java Cloud Service Instance

About Backup and Restoration in Oracle Java Cloud Service 7-1
What are the Contents of a Backup? 7-1
When Do Backups Occur? 7-2
Where are Backups Stored? 7-2
How Long are Backups Retained? 7-3
What Happens When a Backup is Restored? 7-3

Typical Workflow for Backing Up and Restoring a Service Instance 7-3
Add a Backup Configuration to an Oracle Java Cloud Service Instance 7-4
Configure Scheduled Backups for an Oracle Java Cloud Service Instance 7-6
Create an On-Demand Backup 7-7
Enable or Disable Backups 7-8
Delete a Backup 7-9
Restore a Backup 7-9
Restore the Database for a Service Instance 7-10
Return an Instance to Service After Restoration from a Backup 7-11
Access the Contents of a Backup 7-12
Explore the Backup Page 7-12

8 Manage Snapshots and Clones in Oracle Java Cloud Service

About Snapshots and Clones 8-1
What Does a Snapshot Contain? 8-1
What Can I Use Clones For? 8-2
What Happens When I Create a Snapshot? 8-2
How Does Cloning Work? 8-2
Create a Snapshot 8-3
Delete a Snapshot 8-4
Clone an Instance Using a Snapshot 8-5
View Details of Snapshots and Clones 8-16

9 Patch an Oracle Java Cloud Service Instance

About Patching and Rollback 9-1
Typical Workflow for Patching an Oracle Java Cloud Service Instance 9-4
View Patch Details 9-5
Perform Patch Prechecks and Address Patching Issues 9-5
Perform a Patch Precheck 9-6
Address Patch Precheck Issues 9-6
Failure Due to Manual Patching 9-7
Disk Space Shortage 9-7
Missing Database Connectivity 9-8
Node Manager Is Not Running 9-8
Connectivity Issues between Managed Servers and the Administration Server 9-8
Servers Not Running 9-8
Storage Access Failure 9-8
Apply a Patch 9-9
Roll Back a Patch 9-10

Explore the Patching Page 9-10

10 Upgrade the WebLogic Server Release for an Oracle Java Cloud Service Instance

About Upgrading the WebLogic Server Release for an Oracle Java Cloud Service Instance 10-1
Perform Prerequisite Tasks 10-2
Download the Upgrade Software 10-4
Stop All WebLogic Server Processes 10-6
Install the Upgrade Software 10-8
Perform a Readiness Check 10-9
Upgrade the Infrastructure Database Schemas 10-10
Upgrade the WebLogic Server 12c Infrastructure Database Schemas 10-10
Upgrade the WebLogic Server 11g Infrastructure Database Schemas 10-12
Reconfigure the Domain 10-15
Reconfigure the WebLogic Server 12c Domain 10-15
Reconfigure the WebLogic Server 11g Domain 10-16
Upgrade the Domain 10-18
Restart the Administration Server Node 10-19
Update and Restart the Managed Server Nodes 10-20
Perform Post-Upgrade Tasks 10-21
Roll Back an Upgrade 10-22

11 Secure an Oracle Java Cloud Service Instance

About Security in Oracle Java Cloud Service 11-1
Use Oracle Identity Cloud Service with Oracle Java Cloud Service 11-2
About Oracle Identity Cloud Service 11-3
Manage Service Administrators 11-4
Create a Service Instance with Oracle Identity Cloud Service 11-4
Manage WebLogic Server Administrators 11-5
Configure Protected Application Contexts for a New Service Instance 11-6
Update Protected Application Contexts on an Existing Service Instance 11-7
Secure an Application Using Deployment Descriptors 11-7
Oracle Identity Cloud Service Cloudgate Patching on Oracle Java Cloud Service 11-9
Upgrade Cloudgate on Oracle Java Cloud Service 11-9
Change to Previous Cloudgate Version 11-11
Remove the Old Installations 11-11
Configure Network Security 11-12
About the Default Access Ports 11-12
Ports Open to Traffic from Outside the Oracle Cloud Network 11-12

Ports Open to Traffic from Within the Oracle Cloud Network 11-13
Administration Server Deployment Port 11-14
Create an Ingress Rule 11-14
Create an Access Rule 11-15
Enable or Disable an Access Rule 11-17
Delete an Access Rule 11-17
Enable Console Access for a Service Instance 11-18
Enable HTTP Access to a Service Instance 11-18
Enable the HTTP Port on Oracle Traffic Director 11-19
Create an Access Rule for the Oracle Traffic Director HTTP Port 11-20
Enable Communication Between Service Instances 11-21
Enable Communication Between a Compute Instance and a Service Instance 11-22
Configure SSL for a Service Instance 11-23
About SSL in Oracle Java Cloud Service 11-24
Configure SSL for Oracle Traffic Director 11-25
Configure SSL for WebLogic Server 11-29
Configure SSL for Oracle Cloud Infrastructure Load Balancing 11-37
Authenticate Users 11-37
About Users 11-37
Cloud Users and Service Administrators 11-38
WebLogic Server Administrators 11-38
Application Users 11-39
Database Users 11-39
Load Balancer Administrators 11-40
Operating System Users 11-40
About Authentication 11-40
Cloud Authentication 11-41
WebLogic Server Authentication 11-41
Manage Passwords 11-43
Cloud User Password 11-43
WebLogic Server Administrator Password 11-43
WebLogic Node Manager Password 11-44
Database Password 11-44
Oracle Traffic Director Password 11-45
Application User Password 11-45
Relocate Oracle Java Cloud Service to a Different Identity Domain 11-45

12 Use Oracle Coherence in Oracle Java Cloud Service

Overview of Coherence Tasks for Oracle Java Cloud Service 12-1
About Oracle Coherence in Oracle Java Cloud Service 12-2

About Cache Capacity for a Service Instance 12-4
Add a Coherence Data Grid 12-6
Scale Out a Coherence Data Grid 12-6
Scale In a Coherence Data Grid 12-7
Delete a Coherence Data Grid 12-7

13 Administer the Load Balancer for an Oracle Java Cloud Service Instance

About the Load Balancer in Oracle Java Cloud Service 13-1
Overview of Load Balancer Administration Tasks 13-5
Disable or Enable the Load Balancer for an Oracle Java Cloud Service Instance 13-6
Disable and Enable Oracle Traffic Director 13-7
Disable and Enable an Oracle-Managed Load Balancer 13-7
Add a Load Balancer to a Service Instance 13-8
Add a Second Load Balancer Node to a Service Instance 13-10
Remove a Load Balancer Node from a Service Instance 13-11
Configure a Load Balancer for a Service Instance 13-12
Configure Oracle Traffic Director 13-12
Configure an Oracle Cloud Infrastructure Load Balancing Instance 13-12
Configure an Oracle Cloud Infrastructure Load Balancing Classic Instance 13-13
Set Up an Oracle Cloud Infrastructure Load Balancer 13-13
Prepare to Set Up an Oracle Cloud Infrastructure Load Balancer 13-13
Create and Configure an Instance of Oracle Cloud Infrastructure Load Balancing 13-14
Use an Oracle Cloud Infrastructure Load Balancer 13-17
Remove the Oracle-Managed Load Balancer 13-17
Configure an Oracle Cloud Infrastructure Load Balancer 13-18
Verify Access to the Deployed Sample Application 13-19
About the Storage Volumes Attached to the Load Balancer Nodes 13-20

14 About the Infrastructure Resources Used by Oracle Java Cloud Service

About the Deployment Topology 14-1
Compute Topology for Oracle Java Cloud Service Instances 14-3
About the Storage Volumes Attached to the WebLogic Server Nodes 14-4
About the Storage Volumes Attached to the Load Balancer Nodes 14-5

15 Troubleshoot Oracle Java Cloud Service

Before You Begin Troubleshooting 15-2
Find Diagnostic Information to Help with Troubleshooting 15-3
Use the WebLogic Server Administration Console to Find Diagnostic Information 15-3
Use the WebLogic Server Administration Console to Find Log Files 15-3

Find Status Messages for Oracle Java Cloud Service Instances 15-4
Find VM Boot Log Messages 15-5
Problems with Failure of a Running Service when the Schema User Password Expires 15-5
Problems with Creating Service Instances 15-5
Problems with Deploying and Accessing Applications 15-9
Problems with Scaling 15-9
Problems with Patching and Rollback 15-10
Problems with Backup and Restoration 15-11
Problems with Performance of Oracle Java Cloud Service—Coherence Service Instances 15-14
Problems with Restart 15-15
Problems with Connectivity 15-16
My private key is lost or corrupted 15-16
My connection to a VM is refused 15-16
I received a hostname verification error when attempting to connect to Node Manager 15-17
Problems with Database Connectivity When Upgrading the Infrastructure Schema Database 15-17
Problems with the Node Manager 15-17
Problems with a Database Deployment 15-19
Problems with Connection validation when provisioning an Oracle Java Cloud Service-Virtual Image instance with Oracle Database Cloud Service 12.1 VI 15-20
Problems with Transparent Data Encryption Wallet Error when Provisioning an Oracle Java Cloud Service-Virtual Image Instance with Oracle Database Cloud Service 11g Virtual Image 15-21
Problems Opening the WebLogic Server Administration Console from Fusion Middleware Control 15-21
Security Checkup Tool Warnings 15-22
Problems Accessing Applications Via a Custom Web Server 15-25
How can I Fine-Tune Performance? 15-25

A Oracle Fusion Middleware Products Certified on Oracle Java Cloud Service

B Patches Included in Oracle Java Cloud Service

C Effect of Lifecycle and Administration Operations on Billing

D Migrate Applications to Oracle Java Cloud Service with AppToCloud

Typical Workflow for Migrating Applications to Oracle Java Cloud Service with AppToCloud D-1
On-Premises Tasks D-2

- Cloud Tasks D-3
Prerequisites for Using AppToCloud D-4
- Source WebLogic Server Domain D-4
- Destination Oracle Java Cloud Service D-5
AppToCloud Considerations and Limitations D-5
- Database Services D-6
- Multiple Clusters D-6
- 11g Applications D-6
- File System Locations D-7
- Server Classpath D-7
- Exploded Archive Deployments D-8
Migrate an Oracle Database to Oracle Cloud for Oracle Java Cloud Service D-8
Install the On-Premises AppToCloud Tools D-8
Check the Health on an On-Premises WebLogic Domain D-9
Export an On-Premises WebLogic Domain D-13
Create a Service Instance with AppToCloud D-17
- Configure a Foreign JNDI Provider D-29
- Configure a Java Mail Session D-30
- Configure a Foreign JMS Server D-31
- Configure a Remote SAF Context D-33
- Configure a JMS Messaging Bridge Destination D-34
- Disable a JMS Messaging Bridge D-35
Import Applications into a Service Instance D-36
Recreate On-Premises Domain Resources D-37
AppToCloud Command Reference D-39
- Client D-39
- Health Check D-40
- Export D-41
- Wallet Manager D-43

Preface

Administering Oracle Java Cloud Service explains how to provision Oracle Java Cloud Service instances, and ensure reliable functioning of provisioned service instances. This document explains how to perform these tasks by using the Oracle Java Cloud Service web interface.

Topics:

- Audience
- Related Resources
- Conventions

Audience

Administering Oracle Java Cloud Service is intended for Oracle Cloud account administrators and service administrators who want to provision Oracle Java Cloud Service instances, and ensure reliable functioning of provisioned service instances.

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.

Related Resources

For more information, see these Oracle resources:

- Oracle Cloud

  `http://cloud.oracle.com`
- *Getting Started with Oracle Cloud*
- Oracle Java Cloud Service FAQ
- *REST API for Oracle Java Cloud Service*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

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 Get Started with Oracle Java Cloud Service

Service administrators or tenant administrators of Oracle Java Cloud Service , and Java EE application developers can get familiar with the components, interfaces, subscriptions, licenses, service roles, user accounts, instances, and infrastructure of Oracle Java Cloud Service.

Topics:

- About Oracle Java Cloud Service
- About the Components of Oracle Java Cloud Service
- About the Interfaces to Oracle Java Cloud Service
- Before You Begin with Oracle Java Cloud Service
- About Oracle Java Cloud Service Subscriptions and Licenses
- How to Begin with Oracle Java Cloud Service Subscriptions
- Access Oracle Java Cloud Service
- About Oracle Java Cloud Service Roles and User Accounts
- Typical Workflow for Using Oracle Java Cloud Service
- About Java Cloud Service Instances in Oracle Cloud Infrastructure
- Compare Oracle Cloud Services for Deploying Java Applications

See Oracle Cloud Terminology in *Getting Started with Oracle Cloud* for definitions of terms found in this and other documents in the Oracle Cloud library.

About Oracle Java Cloud Service

You can use Oracle Java Cloud Service to quickly create, configure and manage your Java EE application environment in the cloud, including an Oracle WebLogic Server domain, in a fraction of the time it would normally take on-premises.

Topics:

- About Oracle Java Cloud Service Offerings and Oracle WebLogic Server Software Releases
- About Oracle WebLogic Server Editions Available for Oracle Java Cloud Service
- About Certified Oracle Fusion Middleware Products on Oracle Java Cloud Service
- About the Compute Infrastructure for Oracle Java Cloud Service
- About Application and Network Security in Oracle Java Cloud Service

You use a simple wizard to rapidly create an Oracle Java Cloud Service instance, which is a complete application environment provisioned on top of infrastructure provided by Oracle Cloud Infrastructure Compute Classic or Oracle Cloud Infrastructure Compute. The service instance includes Oracle WebLogic Server as the application container, and Oracle Traffic

Director as the software load balancer. Optionally, during provisioning, you can specify Oracle Coherence for caching and data grid functionality. With capabilities like elastic compute and storage, you can run any workload in Oracle Java Cloud Service, and easily scale out your environment based on your current business requirements.

During provisioning, you must associate a database with your Oracle Java Cloud Service instance. Supported databases depend on whether your instance is in an Oracle Cloud Infrastructure or Oracle Cloud Infrastructure Classic region. Supported databases for Oracle Cloud Infrastructure are Oracle Autonomous Database (Oracle Autonomous Transaction Processing), Oracle Cloud Infrastructure Database, and Oracle Database Cloud Service.

> **Note:**
>
> Free tier Oracle Autonomous Transaction Processing database is not supported.

Supported databases for Oracle Cloud Infrastructure Classic are Oracle Database Cloud Service and Oracle Database Exadata Cloud Service. (Not available on Oracle Cloud at Customer)

> **Note:**
>
> For Oracle Cloud at Customer, you can associate Oracle Database Cloud Service 19c with your Oracle Java Cloud Service instance. The Oracle Database Cloud Service 19c support is available from Oracle Cloud at Customer release 21.2.0.

About Oracle Java Cloud Service Offerings and Oracle WebLogic Server Software Releases

When creating an Oracle Java Cloud Service instance, you can choose between an environment that’s designed for high availability needs, such as user acceptance testing, staging and production, or an environment that’s designed for development and testing.

Service Level Offerings

Depending on your needs, you can choose among these Oracle Java Cloud Service service levels:

- Oracle Java Cloud Service

  This service level supports Oracle Java Cloud Service instance creation and monitoring; domain partitions; backup and restoration; patching; cloning; and scaling.

- Oracle Java Cloud Service—Virtual Image

  This service level supports Oracle Java Cloud Service instance creation and monitoring only. It does not support backup and restoration; patching; cloning; or scaling. You cannot provision a domain partition if you specify this service level.

This service level is:

  - Not supported if you have a Universal Credits subscription. This option does not appear on the console.
  - Supported if you have a traditional metered or non-metered subscription
  - Not supported on Oracle Cloud Infrastructure regions

  Oracle recommends using Oracle Java Cloud Service rather than Oracle Java Cloud Service—Virtual Image for better flexibility, administrative control, and availability of new features.

- Oracle Java Cloud Service Fusion Middleware — Oracle WebCenter Portal

  Leverages your Oracle WebCenter Portal license on Oracle Java Cloud Service. Choosing this option downloads additional installation tools. You must install the product yourself after creating this service instance. See Design Considerations for an Oracle Java Cloud Service Instance. This service level is supported on WebLogic Server release 12.2.1.3 only.

  > **Note:**
  >
  > Patching is not supported for service instances where Oracle Java Cloud Service Fusion Middleware—Oracle WebCenter Portal, Oracle Java Cloud Service Fusion Middleware—Oracle Data Integrator, or any other product that modifies the `MW_HOME` directory are installed. If you attempt to patch a service instance where any of these products are installed, patching prechecks issue an error message and patching fails.

  This service level is not supported if you associate an Oracle Autonomous Database with an Oracle Java Cloud Service instance.

- Oracle Java Cloud Service Fusion Middleware — Oracle Data Integrator

  Leverages your Oracle Data Integrator license on Oracle Java Cloud Service. Choosing this option downloads additional installation tools. You must install the product yourself after creating this service instance. See Design Considerations for an Oracle Java Cloud Service Instance. This service level is supported for WebLogic Server release 12.2.1.3 only.

  > **Note:**
  >
  > Patching is not supported for service instances where Oracle Java Cloud Service Fusion Middleware—Oracle WebCenter Portal, Oracle Java Cloud Service Fusion Middleware—Oracle Data Integrator, or any other product that modifies the `MW_HOME` directory are installed. If you attempt to patch a service instance where any of these products are installed, patching prechecks issue an error message and patching fails.

  This service level is not supported if you associate an Oracle Autonomous Database with an Oracle Java Cloud Service instance.

Software Releases

The Oracle WebLogic Server software releases and versions supported at the service levels are:

- Oracle WebLogic Server 12*c* (12.2.1.4) with Java Required Files 12*c* (12.2.1)

  This is the foundation for Oracle Fusion Middleware 12*c* (12.2.1).

  This Oracle WebLogic Server release is Java EE 7 compatible. JDK 8 is supported.

  You can associate an Oracle Autonomous Database with this Oracle WebLogic Server release.

  Oracle Java Cloud Service—Virtual Image does not support this Oracle WebLogic Server release.

  This release is not available on Oracle Cloud at Customer.
- Oracle WebLogic Server 12*c* (12.2.1.3) with Java Required Files 12*c* (12.2.1)

  This is the foundation for Oracle Fusion Middleware 12*c* (12.2.1).

  This Oracle WebLogic Server release is Java EE 7 compatible. JDK 8 is supported.

  You can associate an Oracle Autonomous Database with this Oracle WebLogic Server release.

  Oracle Java Cloud Service—Virtual Image does not support this Oracle WebLogic Server release.
- Oracle WebLogic Server 12*c* (12.2.1.2) with Java Required Files 12*c* (12.2.1)

  The Oracle WebLogic Server 12c (12.2.1.2) release is available only on Oracle Cloud at Customer.

  This Oracle WebLogic Server release is Java EE 7 compatible. JDK 8 is supported.

  Oracle Java Cloud Service—Virtual Image does not support this Oracle WebLogic Server release.
- Oracle WebLogic Server 11*g* (10.3.6) with Java Required Files 11*g* (11.1.1.7).

  This is the foundation for Oracle Fusion Middleware 11*g* (11.1.1.7).

  This Oracle WebLogic Server release is Java EE 5 compatible. JDK 7 is supported.

You can enable Oracle Coherence in Oracle Java Cloud Service when you provision an environment to run Oracle WebLogic Server 12*c* (12.2.1.2) or later. After you enable Oracle Coherence in Oracle Java Cloud Service, the environment provides a predefined cache capacity out-of-the-box for the Coherence applications that you deploy to the cloud environment.

> **Note:**
>
> If you provision the instance with Oracle Weblogic Server 11*g* (11.1.1.7) then Oracle Coherence in Oracle Java Cloud Service will be installed, but it won't be configured. You have to configure Oracle Coherence after the Oracle Java Cloud Service instance is provisioned.

About Oracle WebLogic Server Editions Available for Oracle Java Cloud Service

When you create an Oracle Java Cloud Service instance, you must choose an edition of Oracle WebLogic Servers configured for the service instance: Standard Edition, Enterprise Edition, or Enterprise Edition with Coherence.

The Create New Oracle Java Cloud Service Instance wizard contains a page where you specify the Oracle WebLogic Server edition.

> **Note:**
>
> For Oracle Java Cloud Service— Virtual Image instances based on any edition, backup and restoration, patching, and scaling are not supported.

You can select one of the following Oracle WebLogic Server editions:

WebLogic Server Edition	Description
Standard Edition	Delivers a reliable, manageable runtime platform with industry-leading performance. Includes: • Core Oracle WebLogic Server • Oracle JDeveloper • Oracle TopLink • Oracle Application Development Framework • Oracle Enterprise Pack for Eclipse • Oracle Traffic Director With this edition of WebLogic Service, Oracle Java Cloud Service a service instance with an Administration Server and only one Managed Server. For Oracle Java Cloud Service instances based on this edition, backup and restoration, patching, and scaling a node are supported. Scaling a cluster is not supported. You also cannot provision a domain partition. See Oracle WebLogic Server Standard Edition.
Enterprise Edition	Includes all features and benefits of WebLogic Server Standard Edition, in addition to: • Oracle WebLogic Server Enterprise Edition Clustering • Oracle Java SE Advanced—includes Java Mission Control and Java Flight Recorder for diagnosing problems in development and production For Oracle Java Cloud Service instances based on this edition, backup and restoration, patching, and scaling are supported. This edition supports WebLogic Server Multitenant, so you can create multiple partitions. You must manage these partitions using the WebLogic Server Console or Fusion Middleware Control. See Oracle WebLogic Server Enterprise Edition.

WebLogic Server Edition	Description
High Performance Edition	Delivers an integrated solution for building on-premises cloud infrastructures that span web server, application server, and data grid technology tiers. Includes all features and benefits of WebLogic Server Enterprise Edition, plus: • Oracle Coherence Enterprise Edition data grid for performance and scalability • Oracle DB connectivity thru Active Gridlink for RAC For Oracle Java Cloud Service instances created with this edition, backup and restoration, patching, and scaling are supported. This edition supports WebLogic Server Multitenant, so you can create multiple partitions. You must manage these partitions using the WebLogic Server Console or Fusion Middleware Control. See Oracle WebLogic Suite. **Note**: You must select High Performance Edition if you want to use Oracle Coherence in your Oracle Java Cloud Service instance. (Not available on Oracle Cloud at Customer)

> **Note:**
>
> You cannot change the Weblogic Server edition after the service instance has been created.

About Certified Oracle Fusion Middleware Products on Oracle Java Cloud Service

This topic does not apply to Oracle Cloud at Customer.

Certified Oracle Fusion Middleware products can be used with Oracle Java Cloud Service.

The following Oracle Fusion Middleware products are certified:

- Oracle SOA Suite for Oracle Middleware
- Oracle Service Bus
- Oracle BPEL Process Manager Option
- Oracle Unified Business Process Management Suite
- Oracle Business Intelligence Publisher
- Oracle WebCenter Portal
- Oracle WebCenter Content
- Oracle WebCenter Sites
- Oracle Data Integrator Enterprise Edition
- Oracle Enterprise Data Quality products (not all products certified)

Some products are only supported with Oracle Java Cloud Service—Virtual Image. Other products offer cloud subscriptions and tools to help you quickly provision the product on Oracle Java Cloud Service. Deployment guides are available for most products as well.

For more information, see Oracle Fusion Middleware Products Certified on Oracle Java Cloud Service.

About the Compute Infrastructure for Oracle Java Cloud Service

When you create an Oracle Java Cloud Service instance, the necessary compute infrastructure—virtual machines, storage volumes, and most of the network configuration—is set up for you.

Depending on the region you select while creating the instance, the service is built on Oracle Cloud Infrastructure Compute or Oracle Cloud Infrastructure Compute Classic. The WebLogic Server environment provided by the instance is the same on either infrastructure. But there are differences in the components that make up the infrastructure and in the workflow for creating instances. See About Java Cloud Service Instances in Oracle Cloud Infrastructure.

You can manage most of the infrastructure resources from within Oracle Java Cloud Service. For a few resources, you may need to use other interfaces. At relevant places in the documentation, references are provided to help you identify and access the appropriate interfaces.

About Application and Network Security in Oracle Java Cloud Service

You secure your Java EE applications on Oracle Java Cloud Service in much the same way that you secure any Oracle WebLogic Server environment.

WebLogic Server provides a security realm that controls authentication and authorization for the Java applications deployed to your Oracle Java Cloud Service instance. Administrators and developers can define security roles and policies to protect your applications against unauthorized access. By default, users are authenticated against the local WebLogic identity store.

There are two types of accounts in Oracle Cloud — Traditional and those with Oracle Identity Cloud Service. If cloud account includes Oracle Identity Cloud Service, service instances in Oracle Java Cloud Service can also use it for authentication. As a result, users that access your applications or the administration consoles in a service instance are authenticated against Oracle Identity Cloud Service if they are not found in the local WebLogic identity store.

In addition, Oracle Cloud provides a reliable and flexible network security infrastructure to further control how clients, administrators, and other cloud services access your service instance and its applications. By default, your service instances can only be accessed over secure protocols like HTTPS and SSH.

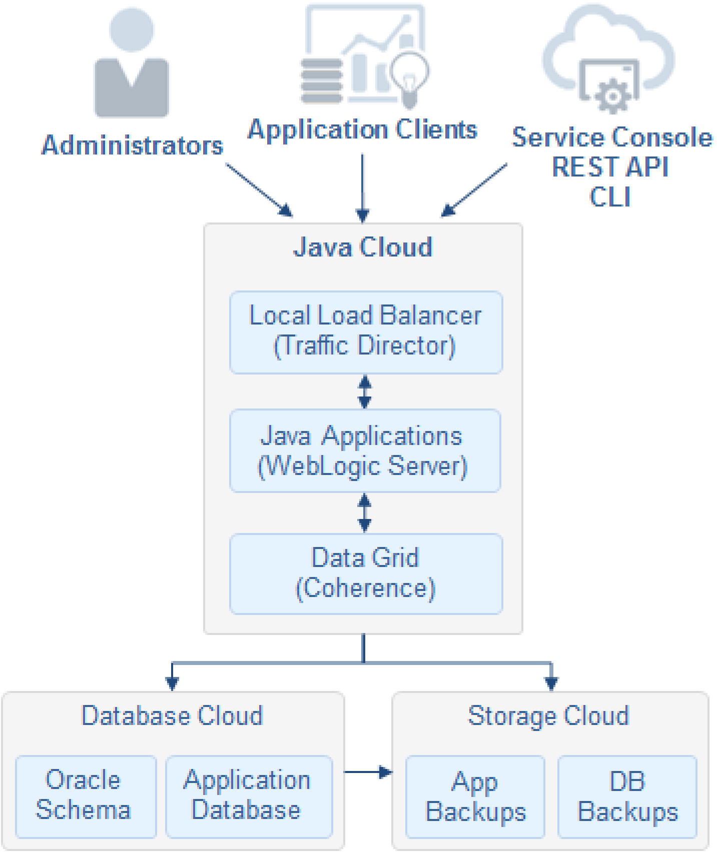
Oracle Cloud uses SSH to access the nodes that comprise your service instances, in order to perform predefined Platform Service actions like backup and patching. You initiate these Platform Service actions from the web console, CLI, or REST API. A separate SSH key pair is used for each service instance to perform this internal communication. This SSH key is not available for ad hoc usage. You cannot delete this key from nodes or it will cause these Platform Service actions to fail. The key is only used under programmatic control and cannot be directly accessed by Oracle employees. All SSH actions performed by Oracle Cloud on your nodes are logged and can be audited. Oracle does not have access to any SSH keys residing on your nodes and has no way to access your nodes, unless you explicitly provide access to the keys for troubleshooting purposes.

See About Security in Oracle Java Cloud Service.

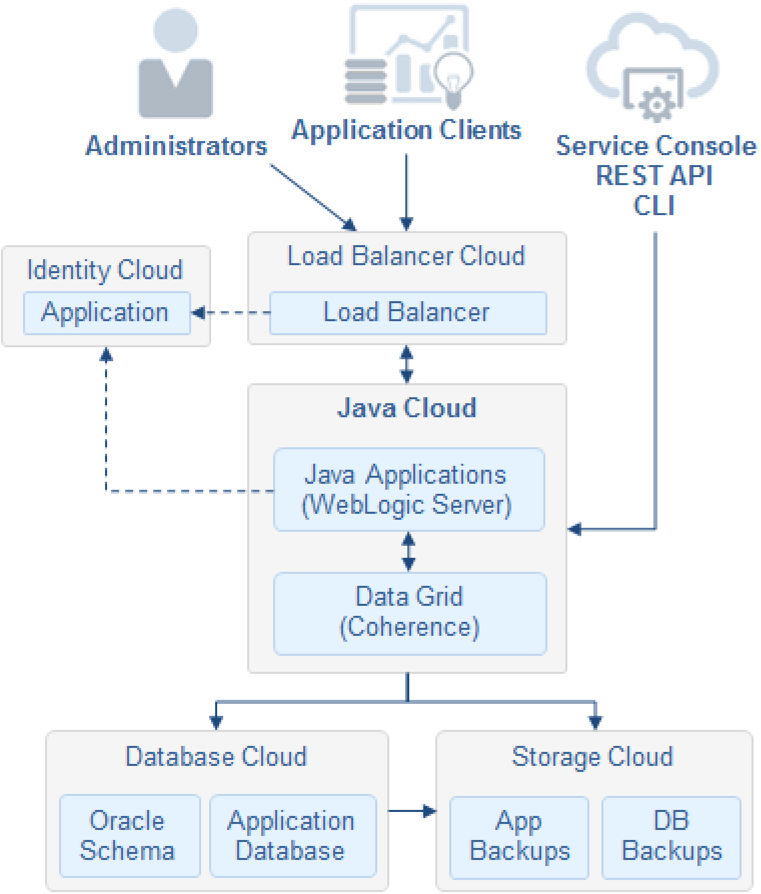
About the Components of Oracle Java Cloud Service

Each Oracle Java Cloud Service instance is comprised of several cloud services and middleware components.

Each service instance has a single Oracle WebLogic Server domain that consists of one WebLogic Administration Server and a cluster of Managed Servers to host your Java application deployments. When Oracle Coherence is enabled for a service instance, there is a second cluster of Managed Servers that provide an in-memory data grid for your applications. Optionally, you can configure a load balancer, particularly if you have configured more than one Managed Server. This figure illustrates the components that make up a typical service instance:



The next figure illustrates a service instance that has been configured to use Oracle Identity Cloud Service and an Oracle-managed load balancer in Oracle Cloud Infrastructure Load Balancing or Oracle Cloud Infrastructure Load Balancing Classic.



The components of Oracle Java Cloud Service and its related Oracle Cloud components that are part of the infrastructure and platform service offerings are described in the following sections.

Infrastructure Console and Infrastructure Classic Console

The Infrastructure Console and Infrastructure Classic Console (depending on your account) are components of Oracle Cloud Portal. These consoles allow account administrators and service administrators to manage and monitor their Oracle Cloud service instances and database deployments, including Oracle Java Cloud Service instances. The Infrastructure Console or Infrastructure Classic Console let administrators monitor and operate all active services within a single identity domain. Administrators can manage users and roles, and manage service notifications. See Overview of Managing Oracle Cloud Accounts and Services in *Managing and Monitoring Oracle Cloud*.

Oracle Java Cloud Service

You use the Oracle Java Cloud Service Console to create Oracle Java Cloud Service instances and perform management activities like scaling and patching.

See Explore the Oracle Java Cloud Service Console.

Databases

Each Oracle Java Cloud Service instance must be associated with a database to host the required Oracle Java Cloud Service schema.

The following databases are supported for service instances based on Oracle Cloud Infrastructure:

- Oracle Autonomous Database (Oracle Autonomous Transaction Processing)

> **Note:**
>
> Free tier Oracle Autonomous Transaction Processing database is not supported.

- Oracle Database Cloud Service (Classic)
- Oracle Cloud Infrastructure Database (DB System)

See Overview of the Database Service in the Oracle Cloud Infrastructure documentation.

The following databases are supported for service instances on Oracle Cloud Infrastructure Classic:

- Oracle Database Cloud Service (Classic)
- Oracle Database Exadata Cloud Service

An Oracle Java Cloud Service instance can optionally be associated with additional Oracle Database Cloud Service or Oracle Database Exadata Cloud Service databases for your application schemas. Oracle Autonomous Database and Oracle Cloud Infrastructure databases are not supported for application schemas.

See About Oracle Database Cloud Service in *Administering Oracle Database Cloud Service*.

Object Storage

You can configure Oracle Java Cloud Service instances to store backups in object storage. Depending on the region that you select when creating an instance, the backup location is a container in Oracle Cloud Infrastructure Object Storage Classic or a bucket in Oracle Cloud Infrastructure Object Storage.

Compute Nodes

Oracle Java Cloud Service instances are hosted on Oracle Linux 7 compute nodes. Depending on the region that you select when creating an instance, the compute nodes are in Oracle Cloud Infrastructure Compute Classic or in Oracle Cloud Infrastructure Compute.

For information about the node deployment topology that is set up and configured for you when you provision an Oracle Java Cloud Service instance, see Compute Topology for Oracle Java Cloud Service Instances.

Oracle Coherence

Oracle Coherence is an in-memory data grid and caching solution that enables organizations to predictably scale applications by providing fast access to frequently used data. When you enable Oracle Coherence for an Oracle Java Cloud Service instance, applications running on Oracle WebLogic Server can use the Coherence API to cache and retrieve data. See About Oracle Coherence in Oracle Java Cloud Service.

Oracle Identity Cloud Service

By default, the WebLogic Server domain in a service instance is configured to use the local WebLogic identity store to maintain administrators, application users, groups and roles. These security elements are used to authenticate users and to also authorize access to tools like the WebLogic Server Administration Console.

There are two types of accounts in Oracle Cloud — Traditional and those with Oracle Identity Cloud Service. If your account includes Oracle Identity Cloud Service, an Oracle Java Cloud Service instance can also use it for authentication. As a result, users that access your applications or the administration consoles are authenticated against Oracle Identity Cloud Service if they are not found in the local WebLogic identity store. See Use Oracle Identity Cloud Service with Oracle Java Cloud Service.

Load Balancer

When creating an instance in an Oracle Cloud Infrastructure region, you can provision the instance with an Oracle-managed load balancer in Oracle Cloud Infrastructure Load Balancing or Oracle Traffic Director nodes, or without any load balancer. When creating an instance in an Oracle Cloud Infrastructure Classic region, you can provision the instance with Oracle Traffic Director nodes or without any load balancer.

If you enable authentication with Oracle Identity Cloud Service for an instance during provisioning, then the instance must use an Oracle-managed load balancer running in either Oracle Cloud Infrastructure Load Balancing or Oracle Cloud Infrastructure Load Balancing Classic (depending on the region you select).

Oracle Fusion Middleware

Oracle Fusion Middleware is a portfolio of products that provide additional enterprise functionality such as web collaboration, content management, data integration and portals. Certified products can be provisioned on your Oracle Java Cloud Service instance after you create it.

Two of these products, Oracle Data Integrator and Oracle WebCenter Portal, offer tools specific to Oracle Java Cloud Service to help automate this provisioning process.

- **Oracle WebCenter Portal** adds functionality such as portlets, personalization and content management to your Oracle Java Cloud Service instance. See:
  - Introduction to Oracle WebCenter Portal (12.2.1.3)
  - Introduction to Oracle WebCenter Portal (12.2.1.2)
- **Oracle Data Integrator** adds data transformation and integration functionality to your Oracle Java Cloud Service instance. It supports high-volume batch loads and event-triggered data loads. See:
  - Introduction to Oracle Data Integrator (12.2.1.3)
  - Introduction to Oracle Data Integrator (12.2.1.2)

Also see and .

- Oracle Fusion Middleware Products Certified on Oracle Java Cloud Service
- Overview of Oracle Fusion Middleware (12.2.1.3)
- Overview of Oracle Fusion Middleware (12.2.1.2)

Oracle Developer Cloud Service

(Not available on Oracle Cloud at Customer)

Oracle Java Cloud Service comes with a complimentary instance of Oracle Developer Cloud Service, which is a cloud-based software development and collaboration platform. It provides source control, issue tracking and continuous integration capabilities. You can use Oracle Developer Cloud Service to automate the deployment of applications to Oracle Java Cloud Service.

See *Using Oracle Developer Cloud Service*.

About the Interfaces to Oracle Java Cloud Service

The entire Oracle Java Cloud Service environment, including the WebLogic domain and cluster, and the storage volumes and network settings, is visible and customizable. The following table summarizes the key interfaces to Oracle Java Cloud Service:

Type of Access	Description	More Information
Web browser	Use the Oracle Java Cloud Service Console to create service instances, and to perform lifecycle operations such as backup, restore, and patch. You can also scale a service instance using the same console.	Access Oracle Java Cloud Service Explore the Oracle Java Cloud Service Console
WebLogic Server Administration Console	Use the WebLogic Server Administration Console to deploy and undeploy Java EE applications, and to manage application users and groups.	Access the Administration Consoles for Oracle Java Cloud Service Oracle WebLogic Server 12c (12.2.1.3) Administration Console Online Help Oracle WebLogic Server 12c (12.2.1.4) Administration Console Online Help Oracle WebLogic Server 12c (12.2.1.2) Administration Console Online Help Oracle WebLogic Server 11g (10.3.6) Administration Console Online Help

Type of Access	Description	More Information
Fusion Middleware Control	Use the Oracle Enterprise Manager Fusion Middleware Control for WebLogic Server to administer your Oracle Fusion Middleware application environments (for example, deploy Oracle ADF applications).	Access the Administration Consoles for Oracle Java Cloud Service Getting Started Using Oracle Enterprise Manager Fusion Middleware Control in *Administering Oracle Fusion Middleware* 12c (12.2.1.3) Getting Started Using Oracle Enterprise Manager Fusion Middleware Control in *Administering Oracle Fusion Middleware* 12c (12.2.1.4) Getting Started Using Oracle Enterprise Manager Fusion Middleware Control in *Administering Oracle Fusion Middleware* 12c (12.2.1.2) Getting Started Using Oracle Enterprise Manager Fusion Middleware Control in *Administering Oracle Fusion Middleware* 11g (11.1.1.7)
Load Balancer Console	If load balancing is enabled for an Oracle Java Cloud Service instance, you can use the web console of the load balancer to configure it.	Configure a Load Balancer for a Service Instance
REST API	Use REST API calls to manage Oracle Java Cloud Service from a terminal, script, or custom program.	*REST API for Oracle Java Cloud Service*
Command Line Interface (CLI)	Use the CLI to manage Oracle Java Cloud Service by using a command-line or script.	psm CLI Reference
Oracle Cloud Stack Manager	Use Oracle Cloud Stack to automate the provisioning of multiple cloud services as a single unit, called a stack. Oracle Cloud Stack includes a template to create both Oracle Java Cloud Service and Oracle Database Cloud Service instances in a single operation.	Create an Oracle Java Cloud Service Instance with Cloud Stack
Secure Shell (SSH)	Access the nodes of an Oracle Java Cloud Service instance through SSH. After you use SSH to access a node, you can run WLST and other command-line applications within the node.	Access a Node with a Secure Shell (SSH)

Type of Access	Description	More Information
Virtual Network Computing (VNC)	Remotely access the graphical desktop of a node in an Oracle Java Cloud Service instance with a combination of VNC client and SSH tunnel.	Connect to a Node with VNC
WebLogic Scripting Tool (WLST)	Use WLST commands locally or remotely, in online or offline mode. • To use WLST commands locally, use SSH to connect to the node on which the Administration Server is running. Then, run the WLST commands from within the node. • To use WLST commands remotely, connect to the administration console through port `7002` (if enabled) or create an SSH tunnel to the node. Then, run the WLST commands remotely from your computer against the service instance.	Use WLST to Administer a Service Instance
Integrated Development Environment (IDE)	Deploy applications to an Oracle Java Cloud Service instance from an IDE such as Oracle Enterprise Pack for Eclipse.	Use an IDE to Deploy and Undeploy an Application
Oracle Developer Cloud Service (Not available on Oracle Cloud at Customer)	Oracle Java Cloud Service comes with a complimentary instance of Oracle Developer Cloud Service, which is a cloud-based software development and collaboration platform. It provides source control, issue tracking and continuous integration capabilities. You can use Oracle Developer Cloud Service to automate the deployment of applications to Oracle Java Cloud Service.	*Using Oracle Developer Cloud Service*

> **Note:**
>
> You provide a user name and password for the WebLogic Administrator when you create an Oracle Java Cloud Service instance. By default, the credentials used to access the WebLogic Server Administration Console and WLST are also used to access the Fusion Middleware Control and the Oracle Traffic Director console.

Before You Begin with Oracle Java Cloud Service

Before you create an Oracle Java Cloud Service instance, you may need to satisfy one or more prerequisites depending on your requirements.

Topics

- Prerequisites for Instances in Oracle Cloud Infrastructure
- Create a Database
- Create an SSH Key Pair
- Create an Object Storage Container
- Select an IP Network for a Service Instance with a Managed Load Balancer

Prerequisites for Instances in Oracle Cloud Infrastructure

Oracle Java Cloud Service instances in Oracle Cloud Infrastructure require certain networking and storage resources that you must create in Oracle Cloud Infrastructure.

To learn about these resources, see Prerequisites for Oracle Platform Services in the Oracle Cloud Infrastructure documentation.

For step-by-step instructions to create these resources, see Creating the Infrastructure Resources Required for Oracle Platform Services.

Create an SSH Key Pair

In order to use Secure Shell (SSH) to access the VMs that make up your Oracle Java Cloud Service instance, you need a public/private key pair.

Choose from one of these options:

- Let Oracle Java Cloud Service generate the keys for you as part of the process of creating a new service instance. You will be prompted to download the generated public key.
- Generate your own keys prior to creating a service instance, and then upload your public key when you create a service instance. See Generate a Key Pair with OpenSSH or Generate a Key Pair with PuTTY.

Create an Object Storage Container

If you enable backups on an Oracle Java Cloud Service instance, backups of the service instance are stored in an object storage container.

The steps for creating an object storage container for an Oracle Java Cloud Service instance vary depending on whether you create the instance in Oracle Cloud Infrastructure Classic or Oracle Cloud Infrastructure.

Oracle Cloud Infrastructure

You must create a storage bucket before you attempt to provision an Oracle Java Cloud Service instance. See Prerequisites for PaaS Services on Oracle Cloud Infrastructure in the Oracle Cloud Infrastructure documentation.

Oracle Cloud Infrastructure Classic

When you create an Oracle Java Cloud Service instance, you are prompted to supply the name of a storage container along with the credentials for an Oracle Cloud user who has read/write access to the container. You can either provide an existing storage container that you previously created, or Oracle Java Cloud Service can create the specified storage container for you.

Before you can create containers and objects, you must have an active subscription to Oracle Cloud Infrastructure Object Storage Classic. Be sure you also select a **Replication Policy** before you create your first storage container. See About Replication Policy in *Using Oracle Cloud Infrastructure Object Storage Classic*.

To create a storage container, choose from one of these options:

- Use Oracle Java Cloud Service to create the container. See Specify the Service Instance Details and refer to the **Create Storage Container** checkbox.
- Use Oracle Cloud Infrastructure Object Storage Classic to create the container. See Creating Containers in *Using Oracle Cloud Infrastructure Object Storage Classic*, or the Creating Oracle Storage Cloud Service Containers Using the REST API tutorial.

> **Note:**
>
> - A storage container is **not** required if you are creating Oracle Java Cloud Service—Virtual Image instances and using the Virtual Image service level of Oracle Database Cloud Service only.
> - Do not use a storage container that you use for backups of Oracle Java Cloud Service instances for any other purpose. For example, do not use it to back up Oracle Database Cloud Service database deployments. Using the container for multiple purposes can result in billing errors.

Select an IP Network for a Service Instance with a Managed Load Balancer

This topic does not apply to Oracle Cloud Infrastructure. Identify the Cloud Infrastructure Used by a Service Instance.

This topic does not apply to Oracle Cloud at Customer.

To select an IP network for a service instance that includes a managed load balancer, you must first attach an internet-facing load balancer to the IP network.

Enabling authentication with Oracle Identity Cloud Service when you provision a Oracle Java Cloud Service instance automatically configures an instance of Oracle Cloud Infrastructure Load Balancing Classic, a managed load balancer. In this case, you must first attach an internet-facing load balancer to the IP network, if one is not already attached. In order for traffic from outside the Cloud to reach the managed load manager, this traffic must first pass through the internet-facing load balancer.

You must create a public or private load balancer on the IP network before you provision an Oracle Java Cloud Service with a public or private load balancer, respectively.

If the IP network selected for the service instance doesn't have a load balancer, but is connected to an IP Network Exchange that has another IP network that does have a load balancer, then that load balancer will be used.

The Create New Instance wizard allows you to select an IP network when an internet-facing load balancer does not exist on the IP network, but an error occurs during the provisioning process. Create the internet-facing load balancer before you attempt to provision the Oracle Java Cloud Service instance again.

See Creating a Load Balancer in *Using Oracle Cloud Infrastructure Load Balancing Classic*.

Create a Database

You must create a database in Oracle Cloud before you provision an Oracle Java Cloud Service instance.

As part of the Oracle Java Cloud Service instance creation process, Oracle Java Cloud Service provisions the required infrastructure schemas in the selected database.

> **Note:**
>
> To ensure that you can restore the database for an Oracle Java Cloud Service instance without risking data loss for other service instances, Oracle recommends that you do not associate the same infrastructure schema database (or the same pluggable database) with multiple service instances. Backups of a database that is used with multiple Oracle Java Cloud Service instances contain data for all the instances. Therefore, if you restore the database from a backup, data for all the service instances is restored, which might not be the intended result.

The supported database services in Oracle Cloud vary by region.

Region Type	Infrastructure Schema Database Options
Oracle Cloud Infrastructure region (Not available on Oracle Cloud at Customer)	• Oracle Cloud Infrastructure Database • Oracle Autonomous Database • Oracle Database Cloud Service
Oracle Cloud Infrastructure Classic region	• Oracle Database Cloud Service • Oracle Database Exadata Cloud Service

Topics:

- Create an Oracle Autonomous Database
- Create an Oracle Cloud Infrastructure Database
- Create an Oracle Database Cloud Service Database Deployment
- Use a Database Cloud Service - Virtual Image Database Deployment
- Use an Oracle Cloud Infrastructure Database on a Different Virtual Cloud Network

Create an Oracle Autonomous Database

If you want to create an Oracle Java Cloud Service instance on Oracle Cloud Infrastructure, you can create and associate an Oracle Autonomous Database (Oracle Autonomous Transaction Processing) with the service instance.

In order to associate an Oracle Autonomous Database with a service instance, it must be running Oracle WebLogic Server 12.2.1.3 or later.

> **Note:**
>
> Free tier Oracle Autonomous Transaction Processing database is not supported.
>
> If the Oracle Autonomous Database is configured with **Virtual Cloud Network** option (private endpoint), you cannot provision an Oracle Java Cloud Service instance in a public subnet. To use the Oracle Autonomous Database configured in a private endpoint with an Oracle Java Cloud Service instance, the Oracle Java Cloud Service instance and the Oracle Autonomous Database using the private subnet should be in the same VCN. Otherwise, configure the Oracle Autonomous Database with **Allow secure access from everywhere** option (public endpoint) to use with an Oracle Java Cloud Service instance.
>
> See About Network Access Options.

Oracle Autonomous Database is fully-managed, offers high-performance, and is elastic. You have all of the performance of the Oracle Database in an environment that is tuned and optimized for transaction processing workloads.

You must create the Oracle Autonomous Database using the serverless option before you begin provisioning your Oracle Java Cloud Service instance. Note that Oracle Java Cloud Service does not yet support a dedicated deployment autonomous database.

You must create a policy in order for your Oracle Autonomous Database to be displayed in the Oracle Java Cloud Service web console.

- Specify this policy if you created the database in a custom compartment:

  ```
  Allow service PSM to inspect autonomous-database in compartment
  compartment_name
  ```

- Specify this policy if you created the database in the root compartment:

  ```
  Allow service PSM to inspect autonomous-database in tenancy
  ```

For information on creating policies, see Creating the Infrastructure Resources Required for Oracle Platform Services .

You cannot create an Oracle Java Cloud Service instance on a public subnet with a database that is configured with an access control list (ACL). You must temporarily remove the ACL from the database before creating the service instance. After creating the service instance, you can recreate the ACL, and add the public IP address of the service instance.

To create an Oracle Java Cloud Service instance on a private subnet with a database that is configured with an ACL, you must first do the following:

- Create a route rule for the private subnet that directs traffic to the database through a service gateway.
- Add the CIDR 240.0.0.0/4 to the database's ACL.

When you provision an Oracle Java Cloud Service instance by using the provisioning wizard, specify the following information:

- **Database Type:** Oracle Autonomous Database
- Compartment where the Oracle Autonomous Database resides
- PDB you created for the Oracle Autonomous Database
- Administrator username is set automatically to `ADMIN`
- Administrator's password

See the following topics in the Oracle Cloud Infrastructure documentation:

- Overview of the Database Service
- Overview of Autonomous Database
- Creating an Autonomous Database

See the tutorial Provisioning Autonomous Database

Create an Oracle Cloud Infrastructure Database

If you want to create an Oracle Java Cloud Service instance on Oracle Cloud Infrastructure, you can create and associate an Oracle Cloud Infrastructure Database with the service instance.

To use an Oracle Cloud Infrastructure Database running Oracle Database 12.2 or later, the service instance must be running Oracle WebLogic Server 12.2.1 or later.

You can use the Oracle Cloud Infrastructure console to create an Exadata-based, VM-based, or Bare Metal-based database to associate with your Oracle Java Cloud Service instance. For a 1-node VM DB system, you can use the fast provisioning option to create the database. Oracle Java Cloud Service supports using Logical Volume Manager as the storage management software for a 1-node VM DB system. See Creating a Database in the Oracle Cloud Infrastructure documentation.

You can use the Oracle Java Cloud Service console to create an instance that uses an Exadata-based or VM-based database. You must use the REST API or CLI to create an instance that uses a Bare Metal-based database.

The Oracle Cloud Infrastructure Database must be in the same region and virtual cloud network (VCN) as the Oracle Java Cloud Service instance you are creating. The instances do not need to be on the same availability domain or subnet. However, you must create the necessary security rules in the VCN to enable communication between the subnets. See VCNs and Subnets in the Oracle Cloud Infrastructure documentation.

You must create a policy in order for your Oracle Cloud Infrastructure Database to be displayed in the Oracle Java Cloud Service web console.

- Specify this policy if you created the database in a custom compartment:

  ```
  Allow service PSM to inspect database-family in compartment
  compartment_name
  ```

- Specify this policy if you created the database in the root compartment:

  ```
  Allow service PSM to inspect database-family in tenancy
  ```

For information on creating policies, see Creating the Infrastructure Resources Required for Oracle Platform Services.

When you provision an Oracle Java Cloud Service instance by using the provisioning wizard, specify the following information:

- **Database Type:** Oracle Cloud Infrastructure Database
- Compartment where the Oracle Cloud Infrastructure Database resides
- Database instance name
- Pluggable database the service instance will connect to
- Administrator user name is set automatically to `SYS`
- Administrator's password

Create an Oracle Database Cloud Service Database Deployment

Prior to creating a custom Oracle Java Cloud Service instance, use your Oracle Database Cloud Service subscription to create a database deployment.

It is not necessary to create a database deployment prior to creating an Oracle Java Cloud Service instance from a QuickStart template. See Create an Oracle Java Cloud Service Instance by Using a QuickStart Template.

For information about subscribing to Oracle Database Cloud Service, provisioning database deployments, and using Oracle RAC database deployments, see Getting Started with Database Cloud Service in *Administering Oracle Database Cloud Service*.

You can optionally associate an Oracle Java Cloud Service instance with up to four additional Oracle Database Cloud Service deployments (or pluggable databases) in order to access your application schemas. This feature is not available for service instances that use the Oracle Java Cloud Service - Virtual Image (BASIC) service level.

Note the following limitations to service instances that use Oracle Database Cloud Service as the infrastructure schema database:

- When creating an Oracle Java Cloud Service instance on a secondary Oracle Identity Cloud Service instance, you can't use an Oracle Database Cloud Service deployment for the infrastructure schema. Instead, you must use an Oracle Cloud Infrastructure Database or Oracle Autonomous Database. When creating an Oracle Java Cloud Service on the primary Oracle Identity Cloud Service instance, you can use an Oracle Database Cloud Service deployment for the infrastructure schema.
- You cannot use an Oracle Database Cloud Service deployment running Oracle Database 18c.

- You can use an Oracle Database Cloud Service deployment running Oracle Database 12.2, but only for service instances running Oracle WebLogic Server 12.2.1 or later.
- Create Oracle Database Cloud Service deployments with a backup option other than `NONE`. This configuration enables Oracle Java Cloud Service to coordinate backups across your service instance and the database. Coordinated backups are not supported for other database services.
- For service instances on Oracle Cloud Infrastructure, the Oracle Database Cloud Service deployment must be in the same region and virtual cloud network (VCN) as the Oracle Java Cloud Service instance. The instances do not need to be on the same availability domain or subnet.
- For service instances on Oracle Cloud Infrastructure Classic, the Oracle Database Cloud Service deployment must be in the same region as the Oracle Java Cloud Service instance.

When you provision an Oracle Java Cloud Service instance by using the provisioning wizard, specify the following information:

- **Database Type:** Oracle Database Cloud Service (Classic)
- Name of a running database deployment
- Pluggable database name (for Oracle Database 12c only)
- Database administrator user name and password
- Connection string to the database deployment (for Virtual Image service level only)
- Application schemas (Optional)

Similar to Oracle Java Cloud Service, Oracle Database Cloud Service supports a standard service level and a Virtual Image service level. The following table summarizes the compatibility between these service levels.

Service Level	Database Cloud Service	Database Cloud Service—Virtual Image
Oracle Java Cloud Service	Supported • This service level must be used if you intend to use an Oracle Real Application Clusters (RAC) database. When creating the database deployment, make sure you select the database edition called **Enterprise Edition - Extreme Performance**. • When creating the database deployment, make sure you do not set the **Backup Destination** to None; instead you should select one of the other available backup options.	Not supported
Oracle Java Cloud Service—Virtual Image	Supported	Supported

Create a Custom Pluggable Database (PDB)

After you create an Oracle Database Cloud Service deployment, you can manually create a custom pluggable database (PDB) for that database deployment. You can then create an

Oracle Java Cloud Service instance based on the custom PDB that you created, rather than on the default PDB.

Topics:

- Before You Begin Creating a Custom PDB
- Create the Custom PDB
- Change the Database Wallet Type
- Configure TDE on the New PDB

Before You Begin Creating a Custom PDB

To create a custom PDB, you must first create an Oracle Database Cloud Service deployment.

See Create an Oracle Database Cloud Service Database Deployment.

Create the Custom PDB

After you create an Oracle Database Cloud Service deployment, create a custom PDB.

To create a custom PDB:

1. SSH to the database's VM.

   ```
   ssh-i <private_key> opc@<database_VM_IP>
   ```

2. Become `oracle` user.

   ```
   sudo su oracle
   ```

3. Connect as root user and get the location of the data files.

   ```
   $sqlplus / as sysdba
   SQL> selectfile_name from dba_data_files where tablespace_name =
   'SYSTEM';
   FILE_NAME
   --------------------------------------------------------------------
   ------------
   /u02/app/oracle/oradata/ORCL/system01.dbf

   SQL> exit
   ```

4. Make a directory for the new PDB data files in `/u02/app/oracle/oradata/ORCL`.

   ```
   mkdir -p /u02/app/oracle/oradata/ORCL/PDB2
   ```

5. Connect as root user again.

   ```
   $sqlplus / as sysdba
   ```

6. Disable restricted session.

```
SQL> alter system disable restricted session;
```

7. Create PDB2 as clone of PDBSEED.

```
SQL> create pluggable database pdb2 admin user pdb2admin identified by
Welcome_1  roles = (DBA)
>FILE_NAME_CONVERT=('/u02/app/oracle/oradata/ORCL/pdbseed/', '/u02/app/
oracle/oradata/ORCL/PDB2/');
create pluggable database pdb2 admin user pdb2admin identified by
Welcome_1 roles = (DBA)FILE_NAME_CONVERT=('/u02/app/oracle/oradata/ORCL/
pdbseed/', '/u02/app/oracle/oradata/ORCL/PDB2/')
*
ERROR at line1:ORA-65005:
missing or invalid filename pattern for file-
/u04/app/oracle/oradata/temp/pdbseed_temp012017-04-25_03-33-20-PM.dbf
```

8. From error message in the previous step, get the temp file name and use it in `file_name_convert`.

```
SQL> create pluggable database pdb2 admin user pdb2admin identified by
Welcome_1  roles = (DBA)
> FILE_NAME_CONVERT=('/u02/app/oracle/oradata/ORCL/pdbseed/', '/u02/app/
oracle/oradata/ORCL/PDB2/',
> '/u04/app/oracle/oradata/temp/pdbseed_temp012017-04-25_03-33-20-
PM.dbf', '/u04/app/oracle/oradata/temp/pdb2_temp.dbf');

Pluggable database created.
SQL> show pdbs;
    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
        2  PDB$SEED                       READ ONLY  NO
        3  PDB1                           READ ONLY  NO
        4  PDB2                           MOUNTED
```

9. Open the new PDB in READ WRITE mode.

```
SQL> alter pluggable database pdb2 open;
Pluggable database altered.
SQL> show pdbs;
    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY  NO
         3 PDB1                           READ ONLY  NO
         4 PDB2                           READ WRITE NO
```

10. Test the connection with new PDB2.

```
SQL> connect sys/Welcome_1@localhost:1521/
pdb2.opcwlaasqa.oraclecloud.internal as sysdba
Connected.
SQL> show pdbs
    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
```

```
---------- ------------------------------ ---------- ----------
         4 PDB2                           READ WRITE NO
```

Change the Database Wallet Type

Change the container database wallet type from AUTO_LOGIN to PASSWORD.

The encryption wallet keystore is of type AUTO_LOGIN instead of PASSWORD in the CDB. This is the default state after Oracle Database Cloud Service deployment provisioning. In order to be able to open the keystore in the new PDB and generate the master encryption key for that PDB, you must change the wallet type to PASSWORD in the container.

Note that the encryption wallet is located at: `/u01/app/oracle/admin/ORCL/tde_wallet`. Check the `sqlnet.ora` file located in the `$ORACLE_HOME/network/admin` path.

From `sqlnet.ora`:

```
ENCRYPTION_WALLET_LOCATION = (SOURCE=(METHOD=FILE)
(METHOD_DATA=(DIRECTORY=/u01/app/oracle/admin/ORCL/tde_wallet)))
```

Use the following script to change the database wallet type.

```
# Remove the auto-open wallet cwallet.sso physically:
$ cd/u01/app/oracle/admin/ORCL/tde_wallet
$ mv cwallet.sso cwallet.sso.bkp

$ sqlplus / as sysdba
SQL> select* from v$encryption_wallet;
WRL_TYPE
--------------------
WRL_PARAMETER
-----------------------------------------------------------------------
---------
STATUS                         WALLET_TYPE          WALLET_OR
FULLY_BAC CON_ID
------------------------------ -------------------- ---------
--------- ----------
FILE
/u01/app/oracle/admin/ORCL/tde_wallet/
OPEN                            AUTOLOGIN            SINGLE
NO        0


SQL> alter system set wallet close;
# If the preceeding command does not work,
# try closing wallet by specifying sys user password
# with the following command:
SQL> ALTER SYSTEM SET ENCRYPTION WALLET CLOSE IDENTIFIED BY
"MyPassword_1";

# This will close the removed auto-open wallet in the database,
# then open the password based wallet and retry the original Set Key
```

```
statement:
SQL> SELECT WRL_PARAMETER, STATUS, WALLET_TYPE FROM V$ENCRYPTION_WALLET;
WRL_PARAMETER
-----------------------------------------------------------------------------
---
STATUS                         WALLET_TYPE
------------------------------ --------------------
/u01/app/oracle/admin/ORCL/tde_wallet/
CLOSED                         UNKNOWN

SQL> administer key management set keystore openidentified by "Welcome_1";
keystore altered.
SQL> SELECT WRL_PARAMETER, STATUS, WALLET_TYPE FROM V$ENCRYPTION_WALLET;
WRL_PARAMETER
-----------------------------------------------------------------------------
---
STATUS                         WALLET_TYPE
------------------------------ --------------------
/u01/app/oracle/admin/ORCL/tde_wallet/
OPEN                           PASSWORD
```

Configure TDE on the New PDB

After you have changed the database wallet type, you can configure Oracle Transparent Data Encryption (TDE) on the new PDB.

To configure TDE on the new PDB:

1. Connect to PDB2.

   ```
   SQL> alter session set container=PDB2;
   Pluggable database altered.
   ```

2. Open the keystore in that PDB and generate master encryption key for the PDB.

   ```
   SQL> SELECT WRL_PARAMETER, STATUS, WALLET_TYPE FROM V$ENCRYPTION_WALLET;
   WRL_PARAMETER
   --------------------------------------------------------------------------
   ------
   STATUS WALLET_TYPE
   ------ -----------
   /u01/app/oracle/admin/ORCL/tde_wallet/
   CLOSED UNKNOWN

   SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY
   "Welcome_1";
   keystore altered.
   SQL> ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY "Welcome_1" with
   backup;
   keystore altered.
   SQL> SELECT WRL_PARAMETER, STATUS, WALLET_TYPE FROM V$ENCRYPTION_WALLET;
   WRL_PARAMETER
   --------------------------------------------------------------------------
   ------
   STATUS WALLET_TYPE
   ```

```
------ -----------
/u01/app/oracle/admin/ORCL/tde_wallet/
OPEN   PASSWORD
```

3. Create encrypted tablespace.

```
SQL> create tablespace enc128_ts
datafile '/u02/app/oracle/oradata/ORCL/PDB2/Test_encrption.dbf'
size 1M autoextend on next 1M
encryption using 'AES128'
default storage (encrypt);
```

4. Verify the new tablespace is encrypted.

```
SQL> select tablespace_name , encrypted from dba_tablespaces;
TABLESPACE_NAME ENC
------------------------------ –
SYSTEM NO
SYSAUX NO
TEMP NO
ENC128_TS YES

SQL> exit
```

5. Verify that the new `pdb2.<network_domain>` service is up.

```
[oracle@<user_name> opc]$ lsnrctl status
LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 07-NOV-2017
19:00:39
Copyright (c)1991, 2017, Oracle. All rights reserved.
Connecting to(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<host_name>)
(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias LISTENER
Version TNSLSNR for Linux: Version 12.1.0.2.0 - Production
Start Date 06-NOV-2017 17:56:44
Uptime 1 days 1 hr. 3 min. 55 sec
Trace Level off Security ON: Local OS Authentication
SNMP OFF Listener Parameter File /u01/app/oracle/product/12.1.0/
dbhome_1/network/admin/listener.ora
Listener Log File /u01/app/oracle/diag/tnslsnr/<user_name>/listener/
alert/log.xml
Listening Endpoints Summary...
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=<host_name>)(PORT=1521)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=<host_name>)(PORT=5500))
(Security=(my_wallet_directory=/u01/app/oracle/admin/ORCL/
xdb_wallet))(Presentation=HTTP)(Session=RAW))
Services Summary... Service "<service_name>" has 1 instance(s).
Instance "ORCL", status READY, has 1 handler(s) for this service...
Service "<service_name>" has 1 instance(s).
Instance "ORCL", status READY, has 1 handler(s) for this service...
Service "pdb1.<network_domain>" has 1 instance(s).
```

```
Instance "ORCL", status READY, has 1 handler(s) for this service...
Service "pdb2.<network_domain>" has 1 instance(s).
Instance "ORCL", status READY, has 1 handler(s) for this service...
The command completed successfully
```

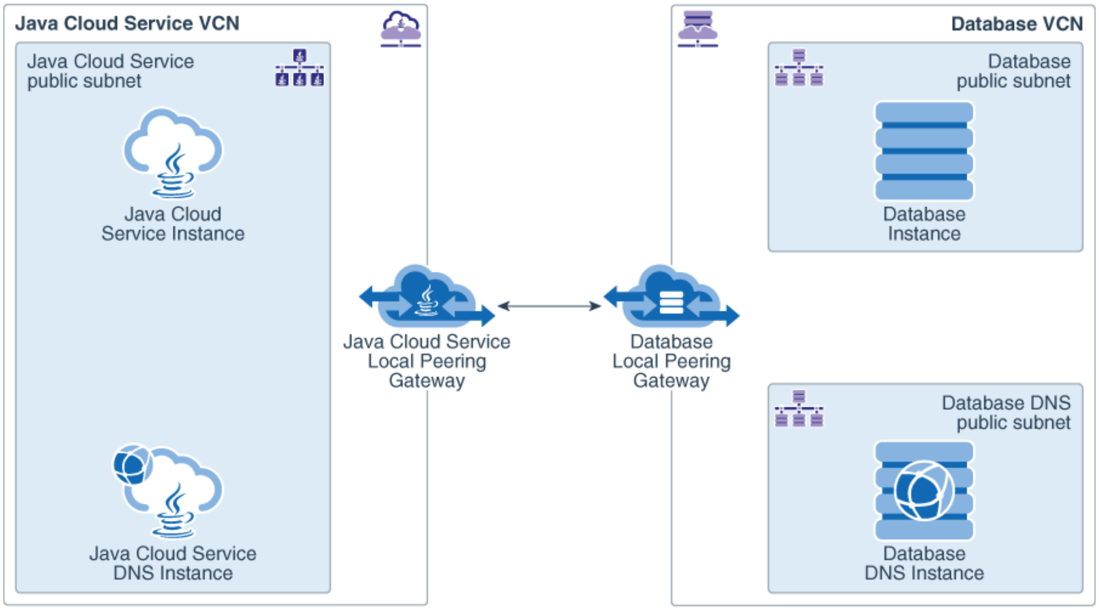
You can now specify your custom PDB when you use the Oracle Java Cloud Service console or the REST API to provision an Oracle Java Cloud Service instance.

Use an Oracle Cloud Infrastructure Database on a Different Virtual Cloud Network

If you want to connect an Oracle Java Cloud Service instance to an Oracle Cloud Infrastructure Database in the same region but in a different Virtual Cloud Network (VCN), then you must configure VCN peering in Oracle Cloud Infrastructure.

As shown in the following illustration, the networking configuration consists of:

- A VCN with a public subnet for the Oracle Java Cloud Service instance and a custom DNS resolver
- A VCN with two public subnets, one for the Oracle Cloud Infrastructure Database instance and the other for a custom DNS resolver
- Two local peering gateways (LPGs)


Java Cloud Service VCN
Java Cloud Service public subnet
Java Cloud Service Instance
Java Cloud Service DNS Instance
Java Cloud Service Local Peering Gateway
Database Local Peering Gateway
Database VCN
Database public subnet
Database Instance
Database DNS public subnet
Database DNS Instance

You must also create these supporting network resources: internet gateways, route table rules, security lists, and dynamic host configuration protocol (DHCP) resources. The VCNs and their resources must be in the same compartment.

If instead of public subnets, you want to use private subnets for the service instance and database, you must create these additional network resources:

- A bastion compute instance on a public subnet so that you can access the private subnet with a secure shell (SSH)
- A NAT gateway so that you can download and install OS packages on the custom DNS resolver
- A service gateway so that the Oracle Java Cloud Service instance can access object storage for backup and restoration (not applicable to the database VCN)

See Create an Oracle Java Cloud Service Instance Attached to a Private Subnet on Oracle Cloud Infrastructure.

To configure the network topology shown in the illustration:

1. Create two VCNs with non-overlapping Classless Inter-Domain Routing (CIDR) in the same region.

   See VCNs and Subnets. You will add subnets to these VCNs later.

2. Create or edit the following resources **in each VCN**.

   a. Create a local peering gateway to allow communication between the resources by using private IP addresses.

      See Local VCN Peering (Within Region).

   b. Create an internet gateway to enable direct connectivity to the internet.

      See Internet Gateway.

   c. Edit the default route table and add a route table rule to enable traffic to flow via the internet gateway. Ensure you select **Internet Gateway** in the **Target Type** and set the destination to `0.0.0.0/0`.

      See Route Tables.

   d. Edit the default security list and create the following ingress and egress rules to control the traffic for your VCN.

Table 1-1 Security list rules

Rule Type	Source Type	Source CIDR	IP Protocol	Source Port Range	Destination Port	Type and Code
Ingress	**CIDR**	*<JCS_VCN_CIDR>*	**UDP**	All	53 (DNS port)	–
Ingress	**CIDR**	*<Database_VCN_CIDR>*	**UDP**	All	53 (DNS port)	–
Ingress	**CIDR**	0.0.0.0/0	**TCP**	All	22 (SSH port)	–
Ingress	**CIDR**	0.0.0.0/0	**ICMP**	–	–	3
Egress	**CIDR**	0.0.0.0/0	**TCP**	All	All	-

See Security Lists.

3. Go to the Oracle Java Cloud Service LPG, click **Establish Connection** and select the Oracle Cloud Infrastructure Database VCN.

4. To enable traffic to flow between the LPGs, create and configure the following resources **in each VCN**.

   a. **Route Rules:** In the default route table, create a route table rule. Select the **Local Peering Gateway** target type, set the destination to the CIDR of the VCN you peered, and select the LPG you created earlier.

      See Route Tables.

b. **Subnets:** Create a public subnet for your VCN using the security list you created earlier. Ensure that the VCN is assigned to a DHCP option whose **DNS Type** is **INTERNET AND VCN RESOLVER**.

   See VCNs and Subnets.

c. **Compute:** Create two compute instances, one using the public subnet you created in the Oracle Java Cloud Service VCN, and the other using the public subnet you created in the Oracle Cloud Infrastructure Database VCN. Select the latest Oracle Linux 7.6 image and write down the provisioned public and private IP address of both custom DNS resolvers. For each compute instance, you must run the following commands.

   i. To open an `SSH` connection, run the following command and replace the `<private_key>` and `<public_IP_address>` placeholders with your own values.

```
$ ssh -i <private_key> opc@<public_IP_address>
```

   ii. Switch to the root user.

```
$ sudo su
```

   iii. Install the BIND tool.

```
$ yum install bind
```

   iv. To allow DNS traffic, open the UDP port `53` on local firewall by running the following commands.

```
$ firewall-cmd --permanent --add-port=53/udp
$ firewall-cmd --permanent --add-port=53/tcp
$ /bin/systemctl restart firewalld
```

   v. Edit the `/etc/named.conf` file.

```
$ vi /etc/named.conf
```

   vi. Replace the `<db_vcn_cidr>`, `<jcs_vcn_cidr>`, `<dbvcn_dns_domain_name>`, `<private_IP_address>`, and `<jcsvcn_dns_domain_name>` placeholders with your own values.
   Example of the `/etc/named.conf` file in the Oracle Java Cloud Service DNS.

```
options {
        listen-on port 53 { any; };
        allow-query    { localhost; <db_vcn_cidr>; 
<jcs_vcn_cidr>; };
        forward        only;
        forwarders     { 169.254.169.254; };
        recursion yes;
};
zone "<dbvcn_dns_domain_name>" {
        type       forward;
        forward    only;
        forwarders { <private_IP_address>; };
};
```

```
zone "<jcsvcn_dns_domain_name>" {
        type       forward;
        forward    only;
        forwarders { 169.254.169.254; };
};
```

Example of the `/etc/named.conf` file in the Oracle Cloud Infrastructure Database DNS.

```
options {
        listen-on port 53 { any; };
        allow-query    { localhost; <db_vcn_cidr>;
<jcs_vcn_cidr; };
        forward        only;
        forwarders     { 169.254.169.254; };
        recursion yes;
};
zone "<jcsvcn_dns_domain_name>" {
        type       forward;
        forward    only;
        forwarders { <private_IP_address>; };
};
zone "<dbvcn_dns_domain_name>" {
        type       forward;
        forward    only;
        forwarders { 169.254.169.254; };
};
```

vii. Restart the service.

```
$ service named restart
```

d. **DHCP Options:** Create a DHCP option whose **DNS Type** is **CUSTOM RESOLVER**. Specify the private IP address of the DNS on the compute instance in your VCN and `169.254.169.254` as DNS Servers.

   See DHCP Options.

e. Associate the public subnet with the DHCP option you created.

5. Create a public subnet in the Oracle Cloud Infrastructure Database VCN. Ensure that your Oracle Cloud Infrastructure Database public subnet is associated with a DHCP option whose **DNS Type** is **INTERNET AND VCN RESOLVER**.

After you create, configure, and peer your VCNs:

- You can create your Oracle Cloud Infrastructure Database instance using the public subnet that you created earlier.
- You can create your Oracle Java Cloud Service instance using the appropriate subnet and Oracle Cloud Infrastructure Database. For the instructions to create an Oracle Java Cloud Service instance, see Create an Oracle Java Cloud Service Instance Attached to a Public Subnet on Oracle Cloud Infrastructure.

About Oracle Java Cloud Service Subscriptions and Licenses

Oracle Java Cloud Service supports multiple subscription types. There are also opportunities to reuse existing on-premises licenses with Oracle Java Cloud Service.
For subscription and license prices, see https://cloud.oracle.com/java/pricing.

Topics

- Subscriptions
- Leveraging On-Premises Licenses

Subscriptions

This topic does not apply to Oracle Cloud at Customer.

You can obtain subscriptions to Oracle Java Cloud Service in several different ways.

- Free Promotion subscription

  You can sign up for a 30–day Oracle Cloud promotion and receive free credits. This promotion applies to eligible Oracle Infrastructure as a Service (Oracle IaaS) and Platform as a Service (Oracle PaaS) services.

  See Requesting and Managing Free Oracle Cloud Promotions in *Getting Started with Oracle Cloud*.
- Universal Credits subscription

  In the Universal Credits subscription model, you commit to pay a certain amount up-front annually, based on a monthly cost estimate. Supports the Bring Your Own License type.

  See About Universal Credits and Buying an Oracle Cloud Subscription in *Getting Started with Oracle Cloud*.
- Non-metered subscription

  A non-metered subscription has a fixed monthly charge.

  See Buying a Nonmetered Subscription to an Oracle Cloud Service in *Getting Started with Oracle Cloud*.
- Government subscription

  A non-metered subscription designed for government customers. You buy resources for each service separately and access only those services you've purchased. Supports the Bring Your Own License type.

  See Billing Models Offered in *Infrastructure and Platform Services (IaaS/PaaS) Billing Guide*.
- Traditional metered subscription

  In a metered subscription, you are only charged for the resources you use per month.

  See Buying a Traditional Metered Subscription to an Oracle Cloud Service in *Getting Started with Oracle Cloud*.

Leveraging On-Premises Licenses

- The **Bring Your Own License** (BYOL) option enables you to bring your on-premises Oracle WebLogic Server licenses to Oracle Cloud. BYOL instances are billed at a lower rate than other instances. See Frequently Asked Questions: Oracle BYOL to PaaS. Before you scale up or scale out a BYOL instance, you must have enough WebLogic Server licenses for the additional OCPUs that will be allocated to the instance after it is scaled.
- You can leverage your on-premises Oracle Fusion Middleware licenses for Oracle Java Cloud Service. Certified products can be provisioned on your Oracle Java Cloud Service instance after you create it. See Oracle Applications Certified on Oracle Java Cloud Service.

How to Begin with Oracle Java Cloud Service Subscriptions

This topic does not apply to Oracle Cloud at Customer.

Obtain a subscription to Oracle Java Cloud Service before signing into Oracle Cloud and accessing the console.

1. Sign up for a free credit promotion or purchase a subscription. Refer to these topics in *Getting Started with Oracle Cloud*:
   - Requesting and Managing Free Oracle Cloud Promotions
   - Buying an Oracle Cloud Subscription
2. Access the Oracle Java Cloud Service console.

   See Access Oracle Java Cloud Service.
3. Optional: Create additional Oracle Cloud users and grant them access to Oracle Java Cloud Service.

   See Add Users, Assign Policies and Roles in *Getting Started with Oracle Cloud*.

To learn more about the roles related to Oracle Java Cloud Service see About Oracle Java Cloud Service Roles and User Accounts.

If your cloud account also includes Oracle Identity Cloud Service, see Use Oracle Identity Cloud Service with Oracle Java Cloud Service.

Access Oracle Java Cloud Service

You access Oracle Java Cloud Service through the web console, REST API or command line interface (CLI).

Depending on how you signed up for Oracle Cloud, you'll be directed to either the Oracle Cloud Infrastructure Console or the Oracle Cloud Infrastructure Classic Console.

Topics

- Access Oracle Java Cloud Service from the Infrastructure Console
- Access Oracle Java Cloud Service from the Infrastructure Classic Console

- Access Oracle Java Cloud Service from Oracle Cloud at Customer

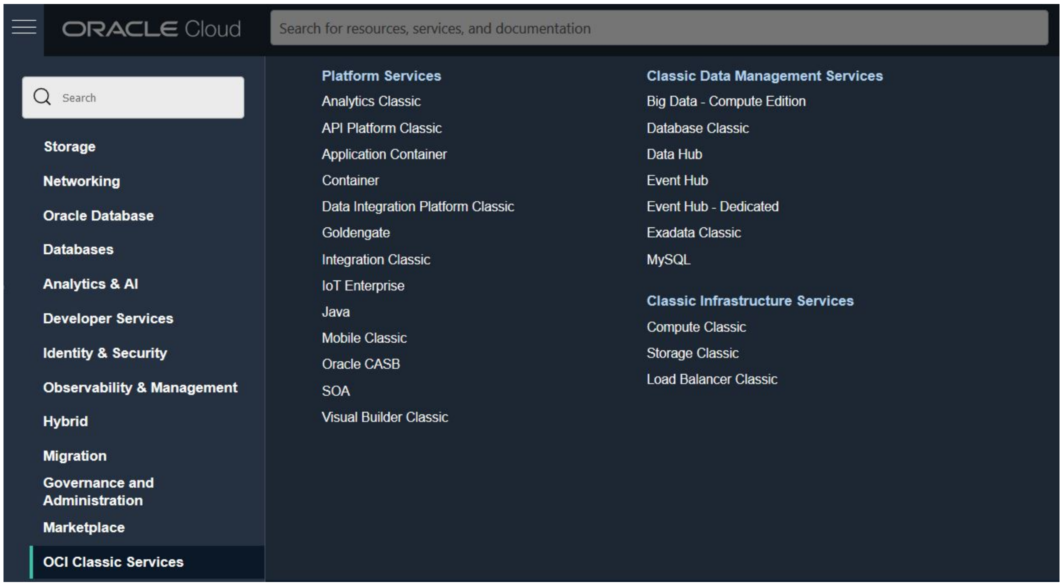
Access Oracle Java Cloud Service from the Infrastructure Console

On most Oracle Cloud accounts, you access the Oracle Java Cloud Service console from the Oracle Cloud Infrastructure Console.

1. Sign in to Oracle Cloud.

   If you received a welcome email, use it to identify the URL, your user name, and your temporary password. After signing in, you will be prompted to change your password.

2. From the Infrastructure Console, click the navigation menu ☰ in the top left corner, and select **OCI Classic Services**. Under **Platform Services**, click **Java**.



3. When you access the Oracle Java Cloud Service console the first time, you see the Welcome page. Click **Instances** or **Go to Console**.

4. From the Instances page, you can create a new Oracle Java Cloud Service, or you can click an existing instance to view or manage it.

To view help for the current page, click the help icon ? at the top of the page.

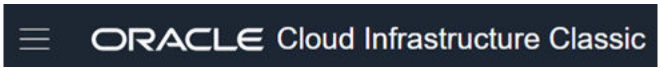
Access Oracle Java Cloud Service from the Infrastructure Classic Console

On some older Oracle Cloud accounts, you access the Oracle Java Cloud Service console from the Oracle Cloud Infrastructure Classic Console.

1. Sign in to Oracle Cloud.

   If you received a welcome email, use it to identify the URL, your user name, and your temporary password. After signing in, you will be prompted to change your password.

2. From the Infrastructure Classic Console, click the navigation menu ☰ in the top left corner, and then click **Java**.



ORACLE Cloud Infrastructure Classic

3. When you access the Oracle Java Cloud Service console the first time, you see the Welcome page. Click **Instances** or **Go to Console**.
4. From the Instances page, you can create a new Oracle Java Cloud Service, or you can click an existing instance to view or manage it.

Access Oracle Java Cloud Service from Oracle Cloud at Customer

On Oracle Cloud at Customer, you access the Oracle Java Cloud Service console from the My Services Dashboard.

1. Sign in to Oracle Cloud at Customer.
2. From the My Services Dashboard, click the navigation menu ≡ in the top left corner, and then click **Java**.

ORACLE Cloud My Services

3. When you access the Oracle Java Cloud Service console the first time, you see the Welcome page. Click **Instances** or **Go to Console**.
4. From the Instances page, you can create a new Oracle Java Cloud Service, or you can click an existing instance to view or manage it.

Typical Workflow for Using Oracle Java Cloud Service

To start using Oracle Java Cloud Service, refer to the following tasks as a guide.

Task	Description	More Information
Sign up for a free credit promotion or purchase a subscription	Provide your information, and sign up for a free trial or purchase a subscription to Oracle Java Cloud Service.	How to Begin with Oracle Java Cloud Service Subscriptions
Add and manage users and roles	Optionally create additional accounts for your cloud users and assign the necessary Oracle Java Cloud Service roles.	Add Users, Assign Policies and Roles in *Getting Started with Oracle Cloud* About Oracle Java Cloud Service Roles and User Accounts
Access the service console	Open the service console after you have signed in.	Access Oracle Java Cloud Service
Create a service instance	Create an instance that meets you precise requirements.	About Creating an Oracle Java Cloud Service Instance About Java Cloud Service Instances in Oracle Cloud Infrastructure

Task	Description	More Information
Enable access to the administration consoles in your service instance	By default (that is, unless you enabled access during instance creation), access to the WebLogic Server Administration Console, Fusion Middleware Control and Load Balancer Console is blocked for security purposes. Enable the corresponding access rules. For instances in Oracle Cloud Infrastructure, the required access rules are enabled automatically.	Enable Console Access for a Service Instance
Create users for the service instance in Oracle Identity Cloud Service	If you created a custom service instance and enabled authentication with Oracle Identity Cloud Service, then you can use Oracle Identity Cloud Service to create additional WebLogic Server users and to secure applications.	Use Oracle Identity Cloud Service with Oracle Java Cloud Service
Deploy applications to the service instance	Use the WebLogic Server Administration Console, the Fusion Middleware Control, WebLogic Scripting Tool commands, or an IDE to deploy and undeploy applications.	Deploy and Undeploy Applications for an Oracle Java Cloud Service Instance
View runtime metrics for a service instance	Access the service metrics graph to view heap usage metrics or request response times (if a load balancer is present).	View the Service Metrics for an Oracle Java Cloud Service Instance
Monitor the service and account balance	Check on the day-to-day operation of your service, monitor OCPU hours, view service details, and access control panels and associated tools.	Topic Overview in *Managing and Monitoring Oracle Cloud*
Patch the service instance	Apply a patch or roll back a patch.	Apply a Patch Roll Back a Patch
Back up the service instance	Initiate on-demand backups, schedule automated backups, set up retention policies and storage for backups, download backups, and manage backups (restore, archive and delete).	Back Up and Restore an Oracle Java Cloud Service Instance
Scale the service instance	Add or remove nodes in preparation for increased or reduced load on a service instance. Change the shape of a node or add storage to a node. When Oracle Coherence is enabled for a service instance: Add or remove Coherence data tier nodes to increase or decrease cache capacity.	About Scaling an Oracle Java Cloud Service Cluster Scale In a Cluster About Scaling an Oracle Java Cloud Service Node Scale Automatically Scale Out a Coherence Data Grid Scale In a Coherence Data Grid
Delete the service instance	Delete a service instance when it’s no longer necessary.	Delete an Oracle Java Cloud Service Instance

About Oracle Java Cloud Service Roles and User Accounts

Oracle Java Cloud Service uses roles to control access to tasks and resources. A role assigned to a user gives certain privileges to the user.

In addition to the roles and privileges described in Learn About Cloud Account Roles in *Getting Started with Oracle Cloud*, the Java Administrator role (`JaaS_Administrator`) is also created for Oracle Java Cloud Service.

When your cloud account is first set up, the service administrator is given the Java Administrator role along with additional service roles that are required to work with Oracle Java Cloud Service. Other users in your account must be assigned these same roles in order to use Oracle Java Cloud Service. Only the identity domain administrator is allowed to create user accounts and assign roles.

Topics:

- Java Administrator
- Related Service Administrators
- Service Instance Users
- Oracle Cloud Infrastructure Policies

Java Administrator

The primary role in Oracle Java Cloud Service is Java Administrator.

The following table summarizes the privileges given to the Java Administrator role.

Description of Privilege	More Information
Can create and delete service instances	Manage the Life Cycle of Oracle Java Cloud Service Instances
Can stop and start service instances, and virtual machines	Stop, Start, and Restart an Oracle Java Cloud Service Instance and Individual Nodes
Can suspend and enable service instances by disabling and enabling the load balancer	Suspend an Oracle Java Cloud Service Instance
Can scale, patch, and back up or restore service instances	Scale an Oracle Java Cloud Service Instance Patch an Oracle Java Cloud Service Instance Back Up and Restore an Oracle Java Cloud Service Instance
Can administer load balancers for service instances	Administer the Load Balancer for an Oracle Java Cloud Service Instance
Can administer the Coherence data tier for service instances	Use Oracle Coherence in Oracle Java Cloud Service
Can monitor and manage service usage in Oracle Cloud	Overview of Managing Oracle Cloud Accounts and Services in *Managing and Monitoring Oracle Cloud*

Related Service Administrators

The following table summarizes the privileges given to other related service administrator roles in Oracle Cloud.

Role	Privileges
`Compute_Operations`	Create Oracle Java Cloud Service instances on Oracle Cloud Infrastructure Classic regions.
`DBaaS_Administrator`	Create and manage Oracle Database Cloud Service deployments. A database deployment must exist prior to creating an Oracle Java Cloud Service instance, unless you create the service instance by using a QuickStart template. See Create an Oracle Java Cloud Service Instance by Using a QuickStart Template.
`Storage_ReadWriteGroup`	Enable backups for an Oracle Java Cloud Service instance, and store the backups in an existing Oracle Cloud Infrastructure Object Storage Classic container.
`Storage_Administrator`	Create Oracle Cloud Infrastructure Object Storage Classic containers to use as backup storage locations for Oracle Java Cloud Service instances.

Service Instance Users

Learn about the operating system and Oracle WebLogic Server administrative user accounts that are created when you create an Oracle Java Cloud Service instance.

User	Description	More Information
OS User	The opc user has root privileges on the OS running on the nodes in a service instance and can: - Connect to a node through SSH for direct OS-level access - Create other OS accounts on a node The oracle user cannot be used to connect to a node through SSH. It has regular OS user permissions and can also access the Oracle product installations on the node. Note that there are no default passwords for either the opc or oracle user. SSH access to the node by the opc user is based on the public key provided at the time the service instance was provisioned. The OS user accounts are not stored or managed in Oracle Cloud.	Access a Node with a Secure Shell (SSH)
WebLogic Administrator	Can manage Oracle WebLogic Server in Oracle Java Cloud Service Can access and use the WebLogic Server Administration Console Can manage users and groups in the embedded LDAP Can configure other identity providers Can deploy and undeploy applications using the WebLogic Server Administration Console	Access the Administration Consoles for Oracle Java Cloud Service Use the WebLogic Server Administration Console to Deploy and Manage Applications Oracle WebLogic Server 12c (12.2.1.3) Administration Console Online Help Oracle WebLogic Server 12c (12.2.1.4) Administration Console Online Help Oracle WebLogic Server 12c (12.2.1.2) Administration Console Online Help Oracle WebLogic Server 11g (10.3.6) Administration Console Online Help

Oracle Cloud Infrastructure Policies

Learn about how to create and manage resources in Oracle Cloud Infrastructure, administrators define policies that grant privileges to users and groups.

To create and manage resources in Oracle Cloud Infrastructure, administrators define policies that grant privileges to users and groups. For example, to create a database for use with Oracle Java Cloud Service in either an Oracle Autonomous Database or Oracle Cloud Infrastructure database, an administrator must create policies that grant you access to these services. See Securing IAM in the Oracle Cloud Infrastructure documentation.

In order to create Oracle Java Cloud Service instances in an Oracle Cloud Infrastructure region, an administrator must create policies that grant specific privileges to Oracle Java Cloud Service.

For example, the administrator must specify the following policy to grant Oracle Java Cloud Service access to Oracle Autonomous Database or Oracle Cloud Infrastructure database:

- Oracle Autonomous Database

  ```
  Allow service PSM to inspect autonomous-database in compartment
  Autonomous Transaction Processing database compartment
  ```

- Oracle Oracle Cloud Infrastructure database

  ```
  Allow service PSM to inspect database-family in compartment Oracle Cloud
  Infrastructure database compartment
  ```

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

About Java Cloud Service Instances in Oracle Cloud Infrastructure

When you create an Oracle Java Cloud Service instance, you can choose the infrastructure that the instance must use: Oracle Cloud Infrastructure or Oracle Cloud Infrastructure Classic. The Oracle WebLogic Server environment that the instance provides in either type of infrastructure is substantially the same. A few differences exist in the supported capabilities and the instance-creation workflows.

Topics:

- Workflow for Creating an Instance in Oracle Cloud Infrastructure
- Differences Between Instances in Oracle Cloud Infrastructure and Oracle Cloud Infrastructure Classic
- Migrating to Oracle Cloud Infrastructure

Workflow for Creating an Instance in Oracle Cloud Infrastructure

Task	More Information
Task 1: Understand the differences between instances created in Oracle Cloud Infrastructure and in Oracle Cloud Infrastructure Classic. Knowing these differences will help you select an appropriate region while creating your Oracle Java Cloud Service instance.	Differences Between Instances in Oracle Cloud Infrastructure and Oracle Cloud Infrastructure Classic
Task 2: Create the required network, object storage, and security resources in the Oracle Cloud Infrastructure region where you intend to create the Oracle Java Cloud Service instance.	**Documentation**: Prerequisites for Oracle Platform Services on Oracle Cloud Infrastructure **Tutorial**: Creating the Infrastructure Resources Required for Oracle Platform Services

Task	More Information
Task 3: Create an Oracle Cloud database instance in the region where you created the Oracle Cloud Infrastructure resources. This database is required to store the schema required for Oracle Java Cloud Service.	Use the appropriate documentation, depending on the Oracle Database service that you want to associate with the Oracle Java Cloud Service instance: **Oracle Database Cloud Service**: Create an Oracle Database Cloud Service Database Deployment **Oracle Cloud Infrastructure Database**: Managing DB Systems Oracle Autonomous Database: Provisioning Autonomous Database in *Using Oracle Autonomous Database on Shared Exadata Infrastructure*
Task 4: Create the Oracle Java Cloud Service instance in the same Oracle Cloud Infrastructure region and virtual cloud network (VCN) as the Oracle Cloud database instance.	Use the appropriate documentation, depending on whether you want to attach the Oracle Java Cloud Service instance to a public subnet or a private subnet: Create an Oracle Java Cloud Service Instance Attached to a Public Subnet on Oracle Cloud Infrastructure Create an Oracle Java Cloud Service Instance Attached to a Private Subnet on Oracle Cloud Infrastructure

Differences Between Instances in Oracle Cloud Infrastructure and Oracle Cloud Infrastructure Classic

Feature	Oracle Cloud Infrastructure	Oracle Cloud Infrastructure Classic
Availability domains	Each region has multiple isolated availability domains, with separate power and cooling. The availability domains within a region are interconnected using a low-latency network. When creating an instance, you can select the availability domain that the instance should be placed in.	Not applicable
Compute shapes	VM.Standard and BM.Standard shapes **Note**: The shapes available might vary by region.	Standard and high memory shapes **Note**: The shapes available might vary by region.
Database options for the infrastructure schema	• Oracle Autonomous Database • Oracle Cloud Infrastructure Database • Oracle Database Cloud Service	• Oracle Database Cloud Service • Oracle Database Exadata Cloud Service
Associating an instance with a RAC database for high availability	Use Oracle Cloud Infrastructure Database	Use Oracle Database Cloud Service

Feature	Oracle Cloud Infrastructure	Oracle Cloud Infrastructure Classic
Networking	You *must* attach each instance to a subnet, in a virtual cloud network created in Oracle Cloud Infrastructure. If you specify a private subnet, the nodes of the instance can't be assigned public IP addresses.	You *can* attach instances to IP networks defined in Oracle Cloud Infrastructure Compute Classic. For such instances, you can choose to not assign public IP addresses.
Public IP addresses	You can’t reserve and assign public IP addresses. The addresses are auto-assigned when the service instance is created, but won’t change when you restart the nodes or the instance.	You can reserve public IP addresses and assign them while creating an instance, or you can have the addresses auto-assigned. With either choice, the public IP addresses won’t change when you restart the nodes or the instance.
Private IP addresses	The private IP addresses are auto-assigned when the service instance is created. The addresses *will not change* when you restart the nodes or the instance.	The private IP addresses are auto-assigned when the service instance is created. The addresses *might change* when you restart the nodes or the instance. For an instance attached to an IP network, when you restart a node, you can assign a fixed private IP address.
Cloning	Supported for all service instances, except instances associated with Oracle Cloud Infrastructure Database or Oracle Autonomous Database.	Supported for all service instances.
Restrictions when using colocated snapshots	No restrictions for colocated snapshots.	Colocated snapshots (stored in block storage) impose the following restrictions: • You can’t delete or scale-in an instance that has a snapshot. • You can’t delete a snapshot if a clone created from it exists. • You can’t create a snapshot of a clone.
Scaling a cluster automatically	Not supported	Supported
Adding block storage	You can perform the add-storage operation up to 29 times for a node. In each operation, you can add 50 GB or a multiple of 50.	You can perform the add-storage operation up to 6 times for a node. In each operation, you can add from 1 to 2048 GB.
Managing access rules	Configure security rules using the Oracle Cloud Infrastructure interfaces.	Use the Oracle Java Cloud Service interfaces to configure access rules.

Feature	Oracle Cloud Infrastructure	Oracle Cloud Infrastructure Classic
Load balancer options available within Oracle Java Cloud Service	While creating an instance, if you enable Oracle Identity Cloud Service as the identity provider, an Oracle-managed load balancer in Oracle Cloud Infrastructure Load Balancing is created and configured automatically for the instance. If you don't enable Oracle Identity Cloud Service, then you can create an instance with an Oracle-managed load balancer or an Oracle Traffic Director load balancer. Or you can choose not to configure a load balancer. If you don't add any load balancer at the time of instance provisioning, you can add an Oracle Traffic Director load balancer to the existing instance later. You cannot use the Oracle Java Cloud Service Console to add an Oracle-managed load balancer to an existing service instance. You cannot use the Oracle Java Cloud Service Console to remove an Oracle-managed load balancer from an existing service instance. You can use the Oracle Java Cloud Service REST API to remove an Oracle-managed load balancer from an existing service instance.	While creating an instance, if you enable Oracle Identity Cloud Service as the identity provider, an Oracle-managed load balancer is created and configured automatically for the instance. If you don't enable Oracle Identity Cloud Service, then you can add Oracle Traffic Director as a user-managed load balancer, either while creating the instance or later. Or you can choose not to configure a load balancer. You cannot use the Oracle Java Cloud Service Console to add an Oracle-managed load balancer to an existing service instance. You cannot use the Oracle Java Cloud Service Console or REST API to remove an Oracle-managed load balancer from an existing service instance.
Object storage for backups	You must create the object storage bucket in Oracle Cloud Infrastructure before creating the instance.	You can create the object storage container either before or during instance creation.
Virtual Image service level	Not supported	Supported
Changing the database association for an instance	Not supported	After creating an instance, you can change the infrastructure database (Oracle Database Cloud Service deployment) that the instance is associated with.

Migrating to Oracle Cloud Infrastructure

If you provisioned an Oracle Java Cloud Service instance on an Oracle Cloud Infrastructure Classic region, tools are available to help you migrate the service instance to an Oracle Cloud Infrastructure region.

See Migrating Oracle Java Cloud Service Instances to Oracle Cloud Infrastructure.

Compare Oracle Cloud Services for Deploying Java Applications

Choose an Oracle Cloud service that best meets the needs of your Java application and development process.

Oracle offers two main cloud services that support Java deployments: Oracle Java Cloud Service and Oracle Application Container Cloud Service. In general, Oracle Java Cloud Service provides a Java solution that is more flexible and customizable, while Oracle Application Container Cloud Service offers a simpler, automated and managed solution for Java applications.

Both services share common capabilities:

- Host your application in a highly-available environment
- Easily scale your application in response to changing capacity requirements
- Cache and retrieve frequently-used data
- Automate deployment though REST APIs, CLI commands, or Oracle Developer Cloud Service

There are important differences between the services:

- Oracle Application Container Cloud Service supports Java Standard Edition applications and Java Enterprise Edition web applications (WAR). Oracle Java Cloud Service supports the full Java EE specification, including enterprise applications (EAR) and Java Message Service (JMS).
- With Oracle Application Container Cloud Service, you can deploy applications that are developed in a variety of languages, including Java, PHP, Python, and Ruby.
- Oracle Application Container Cloud Service cannot be deployed to Oracle Cloud Infrastructure regions. Oracle Java Cloud Service supports both Oracle Cloud Infrastructure and Oracle Cloud Infrastructure Classic regions.
- Oracle Java Cloud Service gives administrators access to Oracle WebLogic Server and the operating system. Oracle Application Container Cloud Service hides this infrastructure from users, and automatically keeps it up-to-date with the latest software and patches.
- When you create an Oracle Java Cloud Service instance, you choose from a list of specific Oracle WebLogic Server releases, including older ones like 11g. With Oracle Application Container Cloud Service, you don’t have to worry about the details of the container.
- Deploying your code to Oracle Application Container Cloud Service is fast and easy, but Oracle Java Cloud Service also integrates with popular Integrated Development Environments (IDE).
- Oracle Java Cloud Service offers tools to automate the migration of existing Oracle WebLogic Server environments to the cloud.
- With Oracle Application Container Cloud Service, you can quickly integrate your Java application with other Oracle Cloud resources like databases and message queues. Oracle Java Cloud Service does not offer a similar data binding feature for your applications, but does provide out-of-the-box integration with Oracle Database Cloud Service, Oracle Cloud Infrastructure Database, and Oracle Autonomous Database.

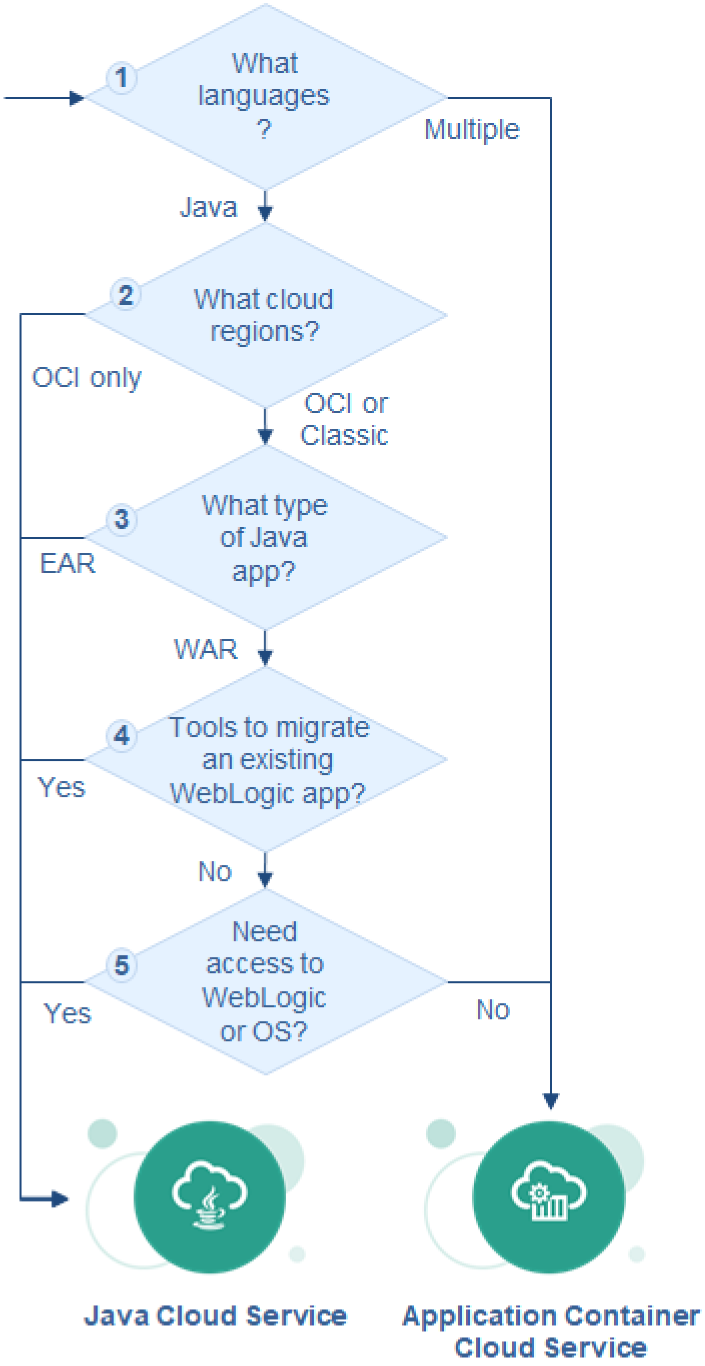
If neither of these services meets your exact requirements, you can create basic compute instances or containers in Oracle Cloud:

- Oracle Cloud Infrastructure Compute
- Oracle Cloud Infrastructure Compute Classic
- Oracle Cloud Infrastructure Container Service Classic
- Oracle Container Engine for Kubernetes
- Oracle Weblogic Server Kubernetes Operator

These infrastructure cloud solutions give you the most flexibility, but you must install, configure, and maintain all of the Java software components.

Decision Tree

Answer the following series of questions to help you choose between Oracle Java Cloud Service and Oracle Application Container Cloud Service.



1. In which language(s) is your application written?

   If the components of your application are written in multiple languages, then use Oracle Application Container Cloud Service.

2. Which regions are available in your Oracle Cloud account?

   If your account has access to Oracle Cloud Infrastructure regions only, then use Oracle Java Cloud Service. Oracle Cloud Infrastructure regions include **us-phoenix-1**, **us-ashburn-1**, **ca-toronto-1eu-frankfurt-1**, and **uk-london-1**.

3. What type of Java EE application are you developing or migrating?

If your application is packaged as an Enterprise Application (EAR), then use Oracle Java Cloud Service.

4. Are you migrating an existing Oracle WebLogic Server application? Would you prefer tools to help automate the migration of your applications and supporting resources?

   If your answer is yes, then use Oracle Java Cloud Service.

5. Do you require administrative access to Oracle WebLogic Server or the operating system, in order to customize the default configuration?

   If your answer is yes, then use Oracle Java Cloud Service.

   If your answer is no, then use Oracle Application Container Cloud Service.

2

Create an Oracle Java Cloud Service Instance

This section describes how you can create an Oracle Java Cloud Service instance with methods ranging from push-button automation to fully custom instance creation, on either the Oracle Cloud Infrastructure Classic or Oracle Cloud Infrastructure platforms.

Topics:

- About Creating an Oracle Java Cloud Service Instance
- About Life Cycle Management of Oracle Java Cloud Service Instances
- Design Considerations for an Oracle Java Cloud Service Instance
- Create an Oracle Java Cloud Service Instance by Using a QuickStart Template
- Create an Oracle Java Cloud Service Instance Attached to a Public Subnet on Oracle Cloud Infrastructure
- Create an Oracle Java Cloud Service Instance Attached to a Private Subnet on Oracle Cloud Infrastructure
- Create a Custom Oracle Java Cloud Service Instance on Oracle Cloud Infrastructure Classic
- Create an Oracle Java Cloud Service Instance with Cloud Stack
- About the Sample Application Deployed to an Oracle Java Cloud Service Instance

About Creating an Oracle Java Cloud Service Instance

There are several ways in which you can create an Oracle Java Cloud Service instance, depending on your requirements and experience level.

Choose from one of the following instance creation methods:

Create Method	More Information
Create a service instance by using a QuickStart template. This method also creates the required Oracle Database Cloud Service instance. This method is not available for Oracle Cloud accounts that include only Oracle Cloud Infrastructure regions, or include a mix of Oracle Cloud Infrastructure and Oracle Cloud Infrastructure Classic regions.	Create an Oracle Java Cloud Service Instance by Using a QuickStart Template

Create Method	More Information
Create a service instance attached to a public subnet in an Oracle Cloud Infrastructure region that meets your precise specifications, and associate it with an existing Oracle Autonomous Database, Oracle Cloud Infrastructure database, or Oracle Database Cloud Service deployment. The Oracle Java Cloud Service instance will be accessible from the public internet.	About Java Cloud Service Instances in Oracle Cloud Infrastructure Create an Oracle Java Cloud Service Instance Attached to a Public Subnet on Oracle Cloud Infrastructure Design Considerations for an Oracle Java Cloud Service Instance
Create a private Oracle Java Cloud Service instance in an Oracle Cloud Infrastructure region, along with a public load balancer.	Create an Oracle Java Cloud Service Instance Attached to a Private Subnet on Oracle Cloud Infrastructure Design Considerations for an Oracle Java Cloud Service Instance
Create a service instance in an Oracle Cloud Infrastructure Classic region that meets your precise specifications, and associate it with an existing Oracle Database Cloud Service or Oracle Database Exadata Cloud Service deployment.	Create a Custom Oracle Java Cloud Service Instance on Oracle Cloud Infrastructure Classic Design Considerations for an Oracle Java Cloud Service Instance
Take a snapshot to create a point-in-time image of a service instance, then quickly create clones of the instance.	About Snapshots and Clones
Use the Oracle Java Cloud Service REST API to create and manage WebLogic Server instances on Oracle Cloud.	REST API for Oracle Java Cloud Service
You can create a service instance by using the Command Line Interface (CLI), which is a wrapper for the REST API. After you have created the service instance, you can use the CLI to perform lifecycle operations such as scaling, patching, and backup.	PaaS Service Manager Command Line Interface Reference

About Life Cycle Management of Oracle Java Cloud Service Instances

With a few clicks of the mouse, you can create a WebLogic Server production environment in the cloud that is based on best practices, optimized for high performance and reliability, and is integrated with your infrastructure schema database and Oracle Cloud Infrastructure Object Storage.

When you create an Oracle Java Cloud ServiceOracle Java Cloud Service instance, you create and configure a Oracle Fusion Middleware Infrastructure domain with the resources defined in the following table.

Resources	Description
Administration Server	Operates as the central control entity for the configuration of the entire domain. It maintains the domain's configuration documents and distributes changes in the configuration documents to Managed Servers. Each Oracle Java Cloud Service instance has one server instance that hosts the Administration Server.

Resources	Description
Managed Servers	Host business applications, application components, Web services, and their associated resources. When creating a service instance, you can configure up to four Managed Servers, then scale out, as needed. Each Oracle Java Cloud Service instance has one or more Managed Servers, each hosted on its own Virtual Machine (node). By default, the Managed Servers are named as follows: `first8charsOfDomainName_wls_n` (where `n` starts with 1 and is incremented by 1 for each additional Managed Server to to ensure that the names are unique).
Cluster	Consists of multiple Managed Servers running simultaneously and working together to provide increased scalability and reliability. In a cluster, most resources and services are deployed identically to each Managed Server (as opposed to a single Managed Server), enabling failover and load balancing. A cluster is configured automatically for a production-level service instance. By default, the cluster name will be generated from the first eight characters of the Oracle Java Cloud Service instance name using the following format: `first8charsOfServiceInstanceName_cluster`.
Load Balancer	It is recommended that you enable a load balancer when you configure more than one Managed Server in your environment. Enabling the load balancer is optional.

When Oracle Coherence is enabled for a service instance, additional resources related to Coherence are defined in a domain.

Resources	Description
Managed Servers (Coherence data tier, storage-enabled)	Each Oracle Java Cloud Service—Coherence instance has a Coherence data tier cluster, in which one or more Virtual Machines (nodes) can have one or more Managed Servers each. By default, the storage-enabled Managed Servers are named as follows: `first8charsOfDomainName_server_n_DG` (where `n` is a number that's incremented by 1 for each additional Managed Server to guarantee unique names). The storage-enabled Managed Servers are responsible for storing and distributing data (both primary and backup) on the cluster. Coherence artifacts (such as Coherence configuration files, POF serialization classes, filters, entry processors, and aggregators) are packaged as a GridARchive (GAR) and deployed on the Managed Servers. Note that when you stop or start a service instance, all the nodes for the Managed Servers on the Coherence data tier will also stop or start. If stopped, all data in the Coherence cache will be lost.
Managed Servers (Application tier, storage-disabled)	The storage-disabled Managed Servers (identified by the name format `first8charsOfDomainName_server_n`) in the first WebLogic Server cluster host Coherence applications (cache clients), and are not responsible for storing data. Clients in the application tier are deployed as EARs. Coherence artifacts (such as Coherence configuration files, POF serialization classes, filters, entry processors, and aggregators) are packaged as a GridARchive (GAR) and deployed within an EAR.

Resources	Description
Cluster (Coherence data tier)	A second WebLogic Server cluster is configured in the domain for storing and distributing data. The Coherence data tier cluster is associated with the Coherence cluster `DataGridConfig`. The cluster members are storage-enabled by default. By default, the cluster name will be generated from the first eight characters of the service instance name using the following format: *`first8charsOfServiceInstanceName`*`_DGCluster`.
Cluster (Application tier)	The first WebLogic Server cluster (identified by the name format *`first8charsOfServiceInstanceName`*`_cluster`) is referred to as the application tier cluster. The cluster is also associated with the Coherence cluster `DataGridConfig`, and the cluster members are storage-disabled by default.
Coherence Cluster	The system-level resource (`CoherenceClusterSystemResource`) has the default name `DataGridConfig`. Both the application tier WebLogic Server cluster (storage-disabled) and the data tier WebLogic Server cluster (storage-enabled) are associated with the Coherence cluster.

For more information about WebLogic domains, see:

- Oracle Fusion Middleware 12.2.1: WebLogic Server Domains in *Understanding Oracle WebLogic Server*
- Oracle Fusion Middleware 11.1.1.7: Understanding Oracle WebLogic Server Domains in *Understanding Domain Configuration for Oracle WebLogic Server*.
- When Oracle Coherence is enabled for a service instance: (Oracle Fusion Middleware 12.2.1) Configuring and Managing Coherence Clusters in *Administering Clusters for Oracle WebLogic Server*.

After the Oracle Java Cloud Service instance is created, the Administration Server in the domain is started automatically. You can deploy applications and manage the domain resources using the standard administration tools, including Enterprise Manager Fusion Middleware Control, Oracle WebLogic Server Administration Console, Oracle WebLogic Scripting Tool (WLST), Node Manager, and Oracle Traffic Director Console.

You can stop, start, or restart a service instance or individual nodes by using the Oracle Java Cloud Service Console, PaaS Service Manager CLI, or REST API. For example, you can stop service instances or individual server nodes to stop metering for these resources, or restart the Administration Server or individual server nodes if reboot is needed.

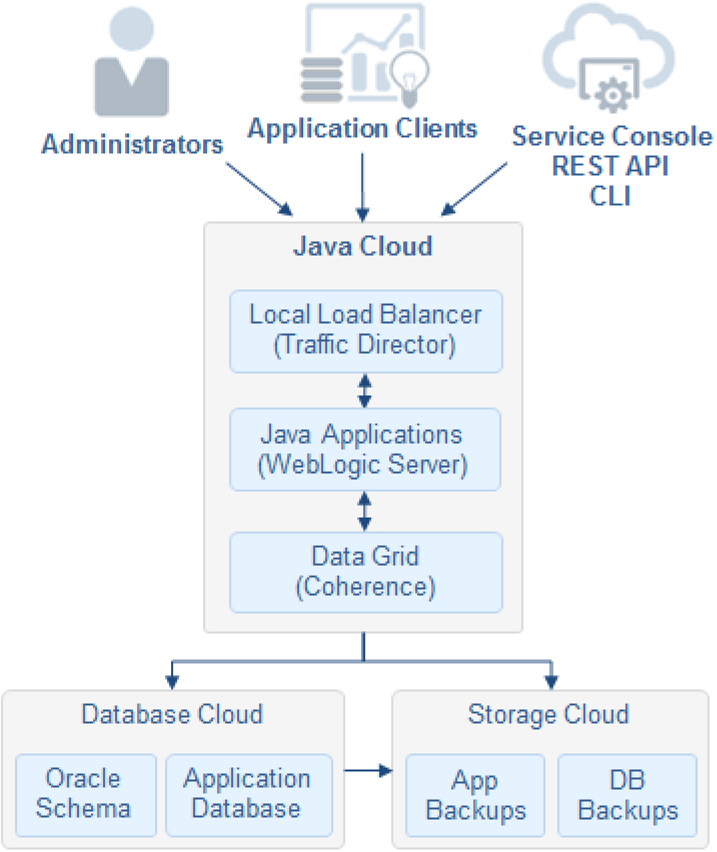
A sample application is deployed automatically when the service instance is created. For more information, see About the Sample Application Deployed to an Oracle Java Cloud Service Instance.

> **Note:**
>
> If you extend your domain using the administration tools, for example, to add an additional cluster, you are responsible for maintaining those additional resources.

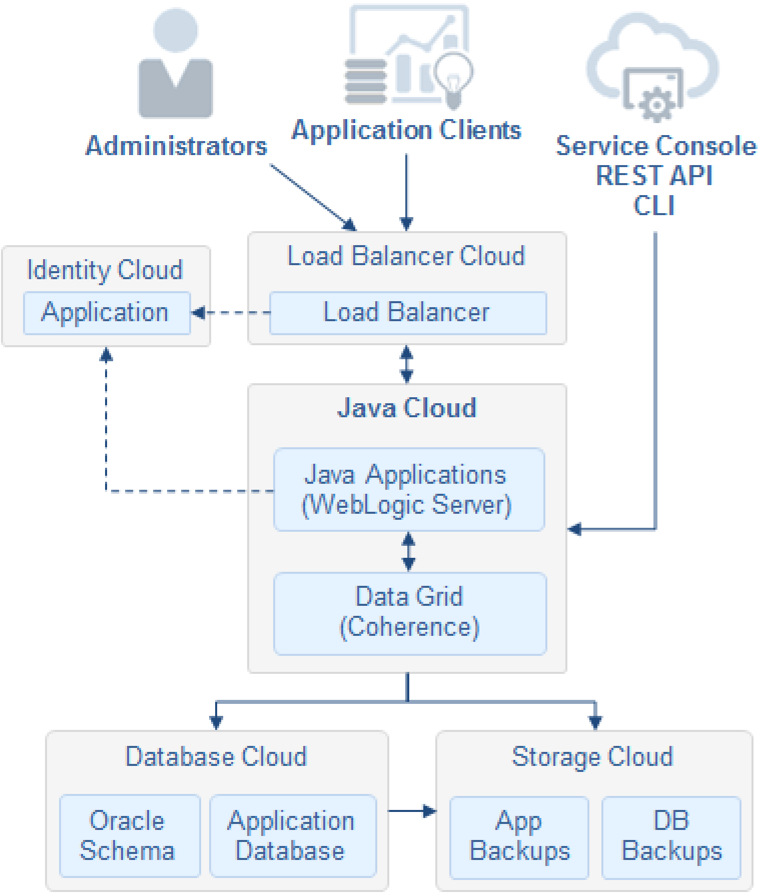
Design Considerations for an Oracle Java Cloud Service Instance

Before creating a custom Oracle Java Cloud Service instance, there are details you should consider in order to create the service instance that best meets your requirements.

This figure illustrates the components that make up a typical service instance:



The next figure illustrates a service instance that has been configured to use Oracle Identity Cloud Service and an Oracle-managed load balancer running in Oracle Cloud Infrastructure Load Balancing or Oracle Cloud Infrastructure Load Balancing Classic.



Topics

- Service Level
- Software Release
- Edition
- License
- Region
- Compute Shape
- WebLogic Cluster
- Availability Domain
- Subnet
- IP Network
- Public IP Address
- Reserved IP Address
- Domain Partition
- User Authentication
- Administrator Access

- Client Access
- Coherence Data Tier
- Database
- Load Balancer
- Backup Location

Service Level

You can select one of these service levels.

- Oracle Java Cloud Service

  This service level supports Oracle Java Cloud Service instance creation and monitoring; domain partitions; backup and restoration; patching; cloning; and scaling.

- Oracle Java Cloud Service Virtual Image (BASIC)

  This service level supports Oracle Java Cloud Service instance creation and monitoring only. It does not support backup and restoration; patching; cloning; or scaling. You cannot provision a domain partition if you specify this service level.

  This service level is:

  - Not supported if you have a Universal Credits subscription. This option does not appear on the console.
  - Supported if you have a traditional metered or non-metered subscription
  - Not supported for Oracle Cloud Infrastructure regions

  Oracle recommends using Oracle Java Cloud Service rather than Oracle Java Cloud Service Virtual Image for better flexibility, administrative control, and availability of new features.

- Oracle Java Cloud Service Fusion Middleware — Oracle WebCenter Portal

  Leverages your Oracle WebCenter Portal license on Oracle Java Cloud Service. Choosing this option downloads additional installation tools. You must install the product yourself after creating this service instance. This service level is supported on WebLogic Server release 12.2.1.3 only.

- Oracle Java Cloud Service Fusion Middleware — Oracle Data Integrator

  Leverages your Oracle Data Integrator license on Oracle Java Cloud Service. Choosing this option downloads additional installation tools. You must install the product yourself after creating this service instance. This service level is supported on WebLogic Server release 12.2.1.3 only.

Patching is not supported for service instances where Oracle Java Cloud Service Fusion Middleware—Oracle WebCenter Portal, Oracle Java Cloud Service Fusion Middleware—Oracle Data Integrator, or any other product that modifies the `MW_HOME` directory are installed. If you attempt to patch a service instance where any of these products are installed, patching prechecks issue an error message and patching fails.

Software Release

You can select one of these Oracle WebLogic Server releases.

- Oracle WebLogic Server 11g (11.1.1.7) — See *Introducing Oracle WebLogic Server*

- Oracle WebLogic Server 12c (12.2.1.2) — (Available only on Oracle Cloud at Customer) See *Understanding Oracle WebLogic Server*
- Oracle WebLogic Server 12c (12.2.1.3) — See *Understanding Oracle WebLogic Server*
- Oracle WebLogic Server 12c (12.2.1.4) — (Not available on Oracle Cloud at Customer) See *Understanding Oracle WebLogic Server* and *What's New in Oracle WebLogic Server*

With Oracle Java Cloud Service you can easily apply patches to an existing service instance. You can also upgrade an existing service instance to Oracle WebLogic Server 12c (12.2.1.3).

For service instances in Oracle Cloud Infrastructure and Oracle Cloud Infrastructure Classic regions, the Fusion Middleware service level is supported only for Oracle WebLogic Server 12c (12.2.1.3). For service instances on Oracle Cloud at Customer, the Fusion Middleware service level is supported only for Oracle WebLogic Server 12c (12.2.1.2).

The Virtual Image (BASIC) service level is not supported for service instances running Oracle WebLogic Server 12c (12.2.1.2) or later.

Oracle Java Cloud Service has a provisioning policy that aligns with the WebLogic Server error correction support policy. Service instance provisioning for a given release ends on the same day as the error correction end date for the corresponding WebLogic Server release. This policy is specific to the provisioning of WebLogic Server instances via Oracle Java Cloud Service, and has no impact on the use of these WebLogic Server releases within on-premises environments or within Oracle Cloud IaaS environments. See Oracle Fusion Middleware Lifetime Support Policy and Error Correction Support Dates for Oracle WebLogic Server.

Edition

You can choose one of these Oracle WebLogic Server editions.

- Standard Edition
- Enterprise Edition
- Enterprise Edition with Coherence (Suite)

Certain WebLogic Server capabilities are only supported in specific editions. To learn about these editions see About Oracle WebLogic Server Editions Available for Oracle Java Cloud Service.

If you select the Oracle Java Cloud Service for Fusion Middleware service level, you cannot select Standard Edition.

License

When you create a service instance, you choose a license type based on the Oracle Java Cloud Service entitlements in your Oracle Cloud account.

The **Bring Your Own License** (BYOL) option enables you to bring your on-premises Oracle WebLogic Server licenses to Oracle Cloud. BYOL instances are billed at a lower rate than other instances. See Frequently Asked Questions: Oracle BYOL to PaaS.

Before you scale up or scale out a BYOL instance, you must have enough WebLogic Server licenses for the additional OCPUs that will be allocated to the instance after it is scaled.

You can also change the license type of an existing instance. (Not available on Oracle Cloud at Customer)

Region

If your identity domain is enabled for regions, you can select a region in which your Oracle Java Cloud Service instance will reside.

A region supports either Oracle Cloud Infrastructure or Oracle Cloud Infrastructure Classic. For a list of available regions, see Data Regions for Platform and Infrastructure Services.

When you select an Oracle Cloud Infrastructure region for a service instance, you must also select an Availability Domain. See Regions and Availability Domains in the Oracle Cloud Infrastructure Services documentation.

When you select an Oracle Cloud Infrastructure Classic region for a service instance, you can also select an IP Network and assign reserved IP addresses to your nodes. If you don’t explicitly select a region (**No Preference**), you cannot select an IP network or use reserved IPs.

Compute Shape

The available shapes for a service instance depend on the type of region that you select. The larger the compute shape, the greater the processing power and the more memory that is available.

Some shapes might not be available in all regions.

If you select an Oracle Cloud Infrastructure region, the `VM.Standard` and `BM.Standard` shapes are supported. The `DenseIO` and `HighIO` shapes are unsupported. See Overview of the Compute Service in the Oracle Cloud Infrastructure Services documentation.

If you select an Oracle Cloud Infrastructure Classic region, Oracle Java Cloud Service provides a set of compute shapes that are optimized for different use cases. Choose from a set of all-purpose and memory-intensive shapes.

All-purpose compute shapes in Oracle Cloud Infrastructure Classic include:

- OC3: 1 OCPU and 7.5 GB memory
- OC4: 2 OCPUs and 15 GB memory
- OC5: 4 OCPUs and 30 GB memory
- OC6: 8 OCPUs and 60 GB memory
- OC7: 16 OCPUs and 120 GB memory
- OC8: 24 OCPUs and 180 GB memory (Not available on Oracle Cloud at Customer)
- OC9: 32 OCPUs and 240 GB memory (Not available on Oracle Cloud at Customer)

Memory-intensive compute shapes in Oracle Cloud Infrastructure Classic include:

- OC1M: 1 OCPUs and 15 GB memory
- OC2M: 2 OCPU and 30 GB memory
- OC3M: 4 OCPUs and 60 GB memory

- OC4M: 8 OCPUs and 120 GB memory
- OC5M: 16 OCPUs and 240 GB memory

See About Shapes in *Using Oracle Cloud Infrastructure Compute Classic* and About JVM Heap Settings.

For a Universal Credits subscription, you will be billed at the Pay-as-you-go rate when you exceed your monthly or annual maximum credit.

WebLogic Cluster

A WebLogic cluster is defined by a compute shape and server count.

You select an initial cluster size of 1, 2, or 4 Managed Servers. In general, the larger the cluster the more application requests that can be processed by your service instance. However, with Oracle Java Cloud Service you can also scale in and out the cluster after you create the service instance.

Another design consideration when selecting the cluster size is continued availability during patching. If the cluster has 2 or more nodes, then during patching, at least 1 node continues to serve requests. This won't be possible with a 1-node cluster.

If you create a service instance with an Oracle-managed load balancer (Oracle Cloud Infrastructure Load Balancing or Oracle Cloud Infrastructure Load Balancing Classic), you can optionally create up to 8 clusters for the instance. You cannot create multiple clusters for service instances that include a user-managed load balancer (Oracle Traffic Director). You configure each cluster with its own compute shape and initial server count (1, 2, or 4 Managed Servers). You might consider creating multiple clusters if, for example, multiple applications or different tiers of your application have different capacity requirements. See Recommended Multi-Tier Architecture in *Administering Clusters for Oracle WebLogic Server*.

Optionally, you can specify a path prefix for a cluster, which is used to configure the load balancer. For example, the load balancer could route traffic from URLs with the prefix `/mystore` to the cluster `cluster1`. If you do not specify a path prefix, then the path prefix is the cluster name.

For more information about clusters see:

- WebLogic Server Clustering in *Understanding Oracle WebLogic Server (12.2.1)*
- Understanding WebLogic Server Clustering in *Using Clusters for Oracle WebLogic Server (10.3.6)*

Availability Domain

This feature is specific to Oracle Cloud Infrastructure regions.

An availability domain consists of a set of data centers within an Oracle Cloud Infrastructure region.

A region can have multiple isolated availability domains with separate power and cooling, for example. The availability domains within a region are interconnected via a low-latency network. See Regions and Availability Domains in the Oracle Cloud Infrastructure Services documentation.

Subnet

This feature is specific to Oracle Cloud Infrastructure regions.

A subnet is a subdivision of a cloud network. Each subnet exists in a single availability domain and consists of a contiguous range of IP addresses that do not overlap with other subnets in the cloud network.

You can create your own subnet before you provision an Oracle Java Cloud Service instance. See VCNs and Subnets in the Oracle Cloud Infrastructure Services documentation.

For convenience, if you do not explicitly select a subnet (**No Preference**), then the service instance is assigned to a subnet in the predefined Virtual Cloud Network (VCN) named `svc-vcn`, which is found in the compartment named `ManagedCompartmentForPaaS`. You cannot modify these predefined subnets, such as assigning a custom security list. If you prefer more control over the network configuration for your service instance, then create a custom subnet.

You must satisfy certain subnet and policy prerequisites when you create a subnet for use with Oracle Java Cloud Service instances. See Prerequisites for PaaS Services on Oracle Cloud Infrastructure in the Oracle Cloud Infrastructure Services documentation.

IP Network

This feature is specific to Oracle Cloud Infrastructure Classic regions.

If you select a specific Oracle Cloud Infrastructure Classic region for your service instance, then you can also select an IP network in that region. Using an IP network gives you more control over the configuration of the network in which your service instance is placed.

By default, if you select an IP network, each underlying node is auto-assigned a public and private IP address. As a result, the IP address might change each time a service instance is started. To assign fixed public IP addresses to instances attached to the IP network, you can create and use IP reservations.

When you select an IP network during provisioning, you must also select a Oracle Database Cloud Service instance that is on an IP network. If the Oracle Java Cloud Service and Oracle Database Cloud Service are attached to different IP networks, then the two IP networks must be connected to the same IP network exchange. The required access rules for the Oracle Java Cloud Service instance and Oracle Database Cloud Service database deployment to communicate are created automatically.

If you want to create a service instance that uses an IP network and also includes an Oracle-managed load balancer running on Oracle Cloud Infrastructure Load Balancing Classic, you must first attach an Internet-facing load balancer to the IP network. A service instance uses an Oracle-managed load balancer when you enable authentication with Oracle Identity Cloud Service.

See:

- Select an IP Network for a Service Instance with a Managed Load Balancer
- Creating an IP Network in *Using Oracle Cloud Infrastructure Compute Classic* (Ignore information in this topic about the Compute API and orchestrations)

Public IP Address

You can choose whether or not to assign public IP addresses to the nodes in your service instance.

By default, any node that is created during instance provisioning, or is later added as part of a scaling operation, will have a public IP address assigned to it. You will be able to directly access the nodes in the service instance, and the Java EE applications deployed to these nodes, from the public Internet.

If you choose not to assign public IP addresses, you will not be able to directly access the nodes in the service instance from the public Internet. This option is for use cases where you only intend to access your Java EE applications from within your private cloud network or from your on-premises data center over a VPN network.

The procedure for creating a service instance with no public IP addresses varies depending on the region type:

- Oracle Cloud Infrastructure (Not available on Oracle Cloud at Customer) – Assign an existing Private Subnet to the instance. You must create the service instance using the CLI or REST API. See Create an Oracle Java Cloud Service Instance Attached to a Private Subnet on Oracle Cloud Infrastructure.
- Oracle Cloud Infrastructure Classic – Assign an existing IP Network to the instance and also explicitly disable public IP addresses on the instance.

When you create a service instance in an Oracle Cloud Infrastructure Classic region, you can choose to create a public or private Oracle-managed load balancer for your service instance. A private load balancer in Oracle Cloud Infrastructure Load Balancing Classic cannot be accessed from the public Internet.

You can further control the nodes and port numbers in your service instance that are accessible from the Internet or other Oracle Cloud resources:

- Oracle Cloud Infrastructure – See Security Lists.
- Oracle Cloud Infrastructure Classic – See Create an Access Rule.

Reserved IP Address

This feature is specific to Oracle Cloud Infrastructure Classic regions.

If you select a specific region for your service instance, you can also assign reserved IP addresses to use for the nodes in your service instance.

Reserved IP addresses are specific to a region.

Reserved IP addresses are persistent. If you create a service instance that uses a set of reserved IP addresses, you can reuse the IP addresses after you delete the instance.

The number of IP addresses you create must match the number of nodes in your service instance cluster. You can either select individual IP addresses for every node or allow Oracle to assign them automatically.

If you have created multiple clusters, the number of IP addresses you create must match the total number of nodes in all the clusters.

See Reserve IP Addresses.

Domain Partition

A WebLogic Server 12c domain can optionally be organized into multiple partitions.

Each partition is dedicated to running specific applications and related resources, and is managed independently of other partitions in the same domain. You can define partitions when you create a service instance, and you can add or remove domain partitions after you create the service instance by using Fusion Middleware Control. These domain partitions will be created with a default resource management policy.

Domain partitions also enable you to create different security realms for the overall WebLogic Server domain and for each partition. Each security realm can have its own identity store with users, credentials and groups.

See About WebLogic Server MT in *Using WebLogic Server Multitenant*.

You cannot configure domain partitions if you select:

- The Oracle Java Cloud Service Virtual Image (BASIC) service level
- The Oracle Java Cloud Service Fusion Middleware — Oracle WebCenter Portal service level
- The Oracle Java Cloud Service Fusion Middleware — Oracle Data Integrator service level
- The Standard Edition of WebLogic Server
- The 11g release of WebLogic Server

User Authentication

By default, the WebLogic Server domain in a service instance is configured to use the local WebLogic identity store to maintain administrators, application users, groups and roles. These security elements are used to authenticate users and to also authorize access to tools like the WebLogic Server Administration Console.

If your cloud account includes Oracle Identity Cloud Service, an Oracle Java Cloud Service instance can also use Oracle Identity Cloud Service for authentication. As a result, users that access your applications or the administration consoles in this service instance are authenticated against Oracle Identity Cloud Service if they are not found in the local WebLogic identity store. See Use Oracle Identity Cloud Service with Oracle Java Cloud Service.

You can also create a service instance within a specific identity domain in Oracle Identity Cloud Service (Not available on Oracle Cloud at Customer). Each identity domain has an independent set of users. For example, you might create separate identity domains for test users and production users. By default, service instances are created in the primary identity domain in Oracle Identity Cloud Service. See About Multiple Instances in *Administering Oracle Identity Cloud Service*.

You cannot configure a service instance to use Oracle Identity Cloud Service if you select:

- The Oracle Java Cloud Service Virtual Image (BASIC) service level
- The 11g release of WebLogic Server

Administrator Access

This feature is specific to Oracle Cloud Infrastructure Classic regions.

By default, remote access to the Administration Server is disabled in a service instance for security purposes.

This includes the use of the WebLogic Server Administration Console and Fusion Middleware Control Console, as well as remote WebLogic Scripting Tool (WLST) commands. You can enable console access either when you create a service instance, or later after it has been created.

Client Access

By default, a service instance can be accessed only over secure protocols like HTTPS and SSH.

If you plan to access an application through the HTTP port, you can enable this port manually after creating a service instance. The HTTP port is disabled by default only when creating the service instance by using the Oracle Java Cloud Service console. The HTTP port is enabled by default if you create the service instance by using the REST API or CLI. See About the Default Access Ports.

From Oracle Java Cloud Service release 20.2.2, new service instances are provisioned with T3 and T3 over SSL (T3S) protocols and tunneling disabled. You'll need to use the Oracle Cloud Infrastructure Console to configure security ingress rules for the administration server VM and the managed server VMs before you can perform certain tasks such as deploying applications via Oracle JDeveloper. See Create an Ingress Rule.

For existing Oracle Java Cloud Service instances created before release 20.2.2, we recommend you restrict Internet access to the WebLogic administration server (and Oracle Traffic Director administration server, if applicable) by configuring security ingress rules using a fixed set of IPs or a CIDR matching your organization's network addresses. See the My Oracle Support Document ID 2664435.1.

> **Note:**
>
> If you provisioned an Oracle Java Cloud Service instance without explicitly specifying a named subnet, the instance is assigned to the predefined Virtual Cloud Network (VCN) named `svc-vcn`, which is found in the `ManagedCompartmentForPaaS` compartment. You cannot modify resources in `svc-vcn`, such as assign security lists or add ingress rules.
>
> If your Oracle Java Cloud Service instance is assigned to `svc-vcn`, submit a Service Request (SR) with Oracle Support Services to obtain access for updating ports and ingress rules in `svc-vcn`.

Coherence Data Tier

If you choose to provision an Oracle Coherence data tier in your service instance, Oracle Java Cloud Service creates a WebLogic Server cluster in the domain to host your in-memory data grid, or cache. This Coherence cluster provides your applications with fast, reliable, and scalable access to frequently used data.

You configure the data grid’s initial cache capacity by configuring the cluster size, the number of nodes, and the number of servers per node. After a service instance is created, you can increase cache capacity by adding more nodes to the data grid cluster. See About Cache Capacity for a Service Instance.

While you can create up to eight application clusters in a new service instance, you can create only one WebLogic Server cluster for the data grid.

Oracle Java Cloud Service can only provision a Coherence data tier in your service instance if you select Enterprise Edition with Coherence (Suite).

Database

Every service instance must be associated with an existing relational database in Oracle Cloud. Oracle Java Cloud Service provisions the required infrastructure schema on the selected database.

The supported database services in Oracle Cloud vary by region.

Region Type	Infrastructure Schema Database Options
Oracle Cloud Infrastructure region (Not available on Oracle Cloud at Customer)	• Oracle Cloud Infrastructure Database • Oracle Autonomous Database • Oracle Database Cloud Service
Oracle Cloud Infrastructure Classic region	• Oracle Database Cloud Service • Oracle Database Exadata Cloud Service

If you specify **No Preference** for region, or if you have an older Oracle Cloud account that doesn't include regions, then you can choose from the same database options as Oracle Cloud Infrastructure Classic.

All databases must be in an active state and not currently in the process of being provisioned. The WebLogic Server domain in a service instance uses Java Database Connectivity (JDBC) to access the databases.

When you associate a service instance with an Oracle Database Cloud Service or Oracle Database Exadata Cloud Service deployment for the infrastructure database schema, you can also associate the service instance with up to four additional database deployments in order to access your application schemas. This feature is not available for service instances that use other database services, but you can also manually configure JDBC data sources for your application schemas after creating the service instance.

This feature is also not available for service instances that use the Virtual Image (BASIC) service level.

To ensure that you can restore the database for an Oracle Java Cloud Service instance without risking data loss for other service instances, Oracle recommends that you do not associate the same infrastructure schema database (or the same pluggable database) with multiple service instances. Backups of a database that is used with multiple Oracle Java Cloud Service instances contain data for all the instances. Therefore, if you restore the database from a backup, data for all the service instances is restored, which might not be the intended result.

The following limitations apply only to service instances that use Oracle Database Cloud Service as the infrastructure schema database.

- When you create an Oracle Java Cloud Service instance on a secondary Oracle Identity Cloud Service domain, you can't use Oracle Database Cloud Service for the infrastructure schema. The only option is to use an Oracle Cloud Infrastructure Database or Oracle Autonomous Database. You can use an Oracle Database Cloud Service deployment for the infrastructure schema for the default Oracle Identity Cloud Service domain only.
- You cannot use an Oracle Database Cloud Service deployment running Oracle Database 18c.
- You can use an Oracle Database Cloud Service deployment running Oracle Database 12.2, but only for service instances running Oracle WebLogic Server 12.2.1 or later.
- Create Oracle Database Cloud Service deployments with a backup option other than `NONE`. This configuration enables Oracle Java Cloud Service to coordinate backups across your service instance and the database. Coordinated backups are not supported for other database services.

The following limitations apply only to service instances on Oracle Cloud Infrastructure regions:

- You must create a security policy in Oracle Cloud Infrastructure in order for your Oracle Autonomous Database or Oracle Cloud Infrastructure Database to be displayed in the Oracle Java Cloud Service web console. See Creating the Infrastructure Resources Required for Oracle Platform Services.
- Database instances in Oracle Database Cloud Service and Oracle Cloud Infrastructure Database must be in the same region and virtual cloud network (VCN) as the Oracle Java Cloud Service instance. The database and service instance do not need to be in the same subnet or availability domain, but it might be necessary to create and assign security rules to the subnets in order to enable communication between them. The database and service instance can be on different VCNs only if you configure VCN peering. See VCNs and Subnets in the Oracle Cloud Infrastructure Services documentation.
- To use Oracle Cloud Infrastructure Database, you must assign a custom subnet to your service instance. The default subnet is not supported.
- To use a Bare Metal database in Oracle Cloud Infrastructure Database, you must create the service instance with the Oracle Java Cloud Service REST API or CLI. The web console supports only VM and Exadata databases in Oracle Cloud Infrastructure Database.
- To use an Oracle Cloud Infrastructure Database running Oracle Database 12.2 or later, the service instance must be running Oracle WebLogic Server 12.2.1 or later.
- Oracle Java Cloud Service supports using Logical Volume Manager as the storage management software for a 1-node VM DB system. You can use the fast provisioning option to create the Oracle Cloud Infrastructure Database.
- Oracle Database Cloud Service does not support Real Application Cluster (RAC) databases containing multiple nodes on Oracle Cloud Infrastructure.
- To use a serverless Oracle Autonomous Database, the service instance must be running WebLogic Server 12.2.1.3 or later, and the service instance cannot use the Fusion Middleware — Oracle WebCenter Portal or Fusion Middleware —

Oracle Data Integrator service levels. Note that Oracle Java Cloud Service does not yet support a dedicated deployment autonomous database.

- The Oracle Java Cloud Service cloning feature is not supported for service instances that use databases in Oracle Cloud Infrastructure Database or Oracle Autonomous Database.

The following limitations apply only to service instances on Oracle Cloud Infrastructure Classic regions:

- The database must be in the same region as the Oracle Java Cloud Service instance.
- If you specify an IP network for a service instance, the infrastructure schema database for the Oracle Java Cloud Service instance must also be attached to an IP network. If the service instance and the database are attached to different IP Networks, the two IP networks must be connected to the same IP network exchange. See Creating an IP Network in *Using Oracle Cloud Infrastructure Compute Classic*.

For more information about the available database services in Oracle Cloud, see:

- Creating a Database Deployment in *Administering Oracle Database Cloud Service*
- Managing Bare Metal and Virtual Machine DB Systems in the Oracle Cloud Infrastructure documentation
- Provisioning Autonomous Database in *Using Oracle Autonomous Database on Shared Exadata Infrastructure*
- Managing Exadata DB Systems in the Oracle Cloud Infrastructure documentation
- Creating a Database Deployment in *Administering Oracle Database Exadata Cloud Service*

Load Balancer

A load balancer routes requests it receives from clients to the WebLogic Servers configured in a service instance.

Using a load balancer within your service instance is recommended if you are configuring more than one Managed Server or more than one cluster. A load balancer also gives you the ability to suspend access to a service instance temporarily to perform routine maintenance.

Oracle Java Cloud Service supports two load balancer options:

- A user-managed load balancer that runs within your service instance. You can access, patch, and administer this type of load balancer like other nodes in your service instance.
- An Oracle-managed load balancer that is automatically patched and maintained by Oracle. This load balancer is provisioned in Oracle Cloud Infrastructure Load Balancing or Oracle Cloud Infrastructure Load Balancing Classic, depending on the region where the service instance is created.

Note:

- The Oracle Cloud Infrastructure Load Balancing Classic Oracle-managed load balancer that is provisioned and configured automatically with an Oracle Java Cloud Service instance cannot be removed after the service instance is created.
- The Oracle Cloud Infrastructure Load Balancing Oracle-managed load balancer that is provisioned and configured automatically with an Oracle Java Cloud Service instance can be removed after the service instance is created. If you create your own instance of Oracle Cloud Infrastructure Load Balancing and configure it yourself to an existing service instance, this load balancer is not considered an Oracle-managed load balancer.

The user-managed load balancer in Oracle Java Cloud Service is an instance of Oracle Traffic Director (OTD) and is administered through the Load Balancer Console. A service instance can include zero, one or two load balancer nodes running OTD. Each load balancer node is assigned a separate public IP address.

The Oracle-managed load balancer is automatically deployed on multiple nodes to provide high availability and is accessed by clients using a single public IP address. The configuration options vary by region:

- On Oracle Cloud Infrastructure regions, you can assign a regional subnet that will be shared by all the load balancer nodes. A regional subnet is not scoped to any particular availability domain, so the subnet contains resources in any of a region's availability domains. Oracle recommends assigning a regional subnet to enable high availability, with automatic failover from one availability domain to another if needed.
- On Oracle Cloud Infrastructure regions, Oracle recommends that you assign a regional subnet, but you can assign a non-regional (availability domain-scoped) subnet to each load balancer node if needed. For high availability, Oracle recommends that each subnet be associated with a different availability domain in the selected region. If the selected region has one availability domain, you can specify only one subnet, which is assigned to both load balancer nodes.
- On Oracle Cloud Infrastructure regions, if you configure the service instance to use Oracle Identity Cloud Service for authentication, then you must also provision an Oracle-managed load balancer. However, you can also create an instance with an Oracle-managed load balancer that does not use Oracle Identity Cloud Service.
- On Oracle Cloud Infrastructure Classic regions, in order to provision an Oracle-managed load balancer, you must also configure the service instance to use Oracle Identity Cloud Service for authentication.
- On Oracle Cloud Infrastructure Classic regions, if you specify an IP Network for your service instance, you can choose to create a public or private Oracle-managed load balancer. A private load balancer cannot be accessed from the public Internet. It is for use cases where you only intend to access your service instance from within your private cloud network or from your on-premises data center over a VPN network.

You cannot configure a service instance to use an Oracle-managed load balancer if you select:

- The Oracle Java Cloud Service Virtual Image (BASIC) service level
- The 11g release of WebLogic Server

See About the Load Balancer in Oracle Java Cloud Service.

Backup Location

When provisioning a service instance, you can choose to enable or disable automated backups.

If you do not enable backups, you will not be able to initiate on-demand backups as well. You can also configure backups for a service instance after its creation.

Backups are recorded to a specified object storage location in Oracle Cloud:

- For a service instance in an Oracle Cloud Infrastructure region, you must create this storage bucket manually.
- For a service instance in an Oracle Cloud Infrastructure Classic region, you can create this storage container manually, or Oracle Java Cloud Service can create one automatically while you are provisioning the service instance.

See Create an Object Storage Container.

Create an Oracle Java Cloud Service Instance by Using a QuickStart Template

QuickStart templates give you the fastest, easiest way to create an Oracle Java Cloud Service instance.

> **Note:**
>
> QuickStart is not available for Oracle Cloud accounts that include only Oracle Cloud Infrastructure regions, or include a mix of Oracle Cloud Infrastructure and Oracle Cloud Infrastructure Classic regions.

> **Note:**
>
> You must have a Universal Credits subscription in order to use QuickStart. This feature is not available to other subscription types.

Video

Tutorial

Topics

- Create a QuickStart Instance
- Simple Java Web App
- Multi-Tier Java EE App with High Availability

- Highly Available Java EE App with Caching
- Compare QuickStart Templates

Create a QuickStart Instance

Use the QuickStart page to choose from one of the available Oracle Java Cloud Service templates. These templates are executed with Oracle Cloud Stack.

The **Bring Your Own License** (BYOL) option enables you to bring your on-premises Oracle WebLogic Server licenses to Oracle Cloud. BYOL instances are billed at a lower rate than other instances. See Frequently Asked Questions: Oracle BYOL to PaaS. If your cloud account includes the BYOL entitlement for Oracle Java Cloud Service, then all QuickStart instances use BYOL. If you prefer not to use BYOL, then create a custom service instance.

> **Note:**
>
> Before you scale up or scale out a BYOL instance, you must have enough WebLogic Server licenses for the additional OCPUs that will be allocated to the instance after it is scaled.

Oracle Java Cloud Service generates the user name and password that you will need to administer the Oracle WebLogic Server and the Load Balancer components in the selected configuration. This same password is also used to administer the Oracle Database. Oracle Java Cloud Service generates an archive file for you to download, which contains:

- A text file with the administrator user name and password for WebLogic Server and the Load Balancer
- A copy of the Secure Shell (SSH) public key that will be associated with each of the nodes in this configuration
- The corresponding SSH private key, which will be necessary to access any of the nodes in this configuration

To create a service instance:

1. Access the Oracle Java Cloud Service console and click the **QuickStarts** link.

   Alternatively, from the Infrastructure Classic Console click **Create Instance**. Within the **Featured Services** tab or **All Services** tab, click the **Create** button for the Java option.
2. On the QuickStarts page, enter an **Instance Name**, or accept the default name.

   This value will also be used as the base name for the Oracle Database Cloud Service deployment: `<instanceName>DBCS`.
3. Click the **Create** button below the template you want to provision:
   - Simple Java Web App
   - Multi-Tier Java EE App with High Availability
   - Highly Available Java EE App with Caching
4. From the confirmation dialog, click the **Download** link. When prompted by your web browser, save this archive file to your local machine.

The **Create** button is now enabled.

5. Click **Create**.

   The Stacks page displays. Your new cloud stack is `<instanceName>QS`.

6. Click the name of the stack.
7. On the Stack Details page, periodically refresh this page to monitor the progress of the new Oracle Database Cloud Service instance and Oracle Java Cloud Service instance.

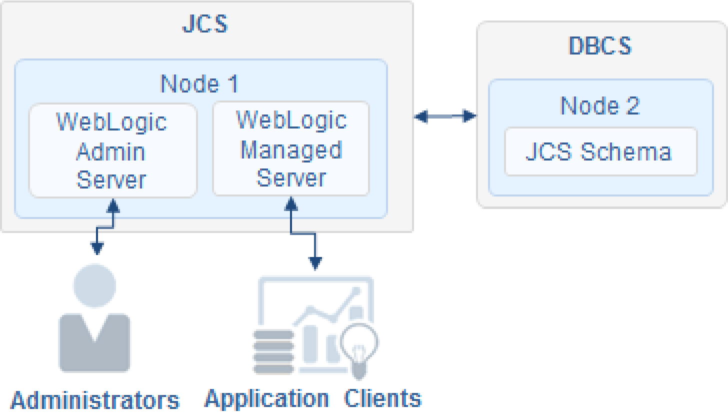
Click the name of your new Oracle Java Cloud Service instance to view its details or perform management operations. To return to the Oracle Cloud Stack console at a later time, click ≡ at the top left corner of the page (next to the Oracle logo), and then choose **Cloud Stack**.

Next steps:

- By default, access to the WebLogic Server and Load Balancer administration consoles is disabled. In order to use these tools to modify the default configuration or to deploy applications, see Enable Console Access for a Service Instance.
- The generated private key file is in OpenSSH format. Before connecting to a node in this service instance with the PuTTY SSH client, you must first convert the key to PuTTY's proprietary format. See Convert a Private Key with PuTTY.
- Service backups are not enabled in QuickStart instances. See Add a Backup Configuration to an Oracle Java Cloud Service Instance.
- In order to delete this Oracle Java Cloud Service instance , you must use Oracle Cloud Stack. See Deleting a Cloud Stack in *Using Oracle Cloud Stack*.

Simple Java Web App

This Oracle Java Cloud Service template is comprised of a single Oracle WebLogic Server node and a single Oracle Database node.



When you execute this template, Oracle Cloud provisions an Oracle Java Cloud Service instance as well as an Oracle Database Cloud Service deployment. This simple template implements a typical development or test Java EE environment, or a production environment for a departmental Java EE application that doesn't require high availability 24 hours a day. It requires a total of **2 OPCUs**.

This template includes:

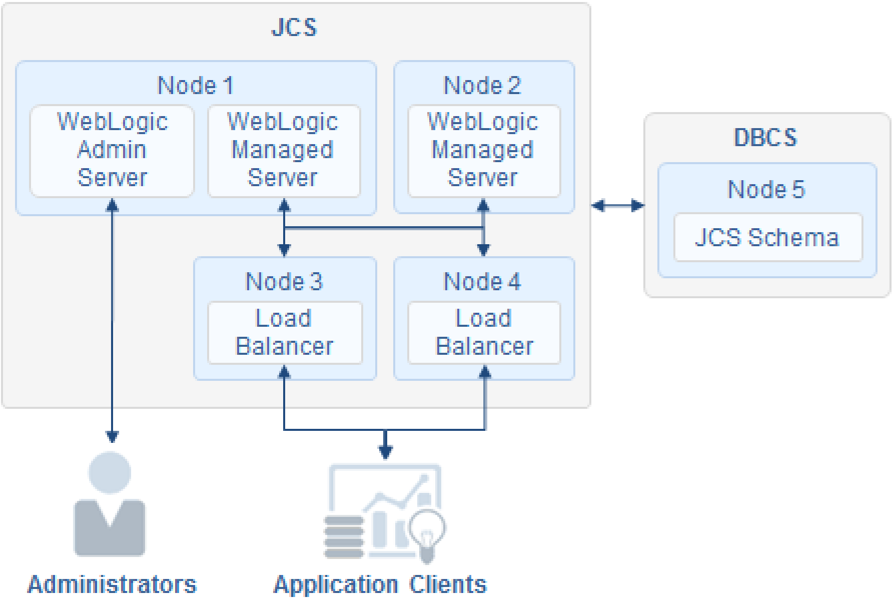
- WebLogic Server 12*c* (12.2) Enterprise Edition installation.
- Oracle Database 12*c* (12.2) Standard Edition installation.
- Single node running a WebLogic Administration Server and a single WebLogic Managed Server. Use the Administration Server to perform administration tasks like configuring Java EE resources and deploying applications. These applications are hosted on the WebLogic Managed Server, and are accessed by end users and other external clients.
- Single node running Oracle Database. This database has **25 GB** of total space and is provisioned with the required schemas for running Oracle Java Cloud Service.

You can scale up your WebLogic Server node at a later time if your applications require more compute or storage capacity. Similarly, you can add more compute or storage capacity to the database. You cannot add more nodes to a Standard Edition instance (scaling out).

You can enable automatic backups on this service instance after creating it.

Multi-Tier Java EE App with High Availability

This Oracle Java Cloud Service template is comprised of multiple Oracle WebLogic Server nodes to ensure maximum availability. External clients access your applications through a load balancer node.



When you execute this template, Oracle Cloud provisions an Oracle Java Cloud Service instance as well as an Oracle Database Cloud Service deployment. This template implements a typical, production-level Java EE environment that requires high availability of the application tier. A separate load balancer tier transparently distributes incoming client requests across the application tier. This template requires a total of **5 OPCUs**.

This template includes:

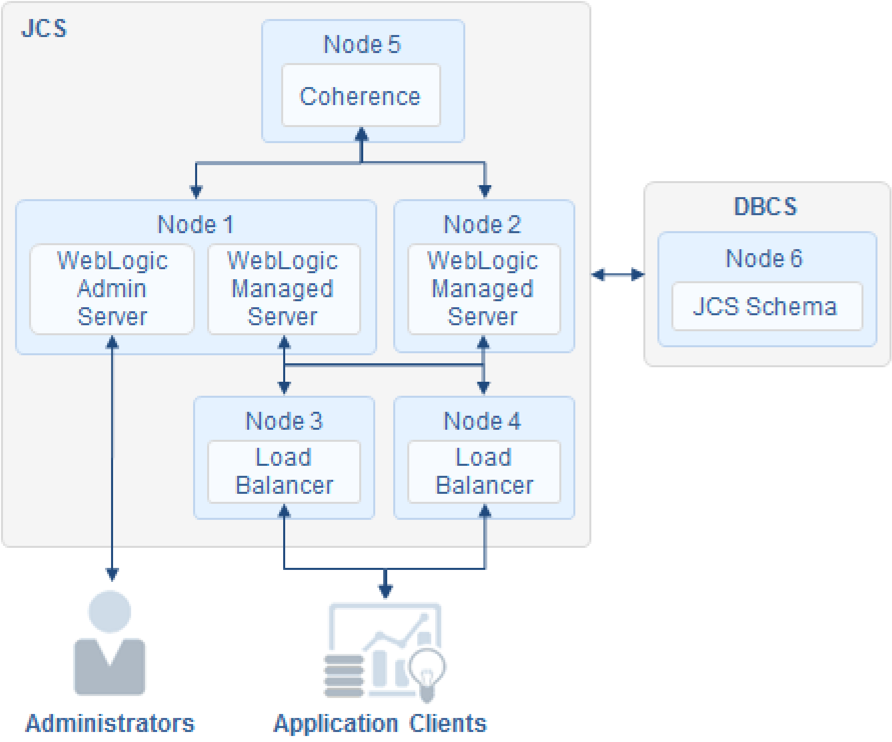
- WebLogic Server 12*c* (12.2) Enterprise Edition installation.
- Oracle Database 12*c* (12.2) Enterprise Edition installation.
- One node running a WebLogic Administration Server and a WebLogic Managed Server. Use the Administration Server to perform administration tasks like configuring Java EE resources and deploying applications.
- One node running a second WebLogic Managed Server. Both Managed Servers are part of a single cluster for high availability. Applications are hosted on these Managed Servers.
- Two nodes running an enterprise-grade load balancer (Oracle Traffic Director). End users and other external clients access applications via the load balancers. Each load balancer node has its own public IP address.
- One node running Oracle Database. This database has **256 GB** of total space and is provisioned with the required schemas for running Oracle Java Cloud Service.

You can scale your service instance at a later time if your applications require additional capacity. Similarly, you can add more compute or storage capacity to the database.

You can enable automatic backups on this service instance after creating it.

Highly Available Java EE App with Caching

This Oracle Java Cloud Service template is comprised of multiple Oracle WebLogic Server nodes to ensure maximum availability. To optimize performance, applications deployed to WebLogic Server can also take advantage of an Oracle Coherence node running an in-memory data cache.



When you execute this template, Oracle Cloud provisions an Oracle Java Cloud Service instance as well as an Oracle Database Cloud Service deployment. This template implements a high-performance, production-level Java EE environment that requires high availability of the application tier. A separate load balancer tier transparently distributes incoming client requests across the application tier. Finally, a data grid tier enables you to predictably scale applications by providing fast access to frequently used data. This template requires a total of **6 OPCUs**.

This template includes:

- WebLogic Server 12*c* (12.2) Suite installation.
- Oracle Database 12*c* (12.2) Enterprise Edition installation.
- One node running a WebLogic Administration Server and a WebLogic Managed Server. Use the Administration Server to perform administration tasks like configuring Java EE resources and deploying applications.
- One node running a second WebLogic Managed Server. Both Managed Servers are part of a single cluster for high availability. Applications are hosted on these Managed Servers.
- Two nodes running an enterprise-grade load balancer (Oracle Traffic Director). End users and other external clients access applications via the load balancers. Each load balancer node has its own public IP address.
- One node running Oracle Coherence and configured with a 1.5 GB in-memory cache. Applications can use the Coherence API to cache and retrieve data.
- One node running Oracle Database. This database has **512 GB** of total space and is provisioned with the required schemas for running Oracle Java Cloud Service.

You can scale this environment in a number of different ways as your applications require additional capacity. Similarly, you can add more compute or storage capacity to the database.

You can enable automatic backups on this service instance after creating it.

Compare QuickStart Templates

Compare the attributes of each Oracle Java Cloud Service QuickStart template, including the amount of cloud resources that each template consumes.

Attribute	Simple Java Web App	Multi-Tier Java EE App with High Availability	Highly Available Java EE App with Caching
OCPUs	2	5	6
Memory	15 GB	38 GB	45 GB
Total Storage (approx.)	225 GB	635 GB	850 GB
Database Storage	25 GB	256 GB	512 GB
Public IP Addresses	2	4	4
WebLogic Server Version	12*c* (12.2.1)	12*c* (12.2.1)	12*c* (12.2.1)
WebLogic Server Edition	Standard	Enterprise	Suite
Oracle Database Version	12*c* (12.2.1)	12*c* (12.2.1)	12*c* (12.2.1)
Oracle Database Edition	Standard	Enterprise	Enterprise

Attribute	Simple Java Web App	Multi-Tier Java EE App with High Availability	Highly Available Java EE App with Caching
Total Nodes	2	5	6
WebLogic Server Nodes	1	2	2
Oracle Database Nodes	1	1	1
Load Balancer Nodes	0	2	2
Coherence Nodes	0	0	1
Backups Enabled	No	No	No

Create an Oracle Java Cloud Service Instance Attached to a Public Subnet on Oracle Cloud Infrastructure

Oracle Java Cloud Service gives you detailed control over how you create your service instance.

There are several methods that you can use to create instances in Oracle Java Cloud Service. One of the easiest methods to create an instance is to use the Create Instance wizard in the web console. The wizard guides you through a short series of screens that present all the parameters that you can configure for your instance, including the WebLogic Server settings, backup and recovery configuration, load balancer parameters, and so on.

If you need to create an instance on a private subnet rather than on a public subnet, see Create an Oracle Java Cloud Service Instance Attached to a Private Subnet on Oracle Cloud Infrastructure.

> **Note:**
>
> You can provision Oracle WebCenter Portal and Oracle Data Integrator on your Oracle Java Cloud Service only using the REST API.

Video

Tutorial

Prerequisites

Before creating a custom Oracle Java Cloud Service instance:

- Review the prerequisites described in Before You Begin with Oracle Java Cloud Service
- Review the options described in Design Considerations for an Oracle Java Cloud Service Instance
- Review About Java Cloud Service Instances in Oracle Cloud Infrastructure

Procedure

- Start the Create New Instance Wizard
- Specify Basic Service Instance Information

- Specify WebLogic Configuration
- Configure the Coherence Data Tier
- Configure the Databases
- Configure Backup and Recovery
- Configure the Load Balancer
- Confirm Your Oracle Java Cloud Service Instance Creation

Start the Create New Instance Wizard

This topic applies only to Oracle Cloud Infrastructure.

To create a service instance from the web console, you use the Create New Instance wizard.

To start the Create New Instance wizard:

1. Access your service console.
2. Click **Create Instance**.

Specify Basic Service Instance Information

This topic applies only to Oracle Cloud Infrastructure.

On the Instance page of the Instance Creation Wizard, enter basic information for your service instance, including service name, region, and software edition.

> **Note:**
>
> You cannot change any of the following options after you have created the service instance.

Complete the following fields:

Field	Description
Instance Name	Specify a name for the Oracle Java Cloud Service instance. The service instance name: • Must contain one or more characters. • Must not exceed 30 characters. • Must start with an ASCII letter: `a` to `z` , or `A` to `Z`. • Must contain only ASCII letters or numbers. • Must not contain a hyphen. • Must not contain any other special characters. • Must be unique within the identity domain.
Description	(Optional) Enter a short description of the Oracle Java Cloud Service instance.

Field	Description
Notification Email	(Optional) Specify an email address where you would like to receive a notification of any events occurring with the service instance, including whether provisioning has succeeded or failed.
Region	(Available only if your account has regions) Select a region if you want to create the service instance in a specific region. A region supports either Oracle Cloud Infrastructure or Oracle Cloud Infrastructure Classic. For a list of available regions, see Data Regions for Platform and Infrastructure Services. The database that you intend to associate with your Oracle Java Cloud Service instance must be in the same region (not applicable to Oracle Autonomous Database).
Availability Domain	Select an availability domain. A region can have multiple isolated availability domains, each with separate power and cooling. The availability domains within a region are interconnected using a low-latency network. Note that the database that you intend to associate with your Oracle Java Cloud Service instance can be in a different availability domain within the selected region.
Subnet	Select the Oracle Cloud Infrastructure subnet to which the nodes of your instance must be attached. This field provides a **No Preference** option and a list of the available subnets. Each subnet is shown in the format *compartmentName* | *vcnName* | *subnetName*. A tooltip lists the compartment name, VCN name, subnet name, and the OCID of the subnet. • To have the subnet assigned automatically, select **No Preference**. The subnet **ManagedCompartmentForPaaS | svc-vcn | svc-subnet-...** is used for your instance. **Note:** Don't select **No Preference** if you plan to associate an Oracle Cloud Infrastructure Database with your service instance. If you want to configure security rules for your instance, don't select **No Preference** or **ManagedCompartmentForPaaS | svc-vcn | svc-subnet-...**. Select a subnet in a VCN that you created. • To assign a subnet explicitly, select a suitable subnet from the available options. • If none of the available subnets meets your networking requirements, then cancel the Create Instance wizard. In Oracle Cloud Infrastructure, create the required VCN and subnets, create policies to allow Oracle Java Cloud Service to use the VCN, and select the appropriate subnet while creating your instance. See Prerequisites for PaaS Services on Oracle Cloud Infrastructure in the Oracle Cloud Infrastructure documentation. Database instances in Oracle Database Cloud Service and Oracle Cloud Infrastructure Database must be in the same region and virtual cloud network (VCN) as the Oracle Java Cloud Service instance. The database and service instance do not need to be in the same subnet. The database and service instance can be on different VCNs only if you configure VCN peering.

Field	Description
Tags	(Optional) Select existing tags or add tags to associate with the service instance. To select existing tags, select one or more check boxes from the list of tags that are displayed on the pull-down menu. To create tags, click **Click to create a tag** to display the **Create Tags** dialog box. In the **New Tags** field, enter one or more comma-separated tags that can be a key or a key:value pair. If you do not assign tags during provisioning, you can create and manage tags after the service instance is created. See Creating, Assigning, and Unassigning Tags.
Identity Domain	(Not available on Oracle Cloud at Customer) Select the identity domain in Oracle Identity Cloud Service in which to create this service instance. By default, the instance is created in the primary identity domain.
The service security administrator	(Not available on Oracle Cloud at Customer) (Optional) Specify the username for the security administrator for the service instance in the selected identity domain. This user gets rights to administer security artifacts (roles, AppId, OAuth IDs, and so on). The username can be the administrator of the selected identity domain or a user in the selected identity domain. You can leave this field blank *only* if you are the administrator of the selected identity domain or a user in the selected identity domain.
License Type	Choose whether you want to leverage the Bring Your Own License (BYOL) option or use your Oracle Java Cloud Service license. • The **Bring Your Own License** (BYOL) option enables you to bring your on-premises Oracle WebLogic Server licenses to Oracle Cloud. BYOL instances are billed at a lower rate than other instances. See Frequently Asked Questions: Oracle BYOL to PaaS. You must own a Universal Credits subscription or Government subscription in order to use BYOL. **Note**: Before you scale up or scale out a BYOL instance, you must have enough WebLogic Server licenses for the additional OCPUs that will be allocated to the instance after it is scaled. • If you choose to use your Oracle Java Cloud Service license, your account will be charged for the new service instance according to your Oracle Java Cloud Service agreement. If you have both BYOL and Oracle Java Cloud Service entitlements, BYOL is selected by default, but you can change the license type. If you have BYOL entitlements only, BYOL is selected and you cannot change the license type. If you do not have BYOL entitlements, the Oracle Java Cloud Service license option is selected and you cannot change the license type.
Software Edition c	Select a WebLogic Server software edition: • **Standard Edition** • **Enterprise Edition** • **High Performance Edition**

Field	Description
Metering Frequency	This option appears only if you have a traditional metered subscription. If you have a Universal Credits subscription, this field is absent. Select a metering frequency to determine how you are billed for this service instance: • **Hourly**—Pay only for the number of hours that this service instance was running during your billing period. • **Monthly**—Pay one price for the full month irrespective of the number of hours that this service instance was running. For services that are started in the middle of a month, the price will be pro-rated; you pay only for the partial month from the day the service instance is created.

Specify the Service Instance Details

This topic applies only to Oracle Cloud Infrastructure.

You must configure the size, shape, and other important details for your Oracle Java Cloud Service instance.

Topics

- Specify WebLogic Configuration
- Configure WebLogic Server Access
- Configure the Coherence Data Tier
- Configure the Databases
- Configure Backup and Recovery
- Configure the Load Balancer

Specify WebLogic Configuration

This topic applies only to Oracle Cloud Infrastructure.

On the second page of the Instance Creation Wizard (Service Details), you start by configuring the size and shape of the Oracle Java Cloud Service instance.

> **Note:**
>
> Two tabs, Simple and Advanced, control which fields appear on the page. Fields that appear when you select the Simple tab also appear when you select the Advanced tab, but some fields appear only when you select the Advanced tab.

Complete the following fields:

Size and Shape Details	Description
WebLogic Clusters	(Advanced option) If you selected **Oracle-Managed Load Balancer**, you can add, edit, or delete up to 8 WebLogic clusters for the service instance, with a maximum of 8 servers per cluster. You specify the cluster name, compute shape, and server count. Optionally, you can specify a path prefix, which determines how the managed load balancer routes traffic to different clusters. If you do not specify a path prefix, the cluster name is used as the path prefix. After you specify these values, you can edit them: • Click **Add** to add a new cluster. • Select a cluster and click **Edit** to update its configuration. • Click **Delete** to delete the cluster. If **Oracle-Managed Load Balancer** is not selected, then a single cluster is created during instance provisioning. You cannot add clusters using the console, but clusters can be added using the REST API.
Compute Shape	Select the compute shape to use for all Administration Server and Managed Server nodes. The compute shape is the number of Oracle Compute Units (OCPUs) and amount of memory (RAM) that you want to allocate to these nodes. The selected shape is not used for Coherence or Load Balancer nodes. The `VM.Standard` and `BM.Standard` shapes are supported. If you purchased a Universal Credits subscription for Oracle Java Cloud Service, you will pay at the Pay-As-You-Go rate when you exceed your monthly or annual maximum credit.
Server Count	Select the initial number of Managed Servers that you want to provision in this service instance. The choices are: `1`, `2`, `4`. • If you configure more than one Managed Server in the cluster, Oracle recommends that you also enable the Load Balancer. • You can also perform scaling operations to increase or decrease the server count after provisioning the service instance.
Domain Partitions	(Advanced option) Select the initial number of WebLogic Server domain partitions that you want to provision in this service instance. The choices are `0`, `1`, `2`, or `4`.

Size and Shape Details	Description
Enable Access to Administration Consoles	(Advanced option) Select this check box if you want to enable access to the WebLogic Service Administration console, Fusion Middleware Control, and Load Balancer console for the service instance. If you do not select this option, these consoles will not be externally accessible, and also will not appear as choices in the service instance's menu ☰. If this check box is enabled, the **This Source CIDR range field can access Admin Consoles** option is displayed. By default, the source CIDR range is `0.0.0.0/0`, so the administration console is accessible from the public internet. You can specify a source CIDR range so that only the IP addresses within the specified range can access the administration console. If you create multiple service instances on a subnet, and you specify a source CIDR range for one service instance, and do not specify a source CIDR range for another service instance, the default source CIDR range that is used by the other service instance, where no CIDR range was specified, is used by both service instances, and you can access the consoles from the public internet. For example: You create two service instances, *service 1* and *service 2* on a subnet. You specify a source CIDR range, `10.0.1.0/24` for *service 1*, and do not specify a source CIDR range for *service 2*; the default source CIDR range `0.0.0.0/0`, is used for *service 2*. In this case, `10.0.1.0/24` is nullified and `0.0.0.0/0` is used. So, you will be able to access the WebLogic Service Administration Console through port 7002 from the public internet, and the Oracle Traffic Director and the Load Balancer console through port 8989 from the public internet.
Deploy Sample Application	(Advanced option) By default, a sample application, `sample-app.war`, is deployed automatically to the Managed Servers in your instance. If you do not want to automatically deploy the sample application, deselect this check box.

Configure WebLogic Server Access

This topic applies only to Oracle Cloud Infrastructure.

On the Service Details page of the Wizard, configure the administrator credentials for the WebLogic Servers.

Complete the following fields:

Access Details	Description
Enable Authentication Using Identity Cloud Service	Select this check box if you want WebLogic Server to authenticate application users and administrators against Oracle Identity Cloud Service in addition to the local WebLogic Server identity store. This field appears only if your cloud account includes Oracle Identity Cloud Service. By default, the WebLogic Server domain in the service instance is configured to use only the local WebLogic Server identity store to maintain administrators, application users, groups, and roles.

Access Details	Description
SSH Public Key	Specify the public key that will be used for authentication when connecting to a node in your instance by using a Secure Shell (SSH) client. Click **Edit** to display the SSH Public Key for VM Access dialog, and then specify the public key using one of the following methods: • Select **Key file name** and use your web browser to select a file on your machine that contains the public key. • Select **Key value** and paste the value of the public key into the text area. Be sure the value does not contain line breaks or end with a line break. • Select **Create a New Key** if you want Oracle to generate a public/private key pair for you. You will be prompted to download these generated keys. If you choose to create a new key, the generated private key file is in OpenSSH format. Before connecting to a node in this service instance with the PuTTY SSH client, you must first convert the key to PuTTY’s proprietary format.
Local Administrative User Name	Enter your choice of user name for the WebLogic Server administrator. The default is `weblogic`. This name is used to access the WebLogic Server Administration Console, Fusion Middleware Control, and Load Balancer Console for the service instance. The name must be between 8 and 128 characters long and **cannot** contain any of the following characters: • Tab • Brackets • Parentheses • These special characters: – Left angle bracket (`<`) – Right angle bracket (`>`) – Ampersand (`&`) – Pound sign (`#`) – Pipe symbol (`|`) – Question mark (`?`) You can also change the user name through the WebLogic Server Administration Console after the service instance is provisioned.
Password	Specify a password for the WebLogic Server administrator and confirm the password. As a best practice, this password must start with a letter, be of 8 to 30 characters in length, and contain at least: • 1 uppercase character • 1 lower case character • 1 digit (0 through 9) • One of the following special characters: _ (underscore), - (hyphen), or # (pound sign or hash) The following basic password criteria are acceptable, but Oracle does not recommend them: • Starts with a letter • Is between 8 and 30 characters long • Contains letters, at least one number, and, optionally, any number of these special characters: – Dollar sign (`$`) – Pound sign (`#`) – Underscore (_) No other special characters are allowed.

Configure the Coherence Data Tier

This topic applies only to Oracle Cloud Infrastructure.

If you want to create a Coherence Data Tier, provide details on the Service Details page of the Wizard.

Complete the following fields:

Coherence Data Tier	Description
Provision Data Grid Cluster	(Advanced option) Select **Yes** to provision a Coherence data grid cluster in your service instance. This option is only available if you selected **High Performance Edition**.
Compute Shape	Select the compute shape to use for all Managed Server nodes in the data grid cluster. The compute shape is the number of Oracle Compute Units (OCPUs) and amount of memory (RAM) that you want to allocate to these nodes. The `VM.Standard` and `BM.Standard` shapes are supported. This option is displayed only if **Provision Data Grid Cluster** is set to `Yes`.
Cluster Size	Set the initial number of Managed Servers that you want to provision in the data grid cluster. Valid values are 1–4. This option is displayed only if **Provision Data Grid Cluster** is set to `Yes`. The number of nodes in the data grid cluster is determined by **Cluster Size** / **Managed Servers Per Node**. If this ratio is a fraction, the number of nodes is rounded up to the next integer. You can also perform scaling operations to increase or decrease the number of Coherence nodes after provisioning the service instance. You cannot specify multiple data grid clusters.
Managed Servers Per Node	Set the number of Coherence Managed Servers to run on each node in the data grid cluster. Valid values are 1–8. This option is displayed only if **Provision Data Grid Cluster** is set to `Yes`.

Configure the Databases

This topic applies only to Oracle Cloud Infrastructure.

On the Service Details page of the Wizard, provide details about the database(s) to use for the Oracle Java Cloud Service instance.

In order for Oracle Autonomous Database (Oracle Autonomous Transaction Processing) and Oracle Cloud Infrastructure Database to be displayed in the Oracle Java Cloud Service web console, you must first create the appropriate policies.

For Oracle Autonomous Database:

- Specify this policy if you created the database in a custom compartment:

  ```
  Allow service PSM to inspect autonomous-database in compartment
  compartment_name
  ```

- Specify this policy if you created the database in the root compartment:

  ```
  Allow service PSM to inspect autonomous-database in tenancy
  ```

For Oracle Cloud Infrastructure Database:

- Specify this policy if you created the database in a custom compartment:

  ```
  Allow service PSM to inspect database-family in compartment
  compartment_name
  ```

- Specify this policy if you created the database in the root compartment:

  ```
  Allow service PSM to inspect database-family in tenancy
  ```

For information on creating policies, see Creating the Infrastructure Resources Required for Oracle Platform Services.

Complete the following fields:

Database Details	Description
Database Type	Select the type of database you want to associate with your service instance: • **Oracle Autonomous Transaction Processing** • **Oracle Cloud Infrastructure Database** • **Oracle Database Cloud Service (Classic)**
Compartment Name	Select the compartment where the Oracle Autonomous Database or Oracle Cloud Infrastructure Database resides.

Database Details	Description
Database Instance Name	Select an Oracle Cloud Infrastructure Database or Oracle Database Cloud Service (Classic) deployment that you want to associate as the infrastructure schema database for your service instance. The list only includes a database deployment if it is in an active state and not currently in the process of being provisioned. Database instances in Oracle Database Cloud Service and Oracle Cloud Infrastructure Database must be in the same region and virtual cloud network (VCN) as the Oracle Java Cloud Service instance. The database and service instance do not need to be in the same subnet or availability domain. The database and service instance can be on different VCNs only if you configure VCN peering. To ensure that you can restore the database for an Oracle Java Cloud Service instance without risking data loss for other service instances, Oracle recommends that you do not associate the same infrastructure schema database (or the same pluggable database) with multiple service instances. Backups of a database that is used with multiple Oracle Java Cloud Service instances contain data for all the instances. Therefore, if you restore the database from a backup, data for all the service instances is restored, which might not be the intended result. Note the following additional constraints and limitations for Oracle Cloud Infrastructure databases: • To use a Bare Metal database, you must create the service instance with the Oracle Java Cloud Service REST API or CLI. The web console supports only VM and Exadata databases. • To use an Oracle Cloud Infrastructure Database running Oracle Database 12.2 or later, the service instance must be running WebLogic Server 12.2.1 or later. • You can select an Oracle Cloud Infrastructure 1-node virtual machine (VM) DB system that was created using the fast provisioning option. Oracle Java Cloud Service supports using Logical Volume Manager as the storage management software for a 1-node VM DB system. Note the following additional constraints and limitations for Oracle Database Cloud Service (Classic) deployments: • You cannot use a database deployment running Oracle Database 18c as the infrastructure schema database. • You can use a database deployment running Oracle Database 12.2 as the infrastructure schema database, but only for service instances running Oracle WebLogic Server 12.2.1 or later. • Create Oracle Database Cloud Service deployments with a backup option other than `NONE`. This configuration enables Oracle Java Cloud Service to coordinate backups across your service instance and the database. Coordinated backups are not supported for other database services.

Database Details	Description
Database Instance	Select the PDB that you created for the Oracle Autonomous Database (Oracle Autonomous Transaction Processing). You must use an Oracle Autonomous Database (Oracle Autonomous Transaction Processing) that is created with the serverless option. Oracle Java Cloud Service does not yet support using a dedicated deployment autonomous database.
PDB Name	Select the pluggable database the service instance will connect to. • For Oracle Cloud Infrastructure databases, the PDB name is populated. If you did not specify a PDB name when you created the Oracle Cloud Infrastructure database, the default PDB name populated in this field is `<dbName>_pdb1`. • For Oracle Database Cloud Service (Classic) databases, if you don't specify a PDB name, Oracle Java Cloud Service uses the default Oracle Database 12c PDB name that was provided when the Oracle Database Cloud Service (Classic) database deployment was originally created.
Administrator User Name	Specify the name of the database administrator that Oracle Java Cloud Service will use to connect to the selected database and to provision the required schemas for this service instance. This value is set automatically for: • Oracle Autonomous Database (Oracle Autonomous Transaction Processing): `ADMIN` • Oracle Cloud Infrastructure Database: `SYS`
Password	Enter the password for the database administrator.
Add Application DB	(Advanced option) Add up to four Oracle Database Cloud Service (Classic) databases for your application schema. You cannot add Oracle Autonomous Database or Oracle Cloud Infrastructure databases. Click **Add** if you want to specify a separate Oracle Database Cloud Service (Classic) database deployment dedicated for your application schema. When you add an application database, the Oracle Java Cloud Service creates an additional data source in your Oracle WebLogic Server domain to connect to this database. Use the Add Database Configuration dialog to select the name of an existing Oracle Database Cloud Service (Classic) deployment, and to provide a user name and password for this database. Click **Add** and repeat this process for up to three more database deployments.

Configure Backup and Recovery

This topic applies only to Oracle Cloud Infrastructure.

On the Service Details page of the Wizard, specify details on the storage used for backup and recovery.

Complete the following fields:

Backup and Recovery Details	Description
Backup Destination	(Advanced option) Select **Both Remote and Disk Storage** if you want to enable automated and on-demand backups for this service instance. Backups will be saved to object storage *and* to block storage volumes that are attached to the nodes of the instance. The default value is **None**, meaning that you cannot use Oracle Java Cloud Service to take backups of this service instance. You can configure backups on a service instance after creating it.
Object Storage Container	This field is displayed only if **Backup Destination** is set to **Both Remote and Disk Storage**. Enter the object storage location where backups of the service instance must be stored. Enter the URL of a bucket in Oracle Cloud Infrastructure Object Storage. See Prerequisites for PaaS Services on Oracle Cloud Infrastructure in the Oracle Cloud Infrastructure documentation. **Format**: `https://swiftobjectstorage.`*`region`*`.oraclecloud.com/v1/`*`namespace`*`/`*`bucket`* To find out your `namespace`, sign in to the Oracle Cloud Infrastructure web console, click the tenancy name, and look for the **Object Storage Namespace** field. **Example**: `https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/myCompany/myBucket`
User Name	This field is displayed only if **Backup Destination** is set to **Both Remote and Disk Storage**. Enter the user name of the Oracle Cloud Infrastructure Object Storage user who created the bucket you specified earlier.
Password	This field is displayed only if **Backup Destination** is set to **Both Remote and Disk Storage**. Enter the Auth Token generated in Oracle Cloud Infrastructure for the user you specified. See Prerequisites for PaaS Services on Oracle Cloud Infrastructure in the Oracle Cloud Infrastructure documentation.

Configure the Load Balancer

This topic applies only to Oracle Cloud Infrastructure.

On the Service Details page of the Wizard, specify details to configure the load balancer(s) for the Oracle Java Cloud Service instance.

Complete the following fields:

Load Balancer Details	Description
Load Balancer	Select the type of load balancer that you want to configure for your service instance: • **Oracle-Managed Load Balancer**: A dual-node, Oracle-managed instance of the Oracle Cloud Infrastructure Load Balancing service, providing active-passive high-availability. Failover from the active load-balancer node to the other node occurs automatically. You can't customize the default listeners, certificates, and so on for an Oracle Cloud Infrastructure Load Balancing instance that is provisioned by Oracle Java Cloud Service. If you need the ability to configure Oracle Cloud Infrastructure Load Balancing, then you must create the load balancer manually. See Set Up an Oracle Cloud Infrastructure Load Balancer. • **Oracle Traffic Director**: One or two Oracle Traffic Director nodes within your service instance. The dual-node configuration is in active-active mode, but failover to the second node is not automatic. • **None**: No load balancer will be configured for this instance. Provisioning a load balancer is recommended if the cluster size is 2 or more. The default value is **None**. If you selected **Enable Authentication Using Identity Cloud Service**, then you cannot configure a user-managed load balancer. You must select **Oracle-Managed Load Balancer**. If you select **Oracle Traffic Director** and configure one Oracle Traffic Director node, you can also add a second Oracle Traffic Director node to a service instance after creating the service instance. If you configured two Oracle Traffic Director nodes during provisioning, you cannot add another Oracle Traffic Director node. If you select **None**, then you can add an Oracle Traffic Director load balancer after creating the service instance.
Compute Shape	This option is displayed only if **Oracle Traffic Director** is selected as the load balancer. Select the compute shape to use for all the load balancer nodes in the service instance. The compute shape is the number of Oracle Compute Units (OCPUs) and amount of memory (RAM) that you want to allocate to these nodes. The `VM.Standard.*` shapes and `BM.Standard.*` shapes are supported. You are billed for Oracle Traffic Director nodes at the same price that you are billed for WebLogic Server nodes in your Oracle Java Cloud Service subscription. See About Oracle Java Cloud Service Subscriptions and Licenses.
Add Another Active OTD Node	This option is displayed only if **Oracle Traffic Director** is selected as the load balancer. Select this check box to provision a second load balancer node running Oracle Traffic Director (OTD) in this service instance. Both load balancer nodes route traffic to the cluster of WebLogic Managed Servers. You can also add a second load balancer node to a service instance after creating the service instance.

Load Balancer Details	Description
Load Balancing Policy	This option is displayed only if you selected **Oracle-Managed Load Balancer** or **Oracle Traffic Director** as the load balancer. If you selected **Oracle Traffic Director**, choose one of the following policies: • **Least Connection Count** (default)—Passes each new request to the Managed Server with the least number of connections. This policy is useful for smoothing distribution when a Managed Server receives more requests than it can handle efficiently. • **Least Response Time**—Passes each new request to the Managed Server with the fastest response time. • **Round Robin**—Evenly distributes requests across all Managed Servers, regardless of the number of connections or response times. If you selected **Oracle-Managed Load Balancer**, choose one of the following policies: • **Round Robin**— (default) Same as above. • **IP Hash**—The IP Hash policy uses an incoming request's source IP address as a hashing key to route traffic to the same backend server. The load balancer routes requests from the same client to the same backend server as long as that server is available. • **Least Connection Count**—Same as above.
Subnet for Load Balancer Node 1 **Subnet for Load Balancer Node 2**	These fields are displayed only if both are true: • **Load Balancer** is set to **Oracle-Managed Load Balancer** • **Subnet** is assigned to a specific subnet, and not the value **No Preference** For regional subnets: • Oracle recommends that you select a regional subnet for the load balancer to support failover to another availability domain if needed. • You can only assign one regional subnet. If you select a regional subnet from either the **Subnet for Load Balancer Node 1** or **Subnet for Load Balancer Node 2** menu, the other menu is not displayed. For non-regional (availability domain-scoped) subnets: • For each load balancer node, select a non-regional subnet from a different availability domain. You must select two non-regional subnets. • If the selected region has only one availability domain, **Subnet for Load Balancer Node 2** is not shown. In this case, you can only select one non-regional subnet, which is assigned to both nodes. • For at least one of the nodes, Oracle recommends selecting a non-regional subnet from the same availability domain as that of the service instance. This ensures that, as long as the service instance is running, the applications deployed on it remain accessible through the load balancer.

Confirm Your Oracle Java Cloud Service Instance Creation

This topic applies only to Oracle Cloud Infrastructure.

On the Confirmation page of the provisioning wizard, review the service details.

If you need to change the service details, use the navigation bar or **Back** button at the top of the wizard to step back through the pages in the wizard. Click **Cancel** to cancel out of the wizard without creating a new service instance. If you are satisfied with your choices on the Confirmation page, click **Create**.

Sample of Options Displayed

If you selected the **Bring Your Own License** option, the Confirmation page will display a message alerting you to the fact that you have chosen to use an existing license. Check to make sure you have the appropriate entitlements.

The compute shape and server count is displayed in the WebLogic Configuration section.

Download the Instance Attributes in JSON Format

(Not available on Oracle Cloud at Customer)

Click **Download** ⤓ to download a JSON-format file containing the parameters you specified in the provisioning wizard. You can use the JSON-formatted file as a sample to construct the request body for creating instances using the REST API.

Note that the file contains placeholders for passwords.

After Confirmation

After the Confirmation page closes, the Oracle Java Cloud Service console opens. Optionally, you can click on the service instance name to view status messages. If provisioning of your service instance fails but there are no fatal errors, the software automatically retries provisioning, after a lag time of 60 minutes. Messages about the auto-retry process and failed compute resources are displayed.

If you provided your email address for the **Notification Email** option, you will receive an email notification when the service instance provisioning has succeeded or failed.

Next Steps

- After the service instance has been created, you can view the system messages logged during the creation process, including error messages. Click **Instance Create and Delete History**, then click the service instance name or **Details**.
- If the provisioning process retried provisioning automatically, some failed resources might still exist. To clean up these failed resources, click the **Complete Cleanup** button. If you click the button once and not all failed resources are cleaned up, the **Complete Cleanup** button will remain. If this is the case, click the button again and wait. Repeat this process until the button is not longer displayed and all failed resources are cleaned up.
- If you selected the **Enable Authentication with Oracle Identity Cloud Service** option, you can use Oracle Identity Cloud Service to create additional WebLogic Server users. See Use Oracle Identity Cloud Service with Oracle Java Cloud Service.
- If you selected the **Deploy Sample Application** option, and want to test the sample application, see About the Sample Application Deployed to an Oracle Java Cloud Service Instance.

Create an Oracle Java Cloud Service Instance Attached to a Private Subnet on Oracle Cloud Infrastructure

When you create an Oracle Java Cloud Service instance in an Oracle Cloud Infrastructure region, you can attach the instance to either a private subnet or a public subnet. If you attach the instance to a private subnet, then the nodes of the instance can’t have public IP addresses. They are isolated from the public Internet.

> **Note:**
>
> For the instructions to create an instance attached to a *public* subnet, see Create an Oracle Java Cloud Service Instance Attached to a Public Subnet on Oracle Cloud Infrastructure.
> You can create an Oracle Java Cloud Service instance using Oracle WebCenter Portal and Oracle Data Integrator only through the REST API.

Task Flow for Creating an Oracle Java Cloud Service Instance Attached to a Private Subnet

1. Create the Required Resources in Oracle Cloud Infrastructure
2. Create an Oracle Cloud Infrastructure Database System Attached to a Private Subnet
3. Create an Oracle Java Cloud Service Instance Attached to a Private Subnet on Oracle Cloud Infrastructure Using the Wizard or Create an Oracle Java Cloud Service Instance Attached to a Private Subnet Using REST API

Create the Required Resources in Oracle Cloud Infrastructure

Before creating an Oracle Java Cloud Service instance attached to a private subnet, you must fulfill certain prerequisites, including creating the required identity, networking, and storage resources in Oracle Cloud Infrastructure.

1. Generate an SSH key pair.

   See Generate a Key Pair with OpenSSH.

   Note the path and name of the files that contain the private and public keys. You’ll need the keys later.

2. Complete the following steps from the tutorial Creating the Infrastructure Resources Required for Oracle Platform Services:

   a. Create a compartment.

      If you want to create the Oracle Cloud Infrastructure resources in an existing compartment, then skip this step.

   b. Create a virtual cloud network (VCN) in the compartment you created or identified.

      If you want to use an existing VCN, then skip this step.

   c. Create a policy to allow Oracle Cloud platform services to use the networking resources in the compartment that you created or identified.

If the required policy exists for the compartment that you want to use, then skip this step.

d. Create a bucket in the Object Storage service to store backups of your Oracle Java Cloud Service instance.

> **Note:**
>
> The user creating the bucket must be either a local user in Oracle Cloud Infrastructure Identity and Access Management (IAM), or a synchronized user created automatically by a federated identity provider.

If you'd like to use a bucket that was created previously, then skip this step.

Note the name of the bucket. You'll need it later while creating the service instances.

e. Generate an authentication token for the user who created the bucket.

If you have the required token already, then skip this step.

Note the authentication token value. You'll need it later while creating the service instances.

3. In the VCN that you created or identified earlier, create the required networking resources:

a. Create a service gateway.

The service gateway is required for the Oracle Java Cloud Service instance to access Oracle Cloud Infrastructure Object Storage.

See Setting Up a Service Gateway in the Oracle Cloud Infrastructure documentation.

b. Create an internet gateway.

The internet gateway enables communication between the public Internet and the bastion node.

See Working with Internet Gateways in the Oracle Cloud Infrastructure documentation.

c. (Optional) Create a NAT gateway.

The NAT gateway is required for the nodes of the Oracle Java Cloud Service instance to access the public Internet. Such access would be useful when (for example) you want to allow the nodes to access the Oracle Yum server to download additional packages or OS patches.

See Setting Up a NAT Gateway in the Oracle Cloud Infrastructure documentation.

d. Create the following route table:

See Working with Route Tables in the Oracle Cloud Infrastructure documentation.

Route Table `route.public` for the Public Subnets

Route Rule	Destination	Target
To route traffic bound for the public Internet through the internet gateway	CIDR: 0.0.0.0/0	Internet gateway

Route Table `route.private` for the Private Subnet

Route Rule	Destination	Target
To route traffic bound for the Object Storage service through the service gateway	Service: OCI *region* Object Storage	Service gateway
(Optional) To route traffic bound for the public Internet through the NAT gateway	CIDR: 0.0.0.0/0	NAT gateway

e. Create the following security lists:

See Working with Security Lists in the Oracle Cloud Infrastructure documentation.

Security List `seclist.bastion` for the Bastion Subnet

Security Rule	Source / Destination	IP Protocol / Port
(Ingress) To allow SSH connections to the bastion node	Source CIDR: 0.0.0.0/0	SSH / 22
(Egress) To allow all outbound traffic	Destination CIDR: 0.0.0.0/0	All protocols / ports

Security List `seclist.lb` for the Load Balancer Subnets

Security Rule	Source / Destination	IP Protocol / Port
(Ingress) To allow traffic from the other compute nodes in the VCN	Source CIDR: 10.0.0.0/16	All protocols / ports
(Egress) To allow all outbound traffic	Destination CIDR: 0.0.0.0/0	All protocols / ports

Security List `seclist.private` for the Private Subnet

Security Rule	Source / Destination	IP Protocol / Port
(Ingress) To allow traffic from the other compute nodes in the VCN	Source CIDR: 10.0.0.0/16	All Protocols
(Egress) To allow all outbound traffic	Destination CIDR: 0.0.0.0/0	All Protocols

f. Create the following subnets:

See Working with VCNs and Subnets in the Oracle Cloud Infrastructure documentation.

Subnet Purpose (`Suggested Name`)	**Availability Domain**	**Attributes**
For the bastion host (`subnet.bastion`)	AD1	Example CIDR[1]: 10.0.1.0/24 Route table: `route.public` Subnet access: Public Security list: `seclist.bastion`
For the primary load balancer node (`subnet.lb1`)	AD1	Example CIDR: 10.0.2.0/24 Route table: `route.public` Subnet access: Public Security list: `seclist.lb`
(Relevant only if the region has multiple availability domains) For the standby load balancer node (`subnet.lb2`)	AD2	Example CIDR: 10.0.3.0/24 Route table: `route.public` Subnet access: Public Security list: `seclist.lb`
For the service instances (`subnet.private`)	AD1	Example CIDR: 10.0.4.0/24 Route table: `route.private` Subnet access: Private Security list: `seclist.private`

[1] Assuming the VCN’s CIDR is 10.0.0.0/16

> **Note:**
>
> Make a note of the OCIDs of the subnets. You’ll need them later while creating the bastion host and the service instances.

4. Create a compute instance and attach it to the public subnet that you created for the bastion host.

   Through this node, administrators can access the administration console of the Oracle Java Cloud Service instance, and they can connect using `ssh` to the compute nodes of the instance.

   See Creating an Instance in the Oracle Cloud Infrastructure documentation.

   After creating the bastion compute instance, note its public IP address.

You’ve created the required resources in Oracle Cloud Infrastructure. You can now create the Oracle Cloud Infrastructure Database and Oracle Java Cloud Service instances.

Create an Oracle Cloud Infrastructure Database System Attached to a Private Subnet

Create an Oracle Cloud Infrastructure Database system that's attached to the private subnet that you plan to use for the Oracle Java Cloud Service instance.

1. Create a DB system by following the steps in Managing Bare Metal and Virtual Machine DB Systems in the Oracle Cloud Infrastructure documentation.

   Note the following:

   - Select the required private subnet in the network settings.
   - You can use a DB system running Oracle Database 12.2 or later as the infrastructure schema database, but only for an Oracle Java Cloud Service instance running WebLogic Server 12.2.1 or later.
   - The PDB name field is optional. If you enter a name, then make a note of it. You'll need it in the next step.

2. Wait for the DB system to be created. When the status displayed in the web console is **AVAILABLE**, construct the connection string. You'll need this string while creating the Oracle Java Cloud Service instance.

   The connection string is in the following format:

   - VM DB system: `//hostNamePrefix-scan.hostDomainName:1521/pdbName.hostDomainName`
   - Bare metal DB system: `//hostNamePrefix.hostDomainName:1521/pdbName.hostDomainName`

   `hostNamePrefix` and `hostDomainName` are the values displayed in the **Hostname Prefix** and **Host Domain Name** fields, respectively, in the Oracle Cloud Infrastructure web console.

   `pdbName` depends on the DB version and the DB shape.

   - 12c (any shape): The PDB name that you entered while creating the DB system (for example, `PDB1`).
     If you didn't enter a PDB name, then use `dbName_PDB1`, where `dbName` is the database name you specified (for example, `dbforjcs_PDB1`).
   - 11g (VM or bare metal): **Database Unique Name** displayed in the web console (for example, `dbforjcs_yyz17v`).
   - 11g (Exadata): The database name you specified (for example, `dbforjcs`).

   The following is an example of a connection string for a 12c VM DB system with the PDB name, `pdb1`:

   ```
   //dbforjcs-scan.privatesubnet.paasvcn.oraclevcn.com:1521/
   pdb1.privatesubnet.paasvcn.oraclevcn.com
   ```

Create an Oracle Java Cloud Service Instance Attached to a Private Subnet on Oracle Cloud Infrastructure Using the Wizard

Use the Create Instance wizard in the web console to create an Oracle Java Cloud Service instance attached to a private subnet.

The wizard guides you through a short series of screens that present all the parameters that you can configure for your instance, including the WebLogic Server settings, backup and recovery configuration, load balancer parameters, and so on.

Prerequisites

Before creating an Oracle Java Cloud Service instance:

- Review the prerequisites described in Before You Begin with Oracle Java Cloud Service

- Review the options described in Design Considerations for an Oracle Java Cloud Service Instance
- Review About Java Cloud Service Instances in Oracle Cloud Infrastructure

Procedure

- Start the Create New Instance Wizard
- Specify Basic Service Instance Information
- Specify WebLogic Configuration
- Configure the Coherence Data Tier
- Configure the Databases
- Configure Backup and Recovery
- Configure the Load Balancer
- Confirm Your Oracle Java Cloud Service Instance Creation

Start the Create New Instance Wizard

This topic applies only to Oracle Cloud Infrastructure.

To create a service instance from the web console, you use the Create New Instance wizard.

To start the Create New Instance wizard:

1. Access your service console.
2. Click **Create Instance**.

Specify Basic Service Instance Information

This topic applies only to Oracle Cloud Infrastructure.

On the Instance page of the Instance Creation Wizard, enter basic information for your service instance, including service name, region, and software edition.

> **Note:**
>
> You cannot change any of the following options after you have created the service instance.

Complete the following fields:

Field	Description
Instance Name	Specify a name for the Oracle Java Cloud Service instance. The service instance name: • Must contain one or more characters. • Must not exceed 30 characters. • Must start with an ASCII letter: `a` to `z` , or `A` to `Z`. • Must contain only ASCII letters or numbers. • Must not contain a hyphen. • Must not contain any other special characters. • Must be unique within the identity domain.
Description	(Optional) Enter a short description of the Oracle Java Cloud Service instance.
Notification Email	(Optional) Specify an email address where you would like to receive a notification of any events occurring with the service instance, including whether provisioning has succeeded or failed.
Region	(Available only if your account has regions) Select a region if you want to create the service instance in a specific region. A region supports either Oracle Cloud Infrastructure or Oracle Cloud Infrastructure Classic. For a list of available regions, see Data Regions for Platform and Infrastructure Services. The database that you intend to associate with your Oracle Java Cloud Service instance must be in the same region (not applicable to Oracle Autonomous Database).
Availability Domain	Select an availability domain. A region can have multiple isolated availability domains, each with separate power and cooling. The availability domains within a region are interconnected using a low-latency network. Note that the database that you intend to associate with your Oracle Java Cloud Service instance can be in a different availability domain within the selected region.
Subnet	Select **Use Private Subnet** to attach the nodes of the instance to a private subnet, and enter the OCID of the private subnet in the text field. **Note:** You must use the OCID of the subnet that you noted when creating the subnet. See Create the Required Resources in Oracle Cloud Infrastructure. Make sure that the database you are using to create the instance is reachable from the private subnet. Database instances in Oracle Database Cloud Service and Oracle Cloud Infrastructure Database must be in the same region and virtual cloud network (VCN) as the Oracle Java Cloud Service instance. The database and service instance do not need to be in the same subnet. The database and service instance can be on different VCNs only if you configure VCN peering.
Tags	(Optional) Select existing tags or add tags to associate with the service instance. To select existing tags, select one or more check boxes from the list of tags that are displayed on the pull-down menu. To create tags, click **Click to create a tag** to display the **Create Tags** dialog box. In the **New Tags** field, enter one or more comma-separated tags that can be a key or a key:value pair. If you do not assign tags during provisioning, you can create and manage tags after the service instance is created. See Creating, Assigning, and Unassigning Tags.

Field	Description
Identity Domain	(Not available on Oracle Cloud at Customer) Select the identity domain in Oracle Identity Cloud Service in which to create this service instance. By default, the instance is created in the primary identity domain.
The service security administrator	(Not available on Oracle Cloud at Customer) (Optional) Specify the username for the security administrator for the service instance in the selected identity domain. This user gets rights to administer security artifacts (roles, AppId, OAuth IDs, and so on). The username can be the administrator of the selected identity domain or a user in the selected identity domain. You can leave this field blank *only* if you are the administrator of the selected identity domain or a user in the selected identity domain.
License Type	Choose whether you want to leverage the Bring Your Own License (BYOL) option or use your Oracle Java Cloud Service license. • The **Bring Your Own License** (BYOL) option enables you to bring your on-premises Oracle WebLogic Server licenses to Oracle Cloud. BYOL instances are billed at a lower rate than other instances. See Frequently Asked Questions: Oracle BYOL to PaaS. You must own a Universal Credits subscription or Government subscription in order to use BYOL. **Note**: Before you scale up or scale out a BYOL instance, you must have enough WebLogic Server licenses for the additional OCPUs that will be allocated to the instance after it is scaled. • If you choose to use your Oracle Java Cloud Service license, your account will be charged for the new service instance according to your Oracle Java Cloud Service agreement. If you have both BYOL and Oracle Java Cloud Service entitlements, BYOL is selected by default, but you can change the license type. If you have BYOL entitlements only, BYOL is selected and you cannot change the license type. If you do not have BYOL entitlements, the Oracle Java Cloud Service license option is selected and you cannot change the license type.
Software Edition c	Select a WebLogic Server software edition: • **Standard Edition** • **Enterprise Edition** • **High Performance Edition**
Metering Frequency	This option appears only if you have a traditional metered subscription. If you have a Universal Credits subscription, this field is absent. Select a metering frequency to determine how you are billed for this service instance: • **Hourly**—Pay only for the number of hours that this service instance was running during your billing period. • **Monthly**—Pay one price for the full month irrespective of the number of hours that this service instance was running. For services that are started in the middle of a month, the price will be pro-rated; you pay only for the partial month from the day the service instance is created.

Specify the Service Instance Details

This topic applies only to Oracle Cloud Infrastructure.

You must configure the size, shape, and other important details for your Oracle Java Cloud Service instance.

Topics

- Specify WebLogic Configuration
- Configure WebLogic Server Access
- Configure the Coherence Data Tier
- Configure the Databases
- Configure Backup and Recovery
- Configure the Load Balancer

Specify WebLogic Configuration

This topic applies only to Oracle Cloud Infrastructure.

On the second page of the Instance Creation Wizard (Service Details), you start by configuring the size and shape of the Oracle Java Cloud Service instance.

> **Note:**
>
> Two tabs, Simple and Advanced, control which fields appear on the page. Fields that appear when you select the Simple tab also appear when you select the Advanced tab, but some fields appear only when you select the Advanced tab.

Complete the following fields:

Size and Shape Details	Description
WebLogic Clusters	(Advanced option) If you selected **Oracle-Managed Load Balancer**, you can add, edit, or delete up to 8 WebLogic clusters for the service instance, with a maximum of 8 servers per cluster. You specify the cluster name, compute shape, and server count. Optionally, you can specify a path prefix, which determines how the managed load balancer routes traffic to different clusters. If you do not specify a path prefix, the cluster name is used as the path prefix. After you specify these values, you can edit them: • Click **Add** to add a new cluster. • Select a cluster and click **Edit** to update its configuration. • Click **Delete** to delete the cluster. If **Oracle-Managed Load Balancer** is not selected, then a single cluster is created during instance provisioning. You cannot add clusters using the console, but clusters can be added using the REST API.
Compute Shape	Select the compute shape to use for all Administration Server and Managed Server nodes. The compute shape is the number of Oracle Compute Units (OCPUs) and amount of memory (RAM) that you want to allocate to these nodes. The selected shape is not used for Coherence or Load Balancer nodes. The `VM.Standard` and `BM.Standard` shapes are supported. If you purchased a Universal Credits subscription for Oracle Java Cloud Service, you will pay at the Pay-As-You-Go rate when you exceed your monthly or annual maximum credit.

Size and Shape Details	Description
Server Count	Select the initial number of Managed Servers that you want to provision in this service instance. The choices are: `1`, `2`, `4`. • If you configure more than one Managed Server in the cluster, Oracle recommends that you also enable the Load Balancer. • You can also perform scaling operations to increase or decrease the server count after provisioning the service instance.
Domain Partitions	(Advanced option) Select the initial number of WebLogic Server domain partitions that you want to provision in this service instance. The choices are `0`, `1`, `2`, or `4`.
Enable Access to Administration Consoles	(Advanced option) Select this check box if you want to enable access to the WebLogic Service Administration console, Fusion Middleware Control, and Load Balancer console for the service instance. If you do not select this option, these consoles will not be externally accessible, and also will not appear as choices in the service instance's menu ☰. If this check box is enabled, the **This Source CIDR range field can access Admin Consoles** option is displayed. By default, the source CIDR range is `0.0.0.0/0`, so the administration console is accessible from the public internet. You can specify a source CIDR range so that only the IP addresses within the specified range can access the administration console. If you create multiple service instances on a subnet, and you specify a source CIDR range for one service instance, and do not specify a source CIDR range for another service instance, the default source CIDR range that is used by the other service instance, where no CIDR range was specified, is used by both service instances, and you can access the consoles from the public internet. For example: You create two service instances, *service 1* and *service 2* on a subnet. You specify a source CIDR range, `10.0.1.0/24` for *service 1*, and do not specify a source CIDR range for *service 2*; the default source CIDR range `0.0.0.0/0`, is used for *service 2*. In this case, `10.0.1.0/24` is nullified and `0.0.0.0/0` is used. So, you will be able to access the WebLogic Service Administration Console through port 7002 from the public internet, and the Oracle Traffic Director and the Load Balancer console through port 8989 from the public internet.
Deploy Sample Application	(Advanced option) By default, a sample application, `sample-app.war`, is deployed automatically to the Managed Servers in your instance. If you do not want to automatically deploy the sample application, deselect this check box.

Configure WebLogic Server Access

This topic applies only to Oracle Cloud Infrastructure.

On the Service Details page of the Wizard, configure the administrator credentials for the WebLogic Servers.

Complete the following fields:

Access Details	Description
Enable Authentication Using Identity Cloud Service	Select this check box if you want WebLogic Server to authenticate application users and administrators against Oracle Identity Cloud Service in addition to the local WebLogic Server identity store. This field appears only if your cloud account includes Oracle Identity Cloud Service. By default, the WebLogic Server domain in the service instance is configured to use only the local WebLogic Server identity store to maintain administrators, application users, groups, and roles.
SSH Public Key	Specify the public key that will be used for authentication when connecting to a node in your instance by using a Secure Shell (SSH) client. Click **Edit** to display the SSH Public Key for VM Access dialog, and then specify the public key using one of the following methods: • Select **Key file name** and use your web browser to select a file on your machine that contains the public key. • Select **Key value** and paste the value of the public key into the text area. Be sure the value does not contain line breaks or end with a line break. • Select **Create a New Key** if you want Oracle to generate a public/private key pair for you. You will be prompted to download these generated keys. If you choose to create a new key, the generated private key file is in OpenSSH format. Before connecting to a node in this service instance with the PuTTY SSH client, you must first convert the key to PuTTY’s proprietary format.
Local Administrative User Name	Enter your choice of user name for the WebLogic Server administrator. The default is `weblogic`. This name is used to access the WebLogic Server Administration Console, Fusion Middleware Control, and Load Balancer Console for the service instance. The name must be between 8 and 128 characters long and **cannot** contain any of the following characters: • Tab • Brackets • Parentheses • These special characters: – Left angle bracket (`<`) – Right angle bracket (`>`) – Ampersand (`&`) – Pound sign (`#`) – Pipe symbol (`|`) – Question mark (`?`) You can also change the user name through the WebLogic Server Administration Console after the service instance is provisioned.

Access Details	Description
Password	Specify a password for the WebLogic Server administrator and confirm the password. As a best practice, this password must start with a letter, be of 8 to 30 characters in length, and contain at least: • 1 uppercase character • 1 lower case character • 1 digit (0 through 9) • One of the following special characters: _ (underscore), - (hyphen), or # (pound sign or hash) The following basic password criteria are acceptable, but Oracle does not recommend them: • Starts with a letter • Is between 8 and 30 characters long • Contains letters, at least one number, and, optionally, any number of these special characters: – Dollar sign (`$`) – Pound sign (`#`) – Underscore (_) No other special characters are allowed.

Configure the Coherence Data Tier

This topic applies only to Oracle Cloud Infrastructure.

If you want to create a Coherence Data Tier, provide details on the Service Details page of the Wizard.

Complete the following fields:

Coherence Data Tier	Description
Provision Data Grid Cluster	(Advanced option) Select **Yes** to provision a Coherence data grid cluster in your service instance. This option is only available if you selected **High Performance Edition**.
Compute Shape	Select the compute shape to use for all Managed Server nodes in the data grid cluster. The compute shape is the number of Oracle Compute Units (OCPUs) and amount of memory (RAM) that you want to allocate to these nodes. The `VM.Standard` and `BM.Standard` shapes are supported. This option is displayed only if **Provision Data Grid Cluster** is set to `Yes`.

Coherence Data Tier	Description
Cluster Size	Set the initial number of Managed Servers that you want to provision in the data grid cluster. Valid values are 1–4. This option is displayed only if **Provision Data Grid Cluster** is set to `Yes`. The number of nodes in the data grid cluster is determined by **Cluster Size** / **Managed Servers Per Node**. If this ratio is a fraction, the number of nodes is rounded up to the next integer. You can also perform scaling operations to increase or decrease the number of Coherence nodes after provisioning the service instance. You cannot specify multiple data grid clusters.
Managed Servers Per Node	Set the number of Coherence Managed Servers to run on each node in the data grid cluster. Valid values are 1–8. This option is displayed only if **Provision Data Grid Cluster** is set to `Yes`.

Configure the Databases

This topic applies only to Oracle Cloud Infrastructure.

On the Service Details page of the Wizard, provide details about the database(s) to use for the Oracle Java Cloud Service instance.

In order for Oracle Autonomous Database (Oracle Autonomous Transaction Processing) and Oracle Cloud Infrastructure Database to be displayed in the Oracle Java Cloud Service web console, you must first create the appropriate policies.

For Oracle Autonomous Database:

- Specify this policy if you created the database in a custom compartment:

```
Allow service PSM to inspect autonomous-database in compartment
compartment_name
```

- Specify this policy if you created the database in the root compartment:

```
Allow service PSM to inspect autonomous-database in tenancy
```

For Oracle Cloud Infrastructure Database:

- Specify this policy if you created the database in a custom compartment:

```
Allow service PSM to inspect database-family in compartment
compartment_name
```

- Specify this policy if you created the database in the root compartment:

```
Allow service PSM to inspect database-family in tenancy
```

For information on creating policies, see Creating the Infrastructure Resources Required for Oracle Platform Services.

Complete the following fields:

Database Details	Description
Database Type	Select the type of database you want to associate with your service instance: • **Oracle Autonomous Transaction Processing** • **Oracle Cloud Infrastructure Database** • **Oracle Database Cloud Service (Classic)**
Compartment Name	Select the compartment where the Oracle Autonomous Database or Oracle Cloud Infrastructure Database resides.

Database Details	Description
Database Instance Name	Select an Oracle Cloud Infrastructure Database or Oracle Database Cloud Service (Classic) deployment that you want to associate as the infrastructure schema database for your service instance. The list only includes a database deployment if it is in an active state and not currently in the process of being provisioned. Database instances in Oracle Database Cloud Service and Oracle Cloud Infrastructure Database must be in the same region and virtual cloud network (VCN) as the Oracle Java Cloud Service instance. The database and service instance do not need to be in the same subnet or availability domain. The database and service instance can be on different VCNs only if you configure VCN peering. To ensure that you can restore the database for an Oracle Java Cloud Service instance without risking data loss for other service instances, Oracle recommends that you do not associate the same infrastructure schema database (or the same pluggable database) with multiple service instances. Backups of a database that is used with multiple Oracle Java Cloud Service instances contain data for all the instances. Therefore, if you restore the database from a backup, data for all the service instances is restored, which might not be the intended result. Note the following additional constraints and limitations for Oracle Cloud Infrastructure databases: • To use a Bare Metal database, you must create the service instance with the Oracle Java Cloud Service REST API or CLI. The web console supports only VM and Exadata databases. • To use an Oracle Cloud Infrastructure Database running Oracle Database 12.2 or later, the service instance must be running WebLogic Server 12.2.1 or later. • You can select an Oracle Cloud Infrastructure 1-node virtual machine (VM) DB system that was created using the fast provisioning option. Oracle Java Cloud Service supports using Logical Volume Manager as the storage management software for a 1-node VM DB system. Note the following additional constraints and limitations for Oracle Database Cloud Service (Classic) deployments: • You cannot use a database deployment running Oracle Database 18c as the infrastructure schema database. • You can use a database deployment running Oracle Database 12.2 as the infrastructure schema database, but only for service instances running Oracle WebLogic Server 12.2.1 or later. • Create Oracle Database Cloud Service deployments with a backup option other than `NONE`. This configuration enables Oracle Java Cloud Service to coordinate backups across your service instance and the database. Coordinated backups are not supported for other database services.

Database Details	Description
Database Instance	Select the PDB that you created for the Oracle Autonomous Database (Oracle Autonomous Transaction Processing). You must use an Oracle Autonomous Database (Oracle Autonomous Transaction Processing) that is created with the serverless option. Oracle Java Cloud Service does not yet support using a dedicated deployment autonomous database.
PDB Name	Select the pluggable database the service instance will connect to. • For Oracle Cloud Infrastructure databases, the PDB name is populated. If you did not specify a PDB name when you created the Oracle Cloud Infrastructure database, the default PDB name populated in this field is `<dbName>_pdb1`. • For Oracle Database Cloud Service (Classic) databases, if you don't specify a PDB name, Oracle Java Cloud Service uses the default Oracle Database 12c PDB name that was provided when the Oracle Database Cloud Service (Classic) database deployment was originally created.
Administrator User Name	Specify the name of the database administrator that Oracle Java Cloud Service will use to connect to the selected database and to provision the required schemas for this service instance. This value is set automatically for: • Oracle Autonomous Database (Oracle Autonomous Transaction Processing): `ADMIN` • Oracle Cloud Infrastructure Database: `SYS`
Password	Enter the password for the database administrator.
Add Application DB	(Advanced option) Add up to four Oracle Database Cloud Service (Classic) databases for your application schema. You cannot add Oracle Autonomous Database or Oracle Cloud Infrastructure databases. Click **Add** if you want to specify a separate Oracle Database Cloud Service (Classic) database deployment dedicated for your application schema. When you add an application database, the Oracle Java Cloud Service creates an additional data source in your Oracle WebLogic Server domain to connect to this database. Use the Add Database Configuration dialog to select the name of an existing Oracle Database Cloud Service (Classic) deployment, and to provide a user name and password for this database. Click **Add** and repeat this process for up to three more database deployments.

Configure Backup and Recovery

This topic applies only to Oracle Cloud Infrastructure.

On the Service Details page of the Wizard, specify details on the storage used for backup and recovery.

Complete the following fields:

Backup and Recovery Details	Description
Backup Destination	(Advanced option) Select **Both Remote and Disk Storage** if you want to enable automated and on-demand backups for this service instance. Backups will be saved to object storage *and* to block storage volumes that are attached to the nodes of the instance. The default value is **None**, meaning that you cannot use Oracle Java Cloud Service to take backups of this service instance. You can configure backups on a service instance after creating it.
Object Storage Container	This field is displayed only if **Backup Destination** is set to **Both Remote and Disk Storage**. Enter the object storage location where backups of the service instance must be stored. Enter the URL of a bucket in Oracle Cloud Infrastructure Object Storage. See Prerequisites for PaaS Services on Oracle Cloud Infrastructure in the Oracle Cloud Infrastructure documentation. **Format**: `https://swiftobjectstorage.`*`region`*`.oraclecloud.com/v1/`*`namespace`*`/`*`bucket`* To find out your `namespace`, sign in to the Oracle Cloud Infrastructure web console, click the tenancy name, and look for the **Object Storage Namespace** field. **Example**: `https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/myCompany/myBucket`
User Name	This field is displayed only if **Backup Destination** is set to **Both Remote and Disk Storage**. Enter the user name of the Oracle Cloud Infrastructure Object Storage user who created the bucket you specified earlier.
Password	This field is displayed only if **Backup Destination** is set to **Both Remote and Disk Storage**. Enter the Auth Token generated in Oracle Cloud Infrastructure for the user you specified. See Prerequisites for PaaS Services on Oracle Cloud Infrastructure in the Oracle Cloud Infrastructure documentation.

Configure the Load Balancer

This topic applies only to Oracle Cloud Infrastructure.

On the Service Details page of the Wizard, specify details to configure the load balancer(s) for the Oracle Java Cloud Service instance.

Complete the following fields:

Load Balancer Details	Description
Load Balancer	Select the type of load balancer that you want to configure for your service instance: • **Oracle-Managed Load Balancer**: A dual-node, Oracle-managed instance of the Oracle Cloud Infrastructure Load Balancing service, providing active-passive high-availability. Failover from the active load-balancer node to the other node occurs automatically. You can't customize the default listeners, certificates, and so on for an Oracle Cloud Infrastructure Load Balancing instance that is provisioned by Oracle Java Cloud Service. If you need the ability to configure Oracle Cloud Infrastructure Load Balancing, then you must create the load balancer manually. See Set Up an Oracle Cloud Infrastructure Load Balancer. • **Oracle Traffic Director**: One or two Oracle Traffic Director nodes within your service instance. The dual-node configuration is in active-active mode, but failover to the second node is not automatic. • **None**: No load balancer will be configured for this instance. Provisioning a load balancer is recommended if the cluster size is 2 or more. The default value is **None**. If you selected **Enable Authentication Using Identity Cloud Service**, then you cannot configure a user-managed load balancer. You must select **Oracle-Managed Load Balancer**. If you select **Oracle Traffic Director** and configure one Oracle Traffic Director node, you can also add a second Oracle Traffic Director node to a service instance after creating the service instance. If you configured two Oracle Traffic Director nodes during provisioning, you cannot add another Oracle Traffic Director node. If you select **None**, then you can add an Oracle Traffic Director load balancer after creating the service instance.
Compute Shape	This option is displayed only if **Oracle Traffic Director** is selected as the load balancer. Select the compute shape to use for all the load balancer nodes in the service instance. The compute shape is the number of Oracle Compute Units (OCPUs) and amount of memory (RAM) that you want to allocate to these nodes. The `VM.Standard.*` shapes and `BM.Standard.*` shapes are supported. You are billed for Oracle Traffic Director nodes at the same price that you are billed for WebLogic Server nodes in your Oracle Java Cloud Service subscription. See About Oracle Java Cloud Service Subscriptions and Licenses.
Add Another Active OTD Node	This option is displayed only if **Oracle Traffic Director** is selected as the load balancer. Select this check box to provision a second load balancer node running Oracle Traffic Director (OTD) in this service instance. Both load balancer nodes route traffic to the cluster of WebLogic Managed Servers. You can also add a second load balancer node to a service instance after creating the service instance.

Load Balancer Details	Description
Load Balancing Policy	This option is displayed only if you selected **Oracle-Managed Load Balancer** or **Oracle Traffic Director** as the load balancer. If you selected **Oracle Traffic Director**, choose one of the following policies: • **Least Connection Count** (default)—Passes each new request to the Managed Server with the least number of connections. This policy is useful for smoothing distribution when a Managed Server receives more requests than it can handle efficiently. • **Least Response Time**—Passes each new request to the Managed Server with the fastest response time. • **Round Robin**—Evenly distributes requests across all Managed Servers, regardless of the number of connections or response times. If you selected **Oracle-Managed Load Balancer**, choose one of the following policies: • **Round Robin**— (default) Same as above. • **IP Hash**—The IP Hash policy uses an incoming request's source IP address as a hashing key to route traffic to the same backend server. The load balancer routes requests from the same client to the same backend server as long as that server is available. • **Least Connection Count**—Same as above.
Subnet for Load Balancer Node 1	This field is displayed only if **Load Balancer** is set to **Oracle-Managed Load Balancer**
Subnet for Load Balancer Node 2	Select the **Use Regional Subnet** check box to select regional subnet. For regional subnets: • Oracle recommends that you specify a regional subnet OCID for the load balancer to support failover to another availability domain if needed. • You can only assign one regional subnet. If you specify a regional subnet OCID for either the **Subnet for Load Balancer Node 1** or **Subnet for Load Balancer Node 2** menu, the other menu is not displayed. For non-regional (availability domain-scoped) subnets: • For each load balancer node, specify non-regional subnet OCID from a different availability domain. You must specify OCIDs for two non-regional subnets. • If the selected region has only one availability domain, **Subnet for Load Balancer Node 2** is not shown. In this case, you can only specify one non-regional subnet OCID, which is assigned to both nodes. • For at least one of the nodes, Oracle recommends specifying a non-regional subnet OCID from the same availability domain as that of the service instance. This ensures that, as long as the service instance is running, the applications deployed on it remain accessible through the load balancer.

Confirm Your Oracle Java Cloud Service Instance Creation

This topic applies only to Oracle Cloud Infrastructure.

On the Confirmation page of the provisioning wizard, review the service details.

If you need to change the service details, use the navigation bar or **Back** button at the top of the wizard to step back through the pages in the wizard. Click **Cancel** to cancel out of the wizard without creating a new service instance. If you are satisfied with your choices on the Confirmation page, click **Create**.

Sample of Options Displayed

If you selected the **Bring Your Own License** option, the Confirmation page will display a message alerting you to the fact that you have chosen to use an existing license. Check to make sure you have the appropriate entitlements.

The compute shape and server count is displayed in the WebLogic Configuration section.

Download the Instance Attributes in JSON Format

(Not available on Oracle Cloud at Customer)

Click **Download** ⬇ to download a JSON-format file containing the parameters you specified in the provisioning wizard. You can use the JSON-formatted file as a sample to construct the request body for creating instances using the REST API.

Note that the file contains placeholders for passwords.

After Confirmation

After the Confirmation page closes, the Oracle Java Cloud Service console opens. Optionally, you can click on the service instance name to view status messages. If provisioning of your service instance fails but there are no fatal errors, the software automatically retries provisioning, after a lag time of 60 minutes. Messages about the auto-retry process and failed compute resources are displayed.

If you provided your email address for the **Notification Email** option, you will receive an email notification when the service instance provisioning has succeeded or failed.

Next Steps

- After the service instance has been created, you can view the system messages logged during the creation process, including error messages. Click **Instance Create and Delete History**, then click the service instance name or **Details**.
- If the provisioning process retried provisioning automatically, some failed resources might still exist. To clean up these failed resources, click the **Complete Cleanup** button. If you click the button once and not all failed resources are cleaned up, the **Complete Cleanup** button will remain. If this is the case, click the button again and wait. Repeat this process until the button is not longer displayed and all failed resources are cleaned up.
- If you selected the **Enable Authentication with Oracle Identity Cloud Service** option, you can use Oracle Identity Cloud Service to create additional WebLogic Server users. See Use Oracle Identity Cloud Service with Oracle Java Cloud Service.
- If you selected the **Deploy Sample Application** option, and want to test the sample application, see About the Sample Application Deployed to an Oracle Java Cloud Service Instance.

Create an Oracle Java Cloud Service Instance Attached to a Private Subnet Using REST API

Use the REST API to create an Oracle Java Cloud Service instance attached to a private subnet.

1. Create a request body in JSON format by using the following template, and save it in a plain-text file (for example, `create-jcs-instance-on-oci.json`).

   > **Note:**
   >
   > This request-body template includes only the minimum set of fields required to create an instance of Oracle Java Cloud Service running Oracle WebLogic Server 12.2.1.3 Enterprise Edition. For information about all the supported fields, see Create a Service Instance in *REST API for Oracle Java Cloud Service*.

```
{
  "serviceName"          : "name",
  "region"               : "region",
  "availabilityDomain"   : "ad",
  "subnet"               : "privateSubnetOCID",
  "vmPublicKeyText"      : "publicKey",
  "components": {
    "WLS": {
      "adminUserName"               : "user",
      "adminPassword"               : "password",
      "sampleAppDeploymentRequested": "true",
      "clusters": [
        {
          "clusterName"             : "name",
          "serverCount"             : "number",
          "shape"                   : "shape",
          "type"                    : "APPLICATION_CLUSTER"
        }
      ],
      "connectString"               : "dbConnectString",
      "dbaName"                     : "SYS",
      "dbaPassword"                 : "password"
    }
  },
  "configureLoadBalancer"           : true
  "loadbalancer": {
    "subnets": [
      "subnetOCID_primaryLBnode",
      "subnetOCID_standbyLBnode"
    ],
    "loadBalancingPolicy"           : "policy"
  },
  "cloudStorageContainer": "https://
swiftobjectstorage.region.oraclecloud.com/v1/namespace/bucket",
  "cloudStorageUser"     : "OCIuser",
  "cloudStoragePassword" : "authToken"
}
```

- `serviceName`: A name that starts with a letter, includes only letters and numbers, and has not more than 30 characters.
- `region`: The Oracle Cloud Infrastructure region in which you want to create the Oracle Java Cloud Service instance (for example, `us-ashburn-1`).
- `availabilityDomain`: The Oracle Cloud Infrastructure availability domain in which you want the Oracle Java Cloud Service instance to be created (for example, `QnsC:US-ASHBURN-AD-1`).
- `subnet`: The OCID of the private subnet to which you want to attach the Oracle Java Cloud Service instance.
- `vmPublicKeyText`: The SSH public key that you want to use for the nodes of the instance.
- `adminUserName`: The user name for the Oracle WebLogic Server administrator.

  The name must be between 8 and 128 characters long. It must not contain any of the following characters: tabs, brackets, parentheses, left angle bracket (<), right angle bracket (>), ampersand (&), pound sign (#), pipe symbol (|), and question mark (?).
- `adminPassword`: The password for the Oracle WebLogic Server administrator.

  The password must start with a letter. It can contain from 8 to 30 characters, and must include at least one number.
- `sampleAppDeploymentRequested`: `true`
- `clusterName`: The name of the Oracle WebLogic Server cluster.

  The name must start with a letter and have not more than 50 characters. It can contain only alphabetical characters, underscores (_), and dashes (-).
- `serverCount`: 1, 2, 4, or 8
- `shape`: Any `VM.Standard` or `BM.Standard` shape that's available in the availability domain that you specified. Check the service limits displayed in the Oracle Cloud Infrastructure web console.
- `type`: `APPLICATION_CLUSTER`
- `connectString`: The connection string for the Oracle Cloud Infrastructure Database system that you created earlier.
- `dbaName`: A database user with the `SYSDBA` privilege. For instances based on Oracle WebLogic Server 12c (any version), you can use the database user `SYS`.
- `dbaPassword`: The password that you specified for the database administrator while creating the Oracle Cloud Infrastructure Database system.
- `configureLoadBalancer`: `true`

> **Note:**
>
> If you need the ability to configure the load balancer (add or modify listeners, use your own certificates, and so on), then don't include this field in the request body. Don't include the fields under `loadbalancer` either. Create an instance of Oracle Cloud Infrastructure Load Balancing manually. See Set Up an Oracle Cloud Infrastructure Load Balancer.

- `loadbalancer.loadBalancingPolicy`: Specify one of the following:
  - `LEAST_CONN`: Each new request is routed to the server with the least number of active connections.
  - `IP_HASH`: Requests from the same client are always routed to the same server, if the server is available.
  - `ROUND_ROBIN`: The load balancer selects the next server for each request by cycling through the available servers in a fixed order.
- `loadbalancer.subnets`: The OCIDs of the subnets for the load-balancer nodes. If the region you've selected has only one availability domain, then specify only one subnet.
- `cloudStorageContainer`: The URL of the Oracle Cloud Infrastructure Object Storage bucket (for example, `https://swiftobjectstorage.us-ashburn-1.oraclecloud.com/v1/mynamespace/jcs_bucket`).
- `cloudStorageUser`: The user name of the user who created the bucket or has access to it.
- `cloudStoragePassword`: The authentication token that you generated.

The following example shows a completed request body.

```
{
  "serviceName"           : "myJCS",
  "region"                : "us-ashburn-1",
  "availabilityDomain"    : "QnsC:US-ASHBURN-AD-1",
  "subnet"                : "ocid1.subnet.oc1.iad.aaaaaaaamgxfkk5...
(truncated)",
  "vmPublicKeyText"       : "ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEA...
(truncated)",
  "components": {
    "WLS": {
      "adminUserName"                : "adminuser",
      "adminPassword"                : "password",
      "sampleAppDeploymentRequested": "true",
      "clusters": [
        {
          "clusterName"              : "myJCScluster",
          "serverCount"              : "2",
          "shape"                    : "VM.Standard2.1",
          "type"                     : "APPLICATION_CLUSTER"
        }
      ],
      "connectString"                : "//dbforjcs-
scan.privatesubnet.paasvcn.oraclevcn.com:1521/
pdb1.privatesubnet.paasvcn.oraclevcn.com",
      "dbaName"                      : "SYS",
      "dbaPassword"                  : "password"
    }
  },
  "configureLoadBalancer"            : true
  "loadbalancer": {
    "subnets": [
      "ocid1.subnet.oc1.iad.aaaaaaaa6j5... (truncated)",
      "ocid1.subnet.oc1.iad.aaaaaaaaj4t... (truncated)"
    ],
    "loadBalancingPolicy"            : "LEAST_CONN"
  },
  "cloudStorageContainer": "https://swiftobjectstorage.us-
ashburn-1.oraclecloud.com/v1/mynamespace/jcs_bucket",
  "cloudStorageUser"     : "john.smith@example.com",
  "cloudStoragePassword" : "sometoken"
}
```

2. Send the REST API request.

```
curl -X POST rest_endpoint/paas/api/v1.1/instancemgmt/
identityServiceID/services/jaas/instances \
-u user:password \
-H 'X-ID-TENANT-NAME: identityServiceID' \
-H 'Content-Type: application/json' \
-d @requestBodyFile
```

- `restEndpoint`: The REST endpoint URL of Oracle Java Cloud Service.
- `identityServiceID`: The identity service ID of your Oracle Cloud account.
- `user`: Your Oracle Cloud user name.
- `password`: Your Oracle Cloud password.
- `requestBodyFile`: The path and name of the file containing the request body.

The following is an example of a REST API request to create an Oracle Java Cloud Service instance.

```
curl -X POST https://jaas.oraclecloud.com/paas/api/v1.1/
instancemgmt/idcs-33e8886d2e6666e7777d14ffa9999e83/services/jaas/
instances \
-u john.smith@example.com:password \
-H 'X-ID-TENANT-NAME: idcs-33e8886d2e6666e7777d14ffa9999e83' \
-H 'Content-Type: application/json' \
-d @create-jcs-instance-on-oci.json
```

A message similar to the following is displayed, indicating that the request was accepted.

```
{
  "details": {
    "message": "Submitted job to create service [myJCS] in domain
[idcs-33e8886d2e6666e7777d14ffa9999e83].",
    "jobId": "50572730"
  }
}
```

3. In the message, note the value in the `jobId` field.

4. Wait for the instance to be created.

   You can check the status in the Oracle Java Cloud Service web console.

   Alternatively, you can send the following REST API request to find out the status of the job.

   ```
   curl rest_endpoint/paas/api/v1.1/activitylog/identityServiceID/job/ID \
   -u user:password \
   -H 'X-ID-TENANT-NAME: identityServiceID'
   ```

   - `restEndpoint`: The REST endpoint URL of Oracle Java Cloud Service.
   - `identityServiceID`: The identity service ID of your Oracle Cloud account.
   - `ID`: The job ID that you noted in the previous step.
   - `user`: Your Oracle Cloud user name.
   - `password`: Your Oracle Cloud password.

   The following is an example of a REST API request to check the status of a request to create an Oracle Java Cloud Service instance.

   ```
   curl https://jaas.oraclecloud.com/paas/api/v1.1/activitylog/
   idcs-33e8886d2e6666e7777d14ffa9999e83/job/50572730 \
   -u john.smith@example.com:password \
   -H 'X-ID-TENANT-NAME: idcs-33e8886d2e6666e7777d14ffa9999e83'
   ```

   In the output, look for the `status` field. It shows `ready` after the instance is created.

> **Note:**
>
> The compute nodes of Oracle Java Cloud Service instances that are attached to private subnets in Oracle Cloud Infrastructure have private IP addresses. So you can’t `ssh` to the nodes or access the administration consoles of such instances from the public Internet.
>
> You can access the administration consoles and connect to the nodes of such instances through a bastion host attached to a public subnet. See Access the Administration Console for a Service Instance Attached to a Private Subnet.

Create a Custom Oracle Java Cloud Service Instance on Oracle Cloud Infrastructure Classic

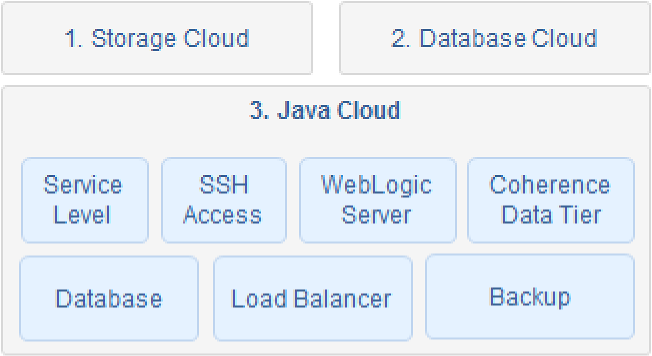
Oracle Java Cloud Service gives you detailed control over how you create your service instance.

There are several methods that you can use to create instances in Oracle Java Cloud Service. One of the easiest methods to create an instance is to use the Create Instance wizard in the web console. The wizard guides you through a short series of screens that present all the parameters that you can configure for your instance, including the WebLogic Server settings, backup and recovery configuration, load balancer parameters, and so on.

You can provision Oracle WebCenter Portal and Oracle Data Integrator on your Oracle Java Cloud Service only using the REST API.

Video

Tutorial



Prerequisites

Before creating a custom Oracle Java Cloud Service instance:

- Review the prerequisites described in Before You Begin with Oracle Java Cloud Service
- Review the options described in Design Considerations for an Oracle Java Cloud Service Instance

Procedure

- Start the Create New Instance Wizard
- Specify Basic Service Instance Information
- Specify WebLogic Configuration
- Assign Reserved IP Addresses for a Service Instance in a Region
- Assign Reserved IP Addresses for an Oracle Database Exadata Cloud Service Database
- Configure WebLogic Server Access
- Configure the Coherence Data Tier
- Configure the Databases
- Configure the Load Balancer
- Configure Backup and Recovery

- Confirm Your Oracle Java Cloud Service Instance Creation

Start the Create New Instance Wizard

This topic does not apply to Oracle Cloud Infrastructure.

To create an service instance from the web console, you use the Create New Instance wizard.

To start the Create New Instance wizard:

1. Access your service console.
2. Click **Create Instance**.

Specify Basic Service Instance Information

This topic does not apply to Oracle Cloud Infrastructure.

On the Instance page of the Instance Creation Wizard, enter basic information for your service instance, including service name, service level, metering frequency, software release, and software edition.

> **Note:**
>
> Except for tags, you cannot change any of the following options after you have created the service instance.

Complete the following fields:

Field	Description
Instance Name	Specify a name for the Oracle Java Cloud Service instance. The service instance name: • Must contain one or more characters. • Must not exceed 30 characters. • Must start with an ASCII letter: `a` to `z` , or `A` to `Z`. • Must contain only ASCII letters or numbers. • Must not contain a hyphen. • Must not contain any other special characters. • Must be unique within the identity domain.
Description	(Optional) Enter a short description of the Oracle Java Cloud Service instance.
Notification Email	(Optional) Specify an email address where you would like to receive a notification of any events occurring with the service instance, including whether provisioning has succeeded or failed.

Field	Description
Region	(Available only if your account has regions) Select a region if you want to create the service instance in a specific region, or if you want to use a custom IP network. You must also select a region if you intend to assign reserved IP addresses to your service instance nodes. A region supports either Oracle Cloud Infrastructure or Oracle Cloud Infrastructure Classic. For a list of available regions, see Data Regions for Platform and Infrastructure Services. The database that you intend to associate with your Oracle Java Cloud Service instance must be in the same region. If you select **No Preference**, Oracle Java Cloud Service will select one of the available Oracle Cloud Infrastructure Classic regions. However, you will not be able to use an IP network or reserved IP addresses for your service instance.
IP Network	(Only if a region is selected) (Not available on Oracle Cloud Infrastructure) Select an IP network if you want to create the service instance in an IP network that you've defined. By default, each node in your instance is auto-assigned a public and a private IP address. The IP addresses might change each time the service instance is restarted. You can reserve and assign fixed public IP addresses. In order to select an IP network if you have selected **Enable Authentication Using Identity Cloud Service**, which automatically configures a managed load balancer, you must first attach an internet-facing load balancer to the IP network. This field is not relevant to Oracle Cloud Infrastructure.
Assign Public IP	(Not available on Oracle Cloud Infrastructure) Choose whether to assign public IP addresses to the nodes in your service instance. You must first select an Oracle Cloud Infrastructure Classic region and specify an IP network. If you select this check box (default), then any node added during instance provisioning, or later added as part of a scaling operation, will have a public IP address assigned to it. You will be able to directly access the nodes from the public Internet. This selection is for use cases where you intend to deploy Java EE applications to the Oracle Java Cloud Service instance and access them from the public Internet. If you deselect this check box, then any node added during instance provisioning, or later added as part of a scaling operation, will not have a public IP address assigned to it. You will not be able to directly access the nodes from the public Internet. This selection is for use cases where you intend to deploy Java EE applications to the Oracle Java Cloud Service instance and access them only within your IP network or from your on-premises data center over a VPN network.
Tags	(Optional) Select existing tags or add tags to associate with the service instance. To select existing tags, select one or more check boxes from the list of tags that are displayed on the pull-down menu. To create tags, click + to display the **Create Tags** dialog box. In the **New Tags** field, enter one or more comma-separated tags that can be a key or a key:value pair. If you do not assign tags during provisioning, you can create and manage tags after the service instance is created.

Field	Description
Identity Domain	(Not available on Oracle Cloud at Customer) Select the identity domain in Oracle Identity Cloud Service in which to create this service instance. By default, the instance is created in the primary identity domain.
The service security administrator	(Not available on Oracle Cloud at Customer) (Optional) Specify the username for the security administrator for the service instance in the selected identity domain. This user gets rights to administer security artifacts (roles, AppId, OAuth IDs, and so on). The username can be the administrator of the selected identity domain or a user in the selected identity domain. You can leave this field blank *only* if you are the administrator of the selected identity domain or a user in the selected identity domain.
License Type	Choose whether you want to leverage the Bring Your Own License (BYOL) option or use your Oracle Java Cloud Service license. • The **Bring Your Own License** (BYOL) option enables you to bring your on-premises Oracle WebLogic Server licenses to Oracle Cloud. BYOL instances are billed at a lower rate than other instances. See Frequently Asked Questions: Oracle BYOL to PaaS. You must own a Universal Credits subscription or Government subscription in order to use BYOL. **Note**: Before you scale up or scale out a BYOL instance, you must have enough WebLogic Server licenses for the additional OCPUs that will be allocated to the instance after it is scaled. • If you choose to use your Oracle Java Cloud Service license, your account will be charged for the new service instance according to your Oracle Java Cloud Service agreement. If you have both BYOL and Oracle Java Cloud Service entitlements, BYOL is selected by default, but you can change the license type. If you have BYOL entitlements only, BYOL is selected and you cannot change the license type. If you do not have BYOL entitlements, the Oracle Java Cloud Service license option is selected and you cannot change the license type.
Software Edition	Select a WebLogic Server software edition: • **Standard Edition** • **Enterprise Edition** • **High Performance Edition** If you purchased a Non-Metered subscription for Oracle Java Cloud Service, you can select a software edition that you did not purchase as part of your subscription. Making this selection will incur additional charges to your account.
Metering Frequency	This option appears only if you have a traditional metered subscription. If you have a Universal Credits subscription, this field is absent. Select a metering frequency to determine how you are billed for this service instance: • **Hourly**—Pay only for the number of hours that this service instance was running during your billing period. • **Monthly**—Pay one price for the full month irrespective of the number of hours that this service instance was running. For services that are started in the middle of a month, the price will be pro-rated; you pay only for the partial month from the day the service instance is created.

Specify the Service Instance Details

This topic does not apply to Oracle Cloud Infrastructure.

You must configure the size, shape, and other important details for your Oracle Java Cloud Service instance.

Topics

- Specify WebLogic Configuration
- Assign Reserved IP Addresses for a Service Instance in a Region
- Assign Reserved IP Addresses for an Oracle Database Exadata Cloud Service Database
- Configure WebLogic Server Access
- Configure the Coherence Data Tier
- Configure the Databases
- Configure Backup and Recovery
- Configure the Load Balancer

Specify WebLogic Configuration

This topic does not apply to Oracle Cloud Infrastructure.

On the second page of the Instance Creation Wizard (Service Details), you start by configuring the size and shape of the Oracle Java Cloud Service instance.

> **Note:**
>
> Two tabs, Simple and Advanced, control which fields appear on the page. Fields that appear when you select the Simple tab also appear when you select the Advanced tab, but some fields appear only when you select the Advanced tab.

Complete the following fields:

Size and Shape Details	Description
WebLogic Clusters	(Advanced option) If you selected **Enable Authentication Using Identity Cloud Service**, your instance will be provisioned with an Oracle-managed load balancer. You can add, edit, or delete up to 8 WebLogic clusters for the service instance, with a maximum of 8 servers per cluster. You specify the cluster name, compute shape, and server count. Optionally, you can specify a path prefix, which determines how the managed load balancer routes traffic to different clusters. If you do not specify a path prefix, the cluster name is used as the path prefix. After you specify these values, you can edit them: • Click **Add** to add a new cluster. • Select a cluster and click **Edit** to update its configuration. • Click **Delete** to delete the cluster. If **Enable Authentication Using Identity Cloud Service** is not selected, then a single cluster is created during instance provisioning. You cannot add clusters using the console, but clusters can be added using the REST API.
Compute Shape	Select the compute shape to use for all Administration Server and Managed Server nodes. The compute shape is the number of Oracle Compute Units (OCPUs) and amount of memory (RAM) that you want to allocate to these nodes. The selected shape is not used for Coherence or Load Balancer nodes. (Advanced option) When you create multiple WebLogic clusters, you can assign a different compute shape for different clusters. This field displays the compute shape of the selected cluster. If you purchased a Universal Credits subscription for Oracle Java Cloud Service, you will pay at the Pay-As-You-Go rate when you exceed your monthly or annual maximum credit.
Server Count	Select the initial number of Managed Servers that you want to provision in this service instance. The choices are: `1`, `2`, `4`. • This field is not relevant if you selected **Standard Edition**. In this case, only one Managed Server is configured. • If you configure more than one Managed Server in the cluster, Oracle recommends that you also enable the Load Balancer. • You can also perform scaling operations to increase or decrease the server count after provisioning the service instance. (Advanced option) When you create multiple clusters, you can assign a different server count to different clusters. You can configure a maximum of 8 servers per cluster. This field displays the server count for the selected cluster.
Reserved IPs	(Not available on Oracle Cloud Infrastructure) Select reserved IP addresses for the nodes in your cluster, or leave the default value as **Assign Automatically** if you want Oracle to automatically assign IP addresses to these nodes. The number of IP addresses that you select must equal the number of nodes in the cluster. This option is displayed only if you selected a specific **Region** for this service instance. You create IP reservations by using the **Reserved IPs** tab in the Oracle Java Cloud Service Console. If you do not see this tab on the console, click the gear icon next to this field and follow the instructions to create your first IP reservation. After creating IP reservations, you need to restart the instance creation wizard.
Domain Partitions	(Advanced option) Select the initial number of WebLogic Server domain partitions that you want to provision in this service instance. The choices are `0`, `1`, `2`, or `4`. This option is also not relevant if you selected **Standard Edition** as the software edition.

Size and Shape Details	Description
Enable Access to Administration Consoles	(Advanced option) Select this check box if you want to enable access to the WebLogic Service Administration Console, Fusion Middleware Control, and Load Balancer Console for the service instance. If you do not select this option, these consoles will not be externally accessible, and also will not appear as choices in the service instance's menu ☰. Alternatively, you can enable access to the administration consoles after creating the service instance. See Enable Console Access for a Service Instance If this check box is enabled, the **This Source CIDR range field can access Admin Consoles** option is displayed. By default, the source CIDR range is `0.0.0.0/0`, so the administration console is accessible from the public internet. You can specify a source CIDR range so that only the IP addresses within the specified range can access the administration console. **Note:** Enabling access to the administration console through port 7002 allows access to the WebLogic Service Administration console from the pubic internet if a source CIDR range is not specified, else it allows access to the IP addresses only within the specified CIDR range. If you have configured Oracle Traffic Director (OTD) for the service instance, enabling access to the administration console through port 8989 allows access to the Oracle Traffic Director and the Load Balancer console.
Deploy Sample Application	(Advanced option) By default, a sample application, `sample-app.war`, is deployed automatically to the Managed Servers in your instance. If you do not want to automatically deploy the sample application, deselect this check box.

Assign Reserved IP Addresses for a Service Instance in a Region

This topic does not apply to Oracle Cloud Infrastructure.

If regions are enabled in your identity domain, you can select a region in which your Oracle Java Cloud Service instance will reside. If a region is selected, you can assign reserved IPs from within that region for your service instance nodes.

Complete the following field:

IP Reservation Details	Description
Reserved IPs	(Not available on Oracle Cloud Infrastructure) Select reserved IP addresses for the nodes in your cluster, or leave the default value as **Assign Automatically** if you want Oracle to automatically assign IP addresses to these nodes. The number of IP addresses that you select must equal the number of nodes in the cluster. This option is displayed only if you selected a specific **Region** for this service instance. You create IP reservations by using the **Reserved IPs** tab in the Oracle Java Cloud Service Console. If you do not see this tab on the console, click the gear icon next to this field and follow the instructions to create your first IP reservation. After creating IP reservations, you need to restart the instance creation wizard.

Assign Reserved IP Addresses for an Oracle Database Exadata Cloud Service Database

This topic does not apply to Oracle Cloud Infrastructure.

If you are provisioning an Oracle Java Cloud Service instance in an Oracle Cloud Infrastructure Classic region, you can associate your service instance with an Oracle Database Exadata Cloud Service database for an infrastructure schema or application database.

The procedure for assigning reserved IP addresses for the Oracle Database Exadata Cloud Service infrastructure schema or application database is the same as the procedure for assigning reserved IP addresses for any other database, except that you must first whitelist the IP addresses you want to assign. See Enabling Network Access to a Compute Node in *Administering Oracle Database Exadata Cloud Service*.

Configure WebLogic Server Access

This topic does not apply to Oracle Cloud Infrastructure.

On the Service Details page of the Wizard, configure the administrator credentials for the WebLogic Servers.

Complete the following fields:

Access Details	Description
SSH Public Key	Specify the public key that will be used for authentication when connecting to a node in your instance by using a Secure Shell (SSH) client. Click **Edit** to display the SSH Public Key for VM Access dialog, and then specify the public key using one of the following methods: • Select **Key file name** and use your web browser to select a file on your machine that contains the public key. • Select **Key value** and paste the value of the public key into the text area. Be sure the value does not contain line breaks or end with a line break. • Select **Create a New Key** if you want Oracle to generate a public/private key pair for you. You will be prompted to download these generated keys. If you choose to create a new key, the generated private key file is in OpenSSH format. Before connecting to a node in this service instance with the PuTTY SSH client, you must first convert the key to PuTTY’s proprietary format.

Access Details	Description
Local Administrative User Name	Enter your choice of user name for the WebLogic Server administrator. The default is `weblogic`. This name is used to access the WebLogic Server Administration Console, Fusion Middleware Control, and Load Balancer Console for the service instance. The name must be between 8 and 128 characters long and **cannot** contain any of the following characters: • Tab • Brackets • Parentheses • These special characters: – Left angle bracket (`<`) – Right angle bracket (`>`) – Ampersand (`&`) – Pound sign (`#`) – Pipe symbol (`|`) – Question mark (`?`) You can also change the user name through the WebLogic Server Administration Console after the service instance is provisioned.
Password	Specify a password for the WebLogic Server administrator and confirm the password. If you selected an Oracle Database Exadata Cloud Service database deployment for **Database Instance Name**, this password must start with a letter, be of 8 to 30 characters in length, and contain at least: • 1 uppercase character • 1 lower case character • 1 digit (0 through 9) • One of the following special characters: _ (underscore), - (hyphen), or # (pound sign or hash) If you did not select an Oracle Database Exadata Cloud Service database deployment, Oracle still recommends following these password requirements as a best practice. However, the following basic password criteria are acceptable: • Starts with a letter • Is between 8 and 30 characters long • Contains letters, at least one number, and, optionally, any number of these special characters: – Dollar sign (`$`) – Pound sign (`#`) – Underscore (`_`) No other special characters are allowed.
Enable Authentication Using Identity Cloud Service	Select this check box if you want WebLogic Server to authenticate application users and administrators against Oracle Identity Cloud Service in addition to the local WebLogic Server identity store. This field appears only if your cloud account includes Oracle Identity Cloud Service and Oracle Cloud Infrastructure Load Balancing Classic. By default, the WebLogic Server domain in the service instance is configured to use only the local WebLogic Server identity store to maintain administrators, application users, groups, and roles.

Configure the Coherence Data Tier

This topic does not apply to Oracle Cloud Infrastructure.

If you want to create a Coherence Data Tier, provide details on the Service Details page of the Wizard.

Complete the following fields:

Coherence Data Tier	Description
Provision Data Grid Cluster	(Advanced option) Select **Yes** to provision a Coherence data grid cluster in your service instance. This option is only available if you selected **High Performance Edition**.
Compute Shape	Select the compute shape to use for all Managed Server nodes in the data grid cluster. The compute shape is the number of Oracle Compute Units (OCPUs) and amount of memory (RAM) that you want to allocate to these nodes. This option is displayed only if **Provision Data Grid Cluster** is set to `Yes`.
Cluster Size	Set the initial number of Managed Servers that you want to provision in the data grid cluster. Valid values are 1–4. This option is displayed only if **Provision Data Grid Cluster** is set to `Yes`. The number of nodes in the data grid cluster is determined by **Cluster Size** / **Managed Servers Per Node**. If this ratio is a fraction, the number of nodes is rounded up to the next integer. You can also perform scaling operations to increase or decrease the number of Coherence nodes after provisioning the service instance. You cannot specify multiple data grid clusters.
Managed Servers Per Node	Set the number of Coherence Managed Servers to run on each node in the data grid cluster. Valid values are 1–8. This option is displayed only if **Provision Data Grid Cluster** is set to `Yes`.

Configure the Databases

This topic does not apply to Oracle Cloud Infrastructure.

On the Service Details page of the Wizard, provide details about the database(s) to use for the Oracle Java Cloud Service instance.

Complete the following fields:

Database Details	Description
Database Instance Name	Select an existing Oracle Database Cloud Service (Classic) deployment or Oracle Database Exadata Cloud Service deployment to connect to this service instance. Oracle Java Cloud Service provisions the selected database with the required schemas for running a service instance. The list only includes a database deployment if it meets the following criteria: • Is in an active state and not currently in the process of being provisioned • Is not configured with a **Backup Destination** set to `None` (not applicable to Oracle Database Cloud Service — Virtual Image deployments). Note the following additional constraints and limitations: • To ensure that you can restore the database for an Oracle Java Cloud Service instance without risking data loss for other service instances, Oracle recommends that you do not associate the same infrastructure schema database (or the same pluggable database) with multiple service instances. Backups of a database that is used with multiple Oracle Java Cloud Service instances contain data for all the instances. Therefore, if you restore the database from a backup, data for all the service instances is restored, which might not be the intended result. • Oracle Java Cloud Service does not support Oracle Database 18c. • If you selected an **IP Network** for this service instance, you must also select an Oracle Database Cloud Service (Classic) database deployment that is attached to an IP network. If the service instance and database deployment are attached to different IP networks, the two IP networks must be connected to the same IP network exchange.
PDB Name	Specify the pluggable database the service instance will connect to. If you don't specify a PDB name, Oracle Java Cloud Service uses the default Oracle Database 12c PDB name that was provided when the Oracle Database Cloud Service (Classic) database deployment was originally created.
Administrator User Name	Enter the name of the database administrator that Oracle Java Cloud Service will use to connect to the selected database deployment and to provision the required schemas for this service instance.
Password	Enter the password for the database administrator.

Database Details	Description
Add Application DB	(Advanced option) Add a up to four database deployments for your application schema. Click **Add** if you want to specify a separate Oracle Database Cloud Service database deployment or Oracle Database Exadata Cloud Service database dedicated for your application schema. When you add an application database, the Oracle Java Cloud Service creates an additional data source in your Oracle WebLogic Server domain to connect to this database. Use the Add Database Configuration dialog to select the name of an existing Oracle Database Cloud Service database deployment or Oracle Database Exadata Cloud Service database, and to provide a user name and password for this database. Click **Add** and repeat this process for up to three more database deployments.

Configure Backup and Recovery

This topic does not apply to Oracle Cloud Infrastructure.

On the Service Details page of the Wizard, specify details on the storage used for backup and recovery.

Complete the following fields:

Backup and Recovery Details	Description
Backup Destination	(Advanced option) Select **Both Remote and Disk Storage** if you want to enable automated and on-demand backups for this service instance. Backups will be saved to object storage *and* to block storage volumes that are attached to the nodes of the instance. The default value is **None**, meaning that you cannot use Oracle Java Cloud Service to take backups of this service instance. You can configure backups on a service instance after creating it. This field is not relevant if you selected **Oracle Java Cloud Service—Virtual Image**.

Backup and Recovery Details	Description
Object Storage Container	This field is displayed only if **Backup Destination** is set to **Both Remote and Disk Storage**. Enter the object storage location where backups of the service instance must be stored by specifying the URL of a container. The object storage container field in the instance creation wizard is auto-populated with a default container URL in the format *`restEndpointUrl`*`/JaaS`, where *`restEndpointUrl`* is the REST endpoint URL of theOracle Cloud Infrastructure Object Storage Classic service in the account, and `JaaS` is the default container name. You can change the container name. Note that if the account doesn't include an Oracle Cloud Infrastructure Object Storage Classic service entitlement, then the container field is not auto-populated. If you have a container, specify the URL of the container in Oracle Cloud Infrastructure Object Storage Classic. **Format**: *`rest_endpoint_url`*`/`*`containerName`* You can find the REST endpoint URL of the Oracle Cloud Infrastructure Object Storage Classic service instance in the Infrastructure Classic Console. **Example**: `https://acme.storage.oraclecloud.com/v1/MyService-acme/MyContainer` **Note**: You can select the **Create Object Storage Container** check box to have a new container created automatically.
User Name	This field is displayed if **Backup Destination** is set to **Both Remote and Disk Storage**, except if you selected **Enable Authentication Using Identity Cloud Service**. Enter the user name of the Oracle Cloud Infrastructure Object Storage Classic service user who created the container you specified earlier. If the container doesn't exist, then enter the user name of a service administrator.
Password	This field is displayed only if **Backup Destination** is set to **Both Remote and Disk Storage**, except if you selected **Enable Authentication Using Identity Cloud Service**. Enter the password of the user you specified.
Create Object Storage Container	This option is displayed only if **Backup Destination** is set to **Both Remote and Disk Storage**. If the Oracle Cloud Infrastructure Object Storage Classic container that you specified doesn't exist, or if you aren't sure whether it exists, then select this check box. If the container doesn't exist, it will be created automatically.

Configure the Load Balancer

This topic does not apply to Oracle Cloud Infrastructure.

On the Service Details page of the Wizard, specify details to configure the load balancer(s) for the Oracle Java Cloud Service instance.

Complete the following fields:

Load Balancer Details	Description
Provision Local Load Balancer	(Advanced option) Select **Yes** to provision a load balancer node running Oracle Traffic Director in this service instance. This user-managed load balancer is configured to distribute client requests to the Managed Servers in the service instance. Provisioning a load balancer is recommended if the cluster size is 2 or more. The default value is **No**. If you selected **Enable Authentication Using Identity Cloud Service**, then you cannot configure a user-managed load balancer. An Oracle-managed load balancer is provisioned for you automatically. You can also add an Oracle Traffic Director load balancer node to a service instance after creating the service instance.
Compute Shape	This option is displayed only if **Provision Local Load Balancer** is set to `Yes`. Select the compute shape to use for all the load balancer nodes in the service instance. The compute shape is the number of Oracle Compute Units (OCPUs) and amount of memory (RAM) that you want to allocate to these nodes. You are billed for load balancer nodes at the same price that you are billed for WebLogic Server nodes in your Oracle Java Cloud Service subscription.
Add Another Active OTD Node	This option is displayed only if **Provision Local Load Balancer** is set to `Yes`. Select this check box to provision a second load balancer node running Oracle Traffic Director (OTD) in this service instance. Both load balancer nodes route traffic to the cluster of WebLogic Managed Servers. You can also add a second load balancer node to a service instance after creating the service instance.
Reserved IPs	Select reserved IP addresses for the load balancer nodes in your cluster, or leave the default value as **Assign Automatically** if you want Oracle to automatically assign IP addresses to these nodes. The number of IP addresses that you select must equal the number of load balancer nodes in the service instance. This option is displayed only if these conditions are true: • You selected a specific **Region** for this service instance. • **Provision Local Load Balancer** is set to `Yes` You create IP reservations by using the **Reserved IPs** tab in the Oracle Java Cloud Service Console. If you do not see this tab on the console, click the gear icon next to this field and follow the instructions to create your first IP reservation. After creating IP reservations, you need to restart the instance creation wizard. See Managing IP Reservations.

Load Balancer Details	Description
Load Balancing Policy	This option is displayed if you selected **Enable Authentication Using Identity Cloud Service** or **Provision Local Load Balancer**. If you selected **Provision Local Load Balancer**, choose one of the following policies: • **Least Connection Count** (default)—Passes each new request to the Managed Server with the least number of connections. This policy is useful for smoothing distribution when a Managed Server receives more requests than it can handle efficiently. • **Least Response Time**—Passes each new request to the Managed Server with the fastest response time. • **Round Robin**—Evenly distributes requests across all Managed Servers, regardless of the number of connections or response times. If you selected **Enable Authentication Using Identity Cloud Service**, choose one of the following policies: • **Round Robin**— (default) Same as above. • **IP Hash**—The IP Hash policy uses an incoming request's source IP address as a hashing key to route traffic to the same backend server. The load balancer routes requests from the same client to the same backend server as long as that server is available. • **Least Connection Count**—Same as above. You can also use the Load Balancer console to modify this policy after creating the service instance.
Load Balancer Type	This field is displayed only if you have specified an Oracle Cloud Infrastructure Classic region and an IP network, and you have selected Oracle Identity Cloud Service as the authentication provider. Select **Public** or **Private** as the load balancer type. • **Public**—Enables access to the service instance from the public Internet. • **Private**—Used to enable access to private workloads over VPN from your on-premises network. This feature is only available forOracle Cloud Infrastructure Classic.

Confirm Your Oracle Java Cloud Service Instance Creation

This topic does not apply to Oracle Cloud Infrastructure.

On the Confirmation page of the provisioning wizard, review the service details.

If you need to change the service details, use the navigation bar or **Back** button at the top of the wizard to step back through the pages in the wizard. Click **Cancel** to cancel out of the wizard without creating a new service instance. If you are satisfied with your choices on the Confirmation page, click **Create**.

Sample of Options Displayed

If you selected the **Bring Your Own License** option, the Confirmation page will display a message alerting you to the fact that you have chosen to use an existing license. Check to make sure you have the appropriate entitlements. If you have selected BYOL license type, a link to BYOL terms appears. Click the link to open the BYOL FAQ.

If you assigned tags to the service instance, the tags are displayed in the **Service** section.

The compute shape and server count is displayed in the WebLogic Configuration section. If you have multiple clusters, this section displays the compute shape and server count for each cluster. To display the compute shape and server count for all clusters, select **Show more**.

Download the Instance Attributes in JSON Format

(Not available on Oracle Cloud at Customer)

Click **Download** ⤓ to download a JSON-format file containing the parameters you specified in the provisioning wizard. You can use the JSON-formatted file as a sample to construct the request body for creating instances using the REST API.

Note that the file contains placeholders for passwords.

After Confirmation

After the Confirmation page closes, the Oracle Java Cloud Service console opens. Optionally, you can click on the service instance name to view status messages. If provisioning of your service instance fails but there are no fatal errors, the software automatically retries provisioning, after a lag time of 60 minutes. Messages about the auto-retry process and failed compute resources are displayed.

If you provided your email address for the **Notification Email** option, you will receive an email notification when the service instance provisioning has succeeded or failed.

Next Steps

- After the service instance has been created, you can view the system messages logged during the creation process, including error messages. Click **Instance Create and Delete History**, then click the service instance name or **Details**.
- If the provisioning process retried provisioning automatically, some failed resources might still exist. To clean up these failed resources, click the **Complete Cleanup** button. If you click the button once and not all failed resources are cleaned up, the **Complete Cleanup** button will remain. If this is the case, click the button again and wait. Repeat this process until the button is not longer displayed and all failed resources are cleaned up.
- If you did not select the **Enable Access to Administration Consoles** option, then in order to use these tools to modify the default configuration or to deploy applications, see Enable Console Access for a Service Instance.
- If you selected the **Enable Authentication with Oracle Identity Cloud Service** option, you can use Oracle Identity Cloud Service to create additional WebLogic Server users. See Use Oracle Identity Cloud Service with Oracle Java Cloud Service.
- If you selected the **Deploy Sample Application** option, and want to test the sample application, see About the Sample Application Deployed to an Oracle Java Cloud Service Instance.
- If you associated an Oracle Real Application Cluster (RAC) database with your service instance, Oracle recommends that you optimize communication between the service instance and the database cluster. See Configure an Oracle Java Cloud Service Instance for an Oracle RAC Database.

Create an Oracle Java Cloud Service Instance with Cloud Stack

Use Oracle Cloud Stack to provision instances of both Oracle Java Cloud Service and Oracle Database Cloud Service as a single operation.

Oracle Cloud Stack is a component of Oracle Cloud that enables you to create multiple cloud resources as a single unit called a stack. You create, delete and manage these resources together as a unit, but you can also access, configure, and manage them through their service-specific interfaces. Stacks also define the dependencies between your stack resources, so that Oracle Cloud Stack creates and destroys the resources in a logical sequence.

Stacks are created from templates. The QuickStart feature of Oracle Java Cloud Service uses stack templates so that you can quickly create service instances based on standard configurations. See Create an Oracle Java Cloud Service Instance by Using a QuickStart Template.

Oracle Cloud Stack also includes a certified Oracle stack template named **`Oracle-JCS-DBCS-Template`**. This template creates a stack that's comprised of these resources:

- A database deployment in Oracle Database Cloud Service
- A service instance in Oracle Java Cloud Service that is connected to the database deployment
- A storage container in Oracle Cloud Infrastructure Object Storage Classic to support cloud backups for the Oracle Java Cloud Service instance and the database deployment

Topics:

- Get Started with Cloud Stack
- Template Parameters
- Create a Stack with the CLI
- Customize the Template

Get Started with Cloud Stack

Learn about documentation, videos, and tutorials to help you get familiar with Oracle Cloud Stack.

Create a stack using the **`Oracle-JCS-DBCS-Template`** template. Refer to these topics in *Using Oracle Cloud Stack Manager*:

- Accessing Oracle Cloud Stack
- Creating a Cloud Stack

A video and a tutorial are also available.

Video

Tutorial

Template Parameters

Certain input parameters can be customized for each stack creation.

The **`Oracle-JCS-DBCS-Template`** template includes these parameters:

- Oracle WebLogic Server and Oracle Database versions
- Oracle WebLogic Server and Oracle Database VM compute shapes (CPU, memory, storage)
- Oracle WebLogic Server user name
- Oracle WebLogic Server and Oracle Database system passwords
- Oracle Database name (SID)
- Oracle Database usable storage in Gigabytes
- SSH public key for all nodes
- Name of the Oracle Cloud Infrastructure Object Storage Classic container to create
- Storage user name and password

The stack name (the predefined parameter `serviceName`) is used to name the new services. This stack name is joined with the text `JCS` and `DBCS`.

Create a Stack with the CLI

In addition to the web console, Oracle Cloud Stack supports the same command line interface (CLI) that you can use to create and manage Oracle Java Cloud Service.

Execute the `stack create` command and specify the template's name, **`Oracle-JCS-DBCS-Template`**. Provide values for the template parameters either as a JSON file or as a command line option. If using the command line option, be sure to properly enclose any values that contain white space or other special characters. For example:

```
psm stack create -n MyStack -t Oracle-JCS-DBCS-Template -p
commonPwd:"password" backupDestination:"BOTH"
backupStorageContainer:"https://acme.storage.oraclecloud.com/v1/MyService-
acme/MyContainer" backupStorageUser:"john@example.com"
backupStoragePassword:"password" publicKeyText:"key_text"
```

To identify the parameter names to use with the CLI, view or export the template. See Viewing a Template in *Using Oracle Cloud Stack Manager*.

Customize the Template

Copy the sample template and add a new parameter, or change the existing parameters used to create the Oracle Java Cloud Service instance.

Use Oracle Cloud Stack to copy and update the **`Oracle-JCS-DBCS-Template`** template in order to customize your stack's behavior. Modify the template's name and contents. Refer to these topics in *Using Oracle Cloud Stack Manager*:

- Copying an Oracle Template

- Creating Resources
- Creating Template Parameters

> **Tip:**
>
> While editing a resource in a stack template, place you mouse over a parameter name to view its description.

See below for some examples of customizing this stack template.

Enable Access to the Administration Console

By default, network access to the WebLogic Server Administration Console in an Oracle Java Cloud Service instance is disabled for security reasons. To enable access to the console after creating a stack, see Enable Console Access for a Service Instance. Alternatively, you can update the template and enable access to the console at the time the service instance is created. Edit the Oracle Java Cloud Service resource and set `enableAdminConsole` to `true`.

Set the WebLogic Server Cluster Size

By default, the Oracle WebLogic Server domain in an Oracle Java Cloud Service instance contains a single Managed Server to host your Java Enterprise applications. This is appropriate for a development environment, but test or production systems may require a larger cluster of Managed Servers. Oracle Java Cloud Service allows users to scale out an existing service instance after creating it, but alternatively you can update the stack template. Edit the Oracle Java Cloud Service resource, expand `components` and `WLS`, and then set `managedServerCount`.

```
components:
  WLS:
    ...
    managedServerCount: 3
```

Create a Separate Application Database

An Oracle Java Cloud Service instance requires at least one Oracle Database Cloud Service deployment in order to host the required Oracle schemas. But a new Oracle Java Cloud Service instance can also connect to a second database deployment (or a second Pluggable Database in the same database deployment) to separate the Oracle schemas from your application schemas. Create a second database resource in your template and associate it with the Oracle Java Cloud Service instance.

1. Add a second Oracle Database Cloud Service resource to your template named `dbcs2`. See Creating Resources in *Using Oracle Cloud Stack Manager*.
2. For the database deployment’s `serviceName` parameter, use the `Join` function to give the resource a unique name. For example:

   ```
   'Fn::Join':
     - ''
     - - 'Fn::GetParam': serviceName
       - DBCSAPP
   ```

3. Edit the Oracle Java Cloud Service resource, expand `components` and `WLS`, and then set `appDBs` to the following value:

```
- dbServiceName:
    'Fn::GetAtt':
      - dbcs2
      - serviceName
  dbaName: sys
  dbaPassword:
    'Fn::GetParam': commonPwd
```

About the Sample Application Deployed to an Oracle Java Cloud Service Instance

When you create an Oracle Java Cloud Service instance, a sample application is deployed automatically to the instance's managed servers and started.

How to Access the Application

The easiest way to access the application, `sample-app.war`, is from the Oracle Java Cloud Service Instance Overview page. Click the link adjacent to **Open Sample Application**.

If you specified a custom **Weblogic Cluster Path Prefix** for the first cluster in a service instance, then this link will not work until you update the service instance. See Configure a Custom URL for the Sample Application.

To access the sample application without using the web console, enter the following URL:

https://*IP_address*:*port*/*cluster_prefix*/sample-app

For example:

```
https://192.0.2.1:443/sample-app
```

IP_address is the public IP address though which you access the application:

- If your service instance has a user-managed load balancer, then use the IP address of the load balancer node.
- If your service instance has an Oracle-managed load balancer, then use the IP address or domain name of the load balancer that was provisioned for your service instance.
- Otherwise, use the IP address of the first node in your service instance.

port is the port number through which you access the application:

- If your service instance has a load balancer or only one Managed Server, use port 443.
- Otherwise, use port 8002.

The default *cluster_prefix* is "/", unless you specified a custom value like `mycluster`.

See:

- Access an Application Deployed to an Oracle Java Cloud Service Instance
- About the Default Access Ports

What the Application Does

When you open the `sample-app` application, the following information is displayed:

- Tweets—You can tweet to @OracleCloudZone and @OracleWebLogic.
- Links— You can access documentation, demos, videos, blogs, FAQs, and related links.

How to Manage the Application

You can verify that the application is deployed and running by viewing the Deployments table in the WebLogic Server Administration Console. From the Deployments table, you can stop, start, and undeploy the application.

3 Manage the Life Cycle of Oracle Java Cloud Service Instances

After you create an Oracle Java Cloud Service instance, you can manage the instance throughout its the life cycle through operations such as shutdown and restart; activity monitoring; suspending requests; and health monitoring.

Topics:

- Typical Workflow for Managing the Life Cycle of Oracle Java Cloud Service Instances
- View All Oracle Java Cloud Service Instances
- Monitor Activity
- Reserve IP Addresses
- View Detailed Information About an Oracle Java Cloud Service Instance
- View the Service Metrics for an Oracle Java Cloud Service Instance
- Suspend an Oracle Java Cloud Service Instance
- Stop, Start, and Restart an Oracle Java Cloud Service Instance and Individual Nodes
- Delete an Oracle Java Cloud Service Instance
- Manage Tags for a Service Instance
- Identify the Cloud Infrastructure Used by a Service Instance
- Explore the Oracle Java Cloud Service Console
- Explore the Oracle Java Cloud Service Welcome Page
- Explore the Oracle Java Cloud Service Instance Overview Page

Typical Workflow for Managing the Life Cycle of Oracle Java Cloud Service Instances

To manage the life cycle of Oracle Java Cloud Service instances, consider the typical workflow described in the following table.

The table provides links to information about how to perform each task by using the web-browser-based Oracle Java Cloud Service Console.

- To use the REST API to manage the life cycle of Oracle Java Cloud Service instances, see Service Instances in *REST API for Oracle Java Cloud Service*.
- To use the Command Line Interface to manage the life cycle of Oracle Java Cloud Service instances, see About the PaaS Service Manager Command Line Interface in *PaaS Service Manager Command Line Interface Reference*.

Task	Description	More Information
Create an Oracle Java Cloud Service instance.	Create a WebLogic Server production environment in the cloud.	About Creating an Oracle Java Cloud Service Instance About Java Cloud Service Instances in Oracle Cloud Infrastructure
View all Oracle Java Cloud Service instances	View status, resource allocation, and other details for all Oracle Java Cloud Service instances.	View All Oracle Java Cloud Service Instances
View detailed information about an Oracle Java Cloud Service instance	View status, resource allocation, and other details for an Oracle Java Cloud Service instance.	View Detailed Information About an Oracle Java Cloud Service Instance
Suspend an Oracle Java Cloud Service instance	Disable the load balancer to block any new traffic to an Oracle Java Cloud Service instance temporarily while maintenance is performed.	Suspend an Oracle Java Cloud Service Instance
Stop, start, or restart a service instance or individual server nodes	Stop service instances or individual server VMs to stop metering for these resources. Restart the Administration Server or individual server nodes if reboot is needed.	Stop, Start, and Restart an Oracle Java Cloud Service Instance and Individual Nodes
Delete an Oracle Java Cloud Service instance	Manage access to an Oracle Java Cloud Service instance by deleting the service instance.	Delete an Oracle Java Cloud Service Instance
Take a snapshot of a service instance	Create a point-in-time image of all the block storage volumes attached to an instance, except the backup volume. The snapshot reflects the state of the volumes at the time when the creation of the snapshot is triggered.	Create a Snapshot
Clone a service instance	Create a new instance based on a snapshot.	Clone an Instance Using a Snapshot

When Oracle Coherence is enabled for a service instance: See also Overview of Coherence Tasks for Oracle Java Cloud Service.

View All Oracle Java Cloud Service Instances

From the Oracle Java Cloud Service Console, you can:

- View the total resources allocated across all Oracle Java Cloud Service instances.
- View the details for each service instance.
- Use the search field to filter the list to include only the service instances that contain the string in their instance name.

To view all Oracle Java Cloud Service instances:

1. Navigate to the Oracle Java Cloud Service Console.
2. Click on **Instances** if this tab is not already selected.

   Your Oracle Java Cloud Service instances are listed on this page.
3. Optional: If you are a member of the primary identity domain in Oracle Identity Cloud Service, you can select a specific **Identity Domain** to view only those

instances in selected identity domain. By default, instances in all identity domains are displayed.

Monitor Activity

You can view all of the cloud operations that have been performed on your Oracle Java Cloud Service instances.

You can restrict the list of activities that are displayed by using search filters. For each activity, you can view the operation, service name, service type, status, start time and end time. You can also view the name of the cloud user that initiated the activity.

1. Access your service console.
2. Click the **Activity** tab.
3. To locate a specific activity, complete these fields in the **Search Activity Log** area, and then click **Search**.

   By default, this page displays all Oracle Java Cloud Service activities that occurred in the previous 24 hours.
4. Optional: Select a value for **Results per page** to limit the maximum number of search results.

Reserve IP Addresses

This topic does not apply to Oracle Cloud Infrastructure. Identify the Cloud Infrastructure Used by a Service Instance.

You can reserve IP addresses for your service by using the IP Reservations tab on the Oracle Java Cloud Service Console.

The **IP Reservations** tab appears in the console if either of these conditions are true:

- Your account already has one or more existing IP reservations.
- You are in the process of creating a new service instance and you have selected a specific **Region** to which to deploy the service instance.

To reserve IP addresses:

1. Navigate to the Oracle Java Cloud Service console.
2. If the **IP Reservations** tab is shown in the console, click this tab and *skip to step 7*.
3. Click **Create Service** and select **Java Cloud Service**.

   The Create Service dialog is displayed.
4. Select a specific **Region** and click **OK**. You cannot reserve IP addresses if you select the **No Preference** option.

   The Service page of the instance creation wizard is displayed.
5. Fill out the fields and click **Next**.

   You are going to reserve IPs, so leave the **IP Network** field at **No Preference**.

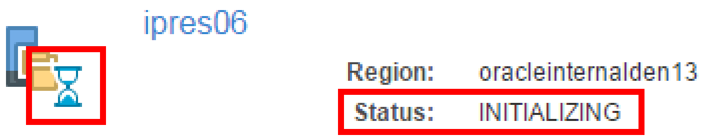
   The Details page of the instance creation wizard is displayed.
6. Click the gear icon beside the **Reserved IPs** field.

You are directed to the IP Reservations page.

7. Create one reserved IP for each node in the service instance you want to create.
   a. On the IP Reservations page, click **Create**.
   b. Enter a **Name** for the reservation.
   c. Select the **Region** in which you want the IP reservation to be created.
   d. If you intend to use this reservation for an instance that you attach to an IP network, select the **On IP Network** check box.

      If you leave this check box deselected, the IP reservation can be assigned to only an instance that you attach to the shared network.
   e. Click **OK**.

      The reservation will be created, a process that takes a few moments. During this time, the reservation icon on the line item will be overlaid by an hourglass and the status will show creation progress:



ipres06

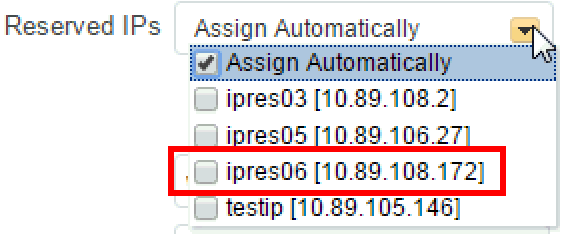
Region: oracleinternalden13

Status: INITIALIZING

   f. Click ⟳ to refresh the page.

      If the creation process has completed, the hourglass will disappear from the IP Reservation icon and the status will change to UNUSED.

Your new IP reservations will be available on the **Reserved IPs** drop-down list in the service instance creation wizard.



Reserved IPs: Assign Automatically

☑ Assign Automatically

□ ipres03 [10.89.108.2]

□ ipres05 [10.89.106.27]

□ ipres06 [10.89.108.172]

□ testip [10.89.105.146]

If an IP reservation is not in use by a service instance, you can also use the IP Reservations page to delete the IP reservation. You cannot delete an IP reservation that is allocated to a service instance.

View Detailed Information About an Oracle Java Cloud Service Instance

From the Oracle Java Cloud Service Instance page, you can:

- View a summary of details for the Oracle Java Cloud Service instance, such as description, subscription mode, and so on.

- View the total resources allocated for the Oracle Java Cloud Service instance.
- View the details and status information for each node.

To view detailed information about an Oracle Java Cloud Service instance:

1. Access your service console.
2. Click on the service instance for which you want to view more information.

   The Oracle Java Cloud Service Instance page is displayed with the Overview tile is in focus, displaying detailed information about the service instance.

To redisplay the information at any time, click the **Overview** tile on the Oracle Java Cloud Service Instance page.

Change the License Type for an Oracle Java Cloud Service Instance

If your account has both Bring Your Own Licesne (BYOL) and Oracle Java Cloud Service entitlements, you can change the license type of an existing service instance.

When you create an Oracle Java Cloud Service, you can choose to leverage your own on-premises middleware license (Bring Your Own License, or BYOL) or use a cloud license you've already subscribed to. You can change the license type from **BYOL** to **Cloud License** or vice versa after the instance is created.

1. Access your service console.
2. Click the name of the service instance for which you want to change the license type.
3. Locate the **License** field on the Instance Overview page and click **Change**.
4. On the Change License Type dialog box, select one of the following options:
   - Subscribe to a new Oracle Java Cloud Service software license and the Oracle Java Cloud Service.
   - My organization already owns Oracle middleware software licenses. Bring my existing middleware software license to the Oracle Java Cloud Service.
5. Click **Change**.

View the Service Metrics for an Oracle Java Cloud Service Instance

You can use run a health check to obtain service metrics such as status and memory usage for your Oracle Java Cloud Service instance.

To run a health check:

1. Access the service console.
2. Click the name of your service instance.
3. On the Instance Overview page, select **Display monitoring information** in the icon bar.
4. Click **Healthcheck** next to the instance name for each node.

A **Healthcheck Details** popup window is displayed, showing:

- How long the node has been up
- Maximum memory
- Minimum memory

5. View the **VM CPU Usage** and **VM Free Memory** information displayed for each node.
6. Roll your mouse over the server icon to display the node status, status message, and cluster.

Suspend an Oracle Java Cloud Service Instance

You can disable the load balancer to suspend the Oracle Java Cloud Service instance temporarily, to block any new traffic from being delivered to the service instance. This is useful when you want to perform routine maintenance on an Oracle Java Cloud Service instance, but do not want to stop the service instance. Once the maintenance activities have been completed, you can re-enable the load balancer to allow traffic to be delivered.

> **Note:**
>
> If a load balancer is not configured, you cannot suspend the Oracle Java Cloud Service instance.

Stop, Start, and Restart an Oracle Java Cloud Service Instance and Individual Nodes

You can stop and start an Oracle Java Cloud Service instance and, when the service instance is running, start, stop, and restart individual server or load balancer nodes.

Topics

- About Stopping, Starting, and Restarting an Oracle Java Cloud Service Instance and Individual Nodes
- Stop or Start an Instance
- Stop or Start a Node

About Stopping, Starting, and Restarting an Oracle Java Cloud Service Instance and Individual Nodes

You can stop and start an Oracle Java Cloud Service instance and, when the service instance is running, stop, start, and restart individual server or load balancer nodes.

Note:

The stop and restart procedures affect entire nodes. If you want to shut down the WebLogic Administration Server or Managed Server processes running on the nodes, without stopping the nodes, see Shut Down and Start Server Processes. You might want to do this if you have other processes besides the servers running on the nodes and you do not want to shut down these other processes.

Why Stop an Oracle Java Cloud Service Instance

Stopping an Oracle Java Cloud Service instance frees up compute resources used by the service instance’s nodes. Metering for those resources stops.

Storage volumes remain intact when the service instance is stopped, and are reattached when your start the service instance. IP address reservations are retained when the service instance is stopped, so the nodes will have the same public IP addresses as before when you start the service instance. Even the public IP addresses that are reserved by the service and assigned automatically to the nodes (during instance creation or when scaling out the instance) are retained.

Why Stop, Start, or Restart an Administration Server, Managed Server, or Load Balancer node

If an Oracle Java Cloud Service instance is running:

- You can restart the nodes on which the Administration Server, Managed Server, or load balancer are running if you are experiencing problems with the server that would warrant a reboot. The restart operation is the same as stopping the server or load balancer node, then starting it immediately.
- You can stop the nodes on which the Managed Server or the load balancer are running to free up resources and stop metering those resources. You might also want to stop the service instance instead of scaling, keeping the server or load balancer ready for a later time. If you stop all Managed Servers nodes except for one, you might want to stop the load balancer node because it is not needed.
- You can start a Managed Server or load balancer node if it is stopped and you want to use it again. Metering begins again.

Note:

You can restart the Administration Server, and stop, start, and restart individual Managed Servers and the load balancer only if you specified Oracle Oracle WebLogic Server 12*c* (12.2.1) when you provisioned the service instance. This feature is not supported if you specified Oracle WebLogic Server 11g.

What Happens When an Oracle Java Cloud Service Instance is Stopped or Started

Stopping and starting an Oracle Java Cloud Service instance has the following results:

- **Stopping the service instance**: The nodes on which the Administration Server, Managed Servers, load balancer, and Coherence Data Tier are running are stopped. You cannot start, stop, or restart the Administration Server, Manager Server, or load balancer nodes individually while the service instance is stopped.

- **Starting the service instance**: All nodes on which the Administration Server, Managed Server, load balancer, and Coherence Data Tier are running are started. You can restart the Administration Server, and stop, start, or restart the Managed Servers and load balancer nodes individually. You cannot do the same for Coherence Data Tier nodes individually.

What Happens to IP addresses when an Instance or Node is Stopped and Started

Instances and nodes are assigned both a public and private IP address when the service instance is created. When an instance or node is stopped and started, IP addresses are released or retained depending on whether the service instance is based on Oracle Cloud Infrastructure or Oracle Cloud Infrastructure Classic.

- Oracle Cloud Infrastructure

  For both public and private IP addresses, the IP address persists when an instance or node stops. When the instance or node starts, the same public and private IP address is assigned.

- Oracle Cloud Infrastructure Classic

  If you create an instance or add a node to an instance on Oracle Cloud Infrastructure Classic, you can reserve public IP addresses instead of having Oracle Java Cloud Service assign them for you.

  The public IP address persists when the instance or node stops. When the instance or node starts, and the same public IP address is assigned.

  The private IP address is released when the instance or node stops. When the instance or node starts, the same address may or may not be assigned to the instance or node.

  If a different private IP address has been assigned to the instance or node when it starts, and you have set up access rules on the private IP address, the access rules no longer apply.

  For a service instance that is attached to an IP network, you can set a static private IP address or remove the static IP configuration when you restart a node or start a node that was stopped. This feature ensures that a node continues to use the same private IP address after it's restarted. This feature is supported by the REST API only. See Stop and Start a Service Instance and Individual VMs in *REST API for Oracle Java Cloud Service*.

What Happens to the Coherence Data Grid When a Service Instance is Stopped or Started

All nodes in a Coherence data grid cluster, including the data grid servers, are stopped when an Oracle Java Cloud Service is stopped, and started if an Oracle Java Cloud Service instance is started.

> **Note:**
>
> When the service instance is stopped, all data in the Coherence cache is lost.

Stopping, starting, and restarting Coherence data grid Managed Server nodes is not supported. The only way you can stop or start the data tier is to stop or start the Oracle Java Cloud Service instance.

How Do I Monitor the Stop, Start, or Restart Operation

You can monitor progress of a stop, start, or restart operation on the Activity page. See Monitor Activity.

You can also monitor the boot progress of individual nodes by using Oracle Cloud Infrastructure Compute Classic. See Viewing the Boot Log of an Instance in *Using Oracle Cloud Infrastructure Compute Classic*. Ignore information in this topic about the Compute API.

> **Note:**
>
> When you restart an Oracle Java Cloud Service instance, WebLogic Server may not restart.
>
> To check the restart status, on the Activity page, for the **Operation Status**, click the **Expand** button and view the details. The operation status shows *Succeeded*, but in the details section, a warning message is displayed that indicates the admin server startup process has *failed*.
>
> This occurs because the JCS restart operation considers the output of the *VM restart status* and not the restart status of the WebLogic Server.

What Happens When a Service Instance Is Stuck in Maintenance Mode While Stopping

When you try to stop an Oracle Java Cloud Service instance, on rare occasions it might become stuck in maintenance mode due to some problem with the service instance.

For six hours, the software will continue to attempt to stop the service instance, then change the instance status from maintenance state to error state. At this point, you can debug the problem causing the error and attempt to stop the service instance again.

Stop or Start an Instance

You can stop, start, or restart an Oracle Java Cloud Service instance (and all of its nodes) with a single operation.

When you stop a service instance, the following changes occur:

- Users and other systems cannot access the service instance.
- You cannot perform any other maintenance operations on the service instance, except to start it or to delete it.
- Scheduled backups of the service instance do not occur.
- Metering of Oracle Compute Unit (OCPU) and memory resources for the service instance stops.
- Other resources and services associated with the service instance, including block storage, object storage, and IP reservations, continue to be metered.

Wait for any maintenance operations on this service instance to complete, such as backup, restoration or scaling operations, before you stop or restart a service instance.

1. Access your service console.
2. Click the name of the service instance that you want to stop or start.
3. On the Overview page, click **Stop Instance** ■, **Start Instance** ▶, or **Restart Instance** ⟳.
4. When prompted for confirmation, click **OK**.
5. Periodically click **Refresh** ↻ until the operation is completed.

   You can also monitor the progress of the operation from the **Activity** page.

Stop or Start a Node

As part of configuring or troubleshooting an Oracle Java Cloud Service instance, you can stop or start individual nodes.

You cannot stop certain Java Cloud Service nodes. For these nodes, you are limited to the following actions:

- Restart the node, to stop the node and then immediately start it again.
- Stop the service instance.

Oracle does not recommend that you perform any other management operations on the service instance during a start or stop operation.

(Not available on Oracle Cloud at Customer) For a service instance that is attached to an IP network, you can use the REST API to set a static private IP address or remove the static IP configuration when you restart a node or start a node that was stopped.

1. Access your service console.
2. Click the name of the service instance that contains the node that you want to start or stop.
3. Under **Resources**, beside the node you want to start or stop, click **Manage this node** ☰, and then select **Stop**, **Start**, or **Restart**.
4. When prompted for confirmation, click **OK**.
5. Periodically click **Refresh** ↻ until the operation is completed.

   You can also monitor the progress of the operation from the **Activity** page.

Delete an Oracle Java Cloud Service Instance

When you no longer require an Oracle Java Cloud Service instance, you can delete it. Your account is no longer charged for the instance.

Only a Java administrator can delete a service instance.

> **Note:**
>
> If you created a service instance by using a QuickStart template, you cannot delete the service instance from the Oracle Java Cloud Service console. Using a QuickStart template creates an entire stack for you , so you must delete the entire stack from the Stack console.

The service can be running or stopped before you attempt to delete it. If the service instance is stopped, you must check Force Delete on the Delete Service dialog for proper schema cleanup.

When you delete an Oracle Java Cloud Service instance:

- Resources such as IP addresses are removed.
- Storage volumes attached to the VMs hosting the Oracle Java Cloud Service instance are removed.
- Oracle Autonomous Database (Oracle Autonomous Transaction Processing), Oracle Cloud Infrastructure database, or Oracle Database Cloud Service (Classic) database deployment is not deleted when you delete the Oracle Java Cloud Service instance, only the database repository and schemas are deleted.

  If you created your instance with one Oracle Database Cloud Service (Classic) database for Oracle required schemas and a second for application schemas, neither database deployment is deleted. Your account continues to be charged for the database instances. You might want to retain these database deployments for use with other service instances; otherwise, you must delete the Oracle Database Cloud Service (Classic) databases manually to avoid being charged for them.
- The object storage container is not deleted when you delete the Oracle Java Cloud Service instance. However, service backups are deleted from the container if the Administration Server VM is in a running state when the instance is deleted. Oracle Database Cloud Service backups are not deleted.

  After the service instance is deleted, your account continues to be charged for the object storage space used. You might want to retain the data in object storage for use with other service instances; otherwise, you must delete the storage container manually to avoid being charged for it..

To delete an Oracle Java Cloud Service instance:

> **Note:**
>
> This option is also available from the ☰ menu on the Oracle Java Cloud Service Instance page.

1. Navigate to the Oracle Java Cloud Service Console.
2. From the ☰ menu for the service instance, select **Delete**.

   The Delete Service dialog is displayed.
3. In the Delete Service dialog box that opens, set the following options and click **Delete**:

- Database Administrator User Name—Enter the name of the database administrator user that was specified when the Oracle Autonomous Database, Oracle Cloud Infrastructure database, or Oracle Database Cloud Service database deployment was created. This user owns the Oracle Required Schema in the database. If your service instance is using multiple database deployments, specify the name of the administrator for the database deployment that hosts the Oracle Required Schema.
- Database Administrator User Password—Enter the Database Administrator user password.
- Force Delete—(Optional) Select this checkbox if you want the Oracle Java Cloud Service instance to be deleted even if the database cannot be reached to delete the database schemas. If enabled, you may need to delete the associated database schemas manually on the database if they are not deleted as part of the service instance delete operation.

Once deleted, the Oracle Java Cloud Service is removed from the list of service instances displayed on the Oracle Java Cloud Service Console.

If there is a problem deleting the service instance, the **Retry Delete** displays. Click the **Retry Delete** button to attempt to clean up any remaining resources and delete the service instance completely. The **Retry Delete** button is displayed for as long as the failed resources exist. Repeat this process, as necessary, until the **Retry Delete** button is no longer displayed.

If the deletion process times out before all cleanup is complete, billing for the service instance stops. Oracle Java Cloud Service periodically retries cleanup until the service instance is successfully deleted. You can try deletion cleanup manually:

1. Click on **Instance create and delete history** on the Oracle Java Cloud Service Console.
2. Select the service instance. The status of the service instance will be Deletion Failed.
3. Click **Retry Delete** to initiate cleanup again.

Manage Tags for a Service Instance

A tag is a key or a key-value pair that you can assign to your Oracle Java Cloud Service instances. You can use tags to organize and categorize your instances, and to search for them.

Topics:

- Create, Assign, and Unassign Tags
- Find Tags and Instances Using Search Expressions

Create, Assign, and Unassign Tags

You can create and assign tags to Oracle Java Cloud Service instances while creating the instances or later. When you no longer need certain tags for an instance, you can unassign them.

To assign tags to an instance or to unassign tags:

1. Navigate to the Overview page for the instance for which you want to assign or unassign tags.
2. Click **Manage this instance** ☰ in the instance name bar at the top.
3. Select **Manage Tags** or **Add Tags**.

   If any tags are already assigned, then the menu shows **Manage Tags**; otherwise, it shows **Add Tags**.
4. In the Manage Tags dialog box, create and assign the required tags, or unassign tags:
   - In the **Assign** section, in the **Tags** field, select the tags that you want to assign to the instance.
   - If the tags that you want to assign don't exist, then select **Create and Assign** in the **Tags** field, and click just above the field. Enter the required new tags in the **Enter New Tags** field.
   - To unassign a tag, in the **Unassign** section, look for the tag that you want to unassign, and click the **X** button next to the tag.

     > **Note:**
     >
     > You might see one or more tags with the key starting with `ora_`. Such tags are auto-assigned and used internally. You can’t assign or unassign them.
   - To exit without changing any tag assignments for the instance, click **Cancel**.
5. After assigning and unassigning tags, click **OK** for the tag assignments to take effect.

List Tags

You can get the details of all the tags available in the account as well as the tags assigned to a service instance by using the REST API.

For each tag, the API request returns the key, the value if it exists, and a list of the instances that are assigned the tag. See Tags and Assignments in *REST API for Oracle Java Cloud Service*.

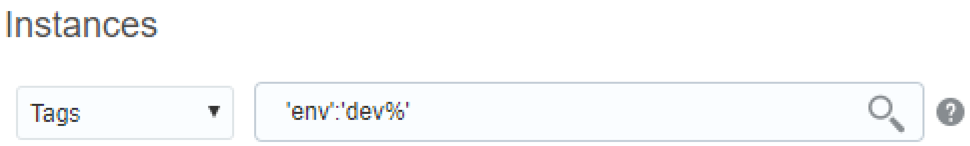
Find Tags and Instances Using Search Expressions

A tag is an arbitrary key or a key-value pair that you can create and assign to your Oracle Java Cloud Service instances. You can use tags to organize and categorize your instances, and to search for them. Over time, you might create dozens of tags, and you might assign one or more tags to your instances. To search for specific tags and to find instances that are assigned specific tags, you can use filtering expressions.

Search for Instances with Tags

From the Instances page of the web console, select **Tags**, and then enter a *search expression* in the **Search** field.

For example, you can search for the instances that are assigned a tag with the key `env` and any value starting with `dev` (example: `env:dev1`, `env:dev2`), by entering the search expression **`'env':'dev%'`**.



Instances

Tags ▼

'env':'dev%'

Similarly, when you use the REST API to find tags or to find instances that are assigned specific tags, you can filter the results by appending the optional `tagFilter=`*`expression`* query parameter to the REST endpoint URL.

- To find specific tags: `GET paas/api/v1.1/tags/{identity_domain}/tags?`**`tagFilter={expression}`**
- To get a list of instances that are assigned specific tags: `GET paas/api/v1.1/instancemgmt/{identity_domain}/instances?`**`tagFilter={expression}`**

Syntax and Rules for Building Tag-Search Expressions

- When using cURL to send tag-search API requests, enclose the URL in double quotation marks.

  Example:

  ```
  curl -s -u username:password -H "X-ID-TENANT-NAME:acme"
  "restEndpointURL/paas/api/v1.1/instancemgmt/acme/instances?
  tagFilter='env'"
  ```

  This request returns all the tags that have the key `env`.

- Enclose each key and each value in single quotation marks. And use a colon (`:`) to indicate a key:value pair.

  Examples:

  ```
  'env'
  'env':'dev'
  ```

- You can include keys or key:value pairs in a tag-filtering expression.

Sample Expression	Description	Sample Search Result

`'env'`	Finds the tags with the key `env`, or the instances that are assigned the tags with that key.	The following tags, or the instances that are assigned any of these tags:
		`env:dev`
		`env:qa`

`'env':'dev'`	Finds the tag with the key `env` and the value `dev`, or the instances that are assigned that tag.	The following tag, or the instances that are assigned this tag `env:dev`

- You can build a tag-search expression by using actual keys and key values, or by using the following wildcard characters.

  `%` (percent sign): Matches any number of characters.

  `_` (underscore): Matches one character.

Sample Expression	Description	Sample Search Result

`'env':'dev%'`	Finds the tags with the key `env` and a value starting with `dev`, or the instances that are assigned such tags.
	Note: When you use `curl` or any command-line tool to send tag-search REST API requests, encode the percent sign as `%25`.
		The following tags, or the instances that are assigned any of these tags:
		`env:dev`
		`env:dev1`

`'env':'dev_'`	Finds the tags with the key `env` and the value `devX` where X can be any one character, or finds the instances that are assigned such tags.	The following tags, or the instances that are assigned any of these tags:
		`env:dev1`
		`env:dev2`

- To use a single quotation mark (`'`), the percent sign (`%`), or the underscore (`_`) as a literal character in a search expression, escape the character by prefixing a backslash (`\`).

Sample Expression	Description	Sample Search Result

`'env':'dev\_%'`	Finds the tags with the key `env` and a value starting with `dev_`, or the instances that are assigned such tags.	The following tags, or the instances that are assigned any of these tags:
		`env:dev_1`
		`env:dev_admin`

- You can use the Boolean operators AND, OR, and NOT in your search expressions:

Sample Expression	Description	Sample Search Result

`'env' OR 'owner'`	Finds the tags with the key `env` or the key `owner`, or the instances that are assigned either of those keys.	The following tags, or the instances that are assigned *any of these tags*:
		`env:dev`
		`owner:admin`

Sample Expression	Description	Sample Search Result

`'env' AND 'owner'`	Finds the instances that are assigned the tags `env` *and* `owner`.
	Note: This expression won’t return any results when used to search for tags, because a tag can have only one key.
		The instances that are assigned *all of the following tags*:
		`env:dev`
		`owner:admin`

`NOT 'env’`	Finds the tags that have a key other than `env`, or the instances that are assigned such tags.
	Note: Untagged instances as well will satisfy this search expression.
		The following tags, or the instances that are assigned *any of these tags* or no tags:
		`owner:admin`
		`department`

`('env' OR 'owner') AND NOT 'department'`	Finds the tags that have the key `env` or the key `owner` but not the key `department`, or the instances that are assigned such tags.	The following tags, or the instances that are assigned *any of these tags*:
		`env:dev`
		`owner:admin`

Delete Tags

You can delete tags by using the REST API.

See Tags and Assignments in *REST API for Oracle Java Cloud Service*.

Identify the Cloud Infrastructure Used by a Service Instance

When you create an Oracle Java Cloud Service instance, you specify the region in which the instance is created. That choice determines the infrastructure that the instance will use: Oracle Cloud Infrastructure or Oracle Cloud Infrastructure Classic.

1. Access the service console.
2. Click the name of the service instance.
3. Click **Instance Details** ⋮, and then locate the **Region** field.
   - If the value is **us-phoenix-1**, **us-ashburn-1**, **ca-toronto-1**, **eu-frankfurt-1**, or **uk-london-1**, then the instance uses Oracle Cloud Infrastructure.
   - If the **Region** field isn’t present, or if the value is not one of the regions listed above, then the instance uses Oracle Cloud Infrastructure Classic.

Explore the Oracle Java Cloud Service Console

You can use the Oracle Java Cloud Service console to view all Oracle Java Cloud Service instances.

Topics:

- What You Can Do from the Oracle Java Cloud Service Console
- What You See on the Oracle Java Cloud Service Console

What You Can Do from the Oracle Java Cloud Service Console

Use the Oracle Java Cloud Service console to perform the tasks described in the following topics:

- About Creating an Oracle Java Cloud Service Instance
- View All Oracle Java Cloud Service Instances
- View Detailed Information About an Oracle Java Cloud Service Instance
- View the Service Metrics for an Oracle Java Cloud Service Instance
- Monitor Activity
- Add an SSH Public Key
- Delete an Oracle Java Cloud Service Instance
- Access the Administration Consoles for Oracle Java Cloud Service
- Configure a Load Balancer for a Service Instance
- Stop or Start an Instance

What You See on the Oracle Java Cloud Service Console

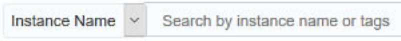
When you access Oracle Java Cloud Service the first time for an account, you will see the Welcome page. Click **Instances** to view the Oracle Java Cloud Service console home page.

There are additional tabs on the Oracle Java Cloud Service console: **Activity** and **SSH Access**. See:

- Monitor Activity
- Add an SSH Public Key

The following table describes the key information shown on the Oracle Java Cloud Service console. The information displayed in the Oracle Java Cloud Service console will vary based on whether or not you have created Oracle Java Cloud Service instances. When you access the Oracle Java Cloud Service console for your account for the first time and there are no Oracle Java Cloud Service instances created, any service instance details will not be displayed. In this case, you can create a service instance by clicking **Create Instance** and access information about the prerequisites and steps for creating an instance.

Element	Description
Identity Domain	View service instances in the selected identity domain, or choose **Multiple** to view service instances in all identity domains.

Element	Description
WE	Click the user menu icon containing the initials of the user in order to access a menu with the following options: • **Help**—Provides links to documentation, videos, tutorials, and troubleshooting information. You can also choose to download the PaaS Service Manager (PSM) Command Line Interface (CLI). • **Accessibility**—Specify whether you use a screen reader, high contrast, and/or large fonts. • **About**—Provides a description of what you can do with Oracle Java Cloud Service, and the version of the service and UI you are using. • **Sign Out**—Logs you out of the service.
?	Access help for this page, including documentation, tutorials, videos, and FAQs. Click the **Contact Use** button to: • Look up Oracle contact phone numbers • Access My Oracle Support • Access Oracle Cloud Discussion Forums • Chat with Oracle Support online
☰ (In the branding bar)	Click and select a choice from the menu to open the service console for one of the Oracle Cloud Services to which you subscribe.
Instances	Click to refresh this page.
Activity	Click to view all operations performed on your service instances. See Monitor Activity.
SSH Access	Click to manage SSH keys for your service instances. See Add an SSH Public Key.
Reserved IPs	Click to reserve IP addresses for your service instance. See Reserve IP Addresses.
Welcome!	Click to return to the Welcome page.
☰ (Adjacent to the **Welcome!** link in the banner)	Click and select a choice from the drop-down menu to open the service console for one of the Oracle Cloud Platform Services to which you subscribe.
Services (Summary panel)	Number of Oracle Java Cloud Service instances in the identity domain.
OCPUs (Summary panel)	Total number of Oracle Compute Units (OCPUs) allocated across all Oracle Java Cloud Service instances.
Memory (Summary panel)	Total amount of memory in GBs allocated across all Oracle Java Cloud Service instances.
Storage (Summary panel)	Total amount of block storage in GBs allocated across all Oracle Java Cloud Service instances.
Public IPs (Summary panel)	Total number of public IP addresses allocated across all Oracle Java Cloud Service instances.
Instances (heading)	All Oracle Java Cloud Service instances in the identity domain.
 Instance Name ⌄ Search by instance name or tags Search field	Select **Instance Name** and enter a full or partial service instance name to filter the list of service instances to include only the instances containing that string in their name. Select **Tags** from the drop-down menu and enter a search expression to filter service instances tagged with the tags you specify. See Find Tags and Instances Using Search Expressions.

Element	Description
⟳	Click to refresh the page. The date and time the page was last refreshed is displayed adjacent to this button.
Create Instance	Select one of the following options: - **Java**—Create a new Oracle Java Cloud Service instance. See About Creating an Oracle Java Cloud Service Instance. - **Java-AppToCloud**—Use the AppToCloud feature to migrate an application. See Typical Workflow for Migrating Applications to Oracle Java Cloud Service with AppToCloud.
☁	Oracle Java Cloud Service instance. Click this icon to view more details.
⌛	Status icon indicating that the Oracle Java Cloud Service instance is being created.
⚠	Status icon indicating the Oracle Java Cloud Service instance is undergoing maintenance or terminating.
❗	Status icon indicating that the Oracle Java Cloud Service instance wasn't created. This icon can also mean that the service instance has stopped. See the Activity section of this page.
service-name	Name of the Oracle Java Cloud Service instance. Click the name to view more details.
Status	Status of the service instance. Valid values include: In Progress, Maintenance, Terminating, Stopped, and Failed. Click the status label to view progress messages. **Note:** Running service instances do not display this field.
Version	Version of Oracle WebLogic Server configured for the Oracle Java Cloud Service instance.
Edition	Software edition. Valid values include: Standard, Enterprise, or Suite.
Tags	Tags assigned to the service instance. The first tag is displayed. To see all tags assigned to the service instance, hover over the tag name and click **More**.
Nodes	Number of nodes allocated for the Oracle Java Cloud Service instance. If your service instance has multiple clusters, the Nodes value is the sum of all the nodes in all the clusters. When Oracle Coherence is enabled for a service instance: This number includes the application tier nodes (storage-disabled) and Coherence data tier nodes (storage-enabled).
Coherence	Flag indicating that Oracle Coherence is configured for the Oracle Java Cloud Service instance. If not configured, this field does not appear.
Submitted On	When status is In Progress, date and time in UTC that the Oracle Java Cloud Service instance creation request was submitted.
Created On	When provisioning is complete, date and time in UTC that the Oracle Java Cloud Service instance was created.

Element	Description
OCPUs	Number of OCPUs allocated for the Oracle Java Cloud Service instance. If your service instance has multiple clusters, the OCPUs value is the sum of all OCPUs used in all the clusters.
Memory	Amount of memory in GBs allocated for the Oracle Java Cloud Service instance.
Storage	Amount of storage in GBs allocated for the Oracle Java Cloud Service instance.
C	Icon indicating that the service instance is a clone.
☰ (adjacent to the service instance name)	Instance menu icon provides the following options: • **Open WebLogic Server Console**—Open the WebLogic Console to administer your application environment. See Access the Administration Consoles for Oracle Java Cloud Service. • **Open Fusion Middleware Control Console**—Open Fusion Middleware Control to administer your application environment. See Access the Administration Consoles for Oracle Java Cloud Service. • **Open Load Balancer Console**—Open the console to administer the load balancer, if the load balancer has been configured for the service instance. See Access the Administration Consoles for Oracle Java Cloud Service and Configure a Load Balancer for a Service Instance. • **Start**—Starts the service instance. • **Stop**—Stops the service instance. • **View Service Metrics**—Displays a graph of heap usage for all servers, overall Managed Servers, Overall Administration Server, and individual Managed Servers. If a load balancer is present, you can select to see response time. See View the Service Metrics for an Oracle Java Cloud Service Instance. • **Manage Access Rules**—Opens the Access Rules page, which enables you to create and manage access rules for selected sources and destinations. See Create an Access Rule. • **Add SSH Access**—Add public SSH keys to the VMs that make up this service instance. See Add an SSH Public Key • **Define Auto Scaling Rules**—Opens a page that enables you to create scaling rules. • **Delete**—Deletes the service instance. See Delete an Oracle Java Cloud Service Instance. The administration console choices will only appear if you have selected to enable administration console access when you created the service instance.

Element	Description
Instance Create and Delete History	Shows details about created or deleted service instances. • **Show only failed attempts**—Check this box if you want to see failed attempts only. • **Details**—Displays system messages logged during the creation or deletion process. Messages include information about auto-retry attempts. • **Complete Cleanup**— This button appears only if there are failed resources created during a successful auto-retry process. If you select this button, the failed resources are deleted. You might have to press the button again and wait, repeating this process until the button is no longer displayed. • **Retry Delete**—This button appears only if an attempt to delete a failed service instances is unsuccessful. The software cleans up failed resources and tries again to delete the service instance. You might have to press the button again and wait, repeating this process until the button is no longer displayed.

Explore the Oracle Java Cloud Service Welcome Page

You can use the Oracle Java Cloud Service Welcome page to get started using Oracle Java Cloud Service.

Topics:

- What You Can Do from the Oracle Java Cloud Service Welcome Page
- What You See on the Oracle Java Cloud Service Welcome Page

What You Can Do from the Oracle Java Cloud Service Welcome Page

Use the Oracle Java Cloud Service Welcome page to perform the following tasks:

- Get started by stepping through the Getting Started Using Oracle Java Cloud Service tutorial.
- Discover Oracle Java Cloud Service by watching video demonstrations of key tasks.
- Learn what's new and noteworthy in the current release of Oracle Java Cloud Service.
- Learn about Oracle Java Cloud Service by selecting your role, and view documentation specifically chosen for your role.
- Navigate to the Oracle Java Cloud Service Console.

What You See on the Oracle Java Cloud Service Welcome Page

The following table describes the key information shown on the Oracle Java Cloud Service Welcome page.

Element	Description
Identity Domain	View service instances in the selected identity domain, or choose **Multiple** to view service instances in all identity domains.

Element	Description
WE	Click the user menu icon containing the initials of the user in order to access a menu with the following options: • **Help**—Provides links to documentation, videos, tutorials, and troubleshooting information. You can also choose to download the PaaS Service Manager (PSM) Command Line Interface (CLI). • **Accessibility**—Specify whether you use a screen reader, high contrast, and/or large fonts. • **About**—Provides a description of what you can do with Oracle Java Cloud Service, and the version of the service and UI you are using. • **Sign Out**—Logs you out of the service.
?	Access help for this page, including documentation, tutorials, videos, and FAQs. Click the **Contact Use** button to: • Look up Oracle contact phone numbers • Access My Oracle Support • Access Oracle Cloud Discussion Forums • Chat with Oracle Support online
Instances	Click to navigate to the Oracle Java Cloud Service Console. See Explore the Oracle Java Cloud Service Console.
Activity	Click to view all operations performed on your service instances. See Monitor Activity.
SSH Access	Click to manage SSH keys for your service instances. See Add an SSH Public Key.
IP Reservations	Click to manage the IP reservations for your service instances. See Reserve IP Addresses.
Watch Video	Click to see a video about how to get started with Oracle Java Cloud Service.
Follow Tutorial	Click to see a video about how to get started with Oracle Java Cloud Service.
Go to Console	Click to navigate to the Oracle Java Cloud Service Console. See Explore the Oracle Java Cloud Service Console.
Welcome!	Click to refresh this page.
See what's new and noteworthy in this release	Click to read What's New in Oracle Java Cloud Service.
Discover	Watch videos that demonstrate how to perform key tasks.
Learn	Click your role to view documentation specifically chosen for your role.

Explore the Oracle Java Cloud Service Instance Overview Page

You can use the Overview tile on the Oracle Java Cloud Service Instance page to view overview information for an Oracle Java Cloud Service instance.

Topics:

- What You Can Do from the Oracle Java Cloud Service Instance Overview Page
- What You See on the Oracle Java Cloud Service Instance Overview Page

What You Can Do from the Oracle Java Cloud Service Instance Overview Page

Use Oracle Java Cloud Service Instance Overview page to perform the tasks described in the following topics:

- View Detailed Information About an Oracle Java Cloud Service Instance
- View the Service Metrics for an Oracle Java Cloud Service Instance

  Note that you can click on the service instance name on the View Metrics screen to return to the Oracle Java Cloud Service Instance Overview page.
- Access the Administration Consoles for Oracle Java Cloud Service
- Stop, Start, and Restart an Oracle Java Cloud Service Instance and Individual Nodes
- Disable or Enable the Load Balancer for an Oracle Java Cloud Service Instance
- Click the Administration tile to backup, restore, and patch an Oracle Java Cloud Service instance. See:
  - Back Up and Restore an Oracle Java Cloud Service Instance
  - Patch an Oracle Java Cloud Service Instance
- Scale up/down, out, and automatically. See:
  - Scale Out a Cluster
  - Scale In a Cluster
  - Scale Out a Coherence Data Grid
  - Scale In a Coherence Data Grid
  - Scale a Node
  - Scale Automatically

What You See on the Oracle Java Cloud Service Instance Overview Page

The following table describes the key information shown on the Oracle Java Cloud Service console.

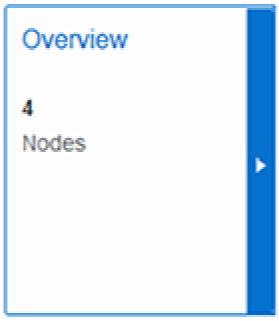
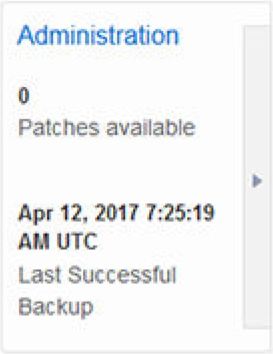
Element	Description
Oracle Java Cloud Service link	Click this link to return to the Oracle Java Cloud Service Console.
Identity Domain	View service instances in the selected identity domain, or choose **Multiple** to view service instances in all identity domains.
?	Access help for this page, including documentation, tutorials, videos, and FAQs. Click the **Contact Use** button to: • Look up Oracle contact phone numbers • Access My Oracle Support • Access Oracle Cloud Discussion Forums • Chat with Oracle Support online

Element	Description
WE	Click the user menu icon containing the initials of the user in order to access a menu with the following options: • **Help**—Provides links to documentation, videos, tutorials, and troubleshooting information. You can also choose to download the PaaS Service Manager (PSM) Command Line Interface (CLI). • **Accessibility**—Specify whether you use a screen reader, high contrast, and/or large fonts. • **About**—Provides a description of what you can do with Oracle Java Cloud Service, and the version of the service and UI you are using. • **Sign Out**—Logs you out of the service.
⋮	Displays information about the service instance: • **Region**—Where the service instance is located. • **Created By**—User who created the service instance. • **Created On**—Date on which the service instance was created. • **License**—Cloud License or BYOL. • **Identity Domain**—Identity domain in which the instance was created. • **IP Network**—The IP network assigned to the service instance. • **Metering Frequency**—Hourly or Monthly, depending on what you selected on the Instance page of the provisioning wizard. • **Subscription Id**—The ID for the entitlement that enabled you to create the service instance.

Element	Description
☰ (in the page header)	**Menu** icon provides the following options: • **Open WebLogic Server Administration Console**—Open the WebLogic Administration Console to administer your application environment. • **Open Fusion Middleware Control Console**—Open Fusion Middleware Control to administer your application environment. • **Open Load Balancer Console**—Open the console to administer the load balancer, if a local load balancer has been configured for the service instance. Note that access to the administrative consoles is disabled by default. When you create a service instance, you can enable consoles by selecting a check box on the Details page of the instance creation wizard. For an instance this is already created, you must create an access rule in order to activate the console choices. See Enabling Console Access in an Oracle Java Cloud Service. • **Start**—Start the nodes for the Administration Server, Managed Servers, load balancer, and Managed Servers on the Coherence data tier (if provisioned). • **Stop**—Stop the nodes for the Administration Server, Managed Servers, load balancer, and Managed Servers on the Coherence data tier (if provisioned). • **Restart**—Stop and then immediately restart all the nodes in the service instance. • **Scale Out**—Adds a managed server node. • **Define Auto Scaling Rules**—Opens the Add Rule dialog box, which opens the Rules page where you can configure auto-scaling rules. • **Change License Type**—Opens the Change License Type dialog box, which enables you to choose whether to leverage your existing on-premises (BYOL) license or use your Oracle Java Cloud Service cloud license. • **Add Load Balancer**—Add a user-managed load balancer to this service instance. • **Disable/Enable Load Balancer**—Depending on the selection, either blocks access to the service instance or forwards the requests it receives from clients to the Oracle WebLogic Server Managed Servers. • **Manage Access Rules**—Create and manage rules to control access to the nodes for this service instance. • **Add SSH Access**—Add public SSH keys to the nodes that make up this service instance. • **Manage Tags/Add Tags**—Either remove or add tags to a service instance. **Manage Tags** appears if a tag already exists for the service instance. **Add Tags** appears if no tags exist for the service instance. • **Enable Backups**—Enable backups for this service instance. • **View Activity**—View all administrative activities that have been performed on your service instances. • **View Instance Metrics**—View performance metrics for this service instance.

Element	Description
☰ (adjacent to the Administration Server)	**Menu** icon provides the following options: • **Restart**—Stop and immediately start the Administration Server VM. • **Scale Up/Down**—Scale the Administration Server node. • **Add Storage**—Increase the volume size or create an additional storage volume. **Note:** This menu is disabled when the service instance is stopped.
☰ (adjacent to a Managed Server)	**Menu** icon provides the following options: • **Remove Node**—Remove the Managed Server node. • **Stop**—Stop the Managed Server VM. • **Start**—Start the Managed Server VM if it is stopped. • **Restart**—Stop the Managed Server VM and immediately start it. • **Scale Up/Down**—Scale the Managed Server node. • **Add Storage**—Increase the volume size or create an additional storage volume. **Note:** This menu is disabled when the service instance is stopped. When the service instance is stopped, you cannot stop, start, or restart a Managed Server VM, or remove or scale a Managed Server node.
WebLogic Server Version	Version of Oracle WebLogic Server configured for the Oracle Java Cloud Service instance.
Description	Description of the Oracle Java Cloud Service instance.
⟳	Click to refresh the page. The date and time the page was last refreshed is displayed adjacent to this button.

Element	Description
▶ ■ ↻ ✚ ♡	Click the start/stop/restart/add node/monitor icons: • **Start Service**—Starts the nodes for the Administration Server, Managed Servers, load balancer, and Managed Servers on the Coherence data tier (if provisioned). • **Stop Service**—Stops the nodes for the Administration Server, Managed Servers, load balancer, and Managed Servers on the Coherence data tier (if provisioned). • **Restart Service**—Restart the nodes for the Administration Server, Managed Servers, load balancer, and Managed Servers on the Coherence data tier (if provisioned). • **Add a node to this service**—Select from two menu options: – **Add Node** —Adds a managed server node. – **Auto Scaling** —Opens the Rules page, which where you can view and configure auto-scaling rules. You cannot scale an Oracle Java Cloud Service—Virtual Image instance. • **Display monitoring information**—Displays similar monitor icons beside each node. Click these individual icons to see: – Date and time each node was last started. – For each server, the percent of heap space used as compared to the total heap space in GBs available. If you click the percentage number, the heap usage graph is displayed.
Status	Status of the service instance. Valid values include: Ready, In Progress, Maintenance, Terminating, Stopped, and Failed.
Backup Destination	Remote, local, or both remote and disk storage.
IDCS Application	Link that opens the Oracle Identity Cloud Service console, specifically to the page pertaining to the Oracle Identity Cloud Service application that's created automatically for each Oracle Java Cloud Service instance. This link appears only if you selected **Enable Authentication with Oracle Identity Cloud Service** when you provisioned the Oracle Java Cloud Service instance. In the Oracle Identity Cloud Service console, you can configure advanced security settings such as users, roles, and web tier policies for the instance.
Object Storage Container	Name of the cloud storage container you specified when you created the service instance.
License	Displays **BYOL** or **Cloud License**, depending on which license model you specified when you created the service instance. **BYOL** (Bring Your Own License)—The service instance leverages your on-premises license. **Cloud License**—The service uses your Oracle Java Cloud Service license. **Change** opens the Change License Type dialog box, which enables you to choose whether to leverage your on-premises license or subscribe to Oracle Java Cloud Service. You can change the license type only if your account has both BYOL and cloud license entitlements. (Not available on Oracle Cloud at Customer)
Version	Version of Oracle WebLogic Server configured for the Oracle Java Cloud Service instance. Also displays whether domain partitions are enabled.
JDK	Java Development Kit version used by the WebLogic Servers.

Element	Description
Open Sample Application	Link to the sample application, if you selected to deploy it when you provisioned the Oracle Java Cloud Service instance. See About the Sample Application Deployed to an Oracle Java Cloud Service Instance.
Tags	Tags assigned to the service instance. The first tag is displayed. To see all tags assigned to the service instance, hover over the tag name and click **More**.
 Overview 4 Nodes	Click the Overview tile to access the Oracle Java Cloud Service Instance Overview page (this page) at anytime. The Overview tile displays the number of running nodes for the Oracle Java Cloud Service instance. When Oracle Coherence is enabled for a service instance, the number of nodes includes the application tier nodes (storage-disabled) and the Coherence data tier nodes (storage-enabled).
 Administration 0 Patches available Apr 12, 2017 7:25:19 AM UTC Last Successful Backup	Click the Administration tile to backup and restore, and patch an Oracle Java Cloud Service instance. You can also create snapshots from a service instance and clone the snapshots. See: • Back Up and Restore an Oracle Java Cloud Service Instance • Patch an Oracle Java Cloud Service Instance • About Snapshots and Clones The Administration tile displays the number of patches applied and the date of last backup.

Element	Description
Administration or Managed Servers	Provides information about the Administration Server or Managed Server. Administration Server • **Administration Server Domain** *`domainName`*—Tag that identifies the node as the Administration Server and lists the domain name. • **Public IP**—Public IP address of the Administration Server. If you chose not to assign public IP addresses to the nodes in the service instance, this field displays the fully-qualified host name of the Administration Server. • **Fault Domain**—Fault domain where the service instance is located. Each fault domain is an isolated grouping of hardware and infrastructure within an availability domain. (Available only on Oracle Cloud Infrastructure) • **Instance**— Names of the servers used by the resource. • **Availability Domain**— The availability domain in the region in which the service instance is located. A region can have multiple isolated availability domains. (Available only on Oracle Cloud Infrastructure) • **OCPU**s—Number of Oracle CPUs allocated to the node. • **Memory**—Amount of memory in GBs allocated to the node. • **Storage**—Amount of storage in GBs allocated to the node. Healthcheck Details for Administration Server Appears after you click the monitoring icon at the top of the screen, and click the icon next to Instance. • **Admin Server Up Since**—Date and time the server started. • **Heap Usage**—The percentage heap space used as compared to the total heap space in GBs available. Appears only after you click the Monitoring icon. • **Managed Server Up Since**—Date and time the server started. • **Heap Usage**—The percentage heap space used as compared to the total heap space in GBs available. Managed Server: • **Host Name**—Host name of the Managed Server. • **Public IP**—Public IP address of the Managed Server. If you chose not to assign public IP addresses to the nodes in the service instance, this field displays the fully-qualified host name of the Managed Server. • **Instance**— Identifies the Managed Server. • **OCPUs**—Number of Oracle CPUs allocated to the node. • **Memory**—Amount of memory in GBs allocated to the node. • **Storage**—Amount of storage in GBs allocated to the node. Healthcheck Details for Managed Server Appears after you click the monitoring icon at the top of the screen, and click the icon next to Instance. • **Managed Server Up Since**—Date and time the server started. • **Heap Usage**—The percentage heap space used as compared to the total heap space in GBs available. (Advanced) If you have created multiple clusters, each server is described separately. You can mouse-over the server icon to see which cluster the server belongs to.

Element	Description
(available only when Oracle Coherence is enabled for the service instance)	Click the icon to view details about the data tier. • **Shape**—Compute shape for all nodes configured by a capacity unit. For example: `oc5`. • **Hosts**—Lists the host names for the Managed Servers configured in the cluster. If you need to identify the IP addresses for the nodes in the data tier cluster, access the Oracle Cloud Infrastructure Compute console.
Coherence Data Tier (available only when Oracle Coherence is enabled for the service instance)	Information about the storage-enabled nodes and capacity configured for the Coherence data tier: • **OCPUs**—Number of Oracle CPUs allocated to the Coherence data tier nodes. • **Memory**—Amount of memory in GBs allocated to the Coherence data tier nodes. • **Storage**—Amount of storage in GBs allocated to the Coherence data tier nodes. • **Data Grid Cluster**—Name of the data grid cluster created if creating a data grid cluster was specified when the service instance was created. • **Instance**— Name of the server used by the data grid. Note that when you stop or start the Oracle Java Cloud Service instance, all the virtual machines for the Managed Servers on the Coherence data tier will also stop or start. You cannot stop or start the data tier virtual machines individually.
Load Balancer	If an Oracle Traffic Director load balancer is provisioned, identifies information about the load balancer on the cluster: • **State**—Shows whether the load balancer is enabled or disabled. • **Host Name**—Name of the Oracle Traffic Director (OTD). • **Public IP**—Public IP address of the load balancer. If you chose not to assign public IP addresses to the nodes in the service instance, this field displays the fully-qualified host name of the load balancer. • **Instance**—Name of the server used by the load balancer. • **OCPUs**—Number of Oracle CPUs allocated to the OTD. • **Memory**—Amount of memory in GBs allocated to the OTD. • **Storage**—Amount of storage in GBs allocated to the OTD.

Element	Description
Load Balancer (Managed)	Shows details about the Oracle Cloud Infrastructure Compute Classic managed load balancer configured for your Oracle Java Cloud Service instance. • **URL**— HTTPS URL of the JCS Service instance including the port number (Default 443). Clients use the base URL to access applications. • **Type**—Shows the endpoint classification: – **Public**—Internet-facing – **Private**—Private endpoint, typically used for internal communication in a private network – **Management**—Private endpoint typically, used for administrative or service management communication • **Status**—Shows whether the load balancer is enabled or disabled. You can disabled by using the ☰ Menu beside the URL. If you disable the load balancer, you will receive an HTTP error. • **Listener**—Name (Read only) – **Path Prefixes**—By default a single path prefix will be set for the default cluster in case of single cluster Oracle Java Cloud Service instance. If multiple clusters are provisioned as part of the JCS instance then each cluster will have seperate LB listener with default path prefix as the cluster name `/`. In case of multiple clusters, you can provide a specific path prefix of your choice. – **Origin Servers**—IP addresses of the origin servers – You can disable listeners by using the ☰ Menu beside the listener. • **Aliases**—The URL Aliases resolve to the same front-end URL of the service – Shows the friendly URL of the pattern : `https://`*`ServiceName-AccountName.Data Jurisdiction`*`.oraclecloud.com` – Permanent URL of the pattern : `https://`*`ServiceGUID.Data Jurisdiction`*`.oraclecloud.com` The permanent URL does not change during the life of the service.

Element	Description
Associations	Information about the Oracle Database Cloud Service database deployments used by the Oracle Java Cloud Service instance, and, if the service instance is a clone, about the source service instance. **Note:** Oracle Autonomous Database (Oracle Autonomous Transaction Processing) and Oracle Cloud Infrastructure databases are not shown here. **Database Deployments** If the instance is based on multiple database deployments (one for the Oracle required schema and up to four for the application schemas), information for all database deployments is displayed. A database deployment can be a clone. • **Instance Name**—Name of the Database Cloud Service database deployments used by the Oracle Java Cloud Service instance. The names were specified during the process of creating the Oracle Java Cloud Service instance. Click the Database Cloud Service name to display the Instance Overview page for the database deployment. • **Service Type**—Database Cloud Service • **Type**—Depends On The Oracle Java Cloud Service instance requires the specified schema database deployment. • **Status**—Displays the state of the database deployment, for example, Ready or Maintenance. • **Manage Associations** ☰ menu —Choose from the following: – **View Details**—Displays the Association Name, Status, Description, Service Name, Type, and Usage Type. – **Reassociate Database**—Opens the Reassociate Database dialog box, which enables you to change the current infrastructure database to a different one. You cannot reassociate an application database. See Associate an Oracle Java Cloud Service Instance with a Different Database. (Available only on Oracle Cloud Infrastructure Classic) **Source for the Clone** The cloned source snapshot instance is shown here. This section is displayed only if service instance is a clone. • **Service Name**—Name of the source instance for the cloned Oracle Java Cloud Service instance. Click the service name to display the Instance Overview page for the source service instance. • **Service Type**—Oracle Java Cloud Service • **Type**—Depends on The cloned instance depends on the Oracle Java Cloud Service instance. You cannot delete a source service instance if it is cloned. • **Association Status**—Displays the state of the cloned instance, for example, Ready or Maintenance.

Element	Description
In-Progress Operation Messages	Details about an operation such as scaling, backup, or patching while the operation is in progress. • **Instance Name**—Service on which the operation is running. • **Operation**—Identifies the operation in progress, scale-out, for example. • **Operation Status**—Reports the status of the current operation, Running, for example. • **Start Time**—Date and time the operation started. • **End Time**—Date and time at which the operation completed. If the operation is in progress, this field is left blank. As the operation runs, messages are displayed with information about the operation’s progress.

4 Administer Oracle Java Cloud Service Software

From an Oracle Java Cloud Service instance, you can access the administration consoles and also individual nodes in order to run utilities such as the WebLogic Scripting Tool (WLST).

Topics:

- Set Up Fast and Secure Connections to Oracle Cloud
- Access the Administration Consoles for Oracle Java Cloud Service
- Access a Node with a Secure Shell (SSH)
- Use WLST to Administer a Service Instance
- Shut Down and Start Server Processes
- About JVM Heap Settings
- About Data Sources
- Manage Associations for a Service Instance
- Connect an Oracle Java Cloud Service Instance to an Application Database
- Configure an Oracle Java Cloud Service Instance for an Oracle RAC Database
- Configure a Vanity Domain Name for a Service Instance
- Configure a Custom URL for an Application Deployed to a Service Instance
- Configure a Custom URL for the WebLogic Server Console
- Configure a Custom URL for the Sample Application
- Monitor Applications with Oracle Java Flight Recorder and Oracle Java Mission Control
- Administration Best Practices

Set Up Fast and Secure Connections to Oracle Cloud

This topic does not apply to Oracle Cloud at Customer.

When you create a service instance, compute nodes are created in Oracle Cloud to host the components of the service instance. Your service users may need to connect to these nodes and the applications running on them. For example, service administrators may create SSH connections to the nodes. And end users may access applications deployed in the service through the ports that the applications listen for requests on.

You can connect to the public IP addresses of the compute nodes in Oracle Cloud over the internet. The connection speed would vary depending on the bandwidth of your internet connection and other factors. The security of your connection would depend on the protocol you use. For example, traffic over SSH and HTTPS connections is encrypted; HTTP traffic isn’t encrypted. Oracle offers the following faster and more secure solutions to connect to Oracle Cloud.

Topics

- Oracle Cloud Infrastructure FastConnect
- VPN

Oracle Cloud Infrastructure FastConnect

With Oracle Cloud Infrastructure FastConnect, traffic between your network and Oracle Cloud uses a direct and dedicated connection, bypassing the internet. You can also set up IPSec VPN tunnels over FastConnect connections.

Speed: When subscribing to FastConnect, you can opt for a port speed based on your business needs.

Cloud endpoints: You can connect to the public and private IP addresses of the compute nodes in Oracle Cloud.

Setup: Requires a subscription to FastConnect, and additional configuration for private peering. You can access all your service accounts within a site or region with a single FastConnect connection. The setup procedure varies depending on the compute infrastructure that your service instance is created in.

- Oracle Cloud Infrastructure: See FastConnect Overview in the Oracle Cloud Infrastructure documentation.
- Oracle Cloud Infrastructure Classic: See *Using Oracle Cloud Infrastructure FastConnect Classic*.

VPN

This topic does not apply to Oracle Cloud Infrastructure. Identify the Cloud Infrastructure Used by a Service Instance.

With Oracle's VPN solutions, all the data to and from Oracle Cloud is transported in an encrypted form over IPSec-based tunnels through the internet.

Speed: The speed depends on the bandwidth of your internet connection.

Cloud endpoints: You can connect to the public and private IP addresses of the compute nodes in Oracle Cloud.

Setup: You must create a VPN gateway in the cloud (one gateway per site or region) and connect it to an on-premises VPN gateway.

The VPN setup procedure depends on the cloud network type (IP network or shared network) that your nodes are attached to and your on-premises gateway device type.

Oracle Cloud Network Type	On-Premises Gateway	Setup Instructions
IP network	Third-party device	• (recommended) Setting Up a VPN Connection Using VPNaaS in *Using Oracle Cloud Infrastructure Compute Classic* • *Setting Up VPN from a Third-Party Gateway to an IP Network in Oracle Cloud*
IP network	Corente Services Gateway	*Setting Up VPN from a Corente Services Gateway to an IP Network in Oracle Cloud*
Shared network	Third-party device	*Setting Up VPN from a Third-Party Gateway On-Premises to the Shared Network*
Shared network	Corente Services Gateway	*Setting Up VPN from Corente Services Gateway On-Premises to the Shared Network*

Access the Administration Consoles for Oracle Java Cloud Service

You can use various consoles to administer the software that an Oracle Java Cloud Service Software instance is running, and to also administer related Oracle Cloud services.

> **Note:**
>
> Security check warnings are displayed at the top of the console. See About the Security Checkup Tool for the warnings and how to handle them.

Topics:

- About the Security Checkup Tool
- Access an Administration Console for a Service Instance
- Access the Console of a Related Oracle Cloud Service
- Access the Administration Console for a Service Instance Attached to a Private Subnet

About the Security Checkup Tool

Oracle WebLogic Server Administration console includes a security checkup tool that displays security check warnings. These security check warnings are displayed for Oracle Java Cloud Service instances that are created using WebLogic Server versions 12.2.1.3 and 12.2.1.4.

In case of Oracle Java Cloud Service instances created after July 20, 2021, or the instances on which the July 2021 PSUs are applied, the message `Security warnings detected. Click here to view the report and recommended remedies` is displayed at the top of the Oracle WebLogic Server Administration console. When you click the message, a list of security warnings are displayed as listed in the following table.

The warning messages listed in the table are examples.

Security Warnings

Warning Message	Resolution
`Tunneling is enabled on server channel channel-dep. Allowing T3 or IIOP to be tunneled on a server channel may allow deserialization of specially crafted, malicious serialized objects that can potentially cause denial of service.` **Note:** This warning is displayed only for existing Oracle Java Cloud Service instances created before release 22.1.1 (January 31, 2022) on which the October 2021 PSUs are applied.	Disable tunneling on `channel-dep` server channel. See Disable Tunneling on Server Channel.
`Remote Anonymous RMI T3 or IIOP requests are enabled. Set the RemoteAnonymousRMIT3Enabled and RemoteAnonymousRMIIIOPEnabled attributes to false.`	Disable the anonymous RMI T3 and IIOP requests in the WebLogic Server Administration Console as soon as possible unless your deployment requires anonymous T3 or IIOP (not typical). See Disable Remote Anonymous RMI T3 and IIOP Requests. **Note**: These attribute settings are also applicable to Oracle Traffic Director, but only for service instances running Oracle Traffic Director 12.2.1.4.

> **Note:**
>
> For existing Oracle Java Cloud Service instances created before release 21.3.2 (August 26, 2021), you see the SSL host name verification and the umask warnings. See Security Checkup Tool Warnings.

After you address the warnings, you must click **Refresh Warnings** to see the warnings removed in the console.

For Oracle Java Cloud Service instances created after July 20, 2021, though the java properties to disable anonymous requests for preventing anonymous RMI access are configured, the warnings still appear. This is a known issue in Oracle WebLogic Server.

If you want to perform anonymous RMI requests,, you must set the java properties for anonymous RMI T3 and IIOP requests. See Set the Java Properties.

Configure the Wildcard Host Name Verifier

To address the SSL hostname verification warnings, you must configure the wildcard host name verifier in the Administration console.

1. Locate the **Change Center** and click **Lock & Edit** to lock the editable configuration hierarchy for the domain.
2. Under **Domain structure**, select **Environment** and then select **Servers**.
3. In the Servers table, select the server instance you want to configure.
4. On the **Configuration** tab, select **SSL** and then expand **Advanced**.
5. Set the **Hostname Verification** field to **Custom Hostname Verifier** and enter *weblogic.security.SSL.HostnameVerifier* in the **Custom Hostname Verifier** field.

   After saving the changes, return to **Change Center** and click **Activate Changes**
6. Repeat steps 3 to 5 for all instances of administration server and managed server.

Update Administration Server Startup Properties

To address the SSL hostname verification warnings, you must update the `startup.properties` file for the administration server.

1. From your computer, run the `ssh` command to create an SSH tunnel to the node as the `opc` user.

   Command format is:

   ```
   ssh -i path_to_private_key opc@<node_IP_address>
   ```
2. Change to the `oracle` user.

   ```
   sudo su - oracle
   ```
3. Navigate to the `nodemanager` directory and list the files in the directory.

   ```
   cd /u01/data/domains/domain_name/servers/admin_server_name/data/
   nodemanager
   ls
   ```

   Where, `domain_name` and `admin_server_name` must be replaced with your domain name and the administration server name.
4. Open the `startup.properties` file in vi editor.
5. For `SSL Arguments`, remove -`Dweblogic.security.SSL.ignoreHostnameVerification=false` and save the file.

   Example of `startup.properties` file:

   ```
   RotateFileCount=7
   FileTimeSpanFactor=3600000
   RestartMax=2
   FileSizeKB=500
   ```

```
AutoRestart=true
NumberOfFilesLimited=true
RestartDelaySeconds=0
SSLArguments=-Dweblogic.ReverseDNSAllowed\=false
RotationType=bySize
RestartInterval=3600
RotationTimeStart=00\:00
FileTimeSpan=24
```

Restart Managed Server Using Node Manager

To address the SSL hostname verification warnings, you must restart the managed sever using node manager.

1. Locate the **Change Center** and click **Lock & Edit** to lock the editable configuration hierarchy for the domain.
2. Under **Domain structure**, select **Environment** and then select **Servers**.
3. Click **Control**, and in the Servers table, select a managed server.
4. In the **Shutdown** drop-down list, select **Force shutdown now**.
5. Click **Yes** to confirm.

   The server may take a while to shut down. You can click the Refresh icon to manually refresh the console page.
6. Select the managed server that you want to shut down and click **Start**.
7. Click **Yes** to confirm.
8. Repeat steps 3 to 7 for all instances of managed server.

After saving the changes, return to **Change Center** and click **Activate Changes**.

Set the Java Properties

You can perform anonymous RMI requests by setting the java properties for anonymous RMI T3 and IIOP requests.

To set the java properties to disable the Remote Anonymous RMI T3 and IIOP Requests in the WebLogic Server Administration console:

1. Locate the **Change Center** and click **Lock & Edit** to lock the editable configuration hierarchy for the domain.
2. Under **Domain structure**, select **Environment** and then select **Servers**.
3. In the Servers table, select the server instance you want to configure.
4. On the **Configuration** tab, select **Server Start**.
5. Remove the following properties from **Arguments**:
   - `Dweblogic.security.remoteAnonymousRMIT3Enabled=false`
   - `Dweblogic.security.remoteAnonymousRMIIIOPEnabled=false`

After saving the changes, return to **Change Center** and click **Activate Changes**.

Disable Remote Anonymous RMI T3 and IIOP Requests

You can disable the anonymous requests from clients.

To disable the remote anonymous RMI T3 and IIOP requests in the WebLogic Server Administration console:

1. Locate the **Change Center** and click **Lock & Edit** to lock the editable configuration hierarchy for the domain.
2. Under **Domain structure**, select the domain name, and then select the **Security** tab.
3. Expand **Advanced** and deselect **Remote anonymous RMI access via IIOP** and **Remote anonymous RMI access via T3**.

After saving the changes, return to **Change Center** and click **Activate Changes**.

Disable Tunneling on Server Channel

To disable tunneling on server channel `channel-dep`:

1. Locate the **Change Center** and click **Lock & Edit** to lock the editable configuration hierarchy for the domain.
2. Under **Domain structure**, select **Environment** and then select **Servers**.
3. In the Servers table, select the administration server instance you want to configure.
4. On the **Protocols** tab, select **Channels**.
5. Select the `channel-dep` network channel and then expand **Advanced**.
6. Clear the **Tunneling Enabled** check box.

   After saving the changes, return to **Change Center** and click **Activate Changes**.

Access an Administration Console for a Service Instance

From an Oracle Java Cloud Service instance, you can access the administration consoles for the software that the service instance is running.

You can access these consoles:

- WebLogic Server Administration Console
- Fusion Middleware Control Console
- Load Balancer Console (Oracle Traffic Director only)

> **Note:**
>
> By default, if you created your service instance in an Oracle Cloud Infrastructure Classic region, external access to these administration consoles is disabled for security purposes. If you did not enable console access while provisioning your service instance, see Enabling Console Access in an Oracle Java Cloud Service Instance. If you created your service instance in an Oracle Cloud Infrastructure region, this procedure is not necessary. Access to the administration consoles is enabled by default in these regions.

> **Note:**
>
> If you created your service instance and chose not to assign public IP addresses, then these administration consoles are not directly accessible from the Internet. They are accessible only from within your private IP network, or from your on-premises data center over a VPN network.

> **Note:**
>
> Prior to modifying the default configuration of these software components, see Administration Best Practices. For example, if you disable a console or modify the default port number used to access it, the shortcuts described here may not work.

To access a console:

1. Access the Oracle Java Cloud Service console.
2. Click **Manage this instance** ☰ for the desired service instance, and then open the console that you want to access:

To access this console	Click this shortcut
WebLogic Server Administration Console	**Open WebLogic Server Administration Console**
Fusion Middleware Control Console	**Open Fusion Middleware Control Console**
Load Balancer Console	**Open Load Balancer Console**

   A new browser opens and you are redirected to the selected console's login page.

   If the server is protected with a self-signed certificate, you will be warned that this certificate is not trusted.
3. Accept the certificate if prompted. These steps are browser-dependent.
   - If you are using Firefox, click **Advanced** , click **Add Exception** and then click **Confirm Security Exception**.
   - If you are using Chrome, click **Advanced** and then click **Proceed**.
4. When the console login page appears, enter the Oracle WebLogic Server user name and password you provided when you created the service instance.

   If you created this service instance from a QuickStart template, these credentials were generated for you and placed in an archive file that you downloaded to your local machine.

Access the Console of a Related Oracle Cloud Service

You can access the consoles for related Oracle Cloud services, such as Oracle Database Cloud Service, from the Oracle Java Cloud Service console.

1. Access the Oracle Java Cloud Service console.

2. Click the ☰ menu at the top left of the page, expand **Services**, and then choose the service that you want to access.

   For example, choose **Database Classic** to access the Oracle Database Cloud Service console.

Access the Administration Console for a Service Instance Attached to a Private Subnet

You can access the administration compute instance of an Oracle Java Cloud Service instance through a bastion host attached to a public subnet.

> **Note:**
>
> For this procedure to work, you must have created a bastion host and configured security rules in Oracle Cloud Infrastructure to allow SSH connections from the public internet to the bastion host, and to allow TCP traffic from the bastion host to the other compute nodes in the VCN.

1. Sign in to the web console of Oracle Java Cloud Service.
2. Locate the instance for which you want to access the administration consoles.
3. Click **Manage this instance** ☰ for the required instance, *hover over* (but don’t click) each console option in the menu (**Open Fusion Middleware Control Console** and **Open WebLogic Server Administration Console**), and then copy the URL shown in the browser’s status bar at the bottom of the window.
4. Open an SSH tunnel to the bastion host on your local computer as `opc` user.

   ```
   ssh -D <Localport> -fCqN -i key <opc@bastionPublicIPaddress>
   ```

   where, `bastionPublicIPaddress` is the public IP address of the bastion host.
5. In your browser settings, set up the SOCKS (version 5) proxy configuration. Specify your local computer and the same port that you used in your SSH command.
6. In the browser, enter the URL of the administration console that you noted earlier.

   Alternatively, sign in to the Oracle Java Cloud Service web console from within the bastion host, locate your service instance, and select the required console menu option from the **Manage this instance** menu.

Access a Node with a Secure Shell (SSH)

You can access the services and resources that an Oracle Java Cloud Service instance's node provides by logging into the node as the `opc` user through SSH. You can use any SSH utility you want. For example, if you are using Windows, you might use PuTTY; if you are using Linux, you might use OpenSSH.

By default, only the `opc` user can remotely connect to your nodes. You cannot use SSH to connect to a node as the `oracle` user. After successfully connecting to a node, tasks such as starting and stopping the server and accessing the administrative logs should only be performed by the `oracle` user.

Oracle Cloud uses SSH to access the nodes that comprise your service instances, in order to perform predefined Platform Service actions like backup and patching. You initiate these Platform Service actions from the web console, CLI, or REST API. A separate SSH key pair is used for each service instance to perform this internal communication. This SSH key is not available for ad hoc usage. You cannot delete this key from nodes or it will cause these Platform Service actions to fail. The key is only used under programmatic control and cannot be directly accessed by Oracle employees. All SSH actions performed by Oracle Cloud on your nodes are logged and can be audited. Oracle does not have access to any SSH keys residing on your nodes and has no way to access your nodes, unless you explicitly provide access to the keys for troubleshooting purposes.

> **Note:**
>
> If you created your service instance in an Oracle Cloud Infrastructure Classic region and chose not to assign public IP addresses, then the nodes in your service instance are not directly accessible from the Internet. They are accessible only from within your private IP network, or from your on-premises data center over a VPN network.

> **Note:**
>
> Prior to making changes to the operating system or the Oracle software on a node, see Administration Best Practices.

Topics:

- Generate a Key Pair with OpenSSH
- Connect to a Node with OpenSSH
- Create an SSH Tunnel to a Node with OpenSSH
- Connect to a Private Node with OpenSSH
- Generate a Key Pair with PuTTY
- Convert a Private Key with PuTTY
- Connect to a Node with PuTTY
- Create an SSH Tunnel to a Node with PuTTY
- Connect to a Node with VNC
- Switch Users on a Node
- Add an SSH Public Key
- Add an SSH User

Generate a Key Pair with OpenSSH

You can generate a secure shell (SSH) key pair for an Oracle Java Cloud Service instance on a UNIX or UNIX-like platform by using the ssh-keygen utility.

1. From your computer, run the `ssh-keygen` utility.

Specify a `filename` for the private key. Also specify the RSA type and a size of 2048.

The command format is: `ssh-keygen -b 2048 -t rsa -f filename`

For example: `ssh-keygen -b 2048 -t rsa -f mykey`

2. When prompted, enter a passphrase for the private key, or press Enter to create a private key without a passphrase.

   ```
   Enter passphrase (empty for no passphrase): YourPassphrase
   ```

   > **Note:**
   >
   > While a passphrase is not required, Oracle recommends using one as a security measure to protect the private key from unauthorized use. When you specify a passphrase, a user must enter the passphrase every time the private key is used.

3. If you provided a passphrase, enter it a second time when prompted.

   ```
   Enter the same passphrase again: YourPassphrase
   ```

The `ssh-keygen` utility creates two files:

- *`filename`* - The private key
- *`filename`*`.pub` - The public key

Connect to a Node with OpenSSH

You can access a node in an Oracle Java Cloud Service instance from a UNIX or UNIX-like platform by using a secure shell (SSH) utility.

After you open an SSH connection to a node, you can issue commands to the Linux OS.

1. Access your service console.
2. Click the name of the service instance that contains the node that you want to access.
3. On the Overview page, identify the **Public IP** address of the node you want to access.

   For example, `203.0.113.13`.
4. From your computer, run the `ssh` command to connect to the node as the `opc` user.

   Provide the path to the private key that corresponds to the public key that you specified when you created this service instance, and the node’s public IP address.

   The command format is: `ssh -i path_to_private_key opc@node_IP_address`

   For example: `ssh -i /home/myuser/id_rsa opc@203.0.113.13`
5. If prompted, enter the passphrase for the private key.

Create an SSH Tunnel to a Node with OpenSSH

If a resource provided by an Oracle Java Cloud Service node uses a port that is not directly accessible through the Internet, you can access that resource by creating a secure shell

(SSH) tunnel to the port. You can create an SSH tunnel from a UNIX or UNIX-like platform by using the SSH utility.

In general, an SSH tunnel can map a remote port to any available port number on your local computer. Some protocols, such as Java Remote Method Invocation (RMI), require that the remote and local port numbers be the same value.

1. Access your service console.
2. Click the name of the service instance that contains the node that you want to access.
3. On the Overview page, identify the **Public IP** address of the node that you want to access.

   For example, `203.0.113.13`.
4. From your computer, run the `ssh` command to create an SSH tunnel to the node as the `opc` user.

   Provide the following:

   - The path to the private key that corresponds to the public key that you specified when you created this service instance.
   - The node’s public IP address.
   - The port number on the node to which you want to connect. The SSH tunnel enables connectivity to this remote port by using the port with same number on your local computer.

   The command format is: `ssh -i` *`path_to_private_key`* `-L` *`port`*`:`*`node_IP_address`*`:`*`port`* `opc@`*`node_IP_address`* `-N`

   For example: `ssh -i /home/myuser/id_rsa -L 9001:203.0.113.13:9001 opc@203.0.113.13 -N`
5. If prompted, enter the passphrase for the private key.

Applications that are running on your local computer can now communicate with the node by using `localhost:`*`port`*, where *`port`* is the local port number.

After your work with the SSH tunnel is completed, press Ctrl+C to close the SSH tunnel.

Connect to a Private Node with OpenSSH

If a node in your Oracle Java Cloud Service instance does not have a public IP address, you can connect to it from a UNIX or UNIX-like platform by using another node as a proxy.

A node that's dedicated to providing administrative access to other private nodes is also referred to as a *bastion*.

Use the `ProxyCommand` option in OpenSSH to specify the node to use as a bastion when making the secure shell (SSH) connection. After you open an SSH connection to the private node, you can issue commands to the Linux OS.

The node that you intend to use as a bastion must be able to access the private node to which you’re connecting. If necessary, create an access rule or a security rule that enables communication between the two nodes prior to connecting to the node with SSH.

1. Identify the **Public IP** address of a node to use as a bastion (proxy).
2. Access your service console.
3. Click the name of the service instance that contains the private node that you want to access.
4. On the Overview page, identify the **Host Name** of the node.
5. From your computer, run the `ssh` command to connect to the private node as the `opc` user, and also specify the command to connect to the proxy node.

   Provide the following:

   - The path to the private key that corresponds to the public key that you specified when you created this service instance.
   - The proxy node's public IP address.
   - The private node's host name.

   The command format is: `ssh -i` *path_to_private_key* `-o ProxyCommand="ssh -W %h:%p -i` *path_to_private_key* `opc@`*proxy_node_IP_address*`" opc@`*node_host_name*

   For example: `ssh -i /home/myuser/id_rsa -o ProxyCommand="ssh -W %h:%p -i /home/myuser/id_rsa opc@203.0.113.13" opc@myinstance-node2`
6. If prompted, enter the passphrase for the private key.

Generate a Key Pair with PuTTY

You can generate a secure shell (SSH) key pair for an Oracle Java Cloud Service instance on a Windows platform by using the PuTTY Key Generator utility.

PuTTYgen is included with your PuTTY installation. To download PuTTY, go to http://www.putty.org/.

1. Start PuTTYgen on your Windows computer.

   The PuTTY Key Generator window is displayed.
2. For **Type of key to generate**, select **SSH-2 RSA**.
3. For **Number of bits in a generated key**, enter 2048.
4. Click **Generate**.
5. When prompted, move the mouse around the specified section of the window.
6. Optional: Enter a passphrase for the private key in **Key passphrase** and **Confirm passphrase**.

   > **Note:**
   >
   > While a passphrase is not required, Oracle recommends using one as a security measure to protect the private key from unauthorized use. When you specify a passphrase, a user must enter the passphrase every time the private key is used.
7. Click **Save private key**.

The `.ppk` file extension indicates that the private key is in PuTTY's proprietary format. You must use a key of this format when using PuTTY as your SSH client. It cannot be used with other SSH client tools.

8. Select the entire contents of **Public key for pasting into OpenSSH authorized_keys file**.

   This window may have a scroll bar. Be sure to select all of the characters.

9. Right-click the selected text, and then select **Copy**.
10. Open a text editor.
11. Paste the copied text into the editor. Do not insert any line breaks.
12. Save the public key file to the same location as the private key file.
13. Optional: Create a copy of the private key in the OpenSSH format.

    a. From the PuTTY Key Generator window, click **Conversions**, and then select **Export OpenSSH key**.

    b. Save the converted private key file to the same location as the `.ppk` file. Use a different file extension such as `.openssh`.

Convert a Private Key with PuTTY

On a Windows platform you can use the PuTTY Key Generator utility to convert an existing private key from OpenSSH format to PuTTY format.

PuTTYgen is included with your PuTTY installation. To download PuTTY, go to http://www.putty.org/.

1. Start PuTTYgen on your Windows computer.

   The PuTTY Key Generator window is displayed.

2. Click **Load**.
3. For the type of file, select **All Files**. Then browse to and select the private key file.
4. If prompted, enter the passphrase for the private key, and then click **OK**.
5. When a notice displays about a foreign key format, click **OK**.
6. Optional: If the original key did not have a passphrase, then enter a value in **Key passphrase** and **Confirm passphrase**.

   > **Note:**
   >
   > While a passphrase is not required, Oracle recommends using one as a security measure to protect the private key from unauthorized use. When you specify a passphrase, a user must enter the passphrase every time the private key is used.

7. Click **Save private key**.
8. Select a location to which to save the converted (`.ppk`) private key file.

Connect to a Node with PuTTY

You can access a node in an Oracle Java Cloud Service instance from a Windows platform by using PuTTY, an open source networking client.

After you open an SSH connection to a node, you can issue commands to the Linux OS.

To download PuTTY, go to http://www.putty.org/.

1. Access your service console.
2. Click the name of the service instance that contains the node that you want to access.
3. On the Overview page, identify the **Public IP** address of the node that you want to access.

   For example, `203.0.113.13`.
4. Start PuTTY on your Windows computer.

   The PuTTY Configuration window is displayed, showing the Session panel.
5. In the **Host Name (or IP address)** field, enter the public IP address of the node.
6. In the Category navigation tree, expand **Connection**, and then click **Data**.
7. In the **Auto-login username** field, enter `opc`.
8. In the **When username is not specified** field, select **Prompt**.
9. In the Category tree, expand **Connection**, expand **SSH**, and then click **Auth**.
10. Under **Private key file for authentication**, click **Browse**.
11. Navigate to the location of your private key file, and select it. Click **Open**.

    This private key corresponds to the public key that you specified when you created this service instance.

    > **Note:**
    >
    > The `.ppk` file extension indicates that the private key is in PuTTY's proprietary format. You must use a key of this format when using PuTTY. If Oracle Cloud generated this key for your service instance, see the PuTTY documentation for information about converting the key format.
12. Optional: To save this session configuration, click **Session** in the Category tree, and then click **Save**.

    To load a saved configuration, select the configuration name, and then click **Load**.
13. Click **Open**.
14. If prompted, enter the passphrase for the private key.

Create an SSH Tunnel to a Node with PuTTY

If a resource provided by an Oracle Java Cloud Service node uses a port that is not directly accessible through the Internet, you can access that resource by creating a secure shell

(SSH) tunnel to the port. You can create an SSH tunnel from a Windows platform by using PuTTY, an open source networking client.

In general, an SSH tunnel can map a remote port to any available port number on your local computer. Some protocols, such as Java Remote Method Invocation (RMI), require that the remote and local port numbers be the same value.

To download PuTTY, go to http://www.putty.org/.

1. Access your service console.
2. Click the name of the service instance that contains the node that you want to access.
3. On the Overview page, identify the **Public IP** address of the node that you want to access.

   For example, `203.0.113.13`.
4. Start PuTTY on your Windows computer.

   The PuTTY Configuration window is displayed, showing the Session panel.
5. In the **Host Name (or IP address)** field, enter the public IP address of the node.
6. In the Category navigation tree, expand **Connection**, and then click **Data**.
7. In the **Auto-login username** field, enter `opc`.
8. In the **When username is not specified** field, select **Prompt**.
9. In the Category tree, expand **Connection**, and then click **SSH**.
10. Under **Protocol options**, select the check box **Don't start a shell command at all**.
11. In the Category tree, expand **SSH**, and then click **Auth**.
12. Under **Private key file for authentication**, click **Browse**.
13. Navigate to the location of your private key file, and select it. Click **Open**.

    This private key corresponds to the public key that you specified when you created this service instance.

    > **Note:**
    >
    > The `.ppk` file extension indicates that the private key is in PuTTY's proprietary format. You must use a key of this format when using PuTTY. If Oracle Cloud generated this key for your service instance, see the PuTTY documentation for information about converting the key format.

14. In the Category tree, expand **SSH**, and then click **Tunnels**.
15. In the **Destination** field, enter `*IP*:*port*`,

    where *IP* is the IP address of the node and *port* is the port number on the node to which you want to connect.
16. In the **Source Port** field, enter the same port number.
17. Click the **Add** button.
18. Optional: To save this session configuration, click **Session** in the Category tree, and then click **Save**.

To load a saved configuration, select the configuration name, and then click **Load**.

19. Click **Open**.
20. If prompted, enter the passphrase for the private key.

Applications that are running on your local computer can now communicate with the node by using `localhost:`*`port`*, where *`port`* is the local port number.

After your work with the SSH tunnel is completed, press Ctrl+C to close the SSH tunnel.

Connect to a Node with VNC

You can access a node in an Oracle Java Cloud Service instance by using a Virtual Network Computing (VNC) client utility.

You can use VNC to work with any OS resources that are accessible from the node, including graphical applications.

From a Windows platform, you can use RealVNC or TightVNC. From a Linux platform, you can use the `vncviewer` utility that is included with your Linux distribution.

By default, the VNC server on a node in Java Cloud Service uses a port that is not directly available through the Internet. An SSH tunnel enables access to the node's VNC server port on your local computer. An SSH tunnel also ensures that all VNC communication uses a secure protocol.

1. Access your service console.
2. Click the name of the service instance that contains the node that you want to access.
3. On the Overview page, identify the **Public IP** address of the node that you want to access.

   For example, `203.0.113.13`.
4. From your computer, run the `ssh` command to connect to the node as the `opc` user.

   Provide the path to the private key that corresponds to the public key that you specified when you created this service instance, and the node's public IP address.

   The command format is: `ssh -i `*`path_to_private_key`*` opc@`*`node_IP_address`*

   For example: `ssh -i /home/myuser/id_rsa opc@203.0.113.13`
5. If prompted, enter the passphrase for the private key.
6. Switch to the `oracle` user.

   ```
   sudo su - oracle
   ```

   The `oracle` user has regular OS user permissions. It is intended to be used to start and stop Oracle products that have been installed on the node, or to run other Oracle applications and utilities on the node.
7. Disable the desktop screensaver lock for this user.

   ```
   gconftool-2 -s -t bool /apps/gnome-screensaver/lock_enabled false
   ```

   This Linux property controls whether or not the desktop prompts you for the user's password when in screensaver mode.

8. Start the VNC server on the node, and if necessary change the screen resolution to match the resolution of your local computer. For example, to 1680 x 1050.

   ```
   vncserver -nolisten tcp -localhost -geometry 1680x1050
   ```

9. When prompted, enter a password for this VNC session.
10. Note the display number for this VNC session, such as `:1`.

    By default, the listen port is 5901 for VNC session `:1` , port 5902 for VNC session `:2`, and so on.

11. Disconnect from the node.

    ```
    exit
    ```

12. Run the following command to open an SSH tunnel to `localhost:`*`vnc_port`* on the node.

    The command format is: `ssh -i` *`path_to_private_key`* `-L` *`vnc_port`*`:localhost:`*`vnc_port`* `opc@`*`node_IP_address`* `-N`

    For example: `ssh -i /home/myuser/id_rsa -L` **`5901:localhost:5901`** `opc@203.0.113.13 -N`

13. If prompted, enter the passphrase for the private key.
14. Start your VNC client application and connect to `localhost:`*`vnc_port`*.
15. When prompted, enter the password that you previously configured for this VNC session.

After your work with the SSH tunnel is completed, press Ctrl+C on your local computer to close the SSH tunnel.

To terminate the VNC server on the node, run the command `vncserver –kill :`*`display_num`*. For example, `vncserver –kill :1`

Switch Users on a Node

After connecting to an Oracle Java Cloud Service node, you can change operating system (OS) users in order to perform specific administration tasks.

By default, you must connect to a node only as the `opc` user. This user has root privileges on the OS running in the node. For example, `opc` can be used to create other OS users on a node. Simply prefix root operations with the `sudo` command. For example:

```
sudo useradd myuser
```

> **Note:**
>
> There is no default password for the `opc` user.

Switch to the Oracle User

The `oracle` user has regular OS user permissions. It is intended to be used to start and stop Oracle products that have been installed on the node, or to run other Oracle applications and utilities on the node.

Type the following to become the `oracle` user:

```
sudo su - oracle
```

> **Note:**
>
> There is no default password for the `oracle` user.

Switch to the Root User

An alternative to using the `sudo` command to perform root OS operations with the `opc` user is to switch to the `root` user.

Type the following to become the `root` user:

```
sudo -s
```

> **Caution:**
>
> Avoid using the `root` user except to perform privileged OS administration tasks.

Add an SSH Public Key

You can add Secure Shell (SSH) public keys to an Oracle Java Cloud Service instance.

You might need to add a new SSH public key to a service instance if the SSH private key that you use to access the service instance becomes lost or corrupted. Or, you might need to comply with your organization's security policies or regulations.

1. Access your service console.
2. Beside the service instance to which you want to add a new SSH public key, click **Manage this instance** ☰, and then select **Add SSH Access**.

   The dialog box displays the value of the most recently added public key.
3. Specify the new public key, by completing one of the following:
   - Select **Upload a new SSH Public Key value from file** and then use your browser to upload a public key file from your local computer.
   - Select **Key Value**. Delete the previous public key value from the input field and then enter or paste the new value. Be sure not to include other characters that aren't part of the key, such as spaces.
4. Click **Add New Key**.

Add an SSH User

You can add an operating system (OS) user to an Oracle Java Cloud Service node, and then use a secure shell (SSH) utility to connect to the node as the new user.

Before you add a new SSH user to a node, connect to the node as the opc user.

> **Note:**
>
> Use caution when making modifications to a node's OS configuration as the `root` user. Certain changes might cause other Java Cloud Service management operations to fail.

1. Switch to the `root` user.

   ```
   sudo su -
   ```

2. Create a new user.

   ```
   useradd username
   ```

   For example: `useradd myadminuser`

3. Create a directory named `.ssh` in the new user's home directory.

   ```
   mkdir /home/username/.ssh
   ```

   For example: `mkdir /home/myadminuser/.ssh`

4. Copy the `authorized_keys` file from the opc user's `.ssh` directory to the new user's `.ssh` directory.

   ```
   cp /home/opc/.ssh/authorized_keys /home/username/.ssh
   ```

   For example: `cp /home/opc/.ssh/authorized_keys /home/myadminuser/.ssh`

   > **Note:**
   >
   > Alternatively, you can create a new SSH key pair for the new user, and paste the contents of the public key into the user's `authorized_keys` file. Do not add extra lines or line breaks.

5. Change the owner of the `/home/username/.ssh` directory.

   ```
   chown -R username:username /home/username/.ssh
   ```

   For example: `chown -R myadminuser:myadminuser /home/myadminuser/.ssh`

6. Edit the file `/etc/ssh/sshd_config`. Add the new user to the list of users in the `AllowUsers` parameter. Separate each user in the list with a space.

   ```
   AllowUsers existing_users username
   ```

   For example: `AllowUsers opc myadminuser`

> **Important:**
>
> The `AllowUsers` parameter must be placed before any `Match` parameters in the `sshd_config` file.

7. Verify that there are no errors in your SSH configuration.

   ```
   /usr/sbin/sshd -t
   ```

   > **Caution:**
   >
   > Correct any errors described in the output. Otherwise, the SSH service will not start properly and you will not be able to reconnect to this node.

8. Restart the SSH service.

   ```
   /sbin/service sshd restart
   ```

9. Optional: Run `visudo` and grant `sudo` privileges to the new user.

   ```
   username ALL=(ALL) NOPASSWD: ALL
   ```

   For example: `myadminuser ALL=(ALL) NOPASSWD: ALL`

10. Disconnect from the node.
11. Connect to the node as the new user.

    For example: `ssh -i /home/myuser/id_rsa` **`myadminuser`**`@203.0.113.13`

12. If prompted, enter the passphrase for the private key.

Use WLST to Administer a Service Instance

You can use the WebLogic Scripting Tool (WLST) to administer the Oracle WebLogic Server domain in your Oracle Java Cloud Service instance from a command line or script.

> **Note:**
>
> Prior to modifying the default WebLogic Server configuration in your service instance, see Administration Best Practices.

Topics

- About WLST Online and Offline
- Run WLST Commands on a Node
- Run WLST Commands from a Different Host

About WLST Online and Offline

You can use WLST as the command-line equivalent to the WebLogic Server Administration Console (WLST online) or as the command-line equivalent to the Configuration Wizard (WLST offline).

Online WLST commands allow you to connect to a running Administration Server and manage the configuration of an active WebLogic Server domain, view performance data about resources in the domain, or manage security data. The commands also allow you to connect to Managed Servers, although you cannot modify configuration data from Managed Servers.

Offline—that is, without connecting to a running WebLogic Server instance—WLST allows you to create domain templates, create a new domain based on existing templates, or extend an existing, inactive domain. You cannot use WLST offline to view performance data about resources in a WebLogic Server domain or modify security data. You cannot run offline commands from a remotely-attached Oracle Java Cloud Service instance because the domain configuration files are not local to your system, and so you cannot manipulate them.

Run WLST Commands on a Node

You can run WLST commands from within any Oracle Java Cloud Service node that includes an Oracle WebLogic Server installation.

Prior to running WLST, identify the public IP address of the node and connect to it with SSH, as described in Access a Node with a Secure Shell (SSH).

1. After connecting to the node, switch to the `oracle` user:

   ```
   sudo su - oracle
   ```

2. Change the directory to the `bin` folder in `DOMAIN_HOME`.

   ```
   cd $DOMAIN_HOME/bin
   ```

   For example, `/u01/data/domains/OurServi_domain/bin`

3. Set up the environment.

   ```
   . ./setDomainEnv.sh
   ```

   You must use the `.` to ensure that the environment variables are set in the current shell.

4. Launch WLST:

   ```
   $COMMON_COMPONENTS_HOME/common/bin/wlst.sh
   ```

5. Connect to the Administration Server:

   ```
   connect('loginID', 'password', 'admin-server-host:admin-server-port')
   ```

   For example:

   ```
   connect('weblogic', 'password', 'service-wls-1:7001')
   ```

You can now use WLST. Refer to the *WLST Command and Variable Reference* in one of the following publications:

- WLST Command Reference for WebLogic Server (12.2.1.4)
- WLST Command Reference for WebLogic Server (12.2.1.3)
- WebLogic Scripting Tool Command Reference (10.3.6)

Run WLST Commands from a Different Host

You can run WLST commands from a host that is not an Oracle Java Cloud Service node (for example, from your laptop or a separate machine running in the Cloud). Use the WLST installation on this remote machine to connect to your Oracle Java Cloud Service Administration Server.

By default, if you created your service instance in an Oracle Cloud Infrastructure Classic region, remote access to the administration console and WLST over port `7002` is disabled for security purposes. If you did not enable console access while provisioning your service instance, see Enabling Console Access in an Oracle Java Cloud Service Instance.

Alternatively, you can create an SSH tunnel to port `9001` on the Administration Server node.

To run WLST commands remotely:

1. Use the Oracle Java Cloud Service console to identify the public IP address of your Administration Server.
2. Launch a command shell on a machine with an Oracle WebLogic Server installation.
3. If you have not updated the default SSL configuration of your Administration Server, set the following environment variable to accept the default SSL certificate:

   ```
   export WLST_PROPERTIES="-
   Dweblogic.security.SSL.ignoreHostnameVerification=true -
   Dweblogic.security.TrustKeyStore=DemoTrust"
   ```

4. Navigate to your Oracle WebLogic Server installation and launch WLST using the `wlst.sh` script.

   ```
   /Middleware_Home/oracle_common/common/bin/wlst.sh
   ```

5. From the WLST prompt, connect to the Administration Server at port `7002`. Specify the WebLogic Server administrative credentials that you specified when you created the service instance.

   ```
   > connect('username', 'password', 't3s://adminServerPublicIP:7002')
   ```

   For example:

   ```
   > connect('weblogic', 'password', '203.0.113.10:7002')
   ```

   If you are accessing the Administration Server node via an SSH tunnel, connect to `localhost:9001` instead.

You can now use WLST to execute additional commands. Refer to the *WLST Command and Variable Reference* in one of the following publications:

- WLST Command Reference for WebLogic Server (12.2.1.4)
- WLST Command Reference for WebLogic Server (12.2.1.3)
- WebLogic Scripting Tool Command Reference (10.3.6)

Shut Down and Start Server Processes

You can shut down and start WebLogic Server processes, including the Administration Server, within your Oracle Java Cloud Service instance.

You can shut down WebLogic Server *processes* without stopping the nodes on which the servers are running. You might shut down a server process for routine maintenance or testing, or to address CPU and memory resource contention. This approach shuts down the server processes only, while other processes you might have running on the nodes continue to run.

Topics

- Use the WebLogic Server Administration Console to Shut Down Servers
- Use WLST Commands to Start the Administration Server
- Use the WebLogic Server Administration Console to Start Managed Servers

Use the WebLogic Server Administration Console to Shut Down Servers

Use the WebLogic Server Administration Console to shut down the server processes for your Oracle Java Cloud Service instance. A WebLogic Server domain includes an administration server and one or more managed servers.

To use the WebLogic Server Console to shut down servers:

1. Access the Oracle Java Cloud Service console.
2. Click **Manage this instance** ☰ for the desired service instance, and then select **Open WebLogic Server Administration Console**.
3. Enter the WebLogic Server administrator user name and password you provided when you created the service instance.
4. From the WebLogic Server Administration Console, under Domain Structure, expand **Environment**.
5. Click **Servers**.

   From the **Configuration** tab of the Summary of Servers page, notice that the state of the Administration Server and Managed Servers is RUNNING.
6. Click the **Control** tab.
7. Click the check box to the left of each server.
8. Click **Shutdown**, and then select **Force Shutdown Now** or **When Work Completes**.

   When you shut down the administration server, a message warns you that the browser session will end.

Use WLST Commands to Start the Administration Server

You start the Administration Server in an Oracle Java Cloud Service instance through the Node Manager by using WLST commands.

Prior to running WLST, identify the public IP address of the Administration Server node and connect to it with a secure shell (SSH) client.

1. After connecting to the node, switch to the `oracle` user:

   ```
   sudo su - oracle
   ```

2. Verify that the Node Manager is running:

   ```
   ps -ef | grep NodeManager
   ```

   You should receive messages showing that the Node Manager is running.

3. Change the directory to the `bin` folder in `DOMAIN_HOME`.

   ```
   cd $DOMAIN_HOME/bin
   ```

   For example, `/u01/data/domains/OurServi_domain/bin`

4. Set up the environment.

   ```
   . ./setDomainEnv.sh
   ```

   You must use the `.` to ensure that the environment variables are set in the current shell.

5. Start WLST:

   ```
   $COMMON_COMPONENTS_HOME/common/bin/wlst.sh
   ```

6. To connect to the Node Manager, use the WLST `nmConnect` command:

   ```
   nmConnect
   ('username','password','host','nmPort','domainName','domainDir','nmType')
   ```

Parameter	Description	Example
`username`	WebLogic Server username you specified when you created the service instance.	
`password`	WebLogic Server password you specified when you created the service instance.	
`host`	The host name of the Node Manager. This is typically of the format *<instanceName>*`-wls-1`.	`ourserviceinstance-wls-1`
`nmPort`	Port number of the node manager. .	`5556`

Parameter	Description	Example
`domainName`	Name of the domain. You can find the domain name on the Oracle Java Cloud Service Instance Overview page.	`OurServi_domain`
`domainDir`	Path to the domain. In Oracle Java Cloud Service, the domain directory is `/u01/data/domains/<domainName>`.	`/u01/data/domains/OurServi_domain`
`nmType`	Use `SSL` for Java-based SSL implementation.	`SSL`

For example:

```
nmConnect ('weblogic','password','ourserviceinstance-
wls-1','5556','OurServi_domain','/u01/data/domains/
OurServi_domain','SSL')
```

7. Use the `nmStart` command to start the Administration Server:

   ```
   nmStart (server_name)
   ```

   For example:

   ```
   nmStart ('OurServi_adminserver')
   ```

8. Exit WLST:

   ```
   exit()
   ```

9. Exit the `oracle` session.

   ```
   exit
   ```

10. Exit the command window:

    ```
    exit
    ```

11. Access the Oracle Java Cloud Service console.
12. Click **Manage this instance** ☰ for the desired service instance, and then select **Open WebLogic Server Administration Console**.
13. When the console login page appears, enter the WebLogic Server administrator user name and password you provided when you created the service instance.
14. From the WebLogic Server Administration Console, under **Domain Structure**, expand **Environment**.
15. Click **Servers**.
16. On the **Configuration** page, check that the Administration Server state is RUNNING.

For more information about `nmConnect` parameters, see:

- For Oracle Fusion Middleware 12.2.1.4: nmConnect in *WLST Command Reference for WebLogic Server*
- For Oracle Fusion Middleware 12.2.1.3: nmConnect in *WLST Command Reference for WebLogic Server*
- For Oracle Fusion Middleware 11.1.1.7: nmConnect in *WLST Scripting Tool Command Reference*

Use the WebLogic Server Administration Console to Start Managed Servers

After the Administration Server is running in your Oracle Java Cloud Service instance, you can start the Managed Servers by using the WebLogic Server Administration Console.

1. Access the Oracle Java Cloud Service console.
2. Click **Manage this instance** ☰ for the desired service instance, and then select **Open WebLogic Server Administration Console**.
3. Enter the WebLogic Server administrator user name and password that you provided when you created the service instance.
4. Under **Domain Structure**, expand **Environment**.
5. Click **Servers**.

   On the Configuration page, notice that the Administration Server state is RUNNING, and the Managed Server state is SHUTDOWN.
6. Click the **Control** tab.
7. Click the check box to the left of each Managed Server name.
8. Click **Start**.
9. When prompted for confirmation, click **Yes**.

   The server state changes to STARTING.
10. Periodically click the **Refresh** icon until the server state changes to RUNNING.

About JVM Heap Settings

When you provision an Oracle Java Cloud Service instance and specify compute shapes, the Java Virtual Machine (JVM) heap size for WebLogic Server and Load Balancer processes is determined automatically.

Topics:

- Default Heap Sizes
- Custom Heap Sizes

Default Heap Sizes

The compute shape you select for a WebLogic Server cluster determines the availability of RAM on the nodes in this cluster, and the amount of available RAM is used to determine the preset heap size for the JVM processes running on the nodes.

Different compute shapes are available for Oracle Java Cloud Service instances in Oracle Cloud Infrastructure and Oracle Cloud Infrastructure Classic regions.

For the default heap settings used in Oracle Java Cloud Service—Coherence instances, see About Cache Capacity for a Service Instance.

The following table shows the Oracle Java Cloud Service JVM heap size settings for some of the available compute shapes.

Compute Shape	Min Heap Size	Max Heap Size	Configured Garbage Collector
OC3	256 MB	2 GB	default
OC4	256 MB	10 GB	Garbage First (-XX:+UseG1GC)
OC5	256 MB	24 GB	Garbage First (-XX:+UseG1GC)
OC6	256 MB	24 GB	Garbage First (-XX:+UseG1GC)
OC7	256 MB	24 GB	Garbage First (-XX:+UseG1GC)
OC1M	256 MB	10 GB	Garbage First (-XX:+UseG1GC)
OC2M	256 MB	24 GB	Garbage First (-XX:+UseG1GC)
OC3M	256 MB	24 GB	Garbage First (-XX:+UseG1GC)
OC4M	256 MB	24 GB	Garbage First (-XX:+UseG1GC)
OC5M	256 MB	24 GB	Garbage First (-XX:+UseG1GC)

Custom Heap Sizes

If you create an Oracle Java Cloud Service instance by using the REST API or CLI, you can specify a custom heap size for the JVMs in the service instance.

You cannot specify a custom heap size when creating a service instance with the web console. See Create a Service Instance in *REST API for Oracle Java Cloud Service*.

After provisioning a service instance, you can also change the heap size by using the WebLogic Server Administration Console. Refer to one of the following publications:

- Increasing the heap size for a managed server in *Administration Console Online Help (12.2.1.4)*
- Increasing the heap size for a managed server in *Administration Console Online Help (12.2.1.3)*
- Set Java options for servers started by Node Manager in *Administration Console Online Help (10.3.6)* (specify the Java option to increase the heap size; for example: `-Xmx3g`)

The heap size is also set when you choose a shape for the load balancer. You cannot change the heap size for the load balancer.

About Data Sources

The Oracle WebLogic Server domain in an Oracle Java Cloud Service instance is automatically configured with several JDBC data sources. You can customize these data sources and also create additional ones.

Java Database Connectivity (JDBC) data sources in WebLogic Server provide database access and database connection management. Each data source contains a pool of reusable database connections that are created when the data source is created and at server startup. Applications reserve a database connection from the data source and then return it back to the pool when finished using it.

Topics

- Predefined Data Sources
- Data Source Types
- Custom Data Sources
- Data Source Network Connectivity

Predefined Data Sources

When you create an Oracle Java Cloud Service instance you provide the locations of one or more databases in Oracle Cloud.

These databases are used for different purposes:

- Infrastructure Schema Database — Oracle Java Cloud Service provisions this Oracle Database or Pluggable Database (PDB) with the required Oracle Fusion Middleware schema and data. By default, this schema is automatically deleted when you delete the service instance.
- Application Schema Database (optional; up to 4) — You can provision this Oracle Database or PDB with any business data that your applications require, and using any standard Oracle Database tools. This feature is supported only if you associate the service instance with an Oracle Database Cloud Service deployment. It is not available for other database services.

During the service instance creation process, Oracle Java Cloud Service creates JDBC data sources in the WebLogic Server domain in order to provide connectivity to these databases.

Data Source Types

WebLogic Server provides several types of data sources.

- *Generic* — Connects to a single database node.
- *Multi* — Provides load balancing and failover across a group of Generic data sources.
- *GridLink* — Provides dynamic load balancing and failover across an Oracle Database RAC cluster, and also receives notifications from the cluster when nodes are added or removed. This type of data source can only be used with Oracle Database.

The type of data source that Oracle Java Cloud Service creates in your domain depends on the following:

- The Software Edition of your service instance.
- Whether or not the selected database is RAC-enabled. See Using Oracle Real Application Clusters (RAC) in Oracle Database Cloud Service.

Software Edition	RAC Database?	Data Source Type
Standard	No	Generic
Standard	Yes	Generic

Software Edition	RAC Database?	Data Source Type
Enterprise	No	Generic
Enterprise	Yes	Multi
Enterprise with Coherence (Suite)	No	Generic
Enterprise with Coherence (Suite)	Yes	GridLink See Configure an Oracle Java Cloud Service Instance for an Oracle RAC Database.

Custom Data Sources

You can modify and tune the existing data sources that were created by Oracle Java Cloud Service, and you can also create additional data sources in your WebLogic Server domain to provide connectivity to other databases or PDBs.

Create and configure data sources using standard tools like the WebLogic Server Administration Console, Fusion Middleware Control, or WebLogic Scripting Tool (WLST).

For more information on the different types of data sources and how to configure them, refer to *Configuring JDBC Data Sources* in one of the following publications:

- Administering JDBC Data Sources for Oracle WebLogic Server (12.2.1.3)
- Configuring and Managing JDBC Data Sources for Oracle WebLogic Server (10.3.6)

Data Source Network Connectivity

When you create an Oracle Java Cloud Service instance, it is automatically configured with the necessary security rules to enable connectivity to the specified databases.

These rules allow the WebLogic Servers in your service instance to connect to your databases through the Oracle Cloud network. If you create additional data sources in your service instance, and these data sources connect to databases that are running outside of the nodes in your Oracle Java Cloud Service instance, then you must create custom security rules to explicitly permit this database network traffic. Security rules are not required if the database to which you are connecting is running on the nodes in your Oracle Java Cloud Service instance.

- For service instances on Oracle Cloud Infrastructure regions, see Security Lists in the Oracle Cloud Infrastructure documentation.
- For service instances on Oracle Cloud Infrastructure Classic regions, see Create an Access Rule.

Manage Associations for a Service Instance

An Oracle Java Cloud Service instance is associated with one or more services in Oracle Cloud, including databases, cloud stacks, and other Oracle Java Cloud Service instances.
Some examples of associations are listed below:

- An Oracle Java Cloud Service instance is connected to an Oracle Database Cloud Service deployment to access the Oracle Required Schemas.
- An Oracle Java Cloud Service instance is connected to an Oracle Database Cloud Service deployment to access your application schemas.
- An Oracle Java Cloud Service instance was cloned from a snapshot that was taken of another Oracle Java Cloud Service instance.
- An Oracle Java Cloud Service instance was created as part of a cloud stack in Oracle Cloud Stack.

You can view details about all associations for a service instance, and quickly navigate to the web console for a related service. You can also update the association between an existing Oracle Java Cloud Service instance and its infrastructure database (the database that is hosting the Oracle Required Schema). You cannot directly modify any of the other associations for a service instance.

Topics:

- View Association Details
- Associate an Oracle Java Cloud Service Instance with a Different Database
- Change the Database Schema Password for an Oracle Java Cloud Service Instance

View Association Details

Learn about and access other Oracle Cloud services with which this Oracle Java Cloud Service instance is associated.

1. Access your service console.
2. Click the name of the service instance whose associations you want to view.
3. At the bottom of the Overview page, expand **Associations**.
4. Click **Manage Association** ☰ for an existing association, and then select **View Details**.
5. After viewing the association's details, click **OK**.
6. Click the **Instance Name** link for the association to access the service's Overview page.
7. To return back to the Overview page for your Oracle Java Cloud Service instance, click the name of your service instance from the navigation links at the top of the page.

Associate an Oracle Java Cloud Service Instance with a Different Database

This topic does not apply to Oracle Cloud Infrastructure. Identify the Cloud Infrastructure Used by a Service Instance.

Update an existing Oracle Java Cloud Service instance and associate it with a different Oracle Database Cloud Service deployment.

You might associate your service instance with a different database if:

- Your service instance requires a database with a different version, edition, or capacity.
- The current database has crashed or become corrupted, and can't be restarted.

Note:

- This operation is not supported for service instances running on Oracle Cloud Infrastructure regions.
- This operation is not supported for service instances created before the Oracle Java Cloud Service release 17.4.1 (October 2017).
- Both the original and the target Oracle Database Cloud Service deployment must be of type **Single Instance**. All other database types are not supported, including *Database Clustering with RAC* and *Single Instance with Data Guard Standby*.
- Oracle Database Exadata Cloud Service is not supported.

You can only modify the association to the **Infrastructure** database for a service instance (the database that hosts the required Oracle WebLogic Server schemas). You cannot directly modify associations to other *Application* databases for a service instance. However, if the service instance’s Infrastructure database and Application database are associated with the same Oracle Database Cloud Service deployment, modifying the association for the Infrastructure database also modifies the association for the Application database.

You must enable backups on the Oracle Java Cloud Service instance prior to associating it with a different database deployment.

Before associating a service instance with a different infrastructure database, you must migrate the existing schema and data from the current database to the new database. Refer to these topics in *Administering Oracle Database Cloud Service*:

- Creating a Database Deployment Using a Cloud Backup
- Creating a Clone Database Deployment from a Snapshot
- Choosing a Migration Method

1. Access your service console.
2. Click the name of the service instance whose database association you want to change.
3. At the bottom of the Overview page, expand **Associations**.
4. From the list of associations for this service instance, identity the association with these characteristics:
   - **Service Type** - Oracle Database Cloud Service
   - **Type** - Depends On
5. Click **Manage Association** ☰ for this association, and then select **Reassociate Database**.

   This menu option is only available for the Infrastructure database association. It it not available for Application databases.
6. For **Database Instance Name**, select the existing Oracle Database Cloud Service deployment to which you migrated the required Oracle Java Cloud Service schema and data.

The list only includes a database deployment if it meets the following criteria:

- Is in an active state and not currently in the process of being provisioned
- Is not configured with a **Backup Destination** set to `None`

7. For **Association Description**, describe the relationship between this Oracle Java Cloud Service instance and the selected database deployment.
8. Enter the **Database Administrator Username** and **Password** of the database administrator that Oracle Java Cloud Service will use to connect to the selected database deployment.

   If the service instance is running Oracle WebLogic Server 12c (12.2.1.4.0 version), you can specify the user `SYS` or any user that has been granted the `SYSDBA` privilege.
9. Click **Reassociate**.

A new association is added to the **Associations** section of the Overview page. You can monitor the operation's progress from this page or from the **Activity** page. After the operation completes successfully, the former database association is removed.

Prior to performing this operation, Oracle Java Cloud Service takes a backup of your service instance. If the operation fails, the service instance is automatically restored from this backup.

Change the Database Schema Password for an Oracle Java Cloud Service Instance

Update the password used by an Oracle Java Cloud Service instance to access the Oracle schemas in the infrastructure database.

You might change the password for the Oracle schemas in order to meet Oracle security policies, corporate security policies or government regulations, or in response to a perceived security threat. By default, this password is set to expire 180 days after your service instance was created.

You can only use Oracle Java Cloud Service to change the password for the Oracle Required Schemas found in the **Infrastructure Schema** database for a service instance. To change the password for schemas hosted in an *Application* database in your service instance, you must directly modify the configuration of both the database and your WebLogic Server domain.

Topics

- Change the Schema Password with the Console
- Change the Schema Password Manually

Change the Schema Password with the Console

Use Oracle Java Cloud Service to change the Oracle schema password in the Oracle Database Cloud Service deployment, and to also update your service instance to use the new password.

You cannot use the console to change the schema password if your service instance uses Oracle Cloud Infrastructure Database or Oracle Autonomous Database (Oracle Autonomous Transaction Processing). You must use the REST API. See Change the Database Schema Password in *REST API for Oracle Java Cloud Service*.

You cannot use Oracle Java Cloud Service to automatically change the schema password if your service instance was created before November 2017. See Change the Schema Password Manually.

1. Access your service console.
2. Click the name of the service instance whose schema password you want to change.
3. At the bottom of the Overview page, expand **Associations**.
4. From the list of associations for this service instance, identity the association with these characteristics:
   - **Service Type** - Oracle Database Cloud Service
   - **Type** - Depends On
5. Click **Manage Association** ☰ for this association, and then select **Update Database Credentials**.

   This menu option is only available for the Infrastructure database association. It it not available for Application databases.
6. Enter the **Database Administrator Username** and **Database Administrator Password** of the system administrator for the selected database deployment.
7. For **New Schema Password**, enter a new password for the Oracle schemas in the selected database deployment.

   The password must start with a letter, be between 8 and 30 characters long, and contain at least one number. The password can optionally include the special characters: $, #, _.
8. Click **Update**.

You can monitor the operation's progress from this page or from the **Activity** page. Oracle Java Cloud Service updates the database credentials, the WebLogic Server domain configuration, and the bootstrap credentials.

Change the Schema Password Manually

If you want to change the password for the Oracle database schemas used by your Oracle Java Cloud Service instance, and your service instance was created prior to November 2017, then you must manually update the database and the service instance to use the new database password.

1. Update the password for each Oracle Java Cloud Service schema in the database.
   a. Connect to your database using a secure shell (SSH) client.
   b. Switch to the `oracle` user.

   ```
   sudo su oracle
   ```

   c. Connect to the database using `sqlplus`.

   ```
   sqlplus / as sysdba
   ```

d. If your database version is 12c, set the name of the pluggable database (PDB).

```
alter session set container=PDB_name;
```

Use the PDB name that you provided during the creation of your Oracle Java Cloud Service instance. The default is `PDB1`.

```
alter session set container=PDB1;
```

e. List all the schemas (users) in the database.

```
select username from dba_users;
```

f. Locate the schema `schema_prefix_IAU`. Note the generated schema prefix.

For example: **`SP737755846`**`_IAU`

g. Unlock and change the password for the following users.

Oracle WebLogic Server 12.2.1.x
*schema_prefix*_IAU
*schema_prefix*_IAU_APPEND
*schema_prefix*_IAU_VIEWER
*schema_prefix*_MDS
*schema_prefix*_OPSS
*schema_prefix*_STB
*schema_prefix*_UMS

```
alter user schema_prefix_user account unlock;
alter user schema_prefix_user identified by new_password;
```

For example:

```
alter user SP737755846_IAU account unlock;
alter user SP737755846_IAU identified by new_password;
```

The password must start with a letter, be between 8 and 30 characters long, and contain at least one number. The password can optionally include the special characters: `$`, `#`, `_`.

2. If your Administration Server is running, use the Administration Console to update the data source passwords.

   If your Administration Server is not running, **skip to step 3**.

   a. Access the Oracle Java Cloud Service console.

   b. Click **Manage this instance** ☰ for your service instance, and then select **Open WebLogic Server Administration Console**.

   c. Log in to the Administration Console.

   d. Click **Lock & Edit**.

e. From the Domain Structure panel, expand **Services**, and then click **Data Sources**.

f. Click **mds-owsm**.

g. Click the **Connection Pool** tab.

h. Update the **Password** and **Confirm Password** fields.

i. Click **Save**.

j. **Repeat from step e** to update the password for the other data sources below.

Oracle WebLogic Server 12c
`mds-owsm`
`LocalSvcTblDataSource`
`opss-audit-DBDS`
`opss-audit-viewDS`
`opss-data-source`

k. Click **Activate Changes**.

l. From the Domain Structure panel, expand **Environment**, and then click **Servers**.

m. Click the **Control** tab.

n. Select the check box for every server. Click **Shutdown**, and then select **Force Shutdown Now**.

o. When prompted for confirmation, click **Yes**.

p. **Skip to step 4**.

3. If your Administration Server is not running or the Administration Console is not accessible, update the domain’s configuration files.

a. Connect to the Administration Server node in your service instance using SSH.

b. Switch to the `oracle` user.

```
sudo su oracle
```

c. Navigate to the domain home directory.

```
cd /u01/data/domains/domain_name
```

d. Stop any WebLogic Server processes if they are running.

```
ps -ef | grep weblogic.Server
kill -9 process_id
```

Do not stop the Node Manager process.

e. Encrypt your new schema password using the `weblogic.security.Encrypt` utility.

```
source bin/setDomainEnv.sh
java weblogic.security.Encrypt
```

When prompted, enter the new password.

f. Copy the encrypted password.

g. Navigate to the directory that contains your domain’s data source configuration files.

```
cd config/jdbc
```

h. Edit the following files and update the `password-encrypted` element with the new encrypted value.

Oracle WebLogic Server 12c
`LocalSvcTblDataSource-jdbc.xml`
`opss-auditview-jdbc.xml`
`mds-owsm-jdbc.xml`
`opss-datasource-jdbc.xml`
`opss-audit-jdbc.xml`

```
<password-encrypted>encrypted_password</password-encrypted>
```

4. Use the WebLogic Scripting Tool (WLST) to update the bootstrap credentials for the OPSS database schema.

a. Connect to the Administration Server node in your service instance using SSH.

b. Switch to the `oracle` user.

```
sudo su oracle
```

c. Identify your domain’s name.

```
ls /u01/data/domains
```

d. Start a WLST prompt.

```
/u01/app/oracle/middleware/oracle_common/common/bin/wlst.sh
```

e. Run the `modifyBootStrapCredential` command. Specify the full path to the `jps-config-jse.xml` file, the OPSS schema name, and your new database password.

```
modifyBootStrapCredential(jpsConfigFile='/u01/data/domains/
domain_name/config/fmwconfig/jps-config-
jse.xml',username='schema_prefix_OPSS',password='new_password')
```

f. Exit WLST.

```
exit()
```

5. Start the servers.

    a. From the Administration Server node, identify its host name.

    ```
    hostname
    ```

    b. Identify the name of the Administration Server.

    ```
    ls /u01/data/domains/domain_name/servers
    ```

    The server’s name ends with the text `adminserver`.

    c. View the `nodemanager.properties` file to determine the Node Manager’s listen port number.

    For Oracle WebLogic Server 12c: `/u01/data/domains/*domain_name*/nodemanager/nodemanager.properties`

    d. Start a WLST prompt.

    ```
    /u01/app/oracle/middleware/oracle_common/common/bin/wlst.sh
    ```

    e. Connect to the Node Manager.

    ```
    nmConnect('weblogic_username','weblogic_password','hostname','nm_
    port','domain_name','/u01/data/domains/domain_name','ssl')
    ```

    f. Start the Administration Server.

    ```
    nmStart('admin_server_name')
    ```

    g. Exit WLST.

    ```
    exit()
    ```

    h. Access the Oracle Java Cloud Service console.

    i. Click **Manage this instance** ☰ for your service instance, and then select **Open WebLogic Server Administration Console**.

    j. Log in to the Administration Console.

    k. From the Domain Structure panel, expand **Environment**, and then click **Servers**.

    l. Click the **Control** tab.

    m. Select the check box for every managed server. Click **Start**.

    n. When prompted for confirmation, click **Yes**.

Connect an Oracle Java Cloud Service Instance to an Application Database

After you provision the service instance, you can create new data sources that enable you to connect to either an Oracle Autonomous Database (Oracle Autonomous Transaction Processing) or an Oracle Cloud Infrastructure database.

Topics:

- Connect to an Oracle Autonomous Database
- Connect to an Oracle Cloud Infrastructure Database

Connect to an Oracle Autonomous Database

You can create an Oracle Autonomous Database (Oracle Autonomous Transaction Processing) using the Oracle Cloud Infrastructure Console, then provision an Oracle Java Cloud Service instance with the Oracle Autonomous Database for the schema infrastructure database, but not the application database. After you provision the Oracle Java Cloud Service, you can configure an additional data source that enables you to connect the provisioned service instance to an Oracle Autonomous Database.

Topics

- Download the Oracle Autonomous Database Wallet
- Copy and Unpack the Wallet
- Create a Data Source in the WebLogic Server Console

Download the Oracle Autonomous Database Wallet

The first step in connecting an Oracle Autonomous Database (Oracle Autonomous Transaction Processing) is to download the database's wallet file, which contains client credentials.

Oracle wallet files are downloaded from Oracle Autonomous Database by a service administrator. If you are not an Oracle Autonomous Database administrator, ask your administrator to provide you with the wallet file.

You must first create an Oracle Java Cloud Service instance and an Oracle Autonomous Database in the same region.

1. Sign in to access the Oracle Cloud Infrastructure Console, if it's not open already. If you are currently in the Oracle Java Cloud Service Console, select **Compute** from the Navigation menu and sign in.
2. On the Oracle Cloud Infrastructure Console, select the region in which your Oracle Autonomous Database and Oracle Java Cloud Service instance are located.
3. Select **Autonomous Transaction Processing** from the Oracle Database section of the Navigation menu.
4. Select the **Compartment** that contains your Oracle Autonomous Database.
5. Click the name of the Oracle Autonomous Database you want to connect to your service instance.

6. On the Oracle Autonomous Transaction Processing details page, click **DB Connection**.
7. On the Database Connection page, click **Download**.
8. In the Download Wallet dialog, enter a wallet password in the **Password** field and confirm the password in the **Confirm Password** field.
9. Click **Download** to save the wallet zip file.

Copy and Unpack the Wallet

After you download the wallet, copy and unpack the wallet to the Administration Server node. If your service instance contains multiple nodes, copy the wallet to each node.

Performing the following steps ensures that each node has client credentials.

First, make a note of the IP address of your Oracle Java Cloud Service Administration Server's IP address. You can locate the IP address on the Overview page for your Oracle Java Cloud Service instance.

1. Use `scp` to copy the wallet file to the service instance's Administration Server node.

   ```
   scp -i ~/<private_key> /<example_zip_directory>/
   wallet_<ATP_db_name>.zip opc@<admin_server_ip>:/tmp
   ```

2. Use `ssh` to access the node where you copied the zip file.

   ```
   ssh -i ~/<private_key> opc@<admin_server_ip>
   ```

3. Change the ownership of the wallet file to `oracle`.

   ```
   sudo chown oracle:oracle /tmp/wallet_<ATP_db_name>.zip
   ```

4. Change to user `oracle`.

   ```
   sudo su oracle
   ```

5. Create a directory where you can copy the wallet zip file. Oracle recommends placing the wallet in the domain home directory.

   ```
   mkdir /u01/data/domains/<service_instance_domain>/config/
   <example_directory>
   ```

6. Change directories to the directory you just created.

   ```
   cd /u01/data/domains/<service_instance_domain>/config/
   <example_directory>
   ```

7. Unpack the wallet zip file.

   ```
   unzip /tmp/wallet_<ATP_db_name>.zip
   ```

8. Repeat these steps for all other Managed Server nodes.

Create a Data Source in the WebLogic Server Console

After you download the wallet file and copy it to all servers in the instance, you can create a data source in the WebLogic Server Console that enables you to connect your Oracle Java Cloud Service instance to your Oracle Autonomous Database (Oracle Autonomous Transaction Processing).

1. Access the Oracle Java Cloud Service console.
2. From the Instances page, click the **Manage this Instance** ☰ icon next to the service instance, and then select **Open WebLogic Server Administration Console**.
3. Sign in to the WebLogic console as the WebLogic Administrator. Enter the same WebLogic Administrator credentials that you specified when you created the service instance.

   The Oracle WebLogic Server Console is displayed.
4. In the Change Center box at the top left corner of the page, click **Lock & Edit**.
5. In the Domain Structure box, expand **Services** (by clicking the + next to it) and click **Data Sources**. The Summary of JDBC Data Sources page is displayed.
6. Click **New**, and then select **Generic Data Source**.
7. In the first page of the **Create a New JDBC Data Source** wizard, enter any values for **Name**:
   a. In the **Name** field, enter any name for your data source.
   b. In the **JNDI Name** field, enter any name.
   c. In the **Database Type** drop-down list, accept default type **Oracle**.
   d. Click **Next**.
8. On the second page, accept all options and click **Next**.
9. On the third page, do the following:
   a. In the **Database Name** field, enter the name of your Oracle Autonomous Database.
   b. In the **Host Name** field, enter the host name for the Oracle Autonomous Database. If you don't know the host name, open the `tsnames.ora` wallet file, and locate the host name there.
   c. In the **Port** field, enter `1522`.
   d. In the **Database User Name** field, enter `ADMIN`.
   e. Enter a password, and then confirm it.
   f. Click **Next**.
10. On the fourth page, update the URL.
    a. View file `tnsnames.ora` in the wallet zip file.
    b. Select an appropriate database service, which uses the format `<database_name>_<priority>`. For example:

```
db1_high = (description=
     (address=(protocol=tcps)(port=1522)(host=mydb.example.com))
(connect_data=(service_name=abcd1234_db1_high.mydb.example.com))
(security=(ssl_server_cert_dn=
    "CN=mydb.example.com,OU=Oracle BMCS US,O=Oracle
Corporation,L=Redwood City,ST=California,C=US"))    )
```

c. Copy the service description, which is all of the text after the equals (=) character:

```
(description= <service_description>)
```

d. Replace the current URL with the following one:

```
jdbc:oracle:thin:@<copied_service_description>
```

For example:

```
jdbc:oracle:thin:@(description=
    (address=(protocol=tcps)(port=1522)(host=mydb.example.com))
(connect_data=(service_name=abcd1234_db1_high.mydb.example.com))
(security=(ssl_server_cert_dn="CN=adwc.uscom-
east-1.oraclecloud.com,OU=Oracle
    BMCS US,O=Oracle Corporation,L=Redwood
City,ST=California,C=US")))
```

> **Note:**
>
> Ensure that all the URL text is all on the same line. The text in the `tnsnames.ora` file is *not* written on the same line, so exercise caution.

11. On the same page where you updated the URL, update the properties in the **Properties** field with the following information and click **Next**:

```
user=ADMIN
oracle.net.tns_admin=/u01/data/domains/
<location_of_unpacked_wallet_zip>/config/<example_directory>
oracle.net.ssl_version=1.2
javax.net.ssl.trustStore=/u01/data/domains/
<location_of_unpacked_wallet_zip>/config/<example_directory>/
truststore.jks
oracle.net.ssl_server_dn_match=true
javax.net.ssl.keyStoreType=JKS
javax.net.ssl.trustStoreType=JKS
javax.net.ssl.keyStore=/u01/data/domains/
<location_of_unpacked_wallet_zip>/config/<example_directory>/
keystore.jks
javax.net.ssl.keyStorePassword=<WalletPassword>
javax.net.ssl.trustStorePassword=<WalletPassword>
```

12. One the fifth page of the wizard, click **Test Configuration** to verify if a connection to the database can be established based on the information that you provided.
    - If the connection test fails, click **Back** and review the entries that you made for the data source and correct any errors. If there are no errors in the entries and the test still fails, make sure that your database is running.
    - If the message *Connection test succeeded* is displayed, click **Next**.

13. On the last page of the wizard, select **All Servers in the Cluster** in the **Select Targets** table and click **Finish**.
14. In the Change Center, click **Activate Changes**.

Connect to an Oracle Cloud Infrastructure Database

You can create an Oracle Cloud Infrastructure database using the Oracle Cloud Infrastructure Console, then provision an Oracle Java Cloud Service instance with the Oracle Cloud Infrastructure database for the schema infrastructure database, but not for the application database. After you provision the Oracle Java Cloud Service instance, you can configure an additional data source that enables you to connect the provisioned service instance to an Oracle Cloud Infrastructure database.

For details, see the Connect an Oracle Java Cloud Service Instance to an Oracle Cloud Infrastructure Database System tutorial.

> **Note:**
>
> In addition to creating a data source, you must also create a security list with the stateless ingress rule to allow the incoming traffic from the Oracle Java Cloud Service instance to the Oracle Cloud Infrastructure database.

Configure an Oracle Java Cloud Service Instance for an Oracle RAC Database

This topic does not apply to Oracle Cloud Infrastructure. Identify the Cloud Infrastructure Used by a Service Instance.

After creating an Oracle Java Cloud Service instance that's associated with an Oracle Real Application Cluster (RAC) database, take steps to further optimize the communication between your service instance and the database cluster.

> **Note:**
>
> This task is applicable only to service instances that meet all of the following criteria:
>
> - You created the service instance in an **Oracle Cloud Infrastructure Classic** region.
> - When you created the service instance, you selected the software edition **Enterprise Edition with Coherence** .
> - You associated the service instance with an Oracle Database Cloud Service deployment that is RAC-enabled.

Oracle Java Cloud Service provisions GridLink data sources in your Oracle WebLogic Server domain to connect to the selected Oracle Database cluster. GridLink provides dynamic load balancing and failover across the nodes in an Oracle Database cluster, and also receives notifications from the cluster when nodes are added or removed. To fully take advantage of these capabilities, Oracle recommends that you make the following modifications to your database and service instance:

- Create an Oracle Database service that supports Cluster Ready Services (CRS) and the Oracle Notification Service (ONS). These services monitor the status of resources in the database cluster and generate notifications when a status changes.
- Update WebLogic Server to connect to this new database service using the Single Client Access Name (SCAN) port 1521. The SCAN service provides WebLogic Server with the specific location of an available database node. WebLogic Server then connects to a specific database node using port 1522.
- If you did **not** assign an IP network to the database deployment, the database service redirects requests to the database nodes' public IP addresses. You must create an access rule for the database deployment that permits traffic from the WebLogic Server nodes to port 1522.

Follow these steps:

1. Create a cluster-enabled service in your existing Oracle Database Cloud Service deployment.
   a. Access the Oracle Java Cloud Service console.
   b. Click the name of the service instance that you want to update.
   c. At the bottom of the Overview page, expand **Associations**.
   d. Click the name of the Oracle Database Cloud Service deployment that's associated with this service instance.
   e. Place your cursor over the **Connect String** field.

      The name of the existing database service displays. For example, `SERVICE_NAME=PDB1.123456789.oraclecloud.internal`.
   f. Copy the name of the database service, and replace the first word with `myservice`.

      For example, **`myservice`**`.123456789.oraclecloud.internal`.

      In subsequent steps, this value will be referred to as *`NEW_SERVICE_NAME`*.
   g. Identify the **Public IP** for the first node in this database deployment.
   h. Connect to this IP address using a Secure Shell (SSH) client.
   i. Switch to the `oracle` user.

      ```
      sudo su - oracle
      ```
   j. Use `svrctl` to define a new database service.

      ```
      srvctl add service -db orcl -service NEW_SERVICE_NAME -preferred
      orcl1,orcl2 -pdb pdb1
      srvctl modify service -db orcl -service NEW_SERVICE_NAME -
      rlbgoal SERVICE_TIME -clbgoal SHORT -pdb pdb1
      srvctl start service -db orcl -service NEW_SERVICE_NAME
      ```
2. Update the boot database URLs in the WebLogic Server domain files.

a. Return to your service instance in the Oracle Java Cloud Service console.

> **Tip:**
>
> From the Oracle Database Cloud Service console, you can use the navigation links at the top of the page to return to your Oracle Java Cloud Service instance.

b. Identify the **Public IP** for the first node in this service instance.

c. Connect to this IP address using an SSH client.

d. Switch to the `oracle` user.

```
sudo su - oracle
```

e. Navigate to the directory `config/fmwconfig` within your domain configuration.

```
cd $DOMAIN_HOME/config/fmwconfig
```

f. Edit the file `jps-config.xml`.

g. Locate this line in the file.

```
<property name="jdbc.url" value="JDBC_URL"/>
```

h. Within the JDBC URL, insert the text `(LOAD_BALANCE=ON)` after the existing text `ADDRESS_LIST=`.

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(LOAD_BALANCE=ON)...
```

i. Within the JDBC URL, change port 1522 to port 1521.

```
jdbc:oracle:thin:...(PORT=1521)...
```

j. Within the JDBC URL, replace the value of `SERVICE_NAME` with your new database service name.

```
jdbc:oracle:thin:...(SERVICE_NAME=NEW_SERVICE_NAME...
```

k. Copy the updated JDBC URL. Then save your changes.

An example JDBC URL is shown below:

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(LOAD_BALANCE=ON)
(ADDRESS=(PROTOCOL=TCP)(HOST=MyDB-scan-int)(PORT=1521)))
(CONNECT_DATA=(SERVICE_NAME=myservice.123456789.oraclecloud.internal))
)
```

l. Edit the file `jps-config-jse.xml`.

m. Locate this line in the file.

```
<property name="jdbc.url" value="JDBC_URL"/>
```

   n. Replace the value with the updated JDBC URL. Then save your changes.

3. Update the default data sources in your WebLogic Server domain.

   a. Return to your service instance in the Oracle Java Cloud Service console.

   b. Click **Manage this instance** ☰, and then select **Open WebLogic Server Administration Console**.

   c. Log into the console as the WebLogic Server administrator.

   d. Click **Lock & Edit**.

   e. From the Domain Structure panel, expand **Services**, and then click **Data Sources**.

   f. Click **LocalSvcTblDataSource**.

   g. Click the **Connection Pool** tab.

   h. Replace the value of **URL** with the updated JDBC URL.

   i. Click **Save**.

   j. Click the **ONS** tab.

   k. Verify that these fields are configured correctly:

      - **Fan Enabled**: Yes
      - **ONS Nodes**: `INSTANCE_NAME-scan-int:6200`

   l. Repeat from **step e** to perform the same modification to all remaining GridLink data sources in the domain, including:

      - **mds-owsm**
      - **opss-audit-DBDS**
      - **opss-audit-viewDS**
      - **opss-data-source**

   m. Click **Activate Changes**.

4. Create access rules to permit traffic from your service instance to port 1522 on the database.

> **Note:**
>
> This step is required only if the Oracle Java Cloud Service instance and the Oracle Database Cloud Service deployment are **not** assigned to an IP network.

   a. Return to your service instance in the Oracle Java Cloud Service console.

   b. Identify the **Public IP** addresses for all nodes in this service instance that will access the RAC database.

   c. At the bottom of the page, expand **Associations**.

   d. Click the name of the Oracle Database Cloud Service deployment that's associated with this service instance.

   e. On the Overview page, click **Manage this instance** ☰, and then select **Access Rules**.
   f. On the Access Rules page, click **Create Rule**.
   g. For **Rule Name**, enter the name of your Oracle Java Cloud Service instance.
   h. For **Source**, select **<custom>**. Enter the public IP addresses of the nodes in your Oracle Java Cloud Service instance as a comma-separated list.

      For example: `203.0.113.13,203.0.113.14,203.0.113.15`

      > **Note:**
      >
      > You will need to create additional access rules each time you scale out your service instance. Alternatively, you can specify multiple IP addresses in CIDR format, such as `203.0.113.1/24`.

   i. For **Destination**, select **DB_1**
   j. For **Destination Port(s)**, enter `1522`.
   k. Accept the default **Protocol** (TCP).
   l. Click **Create**.
   m. Use the navigation links at the top of the page to return to your Oracle Java Cloud Service instance.
5. Restart the service instance.
   a. On the Overview page, click **Restart Service** ⟳.
   b. When prompted for confirmation, click **OK**.

Configure a Vanity Domain Name for a Service Instance

By using the load balancer as the front-end to your Oracle Java Cloud Service instance, you can quickly and easily associate a vanity Internet domain name to your application environment. Rather than accessing your applications using a public IP address, clients can use your custom domain name.

For example, end users might currently access your application with the URL `https://`**`203.0.113.10`**`/getstarted`. But instead you want them to use the vanity URL `https://`**`myapp.example.net`**`/getstarted`.

The steps to configure a vanity URL are different for service instances that use an Oracle-managed load balancer and for service instances that use a user-managed load balancer (Oracle Traffic Director).

Topics:

- Register a Custom Domain Name with a Service Provider
- Add a Vanity URL to an Oracle-Managed Load Balancer
- Delete a Vanity URL from an Oracle-Managed Load Balancer
- Update Oracle Traffic Director to Use a Custom Domain Name

Register a Custom Domain Name with a Service Provider

Third-party vendors enable you to register custom domain names that resolve to your Oracle Java Cloud Service load balancer.

To route external traffic to your load balancer, you must register the domain name (for example, `example.net`) with your Domain Name System (DNS) provider. Oracle Java Cloud Service does not register the domain name with your DNS provider.

Follow the instructions provided by your third-party registration vendor to resolve your custom domain name to the public IP address of the load balancer in your service instance.

To route traffic within Oracle Cloud Infrastructure to your load balancer using the custom domain name, you can register it as a zone using the Oracle Cloud Infrastructure DNS Service. See Managing DNS Service Zones in the Oracle Cloud Infrastructure documentation.

Add a Vanity URL to an Oracle-Managed Load Balancer

If your service instance has an Oracle-managed load balancer, use the Oracle Java Cloud Service console to add a vanity domain name to the load balancer.

When you create a service instance with a Oracle-managed load balancer, the load balancer is accessed from a default URL. You can personalize access to your applications by adding one or more vanity URLs to the load balancer.

Because users access the load balancer with the HTTPS protocol, a vanity URL requires a public certificate and a corresponding private key. The files containing your certificate, private key, and certificate chain must be in PEM format. The private key must not require a password.

If at a later date you need to replace the certificate with a different one, you can delete and recreate the vanity URL.

The following procedure does not apply to an instance of Oracle Cloud Infrastructure Load Balancing that you've created yourself and then configured to use the service instance nodes.

1. Navigate to the Overview page for the instance to which you want to add a vanity URL.
2. Locate and expand the Load Balancer section of the page.

   The load balancer endpoint is displayed.
3. From the **Actions** ☰ menu, select **Add Vanity URL**.
4. For **Public Certificate**, select the file containing your certificate for the custom domain name.
5. For **Private Key**, select the file containing the corresponding private key for the certificate.
6. If you have multiple certificates that form a single certification chain, you must include all relevant certificates in one file. For **Certificate Chain**, select the file containing your certification chain.
7. Enter your **Virtual Host**.

Do not include the protocol (`http://` or `https://`) or port number.
For example: `myapp.example.net`

8. Click **Add**.

The instance is in maintenance mode until the operation is completed.

Delete a Vanity URL from an Oracle-Managed Load Balancer

If your service instance has an Oracle-managed load balancer, use the Oracle Java Cloud Service console to delete a vanity URL from the load balancer.

The following procedure does not apply to an instance of Oracle Cloud Infrastructure Load Balancing that you've created yourself and then configured to use the service instance nodes.

1. Navigate to the Overview page for the instance from which you want to delete a vanity URL.
2. Locate and expand the Load Balancer section of the page.

   The load balancer endpoint is displayed.
3. Click **Delete Vanity URL** ✕ for the vanity URL you want to delete.
4. When prompted for confirmation, click **Delete**.

The instance is in maintenance mode until the operation is completed.

Update Oracle Traffic Director to Use a Custom Domain Name

If your Oracle Java Cloud Service instance uses Oracle Traffic Director as a load balancer, update the load balancer configuration to use the custom domain name.

By default, if you created your service instance in an Oracle Cloud Infrastructure Classic region, external access to the load balancer console is disabled for security purposes. If you did not enable console access while provisioning your service instance, see Enabling Console Access in an Oracle Java Cloud Service Instance.

1. Access the Oracle Java Cloud Service console.
2. Click ☰ for the desired service instance and select **Open Load Balancer Console**.
3. Log in to console using the credentials defined when provisioning your service instance.

   If you created your service instance using the Oracle Java Cloud Service console, the user name and password default to the Oracle WebLogic Server Administration Console user name and password.
4. Access the load balancer configuration (for example, `opc-config`):

   - If your service instance is running Oracle Traffic Director 12*c*, click the ☰ Target Navigation icon. Expand the **Traffic Director** folder and click the name of the Traffic Director configuration.
   - If your service instance is running Oracle Traffic Director 11*g*, click **Configurations** and then click the name of the Traffic Director configuration.
5. Navigate to the Virtual Server in this configuration (for example, `opc-config`):

   - If your service instance is running Oracle Traffic Director 12*c*, click **Traffic Director Configuration** and select **Administration > Virtual Servers**. Click the name of the virtual server.
   - If your service instance is running Oracle Traffic Director 11*g*, expand **Virtual Servers** in the navigation pane and click the name of the virtual server.

6. In the **General Settings** section edit the **Hosts** field. Enter the custom domain name (for example, `example.com`) that you registered.

   If there are multiple entries, separate each by a comma.

7. Activate your changes:
   - If your service instance is running Oracle Traffic Director 12*c*, click **Apply**.
   - If your service instance is running Oracle Traffic Director 11*g*, click **Deploy Changes**.

Configure a Custom URL for an Application Deployed to a Service Instance

You can use Oracle Traffic Director to define a custom "vanity" URL for an application deployed to an Oracle Java Cloud Service instance.

For example, if you have a shopping cart application deployed with the following context root: `/shopping-cart/widgets`, by default users would access the application using a URL that includes the context root details, such as `http://example.com/shopping-cart/widgets`. Let's say that you want to simplify this URL to `http://example.com`. You can accomplish this by modifying the load balancer configuration.

By default, if you created your service instance in an Oracle Cloud Infrastructure Classic region, external access to the load balancer console is disabled for security purposes. If you did not enable console access while provisioning your service instance, see Enabling Console Access in an Oracle Java Cloud Service Instance.

1. Access the Oracle Java Cloud Service Console.
2. Click ☰ for the desired service instance and select **Open Load Balancer Console**.
3. Log in to console using the credentials defined when provisioning your service instance.

   If you created your service instance using the Oracle Java Cloud Service console, the user name and password default to the Oracle WebLogic Server Administration Console user name and password.

4. Access the load balancer configuration (for example, `opc-config`):
   - If your service instance is running Oracle Traffic Director 12*c*, click the ☰ Target Navigation icon. Expand the **Traffic Director** folder and click the name of the Traffic Director configuration.
   - If your service instance is running Oracle Traffic Director 11*g*, click **Configurations** and then click the name of the Traffic Director configuration.
5. Navigate to the Virtual Server in this configuration (for example, `opc-config`):

   - If your service instance is running Oracle Traffic Director 12*c*, click **Traffic Director Configuration**, select **Administration,** and then **Virtual Servers.** Click the name of the virtual server.
   - If your service instance is running Oracle Traffic Director 11*g*, expand **Virtual Servers** in the navigation pane and click the name of the virtual server.

6. Click **Routes** and then click **default-route**.
7. Edit these fields in the **URI Mapping** section:
   a. Select the **Enabled** checkbox.
   b. For **From URI**, enter the URI that should be redirected. For example, enter `/`.
   c. For **Target URI**, enter the context root for the application. For example, enter `/shopping-cart/widgets`.
8. Activate your changes:
   - If your service instance is running Oracle Traffic Director 12*c*, click **Apply**.
   - If your service instance is running Oracle Traffic Director 11*g*, click **Deploy Changes**.

Configure a Custom URL for the WebLogic Server Console

Update the link that's used in the Oracle Java Cloud Service console to provide access to the Oracle WebLogic Server Administration Console for a service instance.

For example, you might want WebLogic Server administrators to access the console with a vanity Internet domain name like `https://`**`cloud.mycompany.com`**`/console`. Third-party vendors enable you to register a custom domain name and map it to the public IP address of Administration Server or load balancer in your service instance.

This operation is available only from the REST API. See Update a Service Instance Configuration in *REST API for Oracle Java Cloud Service*.

Configure a Custom URL for the Sample Application

Update the link that's used in the Oracle Java Cloud Service console to provide access to the sample application for a service instance.

If you specified a custom **Weblogic Cluster Path Prefix** for the first cluster in a service instance, then the **Open Sample Application** link in the Oracle Java Cloud Service console will not work until you update the service instance. For example, if you set the cluster path prefix to `/mycluster` when you created a service instance, then update the service instance and set the sample application URL to `https://`*`public_ip`*`:`*`port`*`/mycluster/sample-app`.

This operation is available only from the Oracle Java Cloud Service REST API.

1. Log into the WebLogic Server Administration Console for your service instance.
2. Click **Lock & Edit**.
3. Click **Deployments**.
4. Click **sample-app**.
5. Click the **Configuration** tab.
6. Update the **Context Root**.

   Format:

```
/prefix/sample-app
```

Example:

```
/mycluster/sample-app
```

7. Click **Save**.
8. Click **Activate Changes**.
9. Use the REST API to update the sample application URL for the Oracle Java Cloud Service console.

   See Update a Service Instance Configuration in *REST API for Oracle Java Cloud Service*.
   A sample payload is shown below:

   ```
   {
      "SAMPLE_ROOT" : "https://192.0.2.10:443/mycluster/sample-app"
   }
   ```

Monitor Applications with Oracle Java Flight Recorder and Oracle Java Mission Control

You can use Oracle's commercial profiling tools, Oracle Java Flight Recorder and Oracle Java Mission Control, to monitor the performance of applications deployed on Oracle Java Cloud Service.

Oracle Java Flight Recorder and Oracle Java Mission Control collect detailed runtime information so that you can analyze incidents after they occur. Oracle Java Flight Recorder, available in Oracle HotSpot Java Virtual Machine (JVM), is a performance monitoring and profiling tool that records diagnostic information on a continuous basis, making it always available, even in the wake of catastrophic failure such as a system crash.

Oracle Java Mission Control enables you to monitor and manage Java applications without introducing the performance overhead that is normally associated with these types of tools. It includes the Oracle Java Flight Recorder user interface, which allows users who are running an Oracle Java Flight Recorder-compliant version of Oracle's HotSpot to view JVM recordings, current recording settings, and runtime parameters. The Oracle Java Flight Recorder interface includes the Events Type View. The Events Type View gives you direct access to event information that is recorded in the .jfr file, such as event producers and types, event logging and graphing, event by thread, event stack traces, and event histograms.

Basic Workflow for Profiling Applications with Oracle Java Flight Recorder and Oracle Java Mission Control

Monitoring applications with Oracle Java Flight Recorder and Oracle Java Mission Control includes the following steps:

1. Enable Oracle Java Flight Recorder in your WebLogic Server domain.
2. Obtain the flight recording by generating a diagnostic image capture.
3. Analyze the recording with the Oracle Java Flight Recorder user interface.

Administration Best Practices

Follow these best practices to ensure that your Oracle Java Cloud Service instances work as expected and remain manageable.

> **Caution:**
>
> Oracle supports customizing the default configuration of the operating system, WebLogic Server, Coherence, Database, and Traffic Director in an Oracle Java Cloud Service instance. While most modifications to these components are acceptable, some changes may cause certain management features to fail or not operate properly, such as backups, patching, or scaling. Use caution, especially when performing root-level or system-level changes on the nodes in your service instance.

Best Practice	More Information
Do not create additional product installations.	Use only the default Oracle WebLogic Server product installation that is provisioned when a service instance is created. If you manually create other installations in your service instance Oracle Java Cloud Service will not be aware of them. Consequently, you will not be able to use Oracle Java Cloud Service to manage, scale, backup or patch these product installations. Similarly, do not manually install Oracle Traffic Director or Oracle Coherence.
Do not create additional WebLogic Server domains.	Use only the default Oracle WebLogic Server domain that is provisioned when a service instance is created. If you manually add domains to the service instance Oracle Java Cloud Service will not be aware of them. Consequently, you will not be able to use Oracle Java Cloud Service to manage, scale, backup or patch these domains.
Ensure unique WebLogic Server domain names.	If you plan to configure cross-domain communication between multiple Oracle Java Cloud Service instances, the Oracle WebLogic Server domains and their associated resources must have unique names. To accomplish this, be sure that the first eight characters of your Oracle Java Cloud Service instance names are unique. By default, the names of the domain and cluster in the Oracle Java Cloud Service instance are generated from the first eight characters of the service instance name and use the following formats, respectively: • `*first8charsOfServiceInstanceName*_domain` • `*first8charsOfServiceInstanceName*_cluster` • `*first8charsOfServiceInstanceName*_DGCluster` (when Oracle Coherence is enabled for a service instance)
Do not modify or delete the default resources and applications in the WebLogic Server domain.	All service instances are configured with several data sources, libraries, applications and other domain resources. Do not modify or delete these default domain resources. Any modifications might cause your servers to fail upon restart. By default, a sample application is deployed to a service instance. You can delete this sample application from the domain.
Do not modify the default administration ports.	Do not change the default ports that Oracle Java Cloud Service configures for the Oracle WebLogic Server administration server and the Oracle Traffic Director administration server. You can open new ports, but closing or modifying existing ports may impair the functionality of the service instance. See About the Default Access Ports.

Best Practice	More Information
Restrict external access.	Oracle recommends you restrict Internet access to the WebLogic administration port and Oracle Traffic Director WebLogic administration port by configuring security ingress rules using a fixed set of IPs or a CIDR matching your organization's network addresses. See the My Oracle Support Document ID 2664435.1.
Do not enable T3 or tunneling.	Oracle does not recommend enabling the T3 and T3 over SSL (T3S) protocols or tunneling on network channels that are accessible from outside of Oracle Cloud.
Use Oracle VPN.	If you continue to use T3, block the ports from external access and make sure the ports are accessible via Oracle VPN only using T3S. See VPN Connect Overview in the Oracle Cloud Infrastructure documentation.
Apply only approved patches.	Apply only patches that are provided through the Oracle Java Cloud Service patching feature. Do not apply patches from any other source. See View Patch Details.
Apply patches promptly.	Apply the most recent patches as soon as they're available in Oracle Java Cloud Service. Delaying the application of patches might cause your service instance to be unsupported for future patches.
Apply Oracle Linux OS patches.	Oracle Java Cloud Service does not provide cloud tooling to patch the Oracle Linux operating system on the nodes in your service instances. You are responsible for installing Oracle Linux OS patches. See About Patching and Rollback. Do not install OS patches for other Linux distributions. Also, if you plan to use your service instance for production applications, Oracle recommends that you avoid installing any test, development, or preview OS packages that might be available in the repository.
Follow the approved SSL configuration procedure.	If you want to use custom SSL certificates in your service instance, see Configure SSL for a Service Instance. This procedure ensures that management operations like restarting, backing up, scaling, and patching continue to function properly.
Use Oracle Java Cloud Service to perform scaling operations.	Scale out an Oracle WebLogic Server cluster within a service instance only by using the scaling capabilities of Oracle Java Cloud Service. Do not use the WebLogic Server administrative interfaces to directly add Managed Servers to a cluster. See About Scaling an Oracle Java Cloud Service Cluster and Scale Out a Coherence Data Grid.
Use the REST API to add a cluster to an existing service instance.	By default, Oracle Java Cloud Service creates a single Oracle WebLogic Server cluster within your service instance, and optionally a second cluster if Oracle Coherence is enabled. However, you can create additional clusters in the service instance if necessary. Add clusters to a service instance only by using the Oracle Java Cloud Service REST API. Do not use the WebLogic Server administrative interfaces to directly add clusters to your domain. See Scale Out a Service Instance in *REST API for Oracle Java Cloud Service*. Oracle Java Cloud Service does not support service instances with multiple Coherence (storage-enabled) clusters.
Use the REST API to change the Node Manager password.	If you want to change the default Node Manager password, do not manually edit the `nm_password.properties` file on a node. This will cause lifecycle and other administrative operations to fail. Instead, you must use the Oracle Java Cloud Service REST API. See Change the Node Manager Credentials in *REST API for Oracle Java Cloud Service*.

Best Practice	More Information
Do not use the default Coherence cluster.	When Oracle Coherence is enabled for a service instance, Oracle Java Cloud Service creates a Coherence cluster in the domain named `DataGridConfig`. Both Oracle WebLogic Server clusters are members of this Coherence cluster. You can customize the configuration of the `DataGridConfig` Coherence cluster if necessary. All Oracle Java Cloud Service domains also include a Coherence cluster named `defaultCoherenceCluster`, which has no members. Do not modify or use this Coherence cluster.
Use Oracle Java Cloud Service to add storage to a node.	If a node within a service instance requires additional storage, add storage by scaling up the node. See About Scaling an Oracle Java Cloud Service Node. Do not attach custom storage volumes to a service instance's nodes. Any custom storage volumes that you attach will be detached if the service instance is restarted.
Do not share databases if backups are enabled.	To ensure that you can restore the database for an Oracle Java Cloud Service instance without risking data loss for other service instances, Oracle recommends that you do not associate the same infrastructure schema database (or the same pluggable database) with multiple service instances. Backups of a database that is used with multiple Oracle Java Cloud Service instances contain data for all the instances. Therefore, if you restore the database from a backup, data for all the service instances is restored, which might not be the intended result.
Use a dedicated storage container for backups.	Do not use an object storage container that you use for backups of Oracle Java Cloud Service instances for any other purpose. Using the container for multiple purposes can result in billing errors. For example, do not use the same object storage container to also back up the database.
Do not directly modify the `MIDDLEWARE_HOME` or `JAVA_HOME` volumes on any node.	All storage volumes mounted under `/u01` should be treated as read-only by administrators, except for the contents of `DOMAIN_HOME` and `APPLICATION_HOME`. Any modifications you make to other volumes such as `MIDDLEWARE_HOME` may be lost when you perform management operations on your Oracle Java Cloud Service instance like applying a patch.
Do not modify a node's file system configuration.	Do not detach, change file access permissions for, or change the mount point of any storage volume that Oracle Java Cloud Service creates and attaches to a service instance's nodes during the creation of the service instance. See About the Disk Volumes.
Do not directly modify a node's user or SSH settings.	Oracle Java Cloud Service configures default OS users and Secure Shell (SSH) access settings during the creation of a service instance. Do not modify these default users and use only the features of Oracle Java Cloud Service to modify the SSH settings. See Add an SSH Public Key.
Avoid certain modifications to a node's network configuration.	Do not close any ports that Oracle Java Cloud Service opened during the creation of your service instance. You can open new ports, but closing existing ports may impair the functionality of the service instance. See Understanding the Default Access Ports. Also, • Do not change the default egress and ingress network and security settings of a service instance's nodes. • Do not detach the public IP addresses from any of a service instance's nodes.

Best Practice	More Information
Do not modify the required database schemas.	Do not modify the Oracle Fusion Middleware component schemas that Oracle Java Cloud Service provisions within the selected infrastructure schema database during the creation of a service instance.
Do not modify any scripts in the `/u01/app/oracle/tools` directory.	Any changes to the scripts in the `/u01/app/oracle/tools` directory can affect the manageability of your instance adversely.
Do not modify the `/u01/app/oracle/tools/paas/state/logs` directory or any files in it.	If the log files in `/u01/app/oracle/tools/paas/state/logs` are lost, then information that's important for diagnosing issues will be lost.
Do not modify the `/u01/app/oracle/tools/paas/state/work` directory or any files in it.	If the files in `/u01/app/oracle/tools/paas/state/work` are lost or modified, failures can occur when performing operations like adding a second Oracle Traffic Director node or rotating the log file.
Use only the `opc-config` Oracle Traffic Director configuration, which is created automatically when the instance is created.	If you replace `opc-config` with a configuration of a different name, then Oracle Cloud won't recognize the configuration. Oracle Traffic Director may not start when the node is rebooted, restarted, or restored. If you create an additional Oracle Traffic Director configuration, then that configuration and its Oracle Traffic Director instance are not recognized by Oracle Cloud. So lifecycle operations for that Oracle Traffic Director instance are not managed by Oracle.
Use the ephemeral port range (1024 - 65535) for Oracle Traffic Director listener port.	By default, Oracle Java Cloud Service instance provisioning always configures Oracle Traffic Director HTTPS listener on port 8081 and redirects the requests on port 443 to port 8081 using `iptables`. If the listener is configured to use ports in the range 1 to 1024, Oracle Traffic Director fails to start as `oracle` user cannot open listen ports in the range 1 to 1024. So, you must configure Oracle Traffic Director listener port in the ephemeral port range and use the `iptables` command to redirect requests from port 443 to the ephemeral port that you have configured.

5

Deploy and Undeploy Applications for an Oracle Java Cloud Service Instance

This section describes deploying and undeploying applications to an Oracle Java Cloud Service instance by using: Fusion Middleware Control, the WebLogic Server Administration Console, WLST commands, and an IDE. You cannot deploy and undeploy applications directly through the Oracle Java Cloud Service console.

Topics:

- Overview of Deploying Applications to Oracle Java Cloud Service Instances
- Use Fusion Middleware Control to Deploy an Application
- Use the WebLogic Server Administration Console to Deploy and Manage Applications
- Use WLST Commands to Deploy and Undeploy an Application
- Use an IDE to Deploy and Undeploy an Application
- Deploy an Application to an Oracle Java Cloud Service Instance with Multiple Clusters
- Access an Application Deployed to an Oracle Java Cloud Service Instance
- Enable the JVM Debug Port on an Oracle Java Cloud Service Instance
- Use Third-Party Frameworks with Oracle Java Cloud Service

Overview of Deploying Applications to Oracle Java Cloud Service Instances

Task	Description	More Information
Deploy and undeploy applications	Deploy and undeploy applications using various tools, just as you would for an on-premises WebLogic Server environment. **Note**: Don't use the *nostage* deployment mode. That mode is not supported. For example, when deploying applications by using the WebLogic Server Administration Console, *don't select* **I will make the deployment accessible from the following location** under **Source Accessibility** on the **Options** page of the deployment wizard.	Use Fusion Middleware Control to Deploy an Application Use the WebLogic Server Administration Console to Deploy and Manage Applications Use WLST Commands to Deploy and Undeploy an Application Use an IDE to Deploy and Undeploy an Application
Access a deployed application	Identify the public IP address of the load balancer and use it to build the URL for the application.	Access an Application Deployed to an Oracle Java Cloud Service Instance

Task	Description	More Information
Access the sample application	Access, view, and manage the sample application deployed automatically when you created your Oracle Java Cloud Service instance.	About the Sample Application Deployed to an Oracle Java Cloud Service Instance
Define a custom domain name for the application environment	Register a custom "vanity" domain name with a registration vendor and associate it to your application environment.	Configure a Vanity Domain Name for a Service Instance
Define a custom URL for a deployed application	Define a custom "vanity" URL for an application deployed to an Oracle Java Cloud Service instance.	Configure a Custom URL for an Application Deployed to a Service Instance
Configure Secure Socket Layer (SSL) for your custom domain	Configure SSL between the client browser and the load balancer to ensure applications are accessed securely using an SSL certificate.	Configure SSL for a Service Instance
Use third-party frameworks with Oracle Java Cloud Service	Use third-party frameworks to extend the functionality of Oracle Java Cloud Service.	Use Third-Party Frameworks with Oracle Java Cloud Service

Use Fusion Middleware Control to Deploy and Undeploy an Application

You can deploy and undeploy an application by using Oracle Enterprise Manager Fusion Middleware Control, just as you would in an on-premises environment.

Topics

- Use Fusion Middleware Control to Deploy an Application
- Use Fusion Middleware Control to Undeploy an Application

Use Fusion Middleware Control to Deploy an Application

You can use Oracle Enterprise Manager Fusion Middleware Control to deploy an application to an Oracle Java Cloud Service instance, just as you would deploy the application to an on-premises Oracle WebLogic Server environment.

The following tutorial and documentation are available to help you learn more about using Fusion Middleware Control to deploy an application:

- Tutorial
- Oracle Fusion Middleware 12.2.1.4

  Deploying, Undeploying, and Redeploying Java EE Applications in *Administering Oracle Fusion Middleware*
- Oracle Fusion Middleware 12.2.1.3

  Deploying, Undeploying, and Redeploying Java EE Applications in *Administering Oracle Fusion Middleware*
- Oracle Fusion Middleware 12.2.1.2

Deploying, Undeploying, and Redeploying Java EE Applications in *Administering Oracle Fusion Middleware*

- Oracle Fusion Middleware 11.1.1.7

  Deploying, Undeploying, and Redeploying Java EE Applications in *Oracle Fusion Middleware Administrator's Guide*

Use Fusion Middleware Control to Undeploy an Application

You can use Oracle Fusion Middleware Control to undeploy an application to an Oracle Java Cloud Service instance, just as you would undeploy the application in an on-premises WebLogic Server environment.

The following tutorial and documentation are available to help you learn more about using Fusion Middleware Control to undeploy an application:

- Tutorial (see the "Undeploying an Application" section in the tutorial)
- Oracle Fusion Middleware 12.2.1.4

  Deploying, Undeploying, and Redeploying Java EE Applications in *Administering Oracle Fusion Middleware*
- Oracle Fusion Middleware 12.2.1.3

  Deploying, Undeploying, and Redeploying Java EE Applications in *Administering Oracle Fusion Middleware*
- Oracle Fusion Middleware 12.2.1.2

  Deploying, Undeploying, and Redeploying Java EE Applications in *Administering Oracle Fusion Middleware*
- Oracle Fusion Middleware 11.1.1.7

  Deploying, Undeploying, and Redeploying Java EE Applications in *Oracle Fusion Middleware Administrator's Guide*

Use the WebLogic Server Administration Console to Deploy and Manage Applications

You can use the WebLogic Server Administration Console to deploy and manage applications in an Oracle Java Cloud Service instance, just as you would deploy and manage applications in an on-premises Oracle WebLogic Server environment.

Besides this documentation, the following video and tutorial are available to help you learn how to use the WebLogic Server Administration Console to deploy applications:

Video

Tutorial

By default, if you created your service instance in an Oracle Cloud Infrastructure Classic region, remote access to the administration console is disabled for security purposes. If you did not enable console access while provisioning your service instance, see Enabling Console Access in an Oracle Java Cloud Service Instance.

Topics

- Use the WebLogic Server Administration Console to Deploy an Application
- Use the WebLogic Server Administration Console to Start an Application
- Use the WebLogic Server Administration Console to Undeploy an Application

Use the WebLogic Server Administration Console to Deploy an Application

You can use the WebLogic Server Administration Console to deploy applications to an Oracle Java Cloud Service instance.

1. Sign in to the web console.
2. Look for the name of the instance on which you want to deploy an application.
3. Click **Manage this instance** ☰, and select **Open WebLogic Server Administration Console**.
4. Sign in using credentials you specified when you created the Oracle Java Cloud Service instance.

   If you did not create the service instance, ask your administrator for the login credentials.
5. In the Change Center of the WebLogic Server Administration Console, click **Lock & Edit** if that button is enabled.
6. Under Domain Structure, select **Deployments**.
7. On the Deployments page, click **Install**.
8. On the Install page, click the **Upload your file(s)** link. This link is in the text just above the **Path** field.
9. In the Install Application Assistant, click **Browse** next to **Deployment Archive**, select the application you want to deploy, and click **Next**.
10. Under **Locate deployment to install and prepare for deployment**, select the application if it is not already selected, and click **Next**.
11. Under **Choose installation type and scope**, select whether you want to install the deployment as an application or as a library, and click **Next**.
12. Under **Select deployment targets**, select the servers or clusters to which you want to deploy the application and click **Next**.
13. (Optional) Update settings for the deployment and click **Next**.

    These settings include:

    - The deployed name of the web application.
    - The security model that is applied to the application.
    - How the source files (WAR or exploded directory contents) are made available to targeted managed servers and clusters.
    - How the deployment plan source files are made available to all targeted managed servers and clusters.

    Typically, the default settings are adequate.

14. Review the configuration settings you specified and click **Finish**.
15. Under Domain Structure, select **Deployments**.

    Look for the application in the Deployments table. It shows the status **distribute Initializing**.
16. In the Change Center, click **Activate Changes**.

    The WebLogic Server Administration Console shows the application in the **Prepared** state.

Your application is now deployed.
Start the application. See Use the WebLogic Server Administration Console to Start an Application.

Use the WebLogic Server Administration Console to Start an Application

You must start the application to make it ready to accept requests.

To start an application:

1. Sign in to the web console.
2. Look for the name of the instance from which you want to undeploy an application.
3. Click **Manage this instance** ☰, and select **Open WebLogic Server Administration Console**.
4. Sign in using credentials you specified when you created the Oracle Java Cloud Service instance.

   If you did not create the service instance, ask your administrator for the login credentials.
5. In the Change Center of the WebLogic Server Administration Console, click **Lock & Edit**.
6. Under Domain Structure, select **Deployments**.
7. Click the **Control** tab.
8. In the Deployments table, select the check box next to the application that you want to start.
9. Click **Start**, and select **Servicing all requests**.
10. On the Start Deployments dialog, click **Yes** to confirm the deployment.

    The application is now in the **Active** state and is ready to accept requests.

Use the WebLogic Server Administration Console to Undeploy an Application

You can use the WebLogic Server Administration Console to undeploy an application from an Oracle Java Cloud Service instance.

To undeploy the application:

1. Sign in to the web console.
2. Look for the name of the instance from which you want to undeploy an application.
3. Click **Manage this instance** ☰, and select **Open WebLogic Server Administration Console**.

4. Sign in using credentials you specified when you created the Oracle Java Cloud Service instance.

   If you did not create the service instance, ask your administrator for the login credentials.

5. In the Change Center of the WebLogic Server Administration Console, click **Lock & Edit**.
6. Under Domain Structure, select **Deployments**.
7. Click the **Control** tab.
8. In the Deployments table, select the check box next to the application you want to undeploy.
9. From the **Stop** menu, select one of the following:
   - **When work completes**
   - **Force stop now**

   Do not select **Stop, but continue servicing administration requests**.
10. Click the **Configuration** tab.
11. When the application is in the **Stopped** state, select the application, and click **Delete**.
12. Click **Yes** to confirm.
13. In the Change Center, click **Activate Changes**.

Use WLST Commands to Deploy and Undeploy an Application

You can use WLST commands to deploy and undeploy an application to an Oracle Java Cloud Service instance. All WLST commands are supported.

For example:

```
deploy('myapp','/u01/apps/myapp.war',upload='true')
```

You can use a secure shell (SSH) to connect to the virtual machine (VM) that hosts the Administration Server and run WLST commands locally. See Run WLST Commands on a Node. You can use either WLST online or offline commands.

Alternatively, you can connect to the Administration Server using another WLST installation and run WLST commands remotely, for example, from a command shell in your local environment. See Running WLST Commands from a Different Host. With this approach you can use WLST online commands only.

Oracle Fusion Middleware documentation is available to help you learn more about using WLST commands to deploy an application.

- Oracle Fusion Middleware 12.2.1.4

  Using WLST Online to Deploy Applications in *Oracle Fusion Middleware Understanding the WebLogic Scripting Tool*

Using WLST Offline to Deploy Applications in *Oracle Fusion Middleware Understanding the WebLogic Scripting Tool*

- Oracle Fusion Middleware 12.2.1.3

  Using WLST Online to Deploy Applications in *Oracle Fusion Middleware Understanding the WebLogic Scripting Tool*

  Using WLST Offline to Deploy Applications in *Oracle Fusion Middleware Understanding the WebLogic Scripting Tool*

- Oracle Fusion Middleware 12.2.1

  Using WLST Online to Deploy Applications in *Oracle Fusion Middleware Understanding the WebLogic Scripting Tool*

  Using WLST Offline to Deploy Applications in *Oracle Fusion Middleware Understanding the WebLogic Scripting Tool*

- Oracle Fusion Middleware 11.1.1.7

  Using WLST Online to Deploy Applications in *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*

  Using WLST Offline to Deploy Applications in *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*

Use an IDE to Deploy and Undeploy an Application

You can use an Integrated Development Environment (IDE) such as Eclipse to deploy and undeploy an application to an Oracle Java Cloud Service instance.

The following tutorials are also available:

- Using Eclipse

  Tutorial

- Using JDeveloper

  Tutorial

Topics

- Prerequisites for Deploying Using an IDE
- Connect the IDE to a Remote WebLogic Server
- Use an IDE to Deploy an Application to a Cluster
- Use an IDE to Deploy an Application to a Cluster

Prerequisites for Deploying Using an IDE

Complete the following tasks before deploying an application:

- Create an Oracle Java Cloud Service instance.
- By default, if you created your service instance in an Oracle Cloud Infrastructure Classic region, remote access to the Administration Server in your instance on port `7002` is disabled for security purposes. If you did not enable console access while creating your service instance, see Enable Console Access for a Service Instance.

- Install the IDE on your local machine.

  If you want to use Eclipse, then also install Oracle Enterprise Pack for Eclipse (download location: http://www.oracle.com/technetwork/developer-tools/eclipse/downloads/index.html)

Connect the IDE to a Remote WebLogic Server

To deploy an application to Oracle Java Cloud Service, you must first establish a connection between the IDE and Oracle WebLogic Server.

1. Use the Oracle Java Cloud Service console to identify the public IP address of your Administration Server.
2. Start Oracle Enterprise Pack for Eclipse.
3. Click **Workbench**.
4. Select **Window – Show View – Others...**
5. Enter `server` in the search box, click the **Servers** entry, and click **OK**.

   The Servers view panel is displayed in the bottom half of the Workbench.
6. In the Servers view panel, click **No servers are available. Click this link to create a new server...**
7. Click **Oracle**, then select the WebLogic Server version of your Oracle Java Cloud Service instance, and click **Next**.
8. In the New Server dialog box, click the browse icon next to **WebLogic home**, and select your WebLogic Server home directory.

   > **Note:**
   >
   > Make sure that the local version of WebLogic Server you're running is the same version as the instance running on the cloud service. If they are not the same version, then you can't make a connection.

9. Click the browse icon next to **Java home**, and select your Java home directory.
10. Select **Remote**.
    a. Set the **Remote Host** to the IP address of your Oracle Java Cloud Service instance.
    b. Set **Port** to **7002**. This is the SSL port of your Oracle Java Cloud Service instance.
    c. Make sure that **Use SSL port** is checked.
    d. Set **User** to the WebLogic Server administrator credentials you specified when you created the Oracle Java Cloud Service instance.
11. Click **Test Connection**.
12. In the Success dialog box, click **OK**.

    In the Eclipse console, a new connection is added to the **Servers** view panel. A **Validating server...** status message is displayed. After the connection is established, the status changes to **Started**.

Use an IDE to Deploy an Application to a Cluster

You can deploy the application to the administration server by using the Eclipse IDE, for example. You can deploy an application to a cluster by setting properties to add the cluster as a target.

To use Eclipse to deploy an application to a cluster in your Oracle Java Cloud Service instance:

1. If you have not done so already, start the Oracle Enterprise Pack for Eclipse (OEPE).
2. If Project Explorer is not visible, select **Window – Show View – Project Explorer**.
3. Under Project Explorer, right-click and select **Import – WAR file** .
4. In the War Import dialog, click the **Browse** button, navigate to the directory where your application resides, and select it. Click **Open**.
5. On the WAR Import dialog, click **Finish**.
6. If an Open Associated Perspective dialog appears, click **Yes** to associate the perspective of the project to Java Enterprise Edition (Java EE).
7. On the Eclipse console, select the **Servers** tab.

   By default, the IDE deploys your application to the Oracle Java Cloud Service administration server. You need to deploy the application to a cluster rather than to the Administration Server.
8. On the Servers tab, right click on the server connection established previously to the administration server.
9. Select **Properties**.

   The Properties dialog is displayed.
10. Select **WebLogic – Publishing – Advanced**.
11. Click on the green plus sign to add the cluster as the target.

    A new line is added under **Targets**.
12. Click on **Browse**.

    The names of the administration server and the cluster are listed on the Target Name dialog.
13. To delete the Administration Server as a target, select the target name and click on the red cross icon next to the Administration Server name.
14. To add the cluster as a target, click on the green plus sign.

    A new line is added under **Targets**.
15. Click **Browse** and select the cluster where you want to deploy the application.

    The Administration Server and the cluster are listed in the Target Name dialog.
16. Click on the cluster and click **OK**.
17. Click **Apply**, and then click **OK**.
18. Click on the Servers tab.
19. On the **Servers** tab, right click on the server connection and select **Add and Remove...**

The application is then listed in the available applications section of the Add and Remove dialog.

20. Select the name of the application and click **Add**.
21. Click **Finish**.

    In the bottom right corner of the Eclipse console, the status of the publish request is displayed.
22. Click the icon next to the publish request status message to see the details of the request.

    The status of the request will become **Active**.
23. On the Servers tab, expand the server connection to see that the application is deployed.

Use an IDE to Undeploy an Application

After you deploy an application, you can undeploy it by using an IDE.

To use the Eclipse IDE to undeploy an application:

1. If you have not done so already, start Oracle Enterprise Pack for Eclipse (OEPE).

   You can click on the icon on your desktop or search for **eclipse** on the Windows Start menu.
2. Select the **Servers** tab of the Eclipse console.
3. Expand the server connection for the server on which the application resides.
4. Locate the application under **Published Modules**.
5. Right-click on the application and select **Remove**.

   The application is removed from the published modules list.

The application is now undeployed.

Deploy an Application to an Oracle Java Cloud Service Instance with Multiple Clusters

If you created an Oracle Java Cloud Service instance with multiple clusters, your application might require additional configuration prior to deployment.

When you create a service instance with multiple clusters, you can provide a path prefix for each cluster, such as `/mycluster`. By default, the default prefix for the first cluster is `/`, and the default prefix for each additional cluster is the name of the cluster. Oracle Java Cloud Service provisions an Oracle-managed load balancer and configures the load balancer to route traffic to each cluster based on the path prefix.

If you want to deploy an application to a cluster whose path prefix is **not** `/`, you must update the application's deployment descriptors.

1. Edit the `WEB-INF/weblogic.xml` file for the application.
2. Update the value of the `context-root` element. Insert the path prefix of the target cluster to the front of the existing path.

   For example: **`/mycluster`**`/myapp`

Deploy or redeploy the application, and target it to the desired cluster. When you access the application, include the cluster path prefix. For example:

```
https://myinstance-myaccount.myregion.oraclecloud.com/mycluster/myapp/
index.html
```

Access an Application Deployed to an Oracle Java Cloud Service Instance

You can access an application deployed to an Oracle Java Cloud Service instance through a URL in a browser.

If your service instance includes multiple WebLogic Server clusters, see Deploy an Application to an Oracle Java Cloud Service Instance with Multiple Clusters.

To access a deployed application:

1. Navigate to the Oracle Java Cloud Service console.
2. Click the service instance where you deployed the application.
3. If your instance has a load balancer, then make a note of the public IP address of the load balancer. Otherwise, note the public IP address of a WebLogic managed server node that hosts your application.
4. Find the context-root of the application.

   The context-root is defined as a project property, or in the `weblogic.xml` file. The context-root might or might not be the same as the internal application name.

   a. Click **Manage this instance** ☰, and select **Open WebLogic Server Administration Console**.
   b. Sign in using credentials you specified when you created the service instance.

      If you did not create the service instance, ask your administrator for the login credentials.
   c. Select *domain* > Deployments, where *domain* is the domain where the application is deployed.
   d. In the Deployments table, click the name of your application.
   e. In the Overview tab, locate the **context-root**.
5. In a browser, specify the URL of the application, in the following format:

   `scheme://host/applicationContextRoot`

   - `scheme`: `http` or `https`

     If you created the service instance by using the Oracle Java Cloud Service console, you can access the application through HTTPS only. The HTTP port is disabled by default. You can open the HTTP port on the load balancer manually. See Enable HTTP Access to a Service Instance.
   - `host`: The public IP address of the load balancer (or the managed server if you're not using a load balancer)

     If you do not want to specify the IP address when you access the application, you can create a custom domain name. To do this, you can use a third-party DNS provider to map the custom domain name. See Configure a Vanity Domain Name for a Service Instance.

> **Note:**
>
> Avoid using IP addresses to access applications running in a production environment. Use a custom domain name, or the default internal host name if a custom domain name has not been provided.

- `applicationContextRoot`: The context root that you noted earlier in this procedure.

  If the application is deployed to a cluster that is assigned a path prefix, the context root must also include the path prefix. For example: `/cluster1/myapp`

  If you want to customize the default URL for your application, see Configure a Custom URL for an Application Deployed to a Service Instance.

6. If you receive a warning, accept the self-signed certificate.

   The application opens in your browser.

Enable the JVM Debug Port on an Oracle Java Cloud Service Instance

Opening a JVM debug port on your Oracle Java Cloud Service nodes will enable you to debug applications remotely from an IDE running on your local machine.

> **Note:**
>
> You will need direct access to the IP address of the server running your application.

Topics

- Set Up the Debug Port in WebLogic Server
- Create an Access Rule for the Debug Port

Set Up the Debug Port in WebLogic Server

To set up the debug port in WebLogic Server, you must update the server’s start-up parameters and then restart the server.

1. Go to the Oracle Java Cloud Service console, and select the service instance that you want to enable the JVM debug port for.
2. In the service menu, click **Open WebLogic Server Administration Console**.
3. On the Welcome screen, enter the administrator login credentials.

   If you did not create the service instance, ask your Java Administrator for the login credentials.

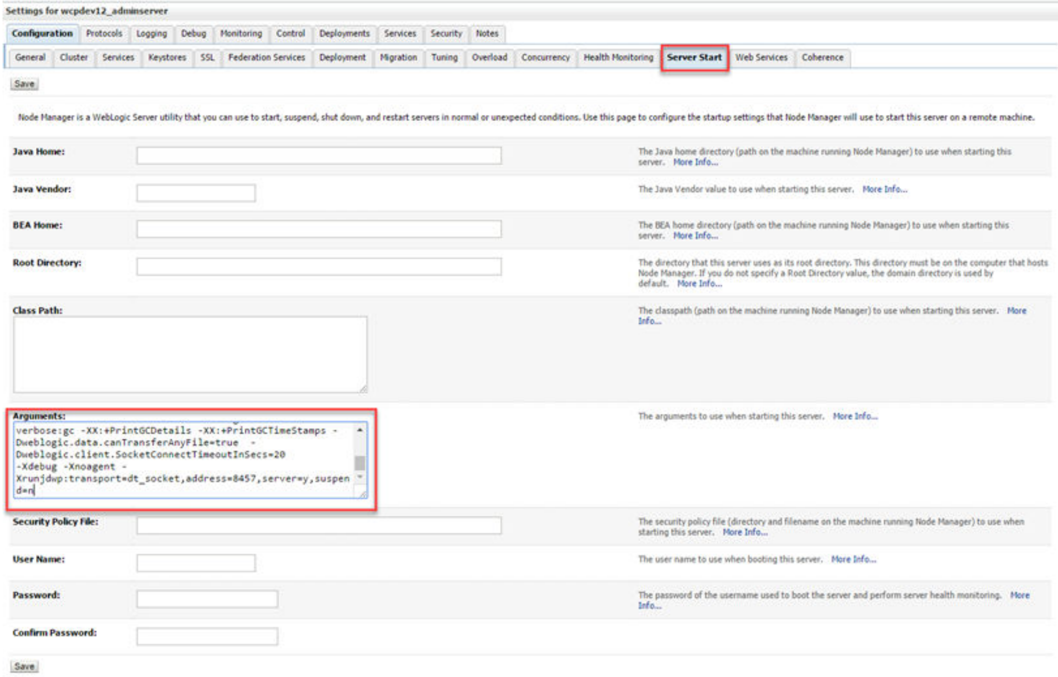
4. In the Change Center pane, click **Lock & Edit**.
5. In the Domain Structure pane, expand **Environment**, and click **Servers**.

   A list of available servers is displayed.

   > **Note:**
   >
   > Make sure that you are on the **Configuration** tab.

6. In the Servers table, click the name of the server running your application.
7. Go to the **Server Start** tab.
8. Append the following in the **Arguments** field. Make sure there are no line breaks when you copy and paste these arguments.

   ```
   -Xdebug -Xnoagent -
   Xrunjdwp:transport=dt_socket,address=8457,server=y,suspend=n
   ```



Settings for wcpdev12_adminserver

Node Manager is a WebLogic Server utility that you can use to start, suspend, shut down, and restart servers in normal or unexpected conditions. Use this page to configure the startup settings that Node Manager will use to start this server on a remote machine.

Java Home:	The Java home directory (path on the machine running Node Manager) to use when starting this server. More Info...
Java Vendor:	The Java Vendor value to use when starting this server. More Info...
BEA Home:	The BEA home directory (path on the machine running Node Manager) to use when starting this server. More Info...
Root Directory:	The directory that this server uses as its root directory. This directory must be on the computer that hosts Node Manager. If you do not specify a Root Directory value, the domain directory is used by default. More Info...
Class Path:	The classpath (path on the machine running Node Manager) to use when starting this server. More Info...
Arguments:	The arguments to use when starting this server. More Info...
Security Policy File:	The security policy file (directory and filename on the machine running Node Manager) to use when starting this server. More Info...
User Name:	The user name to use when booting this server. More Info...
Password:	The password of the username used to boot the server and perform server health monitoring. More Info...
Confirm Password:	

```
verbose:gc -XX:+PrintGCDetails -XX:+PrintGCTimeStamps -
Dweblogic.data.canTransferAnyFile=true  -
Dweblogic.client.SocketConnectTimeoutInSecs=20
-Xdebug -Xnoagent -
Xrunjdwp:transport=dt_socket,address=8457,server=y,suspen
d=n
```

9. Make a note of the debug port address that you specified (`8457` in this example). You'll need this address in the next task.
10. Click **Save**.
11. In the Change Center pane, click **Activate Changes**.
12. In the Domain Structure pane, expand **Environment**, and click **Servers**.
13. Go to the **Control** tab.
14. In the Servers table, select the check box next to the server running your application.
15. Click **Shutdown**, and select **Force shutdown now**. When prompted for a confirmation, click **Yes**.

   Wait for the state of the server to change to SHUTDOWN. Refresh the page to view the current state.

16. Select the check box next to the server, and click **Start**. When prompted for a confirmation, click **Yes**.

    Wait for the state of the server to change to RUNNING. Refresh the page to view the current state.

Create an Access Rule for the Debug Port

You will need to create an access rule so that messages to and from your local debug session will be accepted by the Oracle Java Cloud Service node hosting the application that you want to debug.

You will use the Oracle Java Cloud Service console to create the access rule.

If you provisioned this service instance in an Oracle Cloud Infrastructure region, instead you must use the Oracle Cloud Infrastructure Console to create the access rules (security list). See Security Lists in the Oracle Cloud Infrastructure Services documentation.

1. Navigate to the Oracle Java Cloud Service console
2. Click the ☰ Menu icon adjacent to the service instance name and select **Access Rules**.

   The Access Rules page is displayed, showing the list of all access rules.
3. Click **Create Rule**.

   The Create Access Rule dialog is displayed.
4. Specify a unique **Rule Name**. Optionally specify a rule **Description**.

   The name must begin with a letter, and can contain numbers, hyphens, or underscores. The length cannot exceed 50 characters. When you create a rule, you cannot use prefixes `ora_` or `sys_`.
5. Specify **PUBLIC-INTERNET** for the rule source.
6. Specify **WLS_ADMIN_SERVER** or **WLS_MANAGED_SERVER** depending on the server type where you set up the debug port.
7. The destination port should match the address you set up for the debug port. In our example this would be `8457`.
8. Set the protocol to **TCP**.
9. Now click **Create** to create the new rule.

The Access Rules page displays your new rule.

Use Third-Party Frameworks with Oracle Java Cloud Service

You can use third-party frameworks that conform to the Java EE and Java SE standards to extend the functionality of Oracle Java Cloud Service.

You can use each supported framework with Oracle Java Cloud Service in one of the following ways:

- Packaging the framework with applications that use it
- Deploying the framework as a shared library
  For more information, see Deploy and Undeploy Applications for an Oracle Java Cloud Service Instance.

If multiple applications use a framework, or if you want to simplify updates by minimizing the size of applications that use the framework, deploy the framework as a shared library.

Topics

- Third-Party Application Development Frameworks Tested with Oracle Java Cloud Service
- Information for Configuring Apache Axis/Java
- Omit Checks for Updates to Quartz Job Scheduler

Third-Party Application Development Frameworks Tested with Oracle Java Cloud Service

Oracle Java Cloud Service has been tested with several third-party frameworks. A specific release of each supported framework was tested.

Framework	Release Tested	Purpose
Akka	2.3.9	Build highly concurrent, distributed, and resilient message-driven applications on the JVM.
Apache Axis2/Java	1.6.2	Simplify client-side and server-side programming of Web services. See Information for Configuring Apache Axis/Java.
Apache Commons component BeanUtils	1.9.2	Simplify the use of the Java reflection and introspection APIs.
Apache Commons component Collections	3.2.1	Extend or augment the Java Collections Framework.
Apache Commons component Digester	3.2	Map XML configuration data to Java objects.
Apache Commons component IO	2.4	Help develop functionality for input and output through data streams.
Apache Commons component Logging	1.2	Enable a library to be used with a chosen logging implementation at runtime.
Apache CXF	3.0.4	Build and develop services that use front-end programming APIs, such as JAX-WS and JAX-RS.
Apache Log4j	The following releases: • 1.2.17 • 2.0	Provide a logging framework for Java applications.
Apache MyFaces	2.2.8	Simplify the development of web applications with JavaServer™ Faces by providing: • A JavaServer Faces, implementation • Component libraries of UI widgets for building web applications with JavaServer Faces • Extension packages to JavaServer Faces • Integration modules to other technologies and standards
Apache Struts	2.3.3	Simplify the development of Java web applications that use a Model-View-Controller (MVC) architecture.

Framework	Release Tested	Purpose
Apache Tapestry	5.3.7	Simplify the development of dynamic, robust, highly scalable web applications in Java.
Apache Thrift	0.9.0	Build services that work efficiently and seamlessly between languages including, among other languages: • C++ • C# • Cocoa • Delphi • Erlang • Haskell • Java • JavaScript • Node.js • OCaml • Perl • PHP • Python • Ruby • Smalltalk
Apache Velocity	1.7	Reference objects that are defined in Java code through a template language.
Apache Wicket	6.18.0	Simplify the development of Java web applications by: • Properly separating markup and logic • Using a Plain Old Java Object (POJO) data model • Limiting the use of Extensible Markup Language (XML) configuration files
FreeMarker	2.3.19	Generate text output from templates, for example, web pages for servlet-based applications that follow the MVC pattern.
Google Guava Libraries	15.0	Provide Java libraries for functionality such as: • Caching • Collections • Concurrency • Common annotations • I/O • Primitives • String processing
Google Guice	3.0	Provide dependency injection for Java 6 and above.
GWT	2.5.1	Build and optimize complex browser-based applications without the need to understand the behavior of specific browsers, the `XMLHttpRequest` object, or JavaSrcipt.
Hibernate ORM	4.2.8	Provide Object/Relational Mapping (ORM) to simplify storage of data by object-oriented applications in relational databases.
JBoss Seam	3.1.0	Provide a modular set of extensions to the contexts and dependency injection (CDI) programming model.
Joda-Time	2.1	Replace the date and time class libraries in the Java Platform, Standard Edition (Java SE).
JQuery	2.0.3	Provide a JavaScript library to simplify HTML document traversal and manipulation, event handling, animation, and Ajax.
JRuby	1.7.2	Provide a 100% Java implementation of the Ruby programming language.

Framework	Release Tested	Purpose
Quartz Job Scheduler	2.1.5	Create simple or complex schedules for executing jobs whose tasks are defined as standard Java components. See Omit Checks for Updates to Quartz Job Scheduler.
Simple Logging Facade for Java (SLF4J)	1.7.7	Enable end users to plug in a specific logging framework at deployment time.
Spring	4.0.3	Build simple, portable, fast, and flexible JVM-based systems and applications.

Information for Configuring Apache Axis/Java

The Apache Software Foundation web site provides documentation for using Apache Axis/Java.

For detailed instructions for configuring Apache Axis with Oracle WebLogic Server, see *WebLogic* in Application Server Specific Configuration Guide.

Omit Checks for Updates to Quartz Job Scheduler

By default, Quartz Job Scheduler checks for updates when it starts.

The check for updates involves connecting to a remote server. If the server can’t be reached, the check fails and an exception is written to a log file. The failure does not prevent Quartz Job Scheduler from starting and does not affect the functionality of Quartz Job Scheduler in any way. However, you can prevent this exception by omitting checks for updates to Quartz Job Scheduler.
To omit checks for updates to Quartz Job Scheduler:

1. For each managed server in your Oracle Java Cloud Service instance, set one of the following properties to `true`:

   - The Quartz configuration property `org.quartz.scheduler.skipUpdateCheck`
   - The Java system property `org.terracotta.quartz.skipUpdateCheck`

   For more information about these properties, see the Quartz Job Scheduler documentation.

   For the steps to set a property, see the Administration Console online help for the release of Oracle WebLogic Server that you are using:

2. Restart each managed server for which you set a property in the previous step.

   For more information, see Use the WebLogic Server Administration Console to Start Managed Servers.

6 Scale an Oracle Java Cloud Service Instance

Scaling lets you add or remove resources for an Oracle Java Cloud Service instance on demand in response to changes in load on the service instance.

Topics:

- About Scaling an Oracle Java Cloud Service Instance
- Overview of Scaling Tasks for an Oracle Java Cloud Service Instance
- Scale Out an Oracle Java Cloud Service Cluster
- Scale In a Cluster
- Scale an Oracle Java Cloud Service Node
- Scale Automatically
- View Scaling Requests

About Scaling an Oracle Java Cloud Service Instance

You can scale Oracle Java Cloud Service instance by scaling a cluster or a node. You cannot add a second data grid cluster.
Determine what you need to scale from metrics associated with the service instance. For example, if response times are long, consider scaling out the cluster. Or if memory usage is high, consider scaling up the nodes in the cluster.

You cannot scale a service instance if the service instance is under maintenance, such as during patching or backup.

> **Note:**
>
> If you have a non-metered subscription rate and attempt to use capacity above this rate (called "bursting"), you will shift to the "Pay as You Go" model and be charged per hour and billed monthly in arrears for the increased capacity. Pricing for the increased capacity is based on the current per hour list price for the service, which you can find on the Pricing tab at http://cloud.oracle.com/*<your-service>*. Your total capacity (subscription rate plus "bursting") will not exceed two times (2x) your subscription rate. For example, if you purchased a subscription that allows 4 OCPUs per month, your bursting would be capped at a total of 8 OCPUs for that service.

Topics:

- About Scaling an Oracle Java Cloud Service Cluster
- About Scaling an Oracle Java Cloud Service Node
- About Automatic Scaling

About Scaling an Oracle Java Cloud Service Cluster

Add nodes to or remove nodes from a cluster in your Oracle Java Cloud Service instance, in response to changes in the load on the cluster.

Note the following considerations:

- Scaling a cluster is **not** supported by Oracle Java Cloud Service—Virtual Image instances.
- Scaling out the Coherence data tier cluster requires a different procedure. See Scale Out a Coherence Data Grid.
- If the cluster contains any Managed Servers that were created outside of Oracle Java Cloud Service (for example, by using the WebLogic console), before scaling out the cluster, delete those servers manually or select **Force Remove** during the Remove Node process.
- During a scale-out operation, any custom software installations or file system changes on existing nodes are not automatically propagated to the new node.
- If the service instance was provisioned in a region and assigned reserved IP addresses, you must assign a reserved IP address for the new node.

Topics:

- About Scaling Out a Cluster
- About Scaling In a Cluster
- About Adding a New Cluster

About Scaling Out a Cluster

Scaling out a cluster adds one node to the cluster.

Before scaling out a cluster, ensure that all these conditions are met:

- You have the Service Administrator role.
- The service instance is **not** under maintenance.

If any of these conditions is not met, the scaling operation fails and the service logs an error message.

The service logs a message when scaling out is started or completed, or when a failure is detected. .

If an attempt to scale out a cluster fails, the service does the following:

- Logs any diagnostic information.
- Sets the status of the service instance to `RUNNING` to allow other operations to continue.
- Returns the service instance to its original shape.
- Deletes the node that it created.

About Scaling In a Cluster

Scaling in a cluster removes the selected node from the cluster. You cannot scale in a cluster that contains a single node. If you no longer require that node, you must delete the entire service instance.

By default, the service scales in a cluster gracefully by shutting down any Oracle software running on the node. To ensure that the node is removed even if it is unresponsive, you can choose to forcibly scale in a cluster.

If an attempt to scale in a cluster fails, the service does the following:

- Logs any diagnostic information.
- Sets the status of the service instance to `RUNNING` to allow other operations to continue.
- Cleans up any stale resources.

About Adding a New Cluster

In some cases, you might want to scale out an instance but a cluster doesn't exist.

You can add a cluster to an instance by using the REST API and include the `createClusterIfMissing=true` query parameter on the scale out REST endpoint. This operation will not only add the cluster but will scale it out to include a single node.

About Scaling an Oracle Java Cloud Service Node

You can scale a node in an Oracle Java Cloud Service instance to change its compute shape in response to changes in workload, or to add block storage to a node that is running out of storage. However, you **cannot** remove block storage from a node.

You must scale each node in a cluster individually. You cannot scale all nodes in a cluster in a single operation.

You can scale the Administration Server node and Managed Server nodes in a WebLogic Server cluster. You can also scale load balancer nodes (Not supported on Oracle Cloud at Customer). Oracle Java Cloud Service does not support scaling for nodes that make up the Coherence data tier.

> **Note:**
>
> You can scale a node **only** if a version of Oracle Java Cloud Service that supports scaling a node was used to create your service instance. If the version used to create your Oracle Java Cloud Service instance does not support scaling a node, you **cannot** scale a node.

Topics:

- About Changing the Compute Shape of a Node
- What Happens When a Node is Being Scaled
- What Happens After a Node is Scaled
- About Adding Block Storage to a Node

About Changing the Compute Shape of a Node

You can scale a node in an Oracle Java Cloud Service instance to change its compute shape. You must scale each node in a cluster individually. You cannot scale all nodes in a cluster in a single operation.

You can change the compute shape of a node to adjust capacity in response to changes in workload. The compute shape specifies the number of Oracle Compute Units (OCPUs) and amount of memory (RAM) that you want to allocate to the node.

Some services provide a set of compute shapes that are optimized for different use cases. Choose from a set of all-purpose and memory-intensive shapes. The larger the compute shape, the greater the processing power. For more information about the compute shapes and considerations for selecting the shape that is right for your environment, see:

- Overview of the Compute Service in the Oracle Cloud Infrastructure Services documentation
- About Shapes in *Using Oracle Cloud Infrastructure Compute Classic*

For service instances in Oracle Cloud Infrastructure, scaling a node is only allowed within the same shape series and family. The shape series supported are `VM.Standard` and `BM.Standard`. In the `VM.Standard` series, the shape families supported are:

- `VM.Standard1.x`
- `VM.Standard2.x`
- `VM.Standard.E2`

For example, you cannot scale up from `VM.Standard1.1` to `VM.Standard2.2`. However, you can scale up from `VM.Standard1.1` to `VM.Standard1.2`.

In general, Oracle recommends that the compute shapes of all nodes in a cluster are the same in order to optimize performance. To meet the demands of heavier workloads, scale up by choosing a larger compute shape. To save costs if the workload is lightened, scale down by choosing a smaller compute shape.

What Happens When a Node is Being Scaled

Learn about the access, storage volumes, and others when you are scaling the node.

While a node is being scaled:

- You can't SSH to the node, and apps running on it can't be accessed
- You can't run any admin operations on the instance
- The storage volumes are detached, but remain intact

While Oracle Java Cloud Service is applying your changes, it puts the service instance into Maintenance mode, changes the state of the node to Configuring, and stops any servers running on the node. After applying your changes, Oracle Java Cloud Service starts any servers that should run on the node. At any time during the scaling process, you can check its status by clicking ☰ next to the instance name and the selecting **View Activity**. The Activity page will open and you can see the scale-in status in the activity table.

See Scale an Oracle Java Cloud Service Node.

What Happens After a Node is Scaled

Learn about the access, storage volumes, and others after you scale a node.

After the node is scaled:

- The public IP address of the node won’t change
- The node might get a different private IP address
- The storage volumes are re-attached automatically

If you scale out a cluster in a service instance after scaling any of its nodes, the new node has the compute shape and the amount of storage with which the service instance was originally created. To ensure that all nodes in your cluster have the same shape and storage, you must scale the new node to match the other nodes in your cluster.

About Adding Block Storage to a Node

You can add block storage to a node in the virtual machine. When you add storage to a node that does not have any user-defined partition, a storage volume is created and attached to the node. But, if you subsequently add new storage volumes on the user-defined partition, the existing storage volume expands.

Note:

You **cannot** remove block storage from a node.

Caution:

Before adding storage, Oracle recommends that you back up the service instance to avoid the risk of data loss.

The new storage volume created by scaling remains attached and available to the node even when the service instance is restarted or is stopped and then started. Also, this storage volume exists until you delete the service instance, at which time the storage volume is also deleted.

Storage limits are described here:

Region Type	Storage Limits
Oracle Cloud Infrastructure Classic	You can perform up to 6 add-storage operations. Each time, you can add from 1 to 2048 GB.
Oracle Cloud Infrastructure	You can run up to 29 add-storage operations. In each operation, you can add capacity in 50-GB multiples up to a max of 2000 GB.

About Automatic Scaling

You can configure an Oracle Java Cloud Service instance to automatically scale a cluster in or out by defining an auto-scaling rule. These rules are based on CPU and/or memory utilization and determine when to add or remove nodes.

Topics:

- How It Works
- Prerequisites
- What Are the Rule Components

How It Works

Learn to define a service and the conditions under which it should automatically add or delete a node.

You can define to a service the conditions under which it should automatically add or delete a node. These conditions are called rules (or a "policy") and are applicable to all the clusters in your service. The key component of the rules is the *metric threshold* that you set. You can choose for these metrics a percentage of CPU or memory utilization over a defined interval or the total gigabytes of memory consumed. Once the metric threshold is crossed, auto-scaling receives an alert and either adds or removes a single node, depending up the rule.

After that scaling operation succeeds, auto-scaling goes into a user-defined "cool down" period until the CPU utilization dips below the metric threshold or the cluster size reaches the user defined maximum/minimum cluster size. After cool down, if the alarm is still active (that is, if CPU utilization is still over—or below, depending on the scaling rules—the metric threshold), the service repeats the scaling operation until the CPU utilization dips below the metric threshold.

Prerequisites

Learn about the prerequisites to automatically scale a cluster in or out by defining an auto-scaling rule.

Before auto-scaling occurs, the system checks to ensure that the following prerequisites are met:

- The rule is active.
- The node is configured to handle the conditions of the rule.
- For a scale out, the current cluster size must be smaller than the maximum cluster size defined in the rule
- For a scale in, the current cluster size must be larger than the minimum cluster size defined in the rule.

What Are the Rule Components

Learn about the metric threshold, which consists of either the average, minimum, or maximum percentage of CPU usage,and others in the auto-scaling rule .

In addition to the metric threshold, which consists of either the average, minimum, or maximum percentage of CPU usage, the auto-scaling rule is composed of:

- The scaling operation and, depending on that operation, maximum or minimum cluster size.
- The number of consecutive times per a specific period the threshold must be crossed to trigger an alarm.
- Whether the rule applies to any or all VM instances.
- The duration of the cool-down period.

Overview of Scaling Tasks for an Oracle Java Cloud Service Instance

Perform scaling tasks for an Oracle Java Cloud Service instance as required.

The following table provides one or more links to information about how to perform each task by using the web-browser-based Oracle Java Cloud Service administration console.

Task	More Information
Add a node to a cluster; that is, scale out an Oracle Java Cloud Service cluster to add one node to it.	Scale Out a Cluster
Remove a node from a cluster; that is, scale in an Oracle Java Cloud Service cluster by removing a selected node from it.	Scale In a Cluster
Change the shape of a node or the storage attached to it in response to changes in workload or to add storage to a node that is running out storage.	About Scaling an Oracle Java Cloud Service Node
Scale the Coherence data tier. When Oracle Coherence is enabled for a service instance: you can increase or decrease the Coherence cache capacity for an Oracle Java Cloud Service—Coherence instance.	Scale Out a Coherence Data Grid Scale In a Coherence Data Grid
Configure automating scaling. Automatic scaling scales a cluster in or out by defining an auto-scaling rule that determines when to add or remove nodes. You can then edit or delete existing rules as circumstances dictate.	Scale Automatically
Monitor scaling operations. You can view scaling requests to check the status of ongoing scaling requests, and the success or failure of previous requests.	View Scaling Requests

Scale Out an Oracle Java Cloud Service Cluster

Scale out an Oracle Java Cloud Service cluster to add nodes in response to changes in the load on the cluster.

Topics:

- Scale Out a Cluster

- Add a New Cluster to an Instance

Scale Out a Cluster

To increase resources in response to larger workloads, you can scale out an Oracle Java Cloud Service instance by adding a node.

The new node has the compute shape that you specified when you created the service instance.

If backups are configured for the service instance, Java Cloud Service attempts to create a backup before scaling the instance.

Wait for any maintenance operations on this service instance to complete, such as backup, restoration or patching operations, before you begin.

1. Access your service console.
2. Click the name of the service instance to which you want to add a node.
3. On the Overview page, click **Add a node to this instance** +, and then select **Add Node**.
4. From the Scale Out dialog, select the component you want to scale out. If you are scaling out WebLogic Server (WLS) and your service instance has multiple clusters, select the cluster to which you want to add the node.
5. Click **Scale Out**.
6. Click **Refresh** ⟳ until the node appears on the Overview page, and the status of the node indicates that the scaling operation is completed.

   You can also monitor the progress of the scaling operation from the **Activity** page.

Add a New Cluster to an Instance

You can add a new cluster when scaling out by using the REST API.

In some cases, you might want to scale out an instance but a cluster doesn't exist. You can add a cluster to an instance by using the REST API to add a managed server and include the `createClusterIfMissing=true` parameter in the REST call. This will not only add the cluster but will scale it out to include the new node.

Scaling out by using the REST API, including how to use `createClusterIfMissing`, is discussed in greater detail in Scale Out a Service Instance in *REST API for Oracle Java Cloud Service*.

Scale In a Cluster

To reduce resource usage in response to smaller workloads, you can scale in an Oracle Java Cloud Service instance by removing a node.

You can perform a scale-in operation for a service instance that has at least two nodes. To remove all nodes, you must delete the service instance.

If backups are configured for the service instance, Java Cloud Service attempts to create a backup before scaling the instance.

Wait for any maintenance operations on this service instance to complete, such as backup, restoration or patching operations, before you begin.

1. Access your service console.
2. Click the name of the service instance from which you want to remove a node.
3. Under **Resources**, beside the node that you want to remove, click **Manage this node** ☰, and then select **Remove Node**.
4. Optional: To perform a scale-in operation even if the node is unresponsive, select **Force scale in the VMs**.
5. Click **Remove Node**.
6. Periodically click **Refresh** ⟳ until the node no longer appears on the Overview page.

   You can also monitor the progress of the scaling operation from the **Activity** page.

Scale an Oracle Java Cloud Service Node

Scaling an Oracle Java Cloud Service node allows you to change its compute shape in response to changes in workload or to add storage to a node that is running out of storage. However, you cannot remove block storage from a node.

Topics:

- Scale a Node
- Add Storage to a Node

Scale a Node

To respond to changes in workload requirements in an Oracle Java Cloud Service instance, you can scale up a node to a larger compute shape with more Oracle Compute Units (OCPUs) and memory, or scale down a node to a smaller compute shape.

> **Note:**
>
> - In Oracle Cloud Infrastructure, scaling a node is only allowed within the same shape series and family. For example, you cannot scale up or down between `VM.Standard1.1` and `VM.Standard2.2`. However, scaling within `VM.Standard1.x` is permitted, such as scaling between `VM.Standard1.2` and `VM.Standard1.4`.
> - The **Bring Your Own License** (BYOL) option enables you to bring your on-premises Oracle WebLogic Server licenses to Oracle Cloud. BYOL instances are billed at a lower rate than other instances. See Frequently Asked Questions: Oracle BYOL to PaaS. Before you scale up or scale out a BYOL instance, you must have enough WebLogic Server licenses for the additional OCPUs that will be allocated to the instance after it is scaled.

For example:

- Changing the compute shape of a node in Oracle Cloud Infrastructure from `VM.Standard2.2` to `VM.Standard2.4` doubles the capacity of the node from two OCPUs to four OCPUs, and also doubles the amount of RAM allocated to the node.
- Changing the compute shape of a node in Oracle Cloud Infrastructure Classic from `OC3` to `OC4` doubles the capacity of the node from one OCPU to two OCPUs, and also doubles the amount of RAM allocated to the node.

You must scale each node individually. To optimize performance, Oracle recommends that you scale all nodes within a cluster to the same specifications.

The applications that are running on the node will be temporarily unavailable while the scaling operation is in progress.

Wait for any maintenance operations on this service instance to complete, such as backup, restoration or patching operations, before you begin.

1. Access your service console.
2. Click the name of the service instance that contains the node that you want to scale.
3. Under **Resources**, beside the node that you want to scale, click **Manage this node** ☰, and then select **Scale Up/Down**.
4. Select a new **Compute Shape**.
5. Click **Yes, Scale Up/Down VM**.
6. To check the status of the scaling operation, periodically click **Refresh** ⟳.

   You can also monitor its progress from the **Activity** page.

Add Storage to a Node

You can add more block storage to individual nodes in an Oracle Java Cloud Service instance. You can update an existing storage volume that is already attached to a node, or you can create a new storage volume.

> **Note:**
>
> Do not use Oracle Cloud Infrastructure Compute to attach custom storage volumes to a service instance's nodes. Any custom storage volumes that you attach will be detached if the service instance is restarted.

You must update the storage configuration of each node individually. You cannot remove unused storage from a node.

You cannot perform any other management operations on the service instance while the storage operation is in progress. The applications that are running on the node will be temporarily unavailable.

You cannot add storage to a node while the service instance is under maintenance, such as during a patching or backup operation.

1. Access your service console.

2. Click the name of the service instance that contains the node to which you want to add storage.
3. Under **Resources**, beside the node that you want to update, click **Manage this node** ☰, and then select **Add Storage**.
4. In the Add Storage dialog box, select one of the existing storage volumes. Or, to add a new storage volume to this node, select the **Additional Partition** option.
5. Enter the number of Gigabytes (GB) that you want to add to this volume.

   For instances in Oracle Cloud Infrastructure, the minimum size of a volume added by scaling a node is 50 GB. For instances in Oracle Cloud Infrastructure Classic, the minimum size is 1 GB.
6. Click **Yes, Add Storage**.
7. To check the status of the scaling operation, periodically click **Refresh** ⟳.

   You can also monitor its progress from the **Activity** page.

Scale Automatically

This topic does not apply to Oracle Cloud Infrastructure. Identify the Cloud Infrastructure Used by a Service Instance.

Automatic scaling (auto-scaling) allows you to configure a set of rules–or “policies”–that determine when to add or remove nodes from a cluster and the minimum and maximum number of nodes that can be added to a cluster. You can create rules, edit them or delete them.

The auto-scaling feature allows you to:

- Create a Scaling Rule
- Edit a Scaling Rule
- Delete a Scaling Rule

Create a Scaling Rule

This topic applies only to Oracle Cloud Infrastructure Classic.

Oracle Java Cloud Service can monitor a service instance and, when certain workload conditions are true, perform scaling operations without any user intervention.

> **Note:**
>
> The **Bring Your Own License** (BYOL) option enables you to bring your on-premises Oracle WebLogic Server licenses to Oracle Cloud. BYOL instances are billed at a lower rate than other instances. See Frequently Asked Questions: Oracle BYOL to PaaS. Before you scale up or scale out a BYOL instance, you must have enough WebLogic Server licenses for the additional OCPUs that will be allocated to the instance after it is scaled.

The following types of scaling operations can be performed automatically by using scaling rules:

- Add a new node to the instance (scale out) when the average memory utilization on all nodes is greater than or equal to a specified percentage.
- Remove a node from the instance (scale in) when the maximum amount of memory in use on all nodes is less than a specified size.

Each service instance can be associated with at most one scale-in rule and one scale-out rule.

Wait for any maintenance operations on this service instance to complete, such as backup, restoration or patching operations, before you begin.

1. Access your service console.
2. Click the name of the service instance for which you want to create a scaling rule.
3. At the top of the Overview page, click **Manage this instance** ☰, and then select **Define Auto Scaling Rules**.

   If there are no auto-scaling rules defined, the Rules page displays the message `Auto scaling not configured for this service`.
4. On the Rules page, click **New Rule**.
5. In the New Rule dialog box, define the new rule.

Field	Description
Perform [ ]	Select the type of scaling operation to perform: • **Scale-in** - Remove a node from the cluster. • **Scale-out** - Add a node to the cluster.
to Maximum (Minimum) Cluster Size of [ ]	Enter the cluster size limit: • For **Scale-in** rules, this is the minimum number of nodes that must be present in the cluster after a scaling operation. • For **Scale-out** rules, this is the maximum number of nodes that can be present in the cluster after a scaling operation.
whenever [ ]	Select the basis for calculating the rule threshold: • **Average** CPU or memory usage • **Maximum** CPU or memory usage • **Minimum** CPU or memory usage
of [ ]	Select the type of metric upon which the scaling operation is triggered: • **CPU Utilization** - The percentage of total CPU in use • **Memory Utilization** - The percentage of total memory in use • **Memory** - The amount of memory in use
is >= [ ] % (GB) or **is <= [ ] % (GB)**	Enter the threshold value of the selected metric that, when met or exceeded, will trigger the scaling operation. • For **CPU Utilization** or **Memory Utilization**, enter a percentage between 0 and 100. • For **Memory**, enter a value in Gigabytes.
for at least [ ]	Enter the minimum number of times the threshold condition must be detected before the scaling operation is triggered.

Field	Description
consecutive period(s) of [ ] minutes	Enter the number of minutes during which the threshold condition must be detected before the scaling operation is triggered.
on [ ] instances	Select one of these options: • **Any** - The threshold condition can be detected on any node in this service instance before the scaling operation is triggered. • **All** - The threshold condition must be detected on all of the nodes in this service instance before the scaling operation is triggered.
and wait for [ ] minutes of cool down period	Enter the minimum number of minutes for which Oracle Cloud waits before it reevaluates this scaling rule.

6. Click **Create**.
7. When prompted for confirmation, click **OK**.
8. Periodically click **Refresh** ⟳ until the rule status indicates that the rule configuration is completed.

Edit a Scaling Rule

This topic applies only to Oracle Cloud Infrastructure Classic.

You can edit an existing scaling rule for Oracle Java Cloud Service, to modify the conditions under which it is triggered or the scaling actions that it performs.

1. Access your service console.
2. Click the name of the service instance that defines the scaling rule.
3. At the top of the Overview page, click **Manage this instance** ☰, and then select **Define Auto Scaling Rules**.
4. On the Rules page, for the rule you want to edit, click **Actions** ☰, and then select **Edit**.
5. In the Edit Rule dialog box, edit the rule definition as necessary.
6. Click **Update**.
7. When prompted for confirmation, click **OK**.
8. Periodically click **Refresh** ⟳ until the rule status indicates that the rule configuration is completed.

Delete a Scaling Rule

This topic applies only to Oracle Cloud Infrastructure Classic.

You can delete a scaling rule from an Oracle Java Cloud Service instance when it is no longer required.

1. Access your service console.

2. Click the name of the service instance that defines the scaling rule.
3. At the top of the Overview page, click **Manage this instance** ☰, and then select **Define Auto Scaling Rules**.
4. On the Rules page, for the rule you want to delete, click **Actions** ☰, and then select **Delete**.
5. When prompted for confirmation, click **Yes**, and then click **OK**.
6. Periodically click **Refresh** ↻ until the rule no longer appears on the Rules page.

View Scaling Requests

Check the status of ongoing scaling requests for an Oracle Java Cloud Service instance, and the success or failure of previous requests.

> **Note:**
>
> The initial scale-out activity for an Oracle Java Cloud Service—Coherence instance is the initial request to add Managed Servers for the Coherence data tier when the service instance was first created.

To view ongoing or past scaling requests:

1. Navigate to the Administration or Overview pages for the specific service.
2. Click the service menu (☰) at the top of the page and select **View Activity**.

   The Activity page of the Platform Services Console opens.
3. In the Search Activity Log panel, enter the necessary search criteria:

Option	Description
Start Time Range	The date and time range within which you want to see scaling activity.
Operation Status	The status of the scaling activity you want to view. This option will filter out all scaling activity **not** in this status. To see all status, select **All**.
Service Name	The name of the service instance for which you want to see scaling activity.
Service Type	This should be **Java Cloud Service**.
Operation	The operation for which you want to see activity. In the case of scaling, select any or all of these options: • **Scale Application** • **Scale In** • **Scale Out** • **Scale Up/Down**

4. Click **Search**.

   All scaling activity that meets the search criteria appears in the results table. The Operation Status column will indicate whether the scaling operation succeeded or failed. To see more details about a specific operation, expand the row by clicking the Expand button in the first column.

7

Back Up and Restore an Oracle Java Cloud Service Instance

You can back up and restore your Oracle Java Cloud Service instances to return their software and data to a particular state.

Topics:

- About Backup and Restoration in Oracle Java Cloud Service
- Typical Workflow for Backing Up and Restoring a Service Instance
- Add a Backup Configuration to an Oracle Java Cloud Service Instance
- Configure Scheduled Backups for an Oracle Java Cloud Service Instance
- Create an On-Demand Backup
- Enable or Disable Backups
- Delete a Backup
- Restore a Backup
- Restore the Database for a Service Instance
- Return an Instance to Service After Restoration from a Backup
- Explore the Backup Page

About Backup and Restoration in Oracle Java Cloud Service

Learn how Oracle Java Cloud Service backups are initiated, what backups contain, where backups are stored, and how long backups are retained.

Topics

- What are the Contents of a Backup?
- When Do Backups Occur?
- Where are Backups Stored?
- How Long are Backups Retained?
- What Happens When a Backup is Restored?

What are the Contents of a Backup?

Java Cloud Service ensures that backups contain only the data that is needed for a proper restoration of a service instance.

A backup does not include software installations and other binary files. Java Cloud Service can also take a backup of the database that’s associated with the service instance, if desired.

There are two types of backups. A **full backup** contains all of the artifacts and configuration data that are required to restore a service instance. An **incremental backup** contains only those changes to the configuration data since the last scheduled full backup. Each incremental backup is linked to the last scheduled full backup that was performed before the incremental backup. You cannot delete a full backup that is linked to incremental backups without also deleting the incremental backups

When Do Backups Occur?

Backups run automatically, and you can also run them as needed.

If backups are configured for a service instance, they are **scheduled** to occur automatically. You can also initiate an **on-demand** backup immediately without having to wait for the next scheduled backup.

By default, scheduled backups occur at the following times:

- Full backups are initiated weekly starting 12 hours after backups were enabled on a service instance, rounded to the nearest five-minute interval.

  For example, if a service instance is created with backups at 1:01 PM on a Monday, full backups are initiated at 1:00 AM on Tuesdays.
- Incremental backups are initiated every day, except the day of a full backup, at the same time that full backups are initiated.

  For example, if a service instance is created with backups at 1:01 PM on a Monday, incremental backups are initiated at 1:00 AM every day except Tuesdays.

You can change the default schedule on which automated backups are initiated. Backups do not occur while the instance is stopped. After you start an instance, the next backup occurs at the scheduled time.

After configuring backups for a service instance, you can enable or disable the backup service as needed.

If you provided an email address when you provisioned the service instance, Java Cloud Service automatically disables scheduled backups on a service instance after three consecutive failures of the same type occur with scheduled backups. If you did not specify an email, Java Cloud Service automatically disables backups after three consecutive failures of the same type or seven consecutive failures of any type occur with scheduled backups.

If you provided an email address, both you and the account administrator will receive an email notification when backups are automatically disabled. A reminder notification is sent every week until backups are enabled again.

The activity logs show the last failed backup and all previous failures. You will see that backups are disabled, but you will not see whether you disabled them or they were disabled automatically due to errors.

Where are Backups Stored?

Backups are recorded to multiple locations.

Java Cloud Service records all backups to a specified destination in Oracle Cloud object storage. To speed up restorations, recent backups are also copied to a dedicated storage volume that’s attached to a node in your service instance.

Note:

Do not attempt to download the backup files generated by Java Cloud Service. These files are encrypted and not accessible offline. You must use Java Cloud Service to restore a service instance from a backup.

How Long are Backups Retained?

After completing a scheduled backup, Java Cloud Service deletes any backups in object storage (and any local copies of backups) that are due to be deleted.

By default, backups are retained in object storage for the following time periods:

- Scheduled incremental backups are retained for **30 days**.
- Scheduled full backups are retained until all the incremental backups to which they are linked are no longer available.
- On-demand backups are retained for 30 days, unless you choose to keep a backup forever (it will not expire and will not be deleted automatically).

You can change the default retention policy for backups. You can also manually delete backups that you no longer require.

What Happens When a Backup is Restored?

Java Cloud Service restores your service instance’s configuration from a selected backup.

You can also choose whether to restore the Oracle software (binary files) to its current official patch level, or to leave the software unchanged. Restore the Oracle software to undo a change to the software that you don’t want. For example:

- You accidentally modified or deleted some files in the software installation.
- You installed a patch to the software and it no longer works as required.

If you choose to restore the software, it is restored from an image maintained internally to Oracle Cloud. The software is not restored to the point in time at which the backup was created.

During the restoration process, Java Cloud Service shuts down the server processes that are running in the service instance. After the restoration is complete, these processes are restarted.

If you performed a scaling operation after the backup was created, the topology of the service instance and the topology of the backup might not be the same. The restore operation will not remove nodes that you added after the backup was created, or add nodes that you removed after the backup was created.

Typical Workflow for Backing Up and Restoring a Service Instance

To back up and restore an Oracle Java Cloud Service instance, consider this typical workflow.

Task	Description	More Information
Add a backup configuration to a service instance	If you created an Oracle Java Cloud Service instance without configuring backups, you must add a backup configuration to the instance before performing other backup tasks.	Add a Backup Configuration to an Oracle Java Cloud Service Instance
Configure backups for a service instance	Customize the following properties of backups for a service instance: • When backups are performed • Where backups are stored • How long new backups are retained	Configure Scheduled Backups for an Oracle Java Cloud Service Instance
Initiate an on-demand backup of a service instance	Create a backup immediately without having to wait for the next scheduled backup.	Create an On-Demand Backup
Delete a backup	Delete a backup that you no longer require to release storage or prevent an Oracle Java Cloud Service instance from being restored from the backup.	Delete a Backup
Restore a service instance from a backup	Undo configuration changes you don’t want by returning an Oracle Java Cloud Service instance to a particular state.	Restore a Backup
Restore the database from a backup	Restore the Oracle database that's associated with your Oracle Java Cloud Service instance.	Restore the Database for a Service Instance
Return an instance to service after restoration from a backup	If necessary, manually modify a restored Oracle Java Cloud Service instance to ensure that it is in a consistent and valid state.	Returning an Oracle Java Cloud Service Instance to Service After Restoration from a Backup
Access files within a backup	While testing or troubleshooting, use the REST API to extract the contents of an existing backup on the Administration Server node.	Access the Contents of a Backup

Add a Backup Configuration to an Oracle Java Cloud Service Instance

If you did not configure backups when you created an Oracle Java Cloud Service instance, you can manually add a backup configuration to the existing instance after the service instance is provisioned.

Backups are already configured on a service instance if you set the **Backup Destination** to a value other than `None` when you created the service instance.

Some configuration options vary depending on the type of region in which your service instance was created. See Identify the Cloud Infrastructure Used by a Service Instance.

1. Access your service console.

2. Click the name of the service instance for which you want to configure automated backups.

3. On the Overview page, click **Manage this instance** ☰ beside the instance name, and then select **Enable Backups**.

4. For **Backup Destination**, select the location(s) to which backups will be stored.

5. For **Cloud Storage Container**, enter the object storage location where backups of the service instance will be stored.

    - If your service instance is running in an **Oracle Cloud Infrastructure** region, then enter the URL of an existing bucket in Oracle Cloud Infrastructure Object Storage. See Prerequisites for PaaS Services on Oracle Cloud Infrastructure in the Oracle Cloud Infrastructure documentation.

      Format: `https://swiftobjectstorage.`*`region`*`.oraclecloud.com/v1/`*`namespace`*`/`*`bucket`*

      To find out your `namespace`, sign in to the Oracle Cloud Infrastructure web console, click the tenancy name, and look for the **Object Storage Namespace** field.

      Example: `https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/myCompany/myBucket`

    - If your service instance is running in an **Oracle Cloud Infrastructure Classic** region, then enter the URL of a container in Oracle Cloud Infrastructure Object Storage Classic.

      Format: *`rest_endpoint_url`*`/`*`containerName`*

      You can find the REST endpoint URL of the Oracle Cloud Infrastructure Object Storage Classic service instance in the Infrastructure Classic Console. See Finding the REST Endpoint URL for Your Cloud Account in *Using Oracle Cloud Infrastructure Object Storage Classic*.

      Example: `https://acme.storage.oraclecloud.com/v1/MyService-acme/MyContainer`

6. For **Username** and **Password**, enter the credentials of a cloud user who has access to the specified object storage location.

    - If your service instance is running in an **Oracle Cloud Infrastructure** region, then enter the generated Auth Token for the user that you specified. See Prerequisites for PaaS Services on Oracle Cloud Infrastructure in the Oracle Cloud Infrastructure documentation.

    - If your service instance is running in an **Oracle Cloud Infrastructure Classic** region, then these fields are not displayed if this account includes Oracle Identity Cloud Service, and the current cloud user is entitled to use Oracle Cloud Infrastructure Object Storage Classic.

7. If the Oracle Cloud Infrastructure Object Storage Classic container that you specified doesn’t exist, or if you aren’t sure whether it exists, then select **Create Cloud Storage Container**. If the container doesn’t exist, it will be created automatically.

   This option is not relevant to Oracle Cloud Infrastructure regions. The specified Oracle Cloud Infrastructure Object Storage bucket must be created prior to adding a backup configuration.

8. Click **Enable Backups**.

Configure Scheduled Backups for an Oracle Java Cloud Service Instance

You can control when the backups for an Oracle Java Cloud Service occur, how they are stored, and how long they are retained.

By default, backups are stored in the location that you specified when you enabled backups on a service instance, but you can change this storage location. If the storage user name and password that you originally specified for this service instance have changed, backups will fail until you update this configuration. Oracle Java Cloud Service automatically disables backups on a service instance if consecutive failures occur.

Before updating your backup configuration, you must enable backups on the service instance if you did not enable backups when you originally created the service instance. Similarly, if Oracle Java Cloud Service disabled backups on the service instance, you must enable backups before updating the configuration.

Some configuration options vary depending on the type of region in which your service instance was created. See Identify the Cloud Infrastructure Used by a Service Instance.

1. Access your service console.
2. Click the name of the service instance for which you want to configure backups.
3. On the Overview page, click the **Administration** tile.
4. Click the **Backup** tab.
5. Click **Manage backups for this instance** ☰, and then select **Configure Backups**.
6. For **Full Backup**, select the day of the week and the time of day UTC when you want full backups to occur.

   > **Note:**
   >
   > All times must be for the Coordinated Universal Time (UTC) time zone, not your local time zone.

7. For **Incremental Backup**, select the time of day UTC when you want incremental backups to occur each day.
8. Clear the **Coordinated Backups** check box if you do not want to take backups of the database.

   By default, if your service instance is associated with an Oracle Database Cloud Service deployment, the database deployment is automatically backed up as well. Coordinated backups are not available for service instances that are associated with other database services.
9. For **Storage Container**, enter the object storage location where backups of the service instance will be stored.

Any existing backups in the previous storage location will remain there and be available for service restoration, until the retention period has elapsed.

- If your service instance is running in an **Oracle Cloud Infrastructure** region, then enter the URL of an existing bucket in Oracle Cloud Infrastructure Object Storage. See Prerequisites for PaaS Services on Oracle Cloud Infrastructure in the Oracle Cloud Infrastructure documentation.

  Format: `https://swiftobjectstorage.`*`region`*`.oraclecloud.com/v1/`*`namespace`*`/`*`bucket`*

  To find out your `namespace`, sign in to the Oracle Cloud Infrastructure web console, click the tenancy name, and look for the **Object Storage Namespace** field.

  Example: `https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/myCompany/myBucket`

- If your service instance is running in an **Oracle Cloud Infrastructure Classic** region, then enter the URL of a container in Oracle Cloud Infrastructure Object Storage Classic.

  Format: *`rest_endpoint_url/containerName`*

  You can find the REST endpoint URL of the Oracle Cloud Infrastructure Object Storage Classic service instance in the Infrastructure Classic Console. See Finding the REST Endpoint URL for Your Cloud Account in *Using Oracle Cloud Infrastructure Object Storage Classic*.

  Example: `https://acme.storage.oraclecloud.com/v1/MyService-acme/MyContainer`

10. For **User Name** and **Password**, enter the credentials of a cloud user who has access to the specified object storage location.

    - If your service instance is running in an **Oracle Cloud Infrastructure** region, then enter the generated Auth Token for the user that you specified. See Prerequisites for PaaS Services on Oracle Cloud Infrastructure in the Oracle Cloud Infrastructure documentation.

    - If your service instance is running in an **Oracle Cloud Infrastructure Classic** region, then these fields are not displayed if this account includes Oracle Identity Cloud Service, and the current cloud user is entitled to use Oracle Cloud Infrastructure Object Storage Classic.

11. In the **Set new retention period to** field, enter the number of days that you want new backups to be retained.

    If you decrease the retention period, any existing backups that are older than this period are automatically deleted during the next scheduled backup.

12. Click **Save**.

After updating the backup configuration, Oracle recommends that you create an on-demand backup to verify your configuration.

Create an On-Demand Backup

At any time, you can create a full backup of your Oracle Java Cloud Service instance.

In addition to scheduling regular backups, Oracle recommends that you manually create a backup before and after making major changes to your service instance.

You must enable backups on the service instance, if you did not enable backups when you originally created the service instance.

Wait for any maintenance operations on this service instance to complete, such as patching or scaling operations, before you begin.

1. Access your service console.
2. Click the name of the service instance for which you want to create a backup.
3. On the Overview page, click the **Administration** tile.
4. Click the **Backup** tab.
5. Click **Manage backups for this instance** ☰, and then select **Backup Now**.
6. Select **Include Database** if you want to also back up the database that's associated with this service instance.

   This option is available only if your service instance is associated with an Oracle Database Cloud Service deployment. Coordinated backups are not available for service instances that are associated with other database services.
7. If you select **Keep Forever**, then this backup can only be deleted manually. If not selected, this backup will be deleted at the end of the current backup retention period for this service instance.

   If you select the **Include Database** option and if the Oracle Real Application Clusters (RAC) option is enabled on your database, then this option does not apply to the database backup. The database instance's retention policy determines how long the database backup is kept.
8. For **Notes**, enter up to 255 characters of text to provide additional information about the backup (for example, when to restore from this backup, why the backup was created, or the state of the service instance at the time of the backup).
9. Click **Back Up**.
10. To check the status of the backup operation, periodically click **Refresh** ⟳.

Enable or Disable Backups

You can enable or disable backups for an Oracle Java Cloud Service instance.

Backups are enabled on a service instance if you set the **Backup Destination** to a value other than `None` when you created the service instance. To configure backups for an existing service instance, see Add a Backup Configuration to an Oracle Java Cloud Service Instance.

If backups are disabled for a service instance:

- Scheduled backups do not run.
- You cannot create an on-demand backup.
- You cannot restore an instance from a backup.
- Automated backups do not run prior to other maintenance operations, such as patching.

Backups that were previously created are not affected when backups are disabled.

Java Cloud Service automatically disables backups after consecutive failures occur with scheduled backups. When you enable backups, also update the backup configuration and correct the cause of the failures, such as an incorrect password.

1. Access your service console.
2. Click the name of the service instance for which you want to control backups.
3. On the Overview page, click the **Administration** tile.
4. Click the **Backup** tab.

   If Java Cloud Service automatically disabled backups after consecutive failures, you can identify the cause of the failures from this page.
5. Click **Manage backups for this instance** ☰, and then select **Enable Backups** or **Disable Backups**.
6. When prompted, confirm that you want to enable or disable backups.

Delete a Backup

You can delete an existing backup of an Oracle Java Cloud Service instance.

You might want to delete a backup for the following reasons:

- The backup is no longer needed.
- To prevent users from restoring the service instance from this backup
- To free up storage space

1. Access your service console.
2. Click the name of the service instance for which you want to manage backups.
3. On the Overview page, click the **Administration** tile.
4. Click the **Backup** tab.
5. Under **Available Backups**, beside the backup that you want to delete, click **Menu** ☰, and then select **Delete**.
6. If you selected a full backup that is linked to one or more incremental backups, select the check box to confirm that both the full and incremental backups will be deleted.
7. Click **Delete**.

Restore a Backup

You can restore an Oracle Java Cloud Service instance to a previous state.

Java Cloud Service restores your instance's configuration from a selected backup. You can also choose whether to restore the Oracle software (binary files) to its current official patch level, or to leave the software unchanged. If you choose to restore the software, it is restored from an image maintained internally to Oracle Cloud. The software is not restored to the point in time at which the backup was created.

If you performed a scaling operation after the backup was created, the topology of the service instance and the topology of the backup might not be the same. The restore operation can automatically remove nodes that you added to the service instance after the backup was created. However, the restore operation cannot add missing nodes to the service instance that were removed after the backup was created.

Java Cloud Service does not restore the database associated with your service instance. Prior to restoring a backup, you must restore from the corresponding database backup as identified by its RMAN tag or timestamp.

Wait for any maintenance operations on this service instance to complete, such as patching or scaling operations, before you begin.

1. Access your service console.
2. Click the name of the service instance that you want to restore.
3. On the Overview page, click the **Administration** tile.
4. Click the **Backup** tab.
5. Under **Available Backups**, beside the backup that you want to restore, click **Menu** ☰, and then select **Restore**.
6. If you changed the password of the infrastructure database schema after creating the selected backup, enter the current **Schema Password**.
7. If the backup contains fewer nodes than the current number of nodes in the service instance, then the **Force scale in** check box is selected for you automatically.

   Oracle Java Cloud Service removes these nodes from the service instance prior to performing the restore operation. Alternatively, you can remove the nodes in your service instance that are not found in the backup, and then return to this dialog.
8. If you want to also restore the Oracle software in this service instance to the current official patch level, select **Restore binary files**.
9. For **Notes**, enter any free-form text to provide additional information about the restoration. For example, describe why you are restoring the service instance.
10. Click **Restore**.
11. When prompted for confirmation, perform one of the following steps:
    - If the selected backup has an associated database backup, select the check box to confirm that you have already restored the database, and then click **Continue with Restore**.
    - Click **Yes, Restore Service**.
12. To check the status of the restore operation, periodically click **Refresh** ⟳.

Restore the Database for a Service Instance

Restore the infrastructure schema database for an Oracle Java Cloud Service instance to a previous state.

When you restore an Oracle Java Cloud Service instance from a backup, Oracle Java Cloud Service does not restore the Oracle database associated with your service instance. You must restore from the corresponding database backup as identified by its RMAN tag or timestamp.

The restoration procedure varies with each database service in Oracle Cloud.

Database Service	Documentation
Oracle Database Cloud Service	• Restoring from a Specific Backup • Restoring to a Specific Point in Time
Oracle Cloud Infrastructure Database	• Recovering a Database from Object Storage • Recovering an Exadata Database
Oracle Autonomous Database (Oracle Autonomous Transaction Processing)	Restoring an Autonomous Database
Oracle Database Exadata Cloud Service	• Restoring from a Specific Backup • Restoring to a Specific Point in Time

Return an Instance to Service After Restoration from a Backup

After restoring an Oracle Java Cloud Service instance from a backup, you may need to perform additional steps to return it back to a valid, consistent state.

If a service instance has been scaled since a backup was created, the topology of the service instance and the topology of the backup no longer match. If you restore the service instance's configuration files from the backup, how Oracle Java Cloud Service handles the mismatch depends on the cause of the mismatch.

You may also need to remove any records from the domain’s transaction log in the restored service instance.

To return an Oracle Java Cloud Service instance to service:

1. On the Backup Page, click the text **Status Completed** for the last successful restoration in the Restoration History of the service instance.

   A set of progress messages for the restoration is displayed.

2. Examine the progress messages to determine whether the backup contained any hosts that are not in the service instance.

3. If the backup contains additional nodes that are not in the current service instance, modify the service instance as follows:

   a. Use Oracle WebLogic Server to remove the managed servers on these nodes from the domain configuration.

   b. If you require your service instance to contain the number of nodes in the backup, scale out the service instance.

      When Oracle Coherence is enabled for a service instance: Scale out the Coherence data tier to also match the number of nodes in the backup.

4. If you scaled out the service instance after the backup was taken, and you require your service instance to have the same number of nodes as the backup, scale in the service instance.

   When Oracle Coherence is enabled for a service instance: Scale in the Coherence data tier to also match the number of nodes in the backup.

5. Remove any records from the transaction log in the Oracle WebLogic Server domain.

   Refer to the relevant documentation for the release of Oracle WebLogic Server that your service instance is running:

   - How to Remove Transaction Records in *Developing JTA Applications for Oracle WebLogic Server 12c (12.2.1.3)*

- How to Remove Transaction Records in *Developing JTA Applications for Oracle WebLogic Server 11g (11.1.1.7)*

Access the Contents of a Backup

Extract the contents of an existing backup for an Oracle Java Cloud Service instance.

In general, Oracle Java Cloud Service backups are encrypted and not accessible offline. But if you need to access specific files within a backup, Oracle Java Cloud Service can download, decrypt, and extract the backup to your Administration Server node.

You must use the Oracle Java Cloud Service REST API or CLI to extract the contents of a backup. See Restore a Service Instance in *REST API for Oracle Java Cloud Service*.

In the request body, set `unpackAndDecryptOnly` to `true`. For example:

```
{
    "backupId": "086b01a7-9e80-4292-a2d5-1aa78e7265d7"
    "unpackAndDecryptOnly": "true"
}
```

Locate the required files on the Administration Server node under `/u01/data/backup/work`, and move them to a different location. Subsequent backup and restore operations will remove any existing files under `/u01/data/backup/work`.

If you select an incremental backup, it contains only the files that have been modified since the most recent full backup operation. In order to find a specific file or to obtain the complete contents of a specific directory, you might need to extract the full backup as well.

Explore the Backup Page

You can use the Backup page to back up and restore an Oracle Java Cloud Service instance, and to manage backups for the service instance.

What You Can Do from the Backup Page

You can perform these tasks from the Backup page:

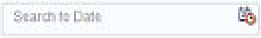
- Add a Backup Configuration to an Oracle Java Cloud Service Instance
- Configure Scheduled Backups for an Oracle Java Cloud Service Instance
- Create an On-Demand Backup
- Enable or Disable Backups
- Delete a Backup
- Restore a Backup

What You See on the Backup Page

The following table describes the key information shown on the Backup page.

Element	Description
Oracle Java Cloud Service link	Click this link to return to the Oracle Java Cloud Service Console.
☰ (in the page header)	**Menu** icon provides the following options: • **Open WebLogic Server Administration Console**—Open the WebLogic Administration Console to administer your application environment. • **Open Fusion Middleware Control Console**—Open Fusion Middleware Control to administer your application environment. • **Open Load Balancer Console**—Open the console to administer the load balancer, if a local load balancer has been configured for the service instance. Note that access to the administrative consoles is disabled by default. When you create a service instance, you can enable consoles by selecting a check box on the Details page of the instance creation wizard. For an instance this is already created, you must create an access rule in order to activate the console choices. See Enabling Console Access in an Oracle Java Cloud Service. • **Start**—Start the nodes for the Administration Server, Managed Servers, load balancer, and Managed Servers on the Coherence data tier (if provisioned). • **Stop**—Stop the nodes for the Administration Server, Managed Servers, load balancer, and Managed Servers on the Coherence data tier (if provisioned). • **Restart**—Stop and then immediately restart all the nodes in the service instance. • **Scale Out**—Adds a managed server node. • **Define Auto Scaling Rules**—Opens the Add Rule dialog box, which opens the Rules page where you can configure auto-scaling rules. • **Change License Type**—Opens the Change License Type dialog box, which enables you to choose whether to leverage your existing on-premises (BYOL) license or use your Oracle Java Cloud Service cloud license. • **Add Load Balancer**—Add a user-managed load balancer to this service instance. • **Disable/Enable Load Balancer**—Depending on the selection, either blocks access to the service instance or forwards the requests it receives from clients to the Oracle WebLogic Server Managed Servers. • **Manage Access Rules**—Create and manage rules to control access to the nodes for this service instance. • **Add SSH Access**—Add public SSH keys to the nodes that make up this service instance. • **Manage Tags/Add Tags**—Either remove or add tags to a service instance. **Manage Tags** appears if a tag already exists for the service instance. **Add Tags** appears if no tags exist for the service instance. • **Enable Backups**—Enable backups for this service instance. • **View Activity**—View all administrative activities that have been performed on your service instances. • **View Instance Metrics**—View performance metrics for this service instance.

Element	Description
Backups on Cloud Storage	The total amount of space, in megabytes or gigabytes, that backups are occupying in the Oracle Cloud Infrastructure Object Storage Classic container for storing backups. This amount includes space that is occupied by backups that have been manually uploaded to the container, if any, in addition to the space occupied by backups that Oracle Java Cloud Service has moved there.
Backup Volume Used	The total amount of space, in megabytes or gigabytes, that local copies of backups are occupying in the backup volume on the block storage of the virtual machine where the Administration Server is running.
Backup Volume Used (%)	The percentage of the available space that backups are occupying in the backup volume on the block storage of the virtual machine where the Administration Server is running.
Incremental Backups	Indicates the schedule for running incremental backups. For information about configuring the incremental backup schedule, see Configure Scheduled Backups for an Oracle Java Cloud Service Instance.
Full Backups	Indicates the schedule for running full backups. For information about configuring the full backup schedule, see Configure Scheduled Backups for an Oracle Java Cloud Service Instance.
Most Recent Backup ⚠	Indicates that the most recent backup failed and the time of its failure. Click the icon for information about why the backup attempt was unsuccessful. Oracle Java Cloud Service automatically disables backups after consecutive failures occur with scheduled backups.
Last Successful Backup	Indicates the time of the last successful backup.
⟳	Click to refresh the page. The date and time the page was last refreshed is displayed adjacent to this button.
Available Backups	List of available backups. By default, only backups for the last seven days are listed. Use the search field to specify a range of dates for which you want backups returned.
Manage backups for this instance ☰	Select from the following options: • **Backup Now** — Create an on-demand backup of the service instance. • **Configure Backups** — Update the backup schedule and where backups are stored. • **Disable Backups** — Disable automated and on-demand backups. • **Enable Backups** — Enable automated and on-demand backups.
	Enter the start date of the period for which you want to filter the list of available backups or the restoration history. By default, the start date is set to seven days before the current date. Enter the date in the format `mm/dd/yyyy`. • *mm* is a one-digit or two-digit month number, for example, 2 for February or 10 for October. • *dd* is a number in the range 1–31 for the day of the month. • *yyyy* is a four-digit year number, for example, 2104. Alternatively, click the calendar icon to select the date from a calendar.

Element	Description
 Search to Date	Enter the end date of the period for which you want to filter the list of available backups or the restoration history. Enter the date in the format `mm/dd/yyyy`. • *mm* is a one-digit or two-digit month number, for example, 2 for February or 10 for October. • *dd* is a number in the range 1–31 for the day of the month. • *yyyy* is a four-digit year number, for example, 2104. Alternatively, click the calendar icon to select the date from a calendar. **Note:** The end date must not be earlier than the start date.
🔍	Click to filter the list of available backups or the restoration history to show only backups or restorations from within the period specified by the Search from Date field and the Search to Date field.
⌛	In-progress backup for the Oracle Java Cloud Service instance. The backup will not be available for use in restoring the service instance until it is completed. The backup is identified by the date and time when the backup operation was started, which is displayed adjacent to icon that represents the backup. Click the icon to see additional information about the backup.
✓	Completed backup for the Oracle Java Cloud Service instance. The backup is available for use in restoring the service instance. The backup is identified by the date and time when the backup was created, which is displayed adjacent to icon that represents the backup. Click the icon to see additional information about the backup, including its start date, complete date, expiration date and size. If a database backup is included, its tag or timestamp is also displayed.
ℹ	Completed backup with a warning message. Oracle Java Cloud Service tried but failed to move or delete one or more older backups. For information about when and why Oracle Java Cloud Service moves or deletes older backups, see About Backup and Restoration in Oracle Java Cloud Service. The backup is still available for use in restoring the service instance. To find out why Oracle Java Cloud Service could not move or remove the older backup, place the cursor over the icon. The presence of the older backup may cause future backups to fail because of insufficient space. For information about how to prevent future backups from failing in this way see One of my backups is showing a warning icon. The backup is identified by the date and time when the backup was created, which is displayed adjacent to icon that represents the backup. Click the icon to see additional information about the backup.

Element	Description
⚠	The backup is in the process of being deleted. Click the icon to see additional information about the backup.
Type	A comma-separated pair of words that describes the type of the backup. The first word in the pair describes the extent of the backup: • Full—The backup contains all the runtime artifacts required to restore the service instance’s configuration data. • Incremental—The backup contains changes to configuration data on all virtual machines since the last scheduled full backup. The second word in the pair indicates how the backup was initiated: • If the backup was initiated automatically at the scheduled time, the text “scheduled" is displayed. • If the backup was initiated by a user, the user name of the user who initiated the backup is displayed. • If the backup was initiated in response to another management operation by a user, the name of the user is displayed. For more information, see About Backup and Restoration in Oracle Java Cloud Service.
Available Until	The date and time until which the backup will be retained.
Contains	A row of up to two icons that indicates the content of the backup: • 🗎—Indicates that the backup contains configuration files. • ☰—Indicates that the backup contains database files. Place your mouse over this icon for additional database information.
Notes	Click the link to display the notes that were provided when the backup was created or the restoration was performed.
Menu ☰	Select from the following options: • **Restore**—Restore the service instance from the backup. See Restore a Backup. • **Delete**—Delete the backup. See Delete a Backup.
Restore History (Last 7 Days)	Click the triangle adjacent to this label to display a list of all the restoration operations on this service instance. By default, only restoration operations for the last seven days are listed. Use the search field to specify a range of dates for which you want restoration operations returned. Click **Select to include unsuccessful restore attempts** to include the unsuccessful restoration operations in the list.
✓	Completed restoration operation for theOracle Java Cloud Service instance. The restoration operation is identified by the date and time when it was started, which is displayed adjacent to icon that represents the restoration operation. Click the icon to see additional information about the restoration operation.

Element	Description
⏳	In-progress restoration operation for the Oracle Java Cloud Service instance. The restoration operation is identified by the date and time when it was started, which is displayed adjacent to icon that represents the restoration operation. Click the icon to see additional information about the restoration operation.
⚠	Unsuccessful restoration attempt for the Oracle Java Cloud Service instance. The restoration attempt is identified by the date and time when it was started, which is displayed adjacent to icon that represents the restoration attempt. Click the icon to see additional information about the restoration attempt.
From Backup	The date and time when the backup from which the service instance was restored was created.
Status	The status of the restoration operation: • Completed • In-Progress • Failed Click the text to see detailed status messages for the operation.
Contains	A row of up to two icons that indicates the items that were restored: • —Indicates that binary files were restored. • —Indicates that configuration files were restored.

8

Manage Snapshots and Clones in Oracle Java Cloud Service

A snapshot is a point-in-time image of a service instance. You can use snapshots to quickly create multiple clones of an instance.

Topics:

- About Snapshots and Clones
- Create a Snapshot
- Delete a Snapshot
- Clone an Instance Using a Snapshot
- View Details of Snapshots and Clones

About Snapshots and Clones

A snapshot is a point-in-time image of a service instance. Use snapshots to create multiple clones of an instance quickly. Each clone is a replica of the original instance, except for certain attributes that you specify or override, such as the instance name and the compute shape.

> **Note:**
>
> Cloning is not supported for instances associated with Oracle Cloud Infrastructure Database or Oracle Autonomous Database (Oracle Autonomous Transaction Processing). Also, you cannot create a clone of a service instance if authentication for the service instance is enabled with Oracle Identity Cloud Service.

Topics

- What Does a Snapshot Contain?
- What Can I Use Clones For?
- What Happens When I Create a Snapshot?
- How Does Cloning Work?

What Does a Snapshot Contain?

A snapshot of an instance includes a point-in-time image of all the block storage volumes attached to the instance, except the boot volume and backup volume.

The snapshot reflects the state of the volumes at the time when the creation of the snapshot is triggered. Changes to the volumes after that point in time won't be included in the snapshot.

Note:

The snapshot does not include the database associated with the instance. You must take a snapshot of the database separately. The source and the cloned instance can't use the same infrastructure database.

What Can I Use Clones For?

Learn about some typical use cases for cloning.

Here are a few typical use cases:

- Move applications from development or testing to production rapidly.

  After deploying your applications to a service instance and testing them, when the applications are ready for production use, you can take a snapshot of the instance, create a clone of the instance using the snapshot, and then scale the cloned instance to the required size.

- Debug issues in a production environment without interrupting service availability

  Say a web application that’s deployed on one of your service instances has a performance issue. You can take a snapshot of the instance, clone it, and use the cloned instance to diagnose and debug the issue. The production instance continues to be available while your engineers debug the performance issue offline in the cloned instance.

What Happens When I Create a Snapshot?

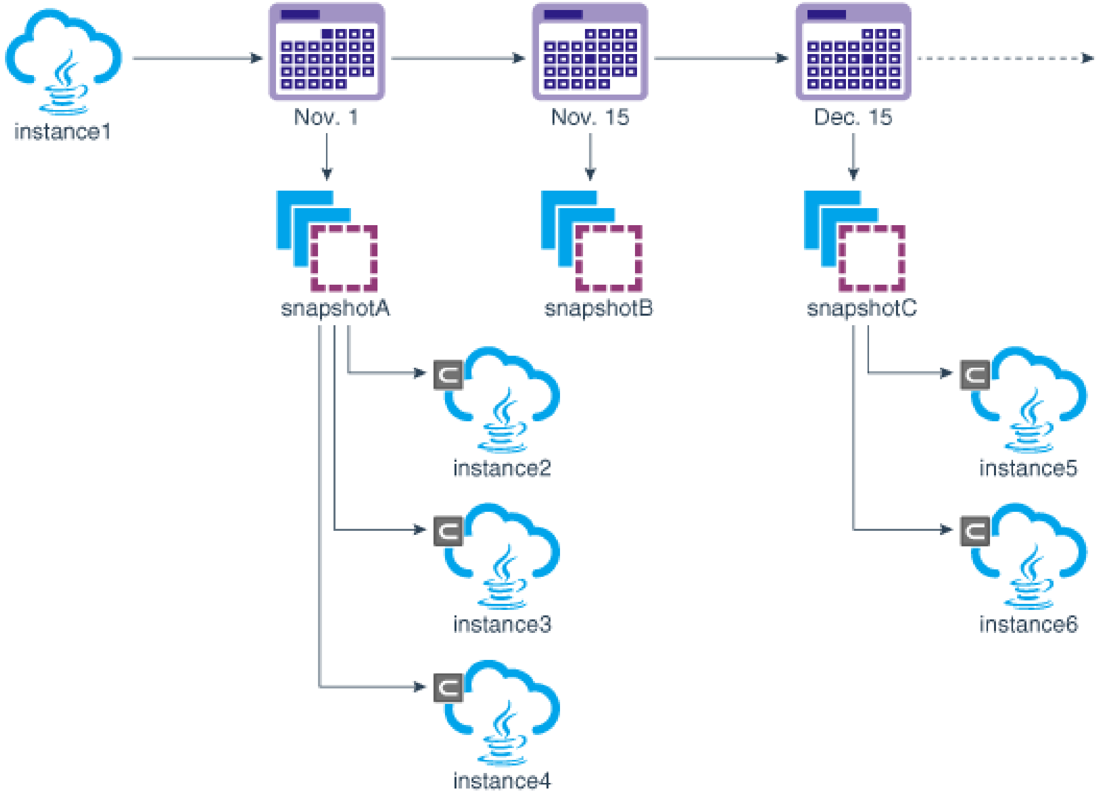
The instance that you’re taking a snapshot of continues to be available while the snapshot is created.

The instance is in maintenance mode, which means that you can't perform administration operations, such as patching and backup. While the snapshot is being created, the applications can continue to access the storage volumes.

How Does Cloning Work?

A snapshot reflects the state of the instance and the attached volumes at the time when the snapshot was taken. A clone created from a snapshot inherits certain attributes of the original instance. You can override some of the attributes, and you must specify a few attributes.

The following diagram shows the relationship between snapshots and clones. In this example, multiple snapshots of an instance are created on different dates. Some of the snapshots were used to create multiple clones of the parent instance.



In the web console, cloned instances are indicated visually by the ☁ icon.

Internally, a snapshot in Oracle Cloud Infrastructure Classic is linked to the parent instance and to the clones created from the snapshot. But snapshots in Oracle Cloud Infrastructure are detached, meaning that they exist independently, with no linkage to the parent instance or the clones.

The service level, software edition, cluster size, and domain partitions of the cloned instance are the same as that of the original instance. They can't be changed. If the original instance has a load balancer, the clone will have a load balancer as well. If Oracle Identity Cloud Service is enabled for the original instance, then the clone will have Oracle Identity Cloud Service enabled as well.

You can change the shape of the compute nodes of the cloned instance. You must specify the instance name, the SSH public key for the compute nodes, the administrator credentials for Oracle WebLogic Server, and the database to be associated with the cloned instance.

At any time, you can view the snapshot from which a given clone was created. You can also view details of the clones created from a given snapshot.

Create a Snapshot

You can use snapshots to quickly create clones of an Oracle Java Cloud Service instance.

Note that snapshots don’t include the database deployment that’s associated with your instance.

> **Note:**
>
> The instance continues to be available while the snapshot is created. But the instance is in maintenance mode, which means that you can't perform administration operations, such as patching and backup. While the snapshot is being created, the applications can continue to access the storage volumes.

To create a snapshot of an instance,

1. Access your service console.
2. Click the name of the instance that you want to take a snapshot of.
3. Click the **Administration** tile.
4. Click the **Snapshots** tab.
5. Click **Create**.
6. In the Create Snapshot dialog box, complete the following steps:
   - **Snapshot Name**: Enter a name for the snapshot.
   - **Snapshot Description**: Enter a description that you can use later to identify the key characteristics of the snapshot.
7. Click **Create**.

After the snapshot is created, you can use it to create clones of the original instance.

Delete a Snapshot

When you no longer need a snapshot of an Oracle Java Cloud Service instance, you can delete it.

> **Note:**
>
> For instances in Oracle Cloud Infrastructure Classic, to delete a snapshot that has clones, you must first delete the clones.

1. Access your service console.
2. Click the name of the service instance for which you want to delete a snapshot.
3. Click the **Administration** tile.
4. Click the **Snapshots** tab.
5. Click ☰ for the snapshot that you want to delete.
6. Click **Delete**.

Clone an Instance Using a Snapshot

After taking a snapshot of an instance, you can use the snapshot to quickly create a clone of the instance.

> **Note:**
>
> Cloning is not supported currently for the following:
>
> - Instances associated with Oracle Cloud Infrastructure Database or Oracle Autonomous Database.
> - Instances where authenication with Oracle Identity Cloud Service is enabled.
> - Instances on Oracle Cloud at Customer created before release 17.4.1.
>   To find out whether an instance was created before 17.4.1:
>   1. Do one of the following:
>   - Send a `GET` REST API request to the instance. See View a Service Instance in *REST API for Oracle Java Cloud Service*.
>   - Run the CLI command `psm jaas service -s` *`serviceInstanceName`* `-of json`. See psm jaas service in *PaaS Service Manager Command Line Interface Reference*.
>   2. In the data that the API request or the CLI command returns, search for the line `"provisionEngine":"Metadata_`*`x`*`_`*`y`*`"` (for example, `"provisionEngine":"Metadata_1_0"`).
>   - If the `provisionEngine` line does not exist, then the instance was created before the 17.4.1 release.
>   - If the line exists, then the instance was created after 17.4.1.

To create a clone of an instance, complete the following steps:

1. Sign in to the web console.
2. Click the name of the instance that you want to create a clone of.
3. Click the **Administration** tile.
4. Click the **Snapshots** tab.
5. Click ☰ for the snapshot that you want to clone from.
6. Click **Create Clone**.

   The instance creation wizard starts.
7. On the Instance page of the wizard, specify the following attributes:

Field	Description
Instance Name	Specify a name for the Oracle Java Cloud Service instance.
Description	(Optional) Enter a short description of the Oracle Java Cloud Service instance.

Field	Description
Notification Email	(Optional) Specify an email address where you would like to receive a notification of any events occurring with the service instance, including whether provisioning has succeeded or failed.
Region	Selected automatically; same as the original instance.
IP Network	(Only if a region is selected) (Not available on Oracle Cloud Infrastructure) Select an IP network if you want to create the service instance in an IP network that you've defined. By default, each node in your instance is auto-assigned a public and a private IP address. The IP addresses might change each time the service instance is restarted. You can reserve and assign fixed public IP addresses. In order to select an IP network if you have selected **Enable Authentication Using Identity Cloud Service**, which automatically configures a managed load balancer, you must first attach an internet-facing load balancer to the IP network. This field is not relevant to Oracle Cloud Infrastructure.
Assign Public IP	(Not available on Oracle Cloud Infrastructure) Choose whether to assign public IP addresses to the nodes in your service instance. You must first select an Oracle Cloud Infrastructure Classic region and specify an IP network. If you select this check box (default), then any node added during instance provisioning, or later added as part of a scaling operation, will have a public IP address assigned to it. You will be able to directly access the nodes from the public Internet. This selection is for use cases where you intend to deploy Java EE applications to the Oracle Java Cloud Service instance and access them from the public Internet. If you deselect this check box, then any node added during instance provisioning, or later added as part of a scaling operation, will not have a public IP address assigned to it. You will not be able to directly access the nodes from the public Internet. This selection is for use cases where you intend to deploy Java EE applications to the Oracle Java Cloud Service instance and access them only within your IP network or from your on-premises data center over a VPN network.
Availability Domain	(Available only on Oracle Cloud Infrastructure) Select an availability domain. A region can have multiple isolated availability domains, each with separate power and cooling. The availability domains within a region are interconnected using a low-latency network. Note that the database that you intend to associate with your Oracle Java Cloud Service instance can be in a different availability domain within the selected region.

Field	Description
Subnet	Select the Oracle Cloud Infrastructure subnet to which the nodes of your instance must be attached. This field provides a **No Preference** option and a list of the available subnets. Each subnet is shown in the format *compartmentName* | *vcnName* | *subnetName*. A tooltip lists the compartment name, VCN name, subnet name, and the OCID of the subnet. • To have the subnet assigned automatically, select **No Preference**. The subnet **ManagedCompartmentForPaaS | svc-vcn | svc-subnet-...** is used for your instance. **Note:** Don't select **No Preference** if you plan to associate an Oracle Cloud Infrastructure Database with your service instance. If you want to configure security rules for your instance, don’t select **No Preference** or **ManagedCompartmentForPaaS | svc-vcn | svc-subnet-...**. Select a subnet in a VCN that you created. • To assign a subnet explicitly, select a suitable subnet from the available options. • If none of the available subnets meets your networking requirements, then cancel the Create Instance wizard. In Oracle Cloud Infrastructure, create the required VCN and subnets, create policies to allow Oracle Java Cloud Service to use the VCN, and select the appropriate subnet while creating your instance. See Prerequisites for PaaS Services on Oracle Cloud Infrastructure in the Oracle Cloud Infrastructure documentation. Database instances in Oracle Database Cloud Service and Oracle Cloud Infrastructure Database must be in the same region and virtual cloud network (VCN) as the Oracle Java Cloud Service instance. The database and service instance do not need to be in the same subnet. The database and service instance can be on different VCNs only if you configure VCN peering.
Tags	(Optional) Select existing tags or add tags to associate with the service instance. To select existing tags, select one or more check boxes from the list of tags that are displayed on the pull-down menu. To create tags, click + to display the **Create Tags** dialog box. In the **New Tags** field, enter one or more comma-separated tags that can be a key or a key:value pair. If you do not assign tags during provisioning, you can create and manage tags after the service instance is created.

Field	Description
Bring Your Own License	The **Bring Your Own License** (BYOL) option enables you to bring your on-premises Oracle WebLogic Server licenses to Oracle Cloud. BYOL instances are billed at a lower rate than other instances. See Frequently Asked Questions: Oracle BYOL to PaaS. You must own a Universal Credits or Government subscription in order to use BYOL. BYOL is enabled by default. If you deselect this option, your account will be charged for the new service instance according to your Oracle Java Cloud Service agreement. **Note**: Before you scale up or scale out a BYOL instance, you must have enough WebLogic Server licenses for the additional OCPUs that will be allocated to the instance after it is scaled.
Software Edition	Selected automatically; same as the original instance.

8. Click **Next.**
9. On the Details page, select the **Advanced** tab, and specify the following attributes:

Field	Description
WebLogic Clusters	Configured automatically; same as the original instance.
Compute Shape	Select the compute shape to use for all Administration Server and Managed Server nodes. The compute shape is the number of Oracle Compute Units (OCPUs) and amount of memory (RAM) that you want to allocate to these nodes. The selected shape is not used for Coherence or Load Balancer nodes. The list of available shapes varies depending on whether you selected an Oracle Cloud Infrastructure Classic or Oracle Cloud Infrastructure region. (Advanced option) When you create multiple WebLogic clusters, you can assign a different compute shape for different clusters. This field displays the compute shape of the selected cluster. If you purchased a Universal Credits subscription for Oracle Java Cloud Service, you will pay at the Pay-As-You-Go rate when you exceed your monthly or annual maximum credit.
Server Count	Selected automatically; same as the original instance.

Field	Description
Reserved IPs	(Not available on Oracle Cloud Infrastructure) Select reserved IP addresses for the nodes in your cluster, or leave the default value as **Assign Automatically** if you want Oracle to automatically assign IP addresses to these nodes. The number of IP addresses that you select must equal the number of nodes in the cluster. This option is displayed only if you selected a specific **Region** for this service instance. You create IP reservations by using the **Reserved IPs** tab in the Oracle Java Cloud Service Console. If you do not see this tab on the console, click the gear icon next to this field and follow the instructions to create your first IP reservation. After creating IP reservations, you need to restart the instance creation wizard.
Domain Partitions	Selected automatically; same as the original instance.
Enable Access to Administration Consoles	(Advanced option) Select this check box if you want to enable access to the WebLogic Service Administration Console, Fusion Middleware Control, and Load Balancer Console for the service instance. If you do not select this option, these consoles will not be externally accessible, and also will not appear as choices in the service instance's menu ☰. Alternatively, you can enable access to the administration consoles after creating the service instance. If you are creating this service instance in Oracle Cloud Infrastructure, access to the administration consoles is enabled by default; selecting or deselecting this check box has no effect.
Deploy Sample Application	Selected or deselected automatically to match the original instance.
Enable Authentication Using Identity Cloud Service	Selected automatically; same as the original instance.
SSH Public Key	Specify the public key that will be used for authentication when connecting to a node in your instance by using a Secure Shell (SSH) client. Click **Edit** to display the SSH Public Key for VM Access dialog, and then specify the public key using one of the following methods: • Select **Key file name** and use your web browser to select a file on your machine that contains the public key. • Select **Key value** and paste the value of the public key into the text area. Be sure the value does not contain line breaks or end with a line break. • Select **Create a New Key** if you want Oracle to generate a public/private key pair for you. You will be prompted to download these generated keys. If you choose to create a new key, the generated private key file is in OpenSSH format. Before connecting to a node in this service instance with the PuTTY SSH client, you must first convert the key to PuTTY's proprietary format.

Field	Description
Local Administrative User Name	Enter your choice of user name for the WebLogic Server administrator. The default is `weblogic`. This name is used to access the WebLogic Server Administration Console, Fusion Middleware Control, and Load Balancer Console for the service instance.
Password	Specify a password for the WebLogic Server administrator and confirm the password.
Provision Local Load Balancer	Selected automatically; same as the original instance.
Load Balancer	This option is displayed only if you selected an Oracle Cloud Infrastructure region. Selected automatically; same as the original instance.
Compute Shape	Select the compute shape to use for all the load balancer nodes in the service instance. The compute shape is the number of Oracle Compute Units (OCPUs) and amount of memory (RAM) that you want to allocate to these nodes. The list of available shapes varies depending on whether you selected an Oracle Cloud Infrastructure Classic or Oracle Cloud Infrastructure region. You are billed for Oracle Traffic Director nodes at the same price that you are billed for Oracle WebLogic Server nodes in your Oracle Java Cloud Service subscription.
Add Another Active OTD Node	This option is displayed only if **Provision Local Load Balancer** is set to `Yes`. Select this check box to provision a second load balancer node running Oracle Traffic Director (OTD) in this service instance. Both load balancer nodes route traffic to the cluster of WebLogic Managed Servers. You can also add a second load balancer node to a service instance after creating the service instance.
Reserved IPs	Select reserved IP addresses for the load balancer nodes in your cluster, or leave the default value as **Assign Automatically** if you want Oracle to automatically assign IP addresses to these nodes. The number of IP addresses that you select must equal the number of load balancer nodes in the service instance. This option is displayed only if these conditions are true: • You selected a specific Oracle Cloud Infrastructure Classic **Region** for this service instance. • **Provision Local Load Balancer** is set to `Yes` You create IP reservations by using the **Reserved IPs** tab in the Oracle Java Cloud Service Console. If you do not see this tab on the console, click the gear icon next to this field and follow the instructions to create your first IP reservation. After creating IP reservations, you need to restart the instance creation wizard.

Field	Description
Load Balancing Policy	If you selected **Provision Local Load Balancer**, choose one of the following policies: • **Least Connection Count** (default)—Passes each new request to the Managed Server with the least number of connections. This policy is useful for smoothing distribution when a Managed Server receives more requests than it can handle efficiently. • **Least Response Time**—Passes each new request to the Managed Server with the fastest response time. • **Round Robin**—Evenly distributes requests across all Managed Servers, regardless of the number of connections or response times.

Field	Description
Database Instance Name	**Note**: Cloning is not supported currently for Oracle Java Cloud Service instances associated with Oracle Cloud Infrastructure Database or Oracle Autonomous Database. The source and the cloned instance can't use the same infrastructure database. Select an existing Oracle Database Cloud Service (Classic) deployment or Oracle Database Exadata Cloud Service deployment to connect to this service instance. Oracle Java Cloud Service provisions the selected database with the required schemas for running a service instance. The list only includes a database deployment if it meets the following criteria: • Is in an active state and not currently in the process of being provisioned • Is not configured with a **Backup Destination** set to `None` (not applicable to Oracle Database Cloud Service — Virtual Image deployments). Note the following additional constraints and limitations: • To ensure that you can restore the database for an Oracle Java Cloud Service instance without risking data loss for other service instances, Oracle recommends that you do not associate the same infrastructure schema database (or the same pluggable database) with multiple service instances. Backups of a database that is used with multiple Oracle Java Cloud Service instances contain data for all the instances. Therefore, if you restore the database from a backup, data for all the service instances is restored, which might not be the intended result. • Oracle Java Cloud Service does not support Oracle Database 18c. • If you selected an **IP Network** for this service instance, you must also select an Oracle Database Cloud Service (Classic) database deployment that is attached to an IP network. If the service instance and database deployment are attached to different IP networks, the two IP networks must be connected to the same IP network exchange.
PDB Name	Specify the pluggable database the service instance will connect to. If you don't specify a PDB name, Oracle Java Cloud Service uses the default Oracle Database 12c PDB name that was provided when the Oracle Database Cloud Service (Classic) database deployment was originally created.
Administrator User Name	Enter the name of the database administrator that Oracle Java Cloud Service will use to connect to the selected database deployment and to provision the required schemas for this service instance.

Field	Description
Password	Enter the password for the database administrator.
Add Application DB	(Advanced option) Add a up to four database deployments for your application schema. Click **Add** if you want to specify a separate Oracle Database Cloud Service database deployment or Oracle Database Exadata Cloud Service database dedicated for your application schema. When you add an application database, the Oracle Java Cloud Service creates an additional data source in your Oracle WebLogic Server domain to connect to this database. Use the Add Database Configuration dialog to select the name of an existing Oracle Database Cloud Service database deployment or Oracle Database Exadata Cloud Service database, and to provide a user name and password for this database. Click **Add** and repeat this process for up to three more database deployments.
Backup Destination	(Advanced option) Select **Both Remote and Disk Storage** if you want to enable automated and on-demand backups for this service instance. Backups will be saved to object storage *and* to block storage volumes that are attached to the nodes of the instance. The default value is **None**, meaning that you cannot use Oracle Java Cloud Service to take backups of this service instance. You can configure backups on a service instance after creating it. This field is not relevant if you selected **Oracle Java Cloud Service—Virtual Image**.

Field	Description
Object Storage Container	This field is displayed only if **Backup Destination** is set to **Both Remote and Disk Storage**. Enter the object storage location where backups of the service instance must be stored. The object storage container field in the instance creation wizard is auto-populated with a default container URL in the format *`restEndpointUrl`*`/JaaS`, where *`restEndpointUrl`* is the REST endpoint URL of the Oracle Cloud Infrastructure Object Storage Classic service in the account, and `JaaS` is the default container name. You can change the container name. Note that if the account doesn't include an Object Storage service entitlement or if the region selected is an Oracle Cloud Infrastructure region, then the container field is not autopopulated. • **Oracle Cloud Infrastructure Classic**: Enter the URL of a container in Oracle Cloud Infrastructure Object Storage Classic. **Format**: *`rest_endpoint_url`*`/`*`containerName`* You can find the REST endpoint URL of the Oracle Cloud Infrastructure Object Storage Classic service instance in the Infrastructure Classic Console. **Example**: `https://acme.storage.oraclecloud.com/v1/MyService-acme/MyContainer` **Note**: You can select the **Create Object Storage Container** check box to have a new container created automatically. • **Oracle Cloud Infrastructure**: Enter the URL of a bucket in Oracle Cloud Infrastructure Object Storage. See Prerequisites for PaaS Services on Oracle Cloud Infrastructure in the Oracle Cloud Infrastructure documentation. **Format**: `https://swiftobjectstorage.`*`region`*`.oraclecloud.com/v1/`*`namespace`*`/`*`bucket`* To find out your `namespace`, sign in to the Oracle Cloud Infrastructure web console, click the tenancy name, and look for the **Object Storage Namespace** field. **Example**: `https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/myCompany/myBucket`

Field	Description
User Name	This field is displayed only if **Backup Destination** is set to **Both Remote and Disk Storage**. In Oracle Cloud Infrastructure Classic regions only, this field is not displayed if you selected **Enable Authentication Using Identity Cloud Service**. • **Oracle Cloud Infrastructure Classic**: Enter the user name of the Oracle Cloud Infrastructure Object Storage Classic service user who created the container you specified earlier. If the container doesn't exist, then enter the user name of a service administrator. • **Oracle Cloud Infrastructure**: Enter the user name of the Oracle Cloud Infrastructure Object Storage user who created the bucket you specified earlier.
Password	This field is displayed only if **Backup Destination** is set to **Both Remote and Disk Storage**. In Oracle Cloud Infrastructure Classic regions only, this field is not displayed if you selected **Enable Authentication Using Identity Cloud Service**. • **Oracle Cloud Infrastructure Classic**: Enter the password of the user you specified. • **Oracle Cloud Infrastructure**: Enter the Auth Token generated in Oracle Cloud Infrastructure for the user you specified. See Prerequisites for PaaS Services on Oracle Cloud Infrastructure in the Oracle Cloud Infrastructure documentation.
Create Object Storage Container	This option is displayed only if **Backup Destination** is set to **Both Remote and Disk Storage**. If the Oracle Cloud Infrastructure Object Storage Classic container that you specified doesn't exist, or if you aren't sure whether it exists, then select this check box. If the container doesn't exist, it will be created automatically. This option is not relevant to Oracle Cloud Infrastructure. The specified Oracle Cloud Infrastructure Object Storage bucket must exist prior to creating a service instance.
Provision Data Grid Cluster	Selected automatically; same as the original instance.
Compute Shape	Selected automatically; same as the original instance.
Cluster Size	Selected automatically; same as the original instance.
Managed Servers Per Node	Selected automatically; same as the original instance.

10. Click **Next.**
11. On the Confirmation page, review the attributes you configured.
    - To make changes, click **Back** and make the necessary changes.
    - To proceed with creating the clone, click **Create**.

After the clone is created, the Snapshots tab shows the cloned instance under **Cloned Services**.

On the Instances page of the web console, cloned instances are indicated by the ☁ icon.

> **Note:**
>
> In Oracle Cloud Infrastructure Classic and Oracle Cloud at Customer, if you create a clone from a colocated snapshot (C), you can't take a snapshot of the cloned instance.

View Details of Snapshots and Clones

You can view the snapshots available for a given instance and the details of the clones created from each snapshot.

1. Access the service console.
2. Click the name of the service instance for which you want to delete a snapshot.
3. Click the **Administration** tile.
4. Click the **Snapshots** tab.

   The page displays the name and creation date of each snapshot available for the selected instance.

   To view the clones created from a specific snapshot, expand **Cloned Services**.

To view all the attributes of a snapshot, use the REST API. See Snapshots REST Endpoints in *REST API for Oracle Java Cloud Service*.

9

Patch an Oracle Java Cloud Service Instance

This section describes how to apply a patch to an Oracle Java Cloud Service instance, and roll back the patch as necessary.

> **Note:**
>
> This section does **not** apply to Oracle Java Cloud Service—Virtual Image instances. Patching within Oracle Java Cloud Service is **not** supported by Oracle Java Cloud Service—Virtual Image instances.

Topics:

- About Patching and Rollback
- Typical Workflow for Patching an Oracle Java Cloud Service Instance
- View Patch Details
- Perform Patch Prechecks and Address Patching Issues
- Apply a Patch
- Roll Back a Patch
- Explore the Patching Page

About Patching and Rollback

You can quickly and easily apply patches to an Oracle Java Cloud Service instance to ensure that it contains the latest bug fixes and performance improvements.

What Kinds of Patches are Available

You can patch different Oracle software components that comprise your service instance. These are often Patch Set Updates (PSU). PSUs are cumulative. Each PSU contains all the previous PSUs bug fixes in addition to any bug fixes released after the previous PSU. You can also apply a tools patch that patches the internal scripts that perform tooling operations such as backup and scaling. A tools patch is available after a PSU becomes available in production, and is applied automatically after a few hours if you do not apply it yourself.

You apply all these types of patches in the same way.

About Operating System (OS) Patching

Java Cloud Service does not provide cloud tooling for OS patching. You are responsible for installing OS patches to existing service instances.

You can obtain Oracle Linux OS patches from the Oracle’s Unbreakable Linux Network if you have an Oracle Linux support subscription. You can also obtain Linux OS patches from Oracle Linux Public Yum server: `http://public-yum.oracle.com`. Java Cloud Service nodes are preconfigured to enable you to install and update packages from the repositories on the Oracle public Yum server. The repository configuration file is in the `/etc/yum.repos.d` directory on the nodes. You can install, update, and remove packages by using the `yum` utility.

> **Note:**
>
> You are responsible to applying the required security updates published through the Oracle public Yum server.

Do not install OS patches for other Linux distributions. Also, if you plan to use your service instance for production applications, Oracle recommends that you avoid installing any test, development, or preview OS packages that might be available in the repository.

> **WARNING:**
>
> You can run yum update only with the repos provided in the OS. If you use the yum repos from other Linux systems to run the yum update, the system might become unstable and it is difficult to recover the OS back to working state.

When to Apply Patches

New approved patches are typically available and displayed on the Patching page of the console on a quarterly basis. Apply the most recent patches promptly. Delaying patches could cause your service to be unsupported for future patching.

> **Note:**
>
> Apply patches only when they become available on the Patching page for your service instance. Applying patches manually between PSUs causes precheck failure. If you have applied patches manually, remove them before patching.

What Happens When Patching Starts

As patching starts, the patching operation first performs internal prechecks for the following types of issues:

- Presence of manually-applied patches
- Disk space shortage
- Missing database connectivity
- Servers not running
- Storage access failure

> **Note:**
>
> Patching is not supported for service instances where Oracle Java Cloud Service Fusion Middleware—Oracle WebCenter Portal, Oracle Java Cloud Service Fusion Middleware—Oracle Data Integrator, or any other product that modifies the `MW_HOME` directory are installed. If you attempt to patch a service instance where any of these products are installed, patching prechecks issue an error message and patching does not start.

If the prechecks fail, the patching operation will fail and leave the Java Cloud Service instance untouched.

Next, the target Oracle software components are restarted to ensure that they can be restarted again after the patching operation performs a binary swap.

The prechecks do not check whether another administration task (backup, restoration, or scaling) is in progress, which would prevent patching.

You can also perform prechecks without attempting to patch, and first remedy any problems found.

If backups are configured and enabled on your service instance, an automatic backup is created only after patching prechecks succeed. If you need to restore the state of the service instance, use the backup and run patching again.

> **Note:**
>
> If automatic backup fails, then the patching operation fails and does not apply the patch.

What Happens During Patching

Most patching operations are rolling operations, so the service functions with very little interruption during the patch process. The patching operation shuts down one node at a time and applies the patch to the server or servers on the node. After each node is patched, it is automatically restarted. If there is a load balancer, the load balancer automatically detects that the server is down and does not send requests to that server. The other servers process application requests without interruption. The patching operation continues patching the servers on one node at a time until all servers are patched.

For example, if you have a two-node cluster, one node keeps running while the other is being patched.

If a load balancer is provisioned, when you patch the load balancer, the other server processes remain running. No requests will be routed to these servers during load balancer patching. The load balancer is stopped while patching is in progress.

What Happens When Patching is Not Fully Successful or Fails

If a patching operation fails, the patch information is displayed in the Patch History section. You can click on the icon to see an error report. After patching operation failure, the operation automatically reverts any change it has made to the service instance. The operation moves the service instance back to the same state it was in before the patching operation started. If patching fails and the operation fails to revert the service back to the previous state, you can use the backup created at the beginning of the patching operation to restore the service manually.

> **Note:**
>
> If you had applied patches from a source other than your service to your service instance, these patches will not be restored if you restore the service manually.

What Happens if Coherence is Enabled

Applying a patch will perform a rolling restart of Managed Coherence Servers on the Coherence data tier. By default, the patching operation checks that the `StatusHA` state for a Coherence member is `NODE-SAFE` before shutting down the node to apply the patch. You might choose to override the default behavior.

When to Roll Back a Patch

You can roll back a patch if you find that a patch is incompatible with applications deployed on your Java Cloud Service, or for any other possible reason.

Patches for the load balancer cannot be rolled back.

What Happens During a Rollback Operation

The rollback operation shuts down the server processes while the patch is rolled back, so the service is temporarily unavailable.

How to View Patching History

The patching history is displayed on the Java Cloud Service Patching page. The history shows the patch number, the name of the administrator who applied the patch, and any notes. When the rollback operation is complete, the patch information stays in the patch history section, but the **Roll Back** button is grayed-out. The patch information also reappears in the list of available patches, so you can try applying the patch again.

How to Restore the Instance Configuration

If backups are configured and enabled on your service instance, each patch, whether failed or successful, has a backup. Rolling back a patch restores the binaries to the specific version recorded at the time of the backup, without modifying the configuration data. If necessary, you can also restore the configuration after rollback by using the Backup page.

Typical Workflow for Patching an Oracle Java Cloud Service Instance

Consider the typical workflow for patching an Oracle Java Cloud Service instance, as described in the following table.

Task	Description	More Information
Learn about approved patches	View approved patches displayed on the Patching page periodically.	View Patch Details
Perform optional prechecks	Learn about problems that would cause patching to fail, so you can address those problems before you try to apply a patch.	Perform Patch Prechecks and Address Patching Issues
Apply a patch	Initiate a patching operation to update the service instance's WebLogic servers, JDKs, or load balancer with minimal impact on the service availability. When Oracle Coherence is enabled for a service instance: By default, the patching operation checks that the `StatusHA` state for a Coherence member is `NODE-SAFE` before shutting down the node to apply the patch. You may choose to override the default behavior.	About Patching and Rollback Apply a Patch
Roll back a patch	Initiate a rollback operation to return a service to its previous patch level.	Roll Back a Patch

View Patch Details

You can routinely check for available patches for your Oracle Java Cloud Service instances.

Oracle recommends that you apply the most recent patches promptly. Delaying the application of patches might cause your service instance to be unsupported for future patching.

1. Access your service console.
2. Click the name of the service instance for which you want to view patches.

   On the Overview page, the **Administration** tile displays the number of available patches for the service instance.
3. If patches are available, click the **Administration** tile.
4. Click the **Patching** tab.
5. Under **Available Patches** , click a patch's icon to view its details.
6. If the patch is associated with one or more `readme` files, to view them expand **Readme**.

Before you apply a patch, Oracle recommends that you perform a precheck.

Perform Patch Prechecks and Address Patching Issues

Optional patching prechecks identify possible sources of patching failure, if any, enabling you to identify and remedy problems before attempting to patch.

Topics

- Perform a Patch Precheck
- Address Patch Precheck Issues

Perform a Patch Precheck

Before you apply a patch on an Oracle Java Cloud Service instance, you can run a precheck to identify and fix potential issues.

The precheck operation helps detect problems such as:

- Disk space shortage
- Missing database connectivity
- Servers not running
- Storage access failure

The precheck operation does not detect whether another maintenance operation, such as a backup, restoration, or scaling operation, is in progress. These operations prevent you from applying a patch.

1. Access your service console.
2. Click the name of the service instance to which you want to apply a patch.
3. Click the **Administration** tile.
4. Click the **Patching** tab.
5. Beside the patch that you want to precheck, click **PSU** ☰, and then select **Precheck**.
6. When prompted for confirmation, click **Yes**.
7. To determine whether the prechecks completed, periodically click **Refresh** ⟳.

   The **Precheck summary** link is displayed.

   - If prechecks pass, a green check mark icon is displayed.
   - If prechecks fail, a red exclamation mark icon is displayed.
8. Click **Precheck summary**.
9. If there are precheck failures, address the issues, and then run **Precheck** again.

After you fix any precheck failures, you are ready to apply the patch.

Address Patch Precheck Issues

Running prechecks prior to patching lets you know of conditions that would prevent successful patching.

By performing prechecks, you can identify problems that you can address before you try to apply a patch.

Accessing Error Messages

Error messages are shown on the Patching Precheck Results dialog displayed when you click the **Patching results** link on the Patching page.

Note:

The values provided in the following messages are examples. Your values and host names will differ.

Topics:

- Failure Due to Manual Patching
- Disk Space Shortage
- Missing Database Connectivity
- Node Manager Is Not Running
- Connectivity Issues between Managed Servers and the Administration Server
- Servers Not Running
- Storage Access Failure

Failure Due to Manual Patching

If you have installed patches manually and then attempt to apply approved patches from the Patching page of the Oracle Java Cloud Service Console, you see messages such as the following.

```
2019-07-23T11:52:26.145+00:00[SEVERE]:PATCHING-59996-[JaaS]-[1]-[FAILURE]:
[PSM-PATCH-60000: The patches applied on the service are not matching the 
expected 
patches. The expected patches are 
["19795066","19154304","18905788","19632480",
"19030178","22754279","19002423","21663638"] and the actual patches 
are 
["19632480","26910516","19002423","19154304","19030178","18905788","19795066"
,
"22754279","18459080","21663638"]. When the patch is applied, the patch 
operation 
will move the service to patch level [12.2.1.2.190617]]
```

Remove the patches you applied manually, and then rerun patching prechecks or apply a patch from within the Patching page of the Oracle Java Cloud Service Console. To avoid this error in the future, apply approved patches only when they become available from the Patching page.

Disk Space Shortage

If patching would fail because of insufficient space on the block storage volume mounted on the Administration Server VM, you see the following error message.

```
Not enough space on vm. Dir size: 457 Free space: 350
```

If there is not enough space, try freeing up disk space by deleting unwanted backups, for example.

Missing Database Connectivity

If patching would fail because of missing database connectivity, you see the following error message.

```
Could not connect to database running at
location.
Please ensure that the database, VM, database instance, and
database listener being used by the service are running.
Currently configured: testdb.example.com:1521:orcl12c
```

Ensure that your database exists and is healthy. Open your Oracle Database Cloud Service Console and check the status of your database deployment.

Node Manager Is Not Running

If patching would fail because the Node Manager is not running, you see the following messages.

```
Nodemanager is not running on test_adminserver.
Could not reach node manager on host: test_adminserver.
```

Start the Node Manager.

Connectivity Issues between Managed Servers and the Administration Server

If patching would fail because there is no SSH connectivity between the Managed Servers and the Administration Server, you see the following messages.

```
Connectivity issue between Managed Server
VM test-wls-2 and Admin Server VM test-wls-1. Please contact
Oracle Support Services to ensure connectivity
is restored.
```

Contact Oracle Support Services to ensure that connectivity is restored.

Servers Not Running

If patching would fail because the Administration Server or Managed Servers are not running, you see the following messages.

```
Admin Server is not running on test-wls-1.
Please ensure Admin Server is running through
Node Manager. Could not reach adminserver: test_adminserver
on host test-wls-1 through NodeManager.

Could not reach managed server: test_server_1 on host test-wls-1
through NodeManager. Please refer to the WebLogic
documentation to start managed server through
node manager using WLS Console or wlst.
```

Start servers that are not running.

Storage Access Failure

If patching would fail because the storage container cannot be accessed, you see the following message.

```
Unable to download patch artifact
from the Oracle Cloud Storage.
```

Although you have a storage container, run a check that tells you whether you have a storage container.

The result can indicate that the container is down.

Apply a Patch

Routinely check for and apply approved Oracle patches for an Oracle Java Cloud Service instance. Oracle recommends that you apply the most recent patches promptly. Delaying the application of patches could cause your service to become unsupported.

> **Note:**
>
> If you have installed Oracle Data Integrator or OracleWebCenter Portal on top of Oracle Java Cloud Service, Java Cloud Service does not provide patches to your service instance. You must use OPatch to apply patches to the service instance and ignore any patches provided in the Java Cloud Service Console.

You must apply only the patches that are displayed on the Patching page for your service instance.

> **Note:**
>
> Applying patches manually rather than from the Patching page causes patching to fail. Remove these patches before applying patches from the Patching page.

Patches are applied in a rolling fashion to each node in your service instance, in order to minimize the impact on your service.

Java Cloud Service does not provide cloud tooling to patch the operating systems for the nodes in your service instance. You are responsible for installing any OS patches and security updates.

Before you apply a patch to a service instance:

- Wait for any maintenance operations on this service instance to complete, such as backup, restoration or scaling operations.
- Enable backups on your service instance if you previously disabled backups.

1. Access your service console.
2. Click the name of the service instance to which you want to apply a patch.
3. Click the **Administration** tile.
4. Click the **Patching** tab.
5. Click **PSU** ☰ beside the patch that you want to apply, and then select **Patch**.
6. Optional: In the **Notes** field, enter a description for this patching operation.

7. If Oracle Coherence is enabled, choose one of these patching options:

   - Do not select the **Confirm** check box. If an Oracle Coherence server never reaches the `NODE-SAFE` state, then the patching operation fails.

   - Select the **Confirm** check box. Oracle Cloud does not wait for an Oracle Coherence server to reach the `NODE-SAFE` state before stopping the server to apply the patch. This action can result in data loss in the Oracle Coherence data tier.

   When there’s only one server in the Oracle Coherence data tier, the check box is automatically selected and disabled.

8. Click **Patch**.

   The patching operation begins. The Administration tile displays a message about the version of patch being applied, and the time of the backup that occurred before the patching operation started.

When the patching operation is completed, information about the patch appears in the **Patch History** section of the Patching page.

Roll Back a Patch

If you experience issues after applying a patch to an Oracle Java Cloud Service instance, you can roll back the patch.

Wait for any maintenance operations on this service instance to complete, such as backup, restoration or scaling operations, before you begin.

1. Access your service console.
2. Click the name of the service instance from which you want to roll back a patch.
3. Click the **Administration** tile.
4. Click the **Patching** tab.
5. Under **Patch and Rollback History**, beside the patch that you want to roll back, click **Roll Back**.
6. When prompted for confirmation, click **Roll Back**.

Explore the Patching Page

You can use the Patching page to view available patches, initiate a patching process, and roll back a patch as necessary.

Topics:

- What You Can Do from the Patching Page
- What You See on the Patching Page

What You Can Do from the Patching Page

You can perform these tasks from the Patching page:

- View a list of available patches. See View Patch Details.
- Perform patching prechecks. See Perform Patch Prechecks and Address Patching Issues.

- Apply a patch. See Apply a Patch.
- Check the patch history.
- Roll back a service to its previous patch level. See Roll Back a Patch.

What You See on the Patching Page

The following table describes the key information shown on the Patching page.

Element	Description
Oracle Java Cloud Service link	Click this link to return to the Oracle Java Cloud Service Console.

Element	Description
☰ (in the page header)	**Menu** icon provides the following options: • **Open WebLogic Server Administration Console**—Open the WebLogic Administration Console to administer your application environment. • **Open Fusion Middleware Control Console**—Open Fusion Middleware Control to administer your application environment. • **Open Load Balancer Console**—Open the console to administer the load balancer, if a local load balancer has been configured for the service instance. Note that access to the administrative consoles is disabled by default. When you create a service instance, you can enable consoles by selecting a check box on the Details page of the instance creation wizard. For an instance this is already created, you must create an access rule in order to activate the console choices. See Enabling Console Access in an Oracle Java Cloud Service. • **Start**—Start the nodes for the Administration Server, Managed Servers, load balancer, and Managed Servers on the Coherence data tier (if provisioned). • **Stop**—Stop the nodes for the Administration Server, Managed Servers, load balancer, and Managed Servers on the Coherence data tier (if provisioned). • **Restart**—Stop and then immediately restart all the nodes in the service instance. • **Scale Out**—Adds a managed server node. • **Define Auto Scaling Rules**—Opens the Add Rule dialog box, which opens the Rules page where you can configure auto-scaling rules. • **Change License Type**—Opens the Change License Type dialog box, which enables you to choose whether to leverage your existing on-premises (BYOL) license or use your Oracle Java Cloud Service cloud license. • **Add Load Balancer**—Add a user-managed load balancer to this service instance. • **Disable/Enable Load Balancer**—Depending on the selection, either blocks access to the service instance or forwards the requests it receives from clients to the Oracle WebLogic Server Managed Servers. • **Manage Access Rules**—Create and manage rules to control access to the nodes for this service instance. • **Add SSH Access**—Add public SSH keys to the nodes that make up this service instance. • **Manage Tags/Add Tags**—Either remove or add tags to a service instance. **Manage Tags** appears if a tag already exists for the service instance. **Add Tags** appears if no tags exist for the service instance. • **Enable Backups**—Enable backups for this service instance. • **View Activity**—View all administrative activities that have been performed on your service instances. • **View Instance Metrics**—View performance metrics for this service instance.
↻	Click to refresh the page. The date and time the page was last refreshed is displayed adjacent to this button.
Available Patches	Displays patches that are available. The patch number is displayed, as well as whether the patches are optional or mandatory. The release date and affected components are displayed. Information about whether a restart is required is displayed.

Element	Description
▣	Represents an available patch.
☰	This menu has two choices: - **Precheck**—Performs prechecks without attempting to patch, and reports any errors found so that you can address them before you patch. - **Patch**–Preforms prechecks, then patches the Oracle Java Cloud Service instance with the selected patch. When Oracle Coherence is enabled for a service instance: Applying a patch will do a rolling restart of the Managed Coherence Servers on the Coherence data tier. By default, the patching operation checks that the `StatusHA` state for a Coherence member is `NODE-SAFE` before shutting down the node to apply the patch. You may choose to override the default behavior by selecting the Confirm checkbox on the Patch Service dialog. If you confirm, this means you accept the possibility of data loss and agree to shut down a server even if `NODE-SAFE` cannot be reached.
✓	Represents a successful patching precheck operation.
❗	Represents a failed patching precheck operation.
▣⧗	Represents a patching operation in progress.
Patch History	Displays the history of patches that have been applied to the service instance.
▣✓	Indicates a successful patching operation. Appears in the Patch History section. Click this icon to obtain more information about the patching operation.
⚠	Displayed on a tools patch, indicates that the patch version for your existing service instance is older than the current version. You will also see a warning stating that the service is on a deprecated tools version. To address this issue, apply the latest tools patch to your service instance.
Roll Back	Initiates an operation to roll back the service to its patch level prior to applying the patch.

10 Upgrade the WebLogic Server Release for an Oracle Java Cloud Service Instance

For an existing Oracle Java Cloud Service instance, you can upgrade the WebLogic Server release from 12.2.1.0, or 12.2.1.2 to WebLogic Server release 12.2.1.4.

The manual upgrade process leverages the basic procedures for the on-premises upgrade, with some additional procedures.

Topics:

- About Upgrading the WebLogic Server Release for an Oracle Java Cloud Service Instance
- Perform Prerequisite Tasks
- Download the Upgrade Software
- Stop All WebLogic Server Processes
- Install the Upgrade Software
- Perform a Readiness Check
- Upgrade the Infrastructure Database Schemas
- Reconfigure the Domain
- Upgrade the Domain
- Restart the Administration Server Node
- Update and Restart the Managed Server Nodes
- Perform Post-Upgrade Tasks
- Roll Back an Upgrade

About Upgrading the WebLogic Server Release for an Oracle Java Cloud Service Instance

You can leverage Fusion Middleware tools to manually upgrade the WebLogic Server release 12c or WebLogic Server release 11g to either WebLogic Server release 12.2.1.4 or 12.2.1.3 for an existing Oracle Java Cloud Service instance.

Upgrading differs from patching. When upgrading you replace an existing WebLogic Server release, whereas patching only applies the latest patch set update (PSU). Unlike in patching, which applies updates in a rolling fashion, requiring no down time, upgrade requires downtime. No servers can be running during the upgrade process.

The following restrictions apply:

- You can't upgrade a service instance created before the Oracle Java Cloud Service release 17.4.1 (October 2017).

- The service instance can't be provisioned with Oracle SOA Suite.
- Only in-place binary installation is supported. The upgrade must use `/u01/app/oracle/middleware` as opposed to a new directory.
- You can upgrade WebLogic Server 12c (12.2.1.2, or 12.2.1.0) or WebLogic Server 11g (11.1.1.7) to WebLogic Server release 12.2.1.4.
- You can upgrade WebLogic Server 12c (12.2.1.3) to WebLogic Server release 12.2.1.4.
- When upgrading to 12.2.1.3, you can't upgrade a service instance that includes an Oracle Traffic Director (OTD) load balancer. You must remove the OTD before performing the upgrade, then add the OTD back after the WebLogic Server nodes have been upgraded. The OTD added back is version 12.2.1.2.
- When upgrading to 12.2.1.4, you can upgrade a service that includes an OTD load balancer. Note, however, the following additional restrictions:
  - The service instance must be running WebLogic Server release 12.2.1.3, 12.2.1.2 or 12.2.1.0. Both WebLogic Server and OTD versions are upgraded at the same time. You can't upgrade one component without upgrading the other component.
  - For a service instance that's based on WebLogic Server 11g, you must remove the OTD before performing the upgrade, then add the OTD back after the WebLogic Server nodes have been upgraded. The OTD added back is version 12.2.1.4.

This upgrade process leverages the following Fusion Middleware tools:

- Reconfiguration Wizard: Helps you reconfigure the WebLogic domain.
- Upgrade Assistant: Helps you perform a readiness check, upgrade the infrastructure schemas, and upgrade the domain component configurations.

During the manual upgrade process, you will be referring to the WebLogic Server documentation. See Introduction to Upgrading Oracle Fusion Middleware Infrastructure to 12c (12.2.1.4.0) or Introduction to Upgrading Oracle Fusion Middleware Infrastructure to 12c (12.2.1.3.0) in *Upgrading to the Oracle Fusion Middleware Infrastructure*.

Perform Prerequisite Tasks

Before upgrading your service instance, check whether there is enough space for temporary backups, back up the database, and back up the `MIDDLEWARE_HOME`, `JDK_HOME`, and domain directories.

To perform prerequisite tasks:

1. Ensure there is enough free disk space on all of the nodes in your Oracle Java Cloud Service instance so that you can download the new 12.2.1.4 or 12.2.1.3 binaries, and back up the current binaries and configurations.

   Verify that there is at least 2GB of available space on the `MIDDLEWARE_HOME` and root volumes. Run the following command on all nodes as the `oracle` user.

   ```
   df -kh /tmp /u01/app/oracle/middleware
   ```

   See Add Storage to a Node.

2. Back up the infrastructure database associated with your service instance. You will need the backup in the unlikely case that the upgrade doesn't succeed and you must roll back the service instance to its original state.
   - For an Oracle Database Cloud Service deployment, see Creating an On-Demand Backup in *Administering Oracle Database Cloud Service*.
   - For an Oracle Cloud Infrastructure database, see *Create an on-demand full backup of a database* in Backing Up to Oracle Cloud Infrastructure Object Storage in the Oracle Cloud Infrastructure documentation.

   > **Note:**
   >
   > Restoring the database that is used with multiple Java Cloud Service instances may risk data loss for other service instances. Oracle recommends that during the upgrade process you bring down all instances that share the database.

3. Back up the `MIDDLEWARE_HOME` directory (`/u01/app/oracle/middleware`), `JDK_HOME` directory (`/u01/jdk`) directory, and the domain directory (`/u01/data/domains/<your_domain_name>`) on all your service's nodes.

   > **Note:**
   >
   > Certain customizations to the environment will be lost during the upgrade. For example, the upgrade process might overwrite your changes to `setDomainEnv.sh`.

   For example:

   ```
   zip -r old_jdk.zip /u01/jdk
   zip -r old_middleware.zip /u01/app/oracle/middleware
   zip -r old_domain.zip /u01/data/domains/Example1_domain
   ```

4. If your service instance is based on WebLogic Server 12.2.1.3, 12.2.1.2, or 12.2.1.0, and you have configured Oracle Traffic Director (OTD) for the service instance:
   - Back up `/u01/data/otd-instance/otd_domain`, the domain that is used to manage and monitor OTD.
     For example:

     ```
     zip -r old_domainotd.zip /u01/data/otd-instance/otd_domain
     ```

5. If your service instance is based on WebLogic Server 11g, and you have configured Oracle Traffic Director (OTD) for the service instance, use the REST API to remove the OTD node.

   ```
   curl -i --user <user>:<password>
             -X PUT -H "Content-Type:application/
   vnd.com.oracle.oracloud.provisioning.Service+json" -d "{}" https://
   <rest_server_url>:/paas/api/v1.1/instancemgmt/<identity-domain>/services/
   jaas/instances/<servicename>/servicecomponent
   ```

> **Note:**
>
> In a multi-cluster service instance you have performed a scale-out operation and added a new cluster, the OTD routing information for the scaled-out server in a new cluster will be lost when you remove the OTD node. After upgrade, you must manually add the routing information back to OTD.

6. Open a VNC session on the WebLogic Administration Server node so that you can run the Reconfiguration Wizard and the Upgrade Assistant. See Connect to a Node with VNC.

   To use the Reconfiguration Wizard and Upgrade Assistant (Fusion Middleware tools) during the upgrade process, you need a graphical user interface (GUI) environment. X11 forwarding may be used to forward the GUI to your local desktop. If you set up VNC, then X11 forwarding is not needed.

   If your service instance is based on WebLogic Server 12.2.1.3, 12.2.1.2, or 12.2.1.0, and you have configured Oracle Traffic Director (OTD) for the service instance, also set up VNC or X11 forwarding on the OTD VM.

   See Running Graphical Applications Securely on Oracle Cloud Infrastructure.

Download the Upgrade Software

Use the REST API to download the WebLogic Server 12.2.1.4 or 12.2.1.3 binaries to the specified Oracle Java Cloud Service instance.

This procedure does not perform the actual process of upgrading the WebLogic Server software to the 12.2.1.4 or 12.2.1.3 version. The upgrading is done later by manually installing the binaries and using the WebLogic Upgrade Assistant. See Upgrade the Domain.

1. Use the REST API to get the patch ID of the WebLogic Server 12.2.1.4 or 12.2.1.3 upgrade patch.

   ```
   curl -i -X GET -u <user>:<password> -H "X-ID-TENANT-
   NAME:<identity_domain>" https://<rest_server_url>/paas/api/v1.1/
   instancemgmt/<identity_domain>/services/jaas/instances/
   <servicename>/patches/available?filter=upgrades
   ```

   For example:

   ```
   curl -i -X GET -u username:password -H "X-ID-TENNANT-
   NAME:ExampleIdentityDomain" https://rest_server_url/paas/api/v1.1/
   instancemgmt/ExampleIdentityDomain/services/jaas/instances/Example1/
   patches/available?filter=upgrades
   ```

   Example output:

   ```
   [{
      "availablePatchGuiMetadata":{
         "supportsPreCheck":true
   ```

```
    },
   "patchId":"wls_upg_12.2.1.3.190115_for_12cRelease212",
   "patchCategory":"MajorPatch",
   "patchSeverity":"Normal",
   "includesConfigUpgrade":false,
   "patchDescription":"WebLogic Server 12.2.1.3.0 with PSU Update 
12.2.1.3.190115",
   "patchReleaseUrl":"https://support.oracle.com/epmos/faces/PatchDetail?
patchID\u003d28710939",
   "serviceType":"JaaS",
   "serviceVersion":"12cRelease212",
   "releaseDate":"2019-01-14T17:40:00.000+0000",
   "entryDate":"2019-02-05T12:04:51.975+0000",
   "entryUserId":"smctl",
   "componentPatches":{

   },
   "patchType":"PSU",
   "requiresRestart":false,
   "isDeleted":false,
   "displayName":"12.2.1.3.190115",
   "releaseVersion":"12.2.1.3.190115",
   "patchCustomActions":[

   ],
   "restartStrategy":"RESTART_AFTER_PATCH",
   "isUpgrade":true,
   "extensionId":0
}]
```

In the response, look for the patch that has `"patchCategory":"MajorPatch"` and copy the `patchId` value. In the example, the upgrade patch ID is `wls_upg_12.2.1.3.190115_for_12cRelease212`.

2. Use the REST API to apply the upgrade patch to the service instance. No parameters are needed in the request payload.

```
curl -i -X PUT -u <user>:<password> -H "X-ID-TENANT-
NAME:<identity_domain>" https://<rest_server_url>/paas/api/v1.1/
instancemgmt/<identity_domain>/services/jaas/instances/<servicename>/
patches/<upgrade-patch>
```

For example:

```
curl -i -X PUT -u username:password -H "X-ID-TENANT-
NAME:ExampleIdentityDomain" https://rest_server_url/paas/api/v1.1/
instancemgmt/ExampleIdentityDomain/services/jaas/instances/Example1/
patches/wls_upg_12.2.1.3.190115_for_12cRelease212
```

Example output:

```
{
   "status":"Completed",
     "details":{
     "message":"Patching  service with patch
[wls_upg_12.2.1.3.190115_for_12cRelease212] is  submitted as an
asynchronous job.",
     "jobId":"107376"
     }
   }
```

> **Note:**
>
> If your service instance is based on WebLogic Server 12.2.1.3, 12.2.1.2, or 12.2.1.0 and Oracle Traffic Director (OTD) is configured, this step needs to be run only once on the service instance. The REST API operation downloads both the WebLogic Server and OTD upgrade binaries and places them on the respective virtual machines.

3. This process may take some time. Wait several minutes, and then check the job status:

```
https://<rest_server_url>/paas/api/v1.1/activitylog/
<identity_domain>/job/{jobId}
```

The upgrade patch downloads the WebLogic Server 12.2.1.4 or 12.2.1.3 binaries to each virtual machine on the specified Oracle Java Cloud Service instance, then updates Oracle Java Cloud Service to indicate that the specified upgrade patch has been applied to the service instance.

> **Note:**
>
> Applying the upgrade patch does not perform the actual process of upgrading the WebLogic Server software to the 12.2.1.4 or 12.2.1.3 version. This is done later by manually installing the binaries and using the WebLogic Upgrade Assistant.

Stop All WebLogic Server Processes

Before you upgrade the service instance, you must stop all Oracle WebLogic Server and Node Manager processes.

If you did not enable console access for this service instance, see Enable Console Access for a Service Instance.

1. Access the Oracle Java Cloud Service console.
2. Click **Manage this instance** ☰ for the service instance, and then select **Open WebLogic Server Administration Console**.
3. Enter the WebLogic Server administrator user name and password.
4. From Domain Structure, expand **Environment**.
5. Click **Servers**.

6. Click the **Control** tab.
7. Click the check box to the left of each server.
8. Click **Shutdown**, and then select **Force Shutdown Now**.

   When you shut down the Administration Server, a message warns you that the browser session will end.
9. Access the Administration Server node as the `oracle` user.
10. If you're upgrading from WebLogic Server release 12.2.1.0, 12.2.1.2, or 12.2.1.3, enter the following command to stop the Node Manager:

    ```
    /u01/data/domains/<domain>/bin/stopNodeManager.sh
    ```
11. If you're upgrading from WebLogic Server 11g (11.1.1.7), use the following procedure to stop the Node Manager:
    a. Start WLST.

       ```
       /u01/app/oracle/middleware/oracle_common/common/bin/wlst.sh
       ```
    b. Connect to the Node Manager.

       ```
       nmConnect('<username>', '<password>', '<host_name>', '5556',
       '<domain>', '/u01/data/domains/<domain>', 'ssl')
       ```

       For example:

       ```
       nmConnect('MyName', '<password>', 'example1-wls-1', '5556',
       'Example1_domain', '/u01/data/domains/Example1_domain', 'ssl')
       ```
    c. Stop the Node Manager.

       ```
       stopNodeManager()
       ```
    d. Quit WLST.

       ```
       exit()
       ```
12. Verify that no `weblogic.Server` or `weblogic.NodeManager` processes are running on this node.

    ```
    jps -l
    ```
13. Repeat Steps 10 through 12 on the Managed Server nodes in this service instance.
14. If your service instance is based on WebLogic Server 12.2.1.3, 12.2.1.2, or 12.2.1.0 and Oracle Traffic Director (OTD) is configured:
    a. Stop the OTD node.

       ```
       /u01/data/otd-instance/otd_domain/config/fmwconfig/components/OTD/
       instances/<lb-instance-name>/bin/stopserv
       ```

b. Stop the NodeManager.

```
/u01/data/otd-instance/otd_domain/bin/stopNodeManager.sh
```

c. Stop the AdminServer.

```
/u01/data/otd-instance/otd_domain/bin/stopWebLogic.sh
```

Install the Upgrade Software

Replace the current Oracle Fusion Middleware (FMW) and Java Development Kit (JDK) software installations in your service instance with the new versions of this software.

These product binaries include tools to upgrade your WebLogic Server domain and infrastructure database schema.

1. Access the Administration Server node as the `oracle` user.
2. Identify the location of the new Oracle Fusion Middleware binaries under the directory `/u01/app/oracle/middleware/jcs/FMW`.

   Example:

   ```
   /u01/app/oracle/middleware/jcs/FMW/12.2.1.3.190115/190115/
   fmiddleware.zip
   ```

3. Identify the location of the new JDK binaries under the directory `/u01/jdk/jcs/JDK`.

   Example:

   ```
   /u01/jdk/jcs/JDK/8.0.201/190115/jdk.zip
   ```

4. Move the new Oracle Fusion Middleware and JDK binaries to a temporary location.

   Example:

   ```
   mkdir /tmp/fmiddleware.zip
   mkdir /tmp/jdk.zip
   mv /u01/app/oracle/middleware/jcs/FMW/12.2.1.3.190115/190115/
   fmiddleware.zip/* /tmp/fmiddleware.zip
   mv /u01/jdk/jcs/JDK/8.0.201/190115/jdk.zip/* /tmp/jdk.zip
   ```

5. Delete the current Oracle Fusion Middleware and JDK installations, `MIDDLEWARE_HOME` and `JAVA_HOME`.

   ```
   rm -rf /u01/app/oracle/middleware/*
   rm -rf /u01/jdk/*
   ```

6. Move the new Oracle Fusion Middleware and JDK binaries from the temporary location to `MIDDLEWARE_HOME` and `JAVA_HOME`.

   ```
   mv /tmp/fmiddleware.zip/* /u01/app/oracle/middleware
   mv /tmp/jdk.zip/* /u01/jdk
   ```

7. Repeat Steps 4 to 6 on all Managed Server nodes in this instance.
8. Edit the `MIDDLEWARE_HOME` `oraInst.loc` file on the Administration Server, which contains the location of the central inventory (`oraInventory`).

   ```
   cat /u01/app/oracle/middleware/oraInst.loc
   #Oracle Installer Location File Location
   #Mon Jan 14 12:43:52 PST 2019
   inst_group=dba
   inventory_loc=/u01/app/oracle/middleware/inventory
   ```

9. If your service instance is based on WebLogic Server 12.2.1.3, 12.2.1.2, or 12.2.1.0, and you have configured Oracle Traffic Director (OTD) for the service instance:
   a. Make directories and move the upgrade binaries.

      For example:

      ```
      mkdir -p /tmp/wls_otd.zip
      mkdir -p /tmp/jdk.zip

      mv /u01/app/oracle/middleware/jcs/lb/12.2.1.4.190910/190910/
      wls_otd.zip/* /tmp/wls_otd.zip/
      mv /u01/jdk/jcs/JDK/8.0.241/200107/jdk.zip/* /tmp/jdk.zip/
      ```

   b. Remove the existing installations and copy the new binaries.

      For example:

      ```
      rm -rf /u01/app/oracle/middleware/*
      rm -rf /u01/jdk/*

      mv /tmp/wls_otd.zip/* /u01/app/oracle/middleware
      mv /tmp/jdk.zip/* /u01/jdk/
      ```

Perform a Readiness Check

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

1. 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, as described in *Upgrading to the Oracle Fusion Middleware Infrastructure*:

- Upgrade from a previous 12c release to 12.2.1.4
- Upgrade from 11g release to 12.2.1.4
- Upgrade from a previous 12c release to 12.2.1.3
- Upgrade from 11g release to 12.2.1.3

If your service instance is based on WebLogic Server 12.2.1.3, 12.2.1.2, or 12.2.1.0, and you have configured Oracle Traffic Director (OTD) for the service instance, see also Starting the Upgrade Assistant for Oracle Traffic Director 12c Specific Upgrade.

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 the Infrastructure Database Schemas

After you perform a readiness check, use the Upgrade Assistant to upgrade the infrastructure database schemas. The procedures for upgrading from a previous WebLogic Server 12c release and a WebLogic Server 11g release differ, so they are presented in different topics.

Topics:

- Upgrade the WebLogic Server 12c Infrastructure Database Schemas
- Upgrade the WebLogic Server 11g Infrastructure Database Schemas

Upgrade the WebLogic Server 12c Infrastructure Database Schemas

Upgrade the previous WebLogic Server 12c database schemas by using the Upgrade Assistant (UA).

> **Note:**
>
> Use this procedure when upgrading from WebLogic Server release 12.2.1.0, 12.2.1.2, or 12.2.1.3.

The Upgrade Assistant creates missing or required schemas by using the default schema settings.

Ensure that you have backed up the database. See Perform Prerequisite Tasks.

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. Complete the steps as described in *Upgrading to the Oracle Fusion Middleware Infrastructure*:

   - Upgrade to 12.2.1.4 - Upgrading Schemas Using the Upgrade Assistant
   - Upgrade to 12.2.1.3 - Upgrading Schemas Using the Upgrade Assistant

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.

7. Verify the schema upgrade was successful by checking that the schemas in `schema_version_registry` have been properly updated. See Verifying the Schema Upgrade (upgrade to 12.2.1.4) or Verifying the Schema Upgrade (upgrade to 12.2.1.3) 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

Upgrade the WebLogic Server 11g Infrastructure Database Schemas

Upgrade the previous WebLogic Server 11g database schemas by using the Upgrade Assistant (UA).

> **Note:**
>
> Use this procedure when upgrading from WebLogic Server release 11g (11.1.1.7).

The Upgrade Assistant creates required WebLogic Server 12c schemas by using the default schema settings.

One of the missing schemas is the Service Table schema (`<prefix>_STB`), which is new in WebLogic Server 12c and is required for domain-based upgrades. The Service Table schema stores basic schema configuration information (for example, schema prefixes and passwords) that can be accessed and used by other Oracle Fusion Middleware components when creating the domain. The WebLogic Services schema (`<schema_prefix>_WLS`) is also new in WebLogic Server 12c and is required for domain-based upgrades.

Before upgrading the infrastructure database schemas, ensure that you have backed up the database. See Perform Prerequisite Tasks.

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. Complete the steps as described in *Upgrading to the Oracle Fusion Middleware Infrastructure*:

- Upgrade to 12.2.1.4 - Creating and Upgrading Schemas Using the Upgrade Assistant
- Upgrade to 12.2.1.3 - Creating and Upgrading Schemas Using the Upgrade Assistant

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 Create Schemas screen:

   a. Accept the default selection for **Create missing schemas for specified domain**.

   b. Enter and confirm the password for the **Common Infrastructure Services** (STB schema) and the **Oracle WebLogic Services** (WLS) components. Use the password you specified when you created the Oracle Java Cloud Service instance.

      - For a service instance on Oracle Cloud Infrastructure Classic, use the WebLogic administrator password.
      - For a service instance on Oracle Cloud Infrastructure, the schema password is encrypted, so you must first change this password. Use the procedure in Change the Schema Password Manually.

   c. Locate the <schema_prefix>_IAU schema and note the generated schema prefix. You will need this value if you need to change the password for the schema later.

5. On the Upgrade Progress screen in the Upgrade Assistant, monitor the schema upgrade progress.

6. 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
     ```

7. Run the Upgrade Assistant a second time to manually upgrade the <schema_prefix>_IAU schema.

   This step is necessary because the Upgrade Assistant only upgrades schemas used by the domain. In WebLogic Server 11g, the <schema_prefix>_IAU schema is not used in the domain by default. Therefore, the schema is not upgraded by Upgrade Assistant and you must upgrade the schema manually.

   a. 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
      ```

   b. Select **Individually Selected Schema**, and then **Oracle Audit Services**.

   c. In the **Database Connect String** field, enter the same connect string as the one used by the other schemas.

d. In the **Schema User Name** field, enter the user name with the same schema prefix used by the other schema.

e. Enter the schema password.

   If you do not know the schema password, change the password by using the procedure in Change the Schema Password Manually. You will need the schema prefix you noted in Step 4.

8. Remedy the database connection failure if one occurs. See Problems with Database Connectivity When Upgrading the Infrastructure Schema Database.

9. Verify the schema upgrade was successful by checking that the schemas in `SCHEMA_VERSION_REGISTRY` have been properly updated. See Verifying the Schema Upgrade (upgrade to 12.2.1.4) or Verifying the Schema Upgrade (upgrade to 12.2.1.3) 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 'SP1556656982';
```

   Example output for the `SCHEMA_VERSION_REGISTRY`:

MRC_NAME	COMP_ID	OWNER	VERSION	STATUS	UPGRADED
SP1556656982	IAU	SP1556656982_IAU	12.2.1.2.0	VALID	Y
SP1556656982	IAUDES	SP1556656982_IAUDS	11.1.1.4.0	VALID	N
SP1556656982	IAU_APPEND	SP1556656982_IAU_APPEND	12.2.1.2.0	VALID	Y
SP1556656982	IAU_VIEWER	SP1556656982_IAU_VIEWER	12.2.1.2.0	VALID	Y
SP1556656982	MDS	SP1556656982_MDS	12.2.1.3.0	VALID	Y
SP1556656982	OPSS	SP1556656982_OPSS	12.2.1.0.0	VALID	Y
SP1556656982	STB	SP1556656982_STB	12.2.1.3.0	VALID	N
SP1556656982	WLS	SP1556656982_WLS	12.2.1.0.0	VALID	N

Reconfigure the Domain

After you upgrade the infrastructure database schemas, use the Reconfiguration Wizard to reconfigure the domain. The procedures for reconfiguring the WebLogic Server 12c (12.2.1.0, 12.2.1.2, or 12.2.1.3) and WebLogic Server 11g (11.1.1.7) domains differ, so they are presented in different topics.

Topics:

- Reconfigure the WebLogic Server 12c Domain
- Reconfigure the WebLogic Server 11g Domain

Reconfigure the WebLogic Server 12c Domain

Reconfigure the domain by using the Reconfiguration Wizard.

> **Note:**
>
> Do not reconfigure the domain if you are upgrading an existing WebLogic Server release 12.2.1.3 to WebLogic Server release 12.2.1.4 for an existing Oracle Java Cloud Service instance. If you skip this step in the upgrade process, the domain `config.xml`, specifically, the `domain-version` is not updated, and you still see the earlier version. This is a known issue and you need not run the upgrade again. See Incorrect Version Numbers After a Reduced Downtime Upgrade . After the upgrade is complete, you see the correct version in the admin server logs and in the WebLogic Server Administration Console.
>
> However, if your service instance is based on WebLogic Server 12.2.1.3, 12.2.1.2, or 12.2.1.0, and you have configured Oracle Traffic Director (OTD) for the service instance, you must reconfigure the domain on your OTD instances. See also Reconfiguring the Existing Oracle Traffic Director Domain in Graphical Mode.

Before you begin, ensure you have backed up the domain on all nodes. See Perform Prerequisite Tasks.

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*.
   - Upgrade to 12.2.1.4 - Reconfiguring the Domain with the Reconfiguration Wizard
   - Upgrade to 12.2.1.3 - 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 Administration Server.
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.

Reconfigure the WebLogic Server 11g Domain

Reconfigure the WebLogic Server 11g domain by using the Reconfiguration Wizard.

> **Note:**
>
> Use this procedure if you are upgrading from WebLogic Server 11g (11.1.1.7).

Before you begin, ensure you have backed up the domain on all nodes. See Perform Prerequisite Tasks.

1. Start the Reconfiguration Wizard as the `oracle` user. Specify the following logging options, including the absolute path of the log file you'd like to create for the domain reconfiguration session.

   For example:

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

   You can use this log file to troubleshoot any reconfiguration problems.
2. Perform the reconfiguration tasks shown in *Upgrading to the Oracle Fusion Middleware Infrastructure*.
   - Upgrade to 12.2.1.4 - Reconfiguring the Domain with the Reconfiguration Wizard
   - Upgrade to 12.2.1.3 - Reconfiguring the Domain with the Reconfiguration Wizard

   Follow the special instructions below for the following screens: Database Configuration Type, Component Datasources, Node Manager, and Advanced Configuration.
3. On the Database Configuration Type screen, enter the RCU data.

The Service Table schema (`<prefix>_STB`) is new in WebLogic Server 12c, so it's not present in WebLogic Server 11g. As a result, you must enter the RCU data manually.

a. Select **RCU Data** to connect to the Server Table (`_STB`) schema. The Reconfiguration Wizard uses this connection to automatically configure the data sources required for components in your domain.

b. Enter the database connection details using the RCU service table (`_STB`) schema credentials. Note that if your service instance is on Oracle Cloud Infrastructure, you changed this password when you upgraded the infrastructure database schema. See Upgrade the WebLogic Server 11g Infrastructure Database Schemas.

c. Click **Get RCU Configuration**.

d. Click **Next**.

4. If your database has `_OPSS` or `_IAU` WebLogic Server 11g database schemas, enter the database connection details on the Component Datasources screen, and then click **Next**.

   - The `_IAU` or `_OPSS` schema displayed on the Component Datasources screen has no connection data, so the default connection data is used.
   - For each schema (`<schema_prefix>_IAU`, `<schema_prefix>_IAU_APPEND`, `<schema_prefix>_IAU_VIEWER`, and `schema_prefix_OPSS`) enter data in the **Host Name**, **DBMS Service**, **Port**, **Schema Owner**, and **Schema Password** fields for each of the `IAU` schemas. The **Host Name**, **DBMS Service**, **Port**, and **Schema Password** are the same as those used for the other schema.

5. Migrate the per-host Node Manager configuration to a per-domain configuration for the reconfigured domain.

   The Node Manager screen is only displayed if the domain you are reconfiguring is currently using a per-host Node Manager. Because you are upgrading your service instance from WebLogic Server 11g, the Node Manager screen appears, and you must migrate the existing per-host Node Manager configuration.

   a. On the Node Manager screen, select **Per Domain Default Location** for the **Node Manager Type**.

   In this configuration, the Node Manager home is redefined to `$domain_name/nodemanager`.

   b. Select **Migrate Existing Configuration** for **Node Manager Configuration**, browse and select your **Node Manager Home**, and then select **Apply Oracle Recommended Defaults**.

   The existing per-host Node Manager configuration will be migrated to a per-domain configuration for the reconfigured domain.

   c. For **Node Manager Credentials**, specify the username and password that you want to use to start Node Manager in the reconfigured domain.

   d. Click **Next**.

6. On the Advanced Configuration screen of the Reconfiguration Wizard, select **Deployment and Services**.

7. Target the `wsm-pm` app to the Administration Server.

8. Click **Reconfig**.

9. 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

Use the Upgrade Assistant to update the domain component configurations to match the updated domain configuration.

Before you begin, you must first run the Reconfiguration Wizard to reconfigure the WebLogic domain to WebLogic Server release 12.2.1.3.

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, as described in *Upgrading to the Oracle Fusion Middleware Infrastructure*.
   - Upgrade from a previous 12c release to 12.2.1.4 - Upgrading the Domain Configurations with the Upgrade Assistant
   - Upgrade from 11g release to 12.2.1.4 - Upgrading the Domain Configurations with the Upgrade Assistant
   - Upgrade from a previous 12c release to 12.2.1.3 - Upgrading the Domain Configurations with the Upgrade Assistant
   - Upgrade from 11g release to 12.2.1.3 - Upgrading the Domain Configurations with the Upgrade Assistant

   If your service instance is based on WebLogic Server 12.2.1.3, 12.2.1.2, or 12.2.1.0, and you have configured Oracle Traffic Director (OTD) for the service instance, see also Upgrading Collocated Oracle Traffic Director in Graphical Mode.

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
     ```

7. If you are upgrading the WebLogic Server release from 11g, connect to the Administration Server and set the `CrashRecoveryEnabled` property to `true` in the `<domain_home>/nodemanager/nodemanager.properties` file.

   ```
   CrashRecoveryEnabled=true
   ```

   Repeat this process for each Managed Server.

   The Node Manager `CrashRecoveryEnabled` configuration property is used to restart servers after a system crash. During domain reconfiguration, this property is reset to `false`, so it's not enabled by default.

Restart the Administration Server Node

After you complete the upgrade of your service instance, restart the Node Manager and WebLogic Server processes on the first node in your instance.

1. Access the Administration Server node as the `oracle` user.
2. Start the Node Manager process.

   ```
   /u01/data/domains/<domain>/bin/startNodeManager.sh >nm.out 2>&1 &
   ```

3. Launch the WebLogic Scripting Tool (WLST).

   ```
   $MIDDLEWARE_HOME/oracle_common/common/bin/wlst.sh
   ```

4. Connect to the Node Manager.

   ```
   nmConnect(username="<nm_user>", password="<nm_password>",
   domainName="<domain>", domainDir="/u01/data/domains/<domain>",
   nmType="ssl", host="<hostname>", port="5556", verbose="false")
   ```

   Example:

   ```
   nmConnect(username="weblogic", password="<nm_password>",
   domainName="MyInstan_domain", domainDir="/u01/data/domains/
   MyInstan_domain", nmType="ssl", host="myinstance-wls-1", port="5556",
   verbose="false")
   ```

   By default, the Node Manager credentials are the same as the WebLogic Server credentials that you specified when you created the instance.

5. Start the Administration Server and the first Managed Server.

   ```
   nmStart('<server>')
   nmServerStatus('<server>')
   ```

   Example:

   ```
   nmStart('MyInstan_adminserver')
   nmServerStatus('MyInstan_adminserver')
   ```

```
nmStart('MyInstan_server_1')
nmServerStatus('MyInstan_server_1')
```

6. Exit WLST.

   ```
   exit()
   ```

7. If your service instance is based on WebLogic Server 12.2.1.3, 12.2.1.2, or 12.2.1.0, and you have configured Oracle Traffic Director (OTD) for the service instance:

   a. Make sure execute permissions are in place for the scripts in `otd_domain`.

      For example:

      ```
      chmod -R +x /u01/data/otd-instance/
      ```

   b. Start the AdminServer.

      ```
      nohup /u01/data/otd-instance/otd_domain/bin/startWebLogic.sh &
      2>&1
      ```

   c. Start the NodeManager.

      ```
      nohup /u01/data/otd-instance/otd_domain/bin/startNodeManager.sh
      & 2>&1
      ```

   d. Start the OTD node.

      ```
      /u01/data/otd-instance/otd_domain/config/fmwconfig/
      components/OTD/instances/<lb-instance-name>/bin/startserv
      ```

Update and Restart the Managed Server Nodes

If your service instance has one or more Managed Server nodes, then update the domain on these nodes before restarting the Managed Servers.

1. Inspect all your managed server nodes to ensure that the 12.2.1.4 or 12.2.1.3 binaries previously downloaded using the REST API are in the `/u01/app/oracle/middleware` directory and the `/u01/jdk` directory. See Install the Upgrade Software.
2. Access the Administration Server node as the `oracle` user.
3. Use the `pack` command to create a Managed Server template from your domain.

   ```
   /u01/app/oracle/middleware/oracle_common/common/bin/pack.sh -
   domain=/u01/data/domains/<domain> -
   template_name=managedServerTemplate -template=/tmp/
   managed_server_template.jar -managed=true -log=/tmp/pack.log
   ```

4. Upload the template file from the Administration Server node to a Managed Server node in this service instance.

   ```
   scp /tmp/managed_server_template.jar <node_host>:/tmp
   ```

   Example:

   ```
   scp /tmp/managed_server_template.jar myinstance-wls-2:/tmp
   ```

5. Connect to the Managed Server node.

   Example:

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

6. Use the `unpack` command to apply the Managed Server template to the domain directory on this node.

   ```
   /u01/app/oracle/middleware/oracle_common/common/bin/unpack.sh -
   domain=/u01/data/domains/<domain> -template=/tmp/
   managed_server_template.jar -overwrite_domain=true -log=/tmp/unpack.log
   ```

   Ignore any warnings about "invalid or missing JDBC datasource connection parameters."

7. Start the Node Manager process on this node.

   ```
   /u01/data/domains/<domain>/bin/startNodeManager.sh >nm.out 2>&1 &
   ```

8. Disconnect from the Managed Server node.
9. Repeat Steps 3 to 6 on all other Managed Server nodes in your service instance.
10. Access the WebLogic Server Administration Console for your service instance.
11. From Domain Structure, expand **Environment**.
12. Click **Servers**.
13. Click the **Control** tab.
14. Click the check box to the left of each server that is not running.
15. Click **Start**. When prompted for confirmation, click **Yes**.

Perform Post-Upgrade Tasks

Perform any needed post-upgrade tasks.

After upgrading the WebLogic Server release for your Oracle Java Cloud Service instance, perform only those tasks described in *Upgrading to the Oracle Fusion Middleware Infrastructure* that apply to your upgraded environment.

- Upgrade to 12.2.1.4 - Tasks to Perform After Upgrade
- Upgrade to 12.2.1.3 - Tasks to Perform After Upgrade

For example, see the topics Using the Upgrade Validation Checklist, Verifying the Domain-Specific-Component Configurations Upgrade, and Reapplying Custom Configuration Settings to setDomainEnv.

Roll Back an Upgrade

If the upgrade fails, you can roll back the upgrade.

Rolling back an upgrade returns the WebLogic Servers for your Oracle Java Cloud Service instance to their previous release number.

1. Shutdown all WebLogic Server processes, including Node Manager, on each node. See Stop All WebLogic Server Processes.
2. Use the REST API to get a list of applied patches.

   ```
   curl -i -X GET -u <user>:<password> -H "X-ID-TENANT-
   NAME:<identitydomain>" https://<rest_server_url>/paas/api/v1.1/
   instancemgmt/<identity-domain>/services/jaas/instances/
   <servicename>/patches/applied
   ```

   For example:

   ```
   curl -i -X GET -u username:password -H "X-ID-TENNANT-
   NAME:ExampleIdentityDomain" https://<rest_server_url>/paas/api/v1.1/
   instancemgmt/ExampleIdentityDomain/services/jaas/instances/Example1/
   patches/applied
   ```

   This example output shows details about applied patches:

   ```
   [
      {
         "backupStatus": "Available",
         "rollbackStatus":"Available",
         "additionalNote": "This note is the default note: Applying
   patch [wls_upg_12.2.1.3.190115_for_12cRelease212].",
         "appliedBy": "weblogic",
         "appliedDate":"2019-03-07T19:07:37.246+0000",
         "totalTime": "2 min, 11 sec",

   "patchId":"wls_upg_12.2.1.3.190115_for_12cRelease212",

         "patchDescription": "WebLogic Server 12.2.1.3.0 with PSU
   Update 12.2.1.3.190115",
         "patchReleaseUrl":"https://support.oracle.com/epmos/faces/PatchDetail?
   patchId=28710939",
         "releaseDate": "2019-01-14T17:40:00.000+0000",
         "lastStatus": "COMPLETED",
         "lastStatusMessage": "No errors",
         "componentPatches": {},
         "patchType": "PSU",
         "patchCategory": "MajorPatch",
         "patchSeverity": "Normal",
         "jobId":"108717",
         "displayName": "12.2.1.3.190115",
         "inProgress": false,
         "operationType":"None",
   ```

```
      "id": 16,
      "patchingResult": {
         "patchingId": 16,
         "versionBeforeThisPatch": "OTD 12.2.1.2.190115,WLS 
12.2.1.2.190128,SERVICEVERSION 12cRelease212",
         "strategy": "Rolling",
         "releaseVersionBeforeThisPatch": 
"12.2.1.2.190128",
         "customRollbackId": "108717_1551985657246",
         "startDate": "2019-03-07T19:07:37.246+0000",
         "endDate": "2019-03-07T19:09:48.411+0000",
         "patchingStatus": "COMPLETED","resultMessage":"No errors",
         "additionalNote": "This note is the default note:Applying patch 
[wls_upg_12.2.1.3.190115_for_12cRelease212].",
         "appliedBy":"weblogic",
         "jobId": "108717",
         "completeLog": "",
         "progressMessages": [
            "7:07:37.153 PM Phase initialize 
started",
            "7:07:37.445 PM Phase initialize completed",
            "7:07:37.718 PM patching.action.tools.phase_started",
            "7:07:37.861 PM 
patching.action.tools.phase_completed",
            "7:07:38.067 PM Phase backup started",
            "7:07:38.214 PM Phase backup 
completed",
            "7:07:38.529 PM patching.action.patch-pre-
action.phase_started",
            "7:07:38.683 PM patching.action.patch-pre-
action.phase_completed",
            "7:07:38.949 PM Phase patch started",
            "7:07:31.043 PM PSM-PATCH-60099: Removing any retained old 
artifacts",
            "7:07:36.767 PM 
patching.progress.remove.left.over.binary.completed$jcs/FMW/
12.2.1.3.190115/190115/fmiddleware.zip$pltinstance-wls-1",
            "7:07:36.767 PM Retrieving pre-patched binary artifact 
[jcs/FMW/12.2.1.3.190115/190115/fmiddleware.zip] on vm [pltinstance-
wls-1]",
            "7:07:49.404 PM Retrieved pre-patched binary artifact 
[jcs/FMW/12.2.1.3.190115/190115/fmiddleware.zip] on vm [pltinstance-
wls-1]",
            "7:07:49.404 PM Unpacking binary [jcs/FMW/
12.2.1.3.190115/190115/fmiddleware.zip] on vm [pltinstance-wls-1]",
            "7:08:35.616 PM Unpacked binary [jcs/FMW/
12.2.1.3.190115/190115/fmiddleware.zip] on vm [pltinstance-wls-1]",
            "7:08:35.616 PM PSM-PATCH-60099: Removing any retained old 
artifacts",
            "7:08:36.283 PM 
patching.progress.remove.left.over.binary.completed$jcs/JDK/
8.0.201/190115/jdk.zip$pltinstance-wls-1",
            "7:08:36.283 PM Retrieving pre-patched binary artifact 
[jcs/JDK/8.0.201/190115/jdk.zip] on vm [pltinstance-wls-1]",
            "7:08:38.250 PM Retrieved pre-patched binary artifact 
[jcs/JDK/8.0.201/190115/jdk.zip] on vm         [pltinstance-wls-1]",
            "7:08:38.250 PM Unpacking binary [jcs/JDK/
8.0.201/190115/jdk.zip] on vm [pltinstance-wls-1]",
            "7:08:44.180 PM Unpacked binary [jcs/JDK/8.0.201/190115/
jdk.zip] on vm [pltinstance-wls-1]",
            "7:08:44.180 PM Retrieving pre-patched binary artifact
[jcs/lb/12.2.1.2.0/161019/wls_otd.zip] on vm         [pltinstance-
wls-1]",
            "7:08:44.181 PM Retrieved pre-patched binary artifact
[jcs/lb/12.2.1.2.0/161019/wls_otd.zip] on vm         [pltinstance-
wls-1]",
            "7:08:44.181 PM Retrieving pre-patched binary artifact
[jcs/JDK/8.0.201/190115/jdk.zip] on vm         [pltinstance-wls-1]",
            "7:09:46.509 PM Phase patch completed",
            "7:09:46.771 PM patching.action.patch-post-
action.phase_started",
            "7:09:46.937 PM patching.action.patch-post-
action.phase_completed",
            "7:09:48.227 PM Phase finalize started",
            "7:09:48.352 PM Completed"
         ]
      },
      "rollbackId":"16",
      "rollbackVersion":"12.2.1.2.190128",
      "currentPatchLevel":"12.2.1.3.190115",
      "isUpgrade": true,
      "appliedPatchGuiMetadata": {
         "supportsPreRollbackCheck": false
      }
   }
]
```

3. From the response, find the previously applied upgrade patch, `patchId`, and then find the value of `rollbackId`. In this example, the `rollbackId` is `16`.

4. Use the REST API to roll back the patch.

   ```
   curl -i -X PUT -u <user>:<password> -d {} -H "Content-
   Type:application/json" -H "X-ID-TENANT-NAME:<identitydomain>"
   https://<rest_server_url>/paas/api/v1.1/instancemgmt/
   <identitydomain>/services/jaas/instances/<servicename>/patches/
   <rollbackid>/rollback
   ```

   For example:

   ```
   curl -i -X PUT -u username:password -d {} -H "Content-
   Type:application/json" -H "X-ID-TENANT-NAME:ExampleIdentityDomain"
   https://<rest_server_url>/paas/api/v1.1/instancemgmt/
   ExampleIdentityDomain/services/jaas/instances/Example1/patches/16/
   rollback
   ```

Example output:

```
{
   "status":"Completed",
   "details":{
     "message":"PSM-PATCH-50038:  Rollback of service from patch
[wls_upg_12.2.1.3.190115_for_12cRelease212] is submitted as an
asynchronous job.",
     "jobId":"206236"
   }
}
```

5. Restore the backup of the database instance.
   - If you backed up an Oracle Database Cloud Service deployment, use the procedure shown in Restoring from a Specific Backup in *Administering Oracle Database Cloud Service*.
   - If you backed up an Oracle Cloud Infrastructure database, use the procedure shown in *To restore a database from a specific backup from Object Storage* in Recovering a Database from Object Storage in the Oracle Cloud Infrastructure documentation.
6. Replace the files in `MIDDLEWARE_HOME` (`/u01/app/oracle/middleware`), `JDK_HOME` (`/u01/jdk`), and domain directory (`/u01/data/domains`) on each node with those you saved earlier. See Perform Prerequisite Tasks.
7. Restart the Node Manager and WebLogic Server processes on the Administration Server node. See Restart the Administration Server Node.
8. Restart the Node Manager and WebLogic Server processes on the Managed Server nodes. See Update and Restart the Managed Server Nodes.

The WebLogic Server installation and domain for your Oracle Java Cloud Service instance have been returned to their previous release number.

11
Secure an Oracle Java Cloud Service Instance

Security in Oracle Java Cloud Service spans many topics, including authentication, authorization, password management and network security.

Topics

- About Security in Oracle Java Cloud Service
- Use Oracle Identity Cloud Service with Oracle Java Cloud Service
- Configure Network Security
- Authenticate Users

About Security in Oracle Java Cloud Service

You secure applications deployed to your Oracle Java Cloud Service instance through the capabilities of Oracle Cloud, Java EE standards and Oracle WebLogic Server.

An Oracle Java Cloud Service instance includes an Oracle WebLogic Server domain, which is comprised of an Administration Server and one or more Managed Servers. A domain also defines a security realm that controls authentication, authorization, role mapping, credential mapping and security auditing across all of the servers in the domain. Java applications deployed to this WebLogic Server domain can be associated with security roles and policies that protect the applications against unauthorized access. WebLogic Server supports various security providers that assign an identity to the requesting user. By default, users, groups, roles and policies are all maintained in WebLogic Server's embedded LDAP server.

Alternatively, Oracle Java Cloud Service instances can use Oracle Identity Cloud Service as an identity store in addition to the embedded LDAP server. If either of these security configurations does not meet your requirements, you can modify the security realm or create a new one with any combination of security providers. For large production applications, Oracle recommends that you use a proper identity management system such as Oracle Identity Cloud Service instead of the embedded LDAP server.

To provide the highest level of network security, Oracle Java Cloud Service implements an "access by exception" architecture. You must explicitly grant network access to your service instance for administrators, application users or other cloud services. By default, a service instance is accessible only through secure protocols like HTTPS and SSH, and only using specific ports. You're also able to customize the default network security configuration to support different access rules and security policies.

To learn more about Oracle Java Cloud Service security see:

- Use Oracle Identity Cloud Service with Oracle Java Cloud Service
- About Users
- About Authentication
- About the Default Access Ports

- Create an Access Rule

To learn more about the Java EE and WebLogic Server security architecture see:

- *Understanding Security for Oracle WebLogic Server* (12.2.1.3)
- *Understanding Security for Oracle WebLogic Server* (12.2.1.2)
- *Understanding Security for Oracle WebLogic Server* (11.1.1.7)

To learn more about the security capabilities of an Oracle Coherence data grid see these topics in *Securing Oracle Coherence*:

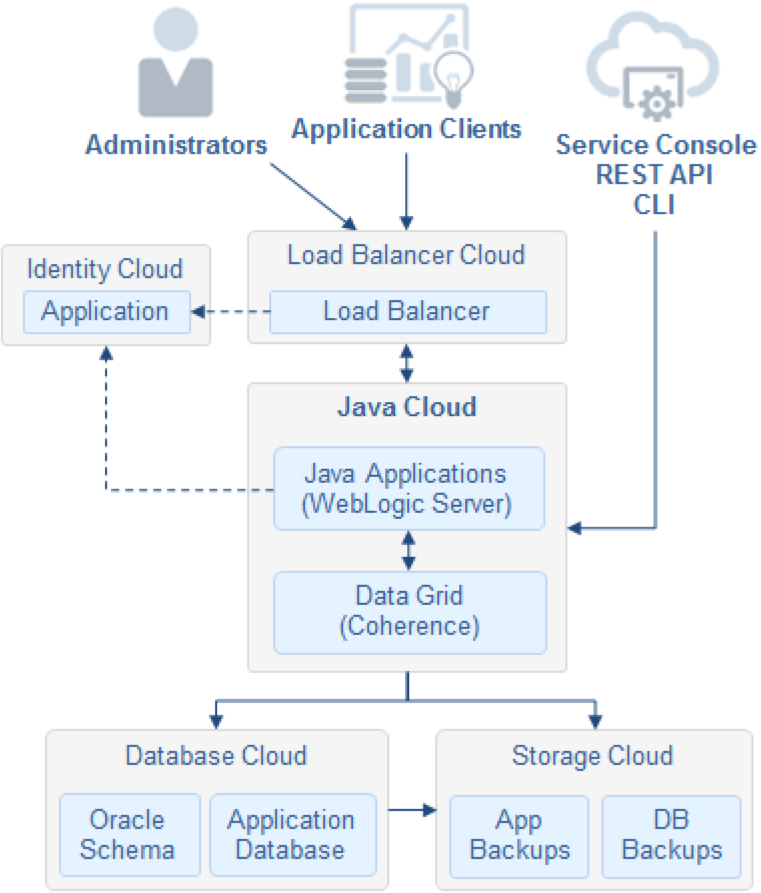
- Securing Oracle Coherence in Oracle WebLogic Server (12.2.1.3)
- Securing Oracle Coherence in Oracle WebLogic Server (12.2.1.2)

Use Oracle Identity Cloud Service with Oracle Java Cloud Service

Oracle Java Cloud Service instances can use Oracle Identity Cloud Service to authenticate Oracle WebLogic Server administrators and application users.

A series of Tutorials is also available on using Oracle Java Cloud Service with Oracle Identity Cloud Service.

This figure illustrates a service instance that has been configured to use Oracle Identity Cloud Service and an Oracle-managed load balancer.



Topics

- About Oracle Identity Cloud Service
- Manage Service Administrators
- Create a Service Instance with Oracle Identity Cloud Service
- Manage WebLogic Server Administrators
- Configure Protected Application Contexts for a New Service Instance
- Update Protected Application Contexts on an Existing Service Instance
- Secure an Application Using Deployment Descriptors
- Oracle Identity Cloud Service Cloudgate Patching on Oracle Java Cloud Service

About Oracle Identity Cloud Service

Oracle Identity Cloud Service provides Oracle Cloud administrators with a central security platform to manage the relationships that your users have with your applications, including with other Oracle Cloud services like Oracle Java Cloud Service.

With Oracle Identity Cloud Service you can create custom password policies and email notifications, onboard new users, assign users and groups to applications, and run security reports. See these topics in *Administering Oracle Identity Cloud Service*:

- About Oracle Identity Cloud Service Concepts
- How to Access Oracle Identity Cloud Service

Manage Service Administrators

If your Oracle Cloud account includes Oracle Identity Cloud Service, use it to create users and groups and to give them access to Oracle Java Cloud Service.

Services in your account can be associated with Oracle Identity Cloud Service security *applications*. Each security application defines one or more *roles*. Assign users and groups to these application roles in order to grant them administrative access to these services. For example, in order to create and manage Oracle Java Cloud Service instances, assign users and groups to the `JaaS_Administrator` role.

In Oracle Cloud Infrastructure Classic regions, you can also assign users and groups to these related roles:

- `Compute_Operations` — Create Oracle Java Cloud Service instances on Oracle Cloud Infrastructure Classic regions.
- `DBaaS_Administrator` (optional) — Create and manage Oracle Database Cloud Service deployments.
- `Storage_ReadWriteGroup` (optional) — Enable backups for an Oracle Java Cloud Service instance, and store the backups in an existing Oracle Cloud Infrastructure Object Storage Classic container.
- `Storage_Administrator` (optional) — Create Oracle Cloud Infrastructure Object Storage Classic containers to use as backup storage locations for Oracle Java Cloud Service instances.

See these topics in *Administering Oracle Identity Cloud Service*:

- Creating User Accounts
- Creating Groups
- Assigning Users to Oracle Applications
- Assigning Groups to Oracle Applications

To create and manage resources in Oracle Cloud Infrastructure regions like databases and storage, administrators define *policies* that grant privileges to users and groups. See Securing IAM in the Oracle Cloud Infrastructure documentation.

Create a Service Instance with Oracle Identity Cloud Service

When you create an Oracle Java Cloud Service instance, you have the option to integrate the Oracle WebLogic Server domain in the instance with Oracle Identity Cloud Service.

When you select the **Enable Authentication Using Identity Cloud Service** option for a service instance, Oracle Java Cloud Service creates a new security application in Oracle Identity Cloud Service that's integrated with your WebLogic Server domain. For convenience, the **Overview** page of your service instance includes a link to this security application.

You can also create a service instance within a specific **Identity Domain** in Oracle Identity Cloud Service (Not available on Oracle Cloud at Customer). Each identity domain has an independent set of users. For example, you might create separate identity domains for test users and production users. By default, the security application for a service instance is created in the primary identity domain in Oracle Identity Cloud Service.

One way to create an Oracle Java Cloud Service instance is to clone an existing service instance. However, you cannot clone a service instance if authentication with Oracle Identity Cloud Service is enabled for the service instance.

Oracle Java Cloud Service provisions and configures an Oracle-managed load balancer for service instances that use Oracle Identity Cloud Service. For instances created in Oracle Cloud Infrastructure regions, you cannot update the default configuration for a load balancer that is provisioned automatically during the creation of the service instance.

Manage WebLogic Server Administrators

An Oracle Java Cloud Service instance includes an Oracle WebLogic Server domain. This domain defines a security realm that controls authentication, authorization, role mapping, credential mapping and security auditing across all of the servers in the domain.

All domains include predefined roles such as `Administrators`, `Operators`, `Deployers` and `Monitors`. The WebLogic Server administrative user whose credentials you initialize when you create a service instance (the default name is `weblogic`) is assigned the `Administrators` role, meaning that this user can perform all WebLogic Server administrative operations through either the WebLogic Server Administration Console, Fusion Middleware Control, WebLogic Scripting Tool (WLST) or WebLogic REST API.

When you create an Oracle Java Cloud Service instance and enable authentication with Oracle Identity Cloud Service:

- The WebLogic Server security realm in the service instance is configured to use Oracle Identity Cloud Service for authentication in addition to the default WebLogic Server identity store (embedded LDAP).
- A new security application is defined in Oracle Identity Cloud Service and associated with the service instance. This security application includes the same predefined WebLogic Server roles that are also found in the embedded LDAP.
- Your Oracle Cloud user name is assigned to the `Administrators` role in your domain’s security application in Oracle Identity Cloud Service.
- An Oracle-managed load balancer is provisioned in Oracle Cloud Infrastructure Load Balancing or Oracle Cloud Infrastructure Load Balancing Classic. Clients access applications on WebLogic Server via this load balancer.

As a result, you can use your Oracle Cloud user name to log into the WebLogic Server Administration Console and other WebLogic tools for your service instance. You can also use Oracle Identity Cloud Service to create additional users or groups, and assign them to the WebLogic Server administrator roles. These users and groups will then be able to perform WebLogic Server administrative operations in the service instance, depending on their assigned roles. For example, users with the `Deployer` role can deploy Java applications to the domain. See these topics in *Administering Oracle Identity Cloud Service*:

- Creating User Accounts
- Creating Groups
- Assigning Users to Oracle Applications
- Assigning Groups to Oracle Applications

The default WebLogic Server administrative user that you specify when creating your Oracle Java Cloud Service instance remains in the embedded LDAP. Use standard WebLogic Server tools like the Administration Console in order to modify this user or to change its password. Other users that you create with Oracle Identity Cloud Service are not stored in the embedded LDAP.

Configure Protected Application Contexts for a New Service Instance

When you create an Oracle Java Cloud Service instance with the REST API or CLI, you can configure the application contexts that require authentication via Oracle Identity Cloud Service. This feature is not available in the web console.

Users access Java applications that are deployed to your Oracle WebLogic Server domain through a load balancer hosted on Oracle Cloud Infrastructure Load Balancing or Oracle Cloud Infrastructure Load Balancing Classic. By default, this load balancer is configured to use Oracle Identity Cloud Service for authentication for all requests whose URI begins with `/__protected` (two underscore characters followed by the word "protected"). For example, a client request to the URL `https://`*`lb_host`*`/`**`__protected`**`/myapp/doaction` requires authentication, while a request to `https://`*`lb_host`*`/myapp/doaction` does not.

When you create a service instance by using the REST API or CLI, you can specify additional URL patterns that require authentication. See Create a Service Instance in *REST API for Oracle Java Cloud Service*. In the request body, use `protectedRootContexts` to specify one or more URL patterns. Begin each pattern with the `/` character, use the `*` character as a wildcard, and separate multiple patterns with a comma.

For example, suppose your existing Java applications are configured to use the context roots `store` and `marketplace`. To protect all resources in the `marketplace` application, and also those resources in `/store/cart`, use the following URI patterns when creating your service instance:

```
...
"useIdentityService": "true",
"protectedRootContexts": "/store/cart/*,/marketplace/*"
```

If you create an Oracle Java Cloud Service instance with multiple WebLogic Server clusters, each cluster is assigned a path prefix. Java applications are deployed to a specific cluster, and clients must access the applications by using the cluster's path prefix:

```
https://lb_host/cluster_path_prefix/path
```

By default, the path prefix for a cluster is the cluster's name. One cluster in your service instance can be assigned the path prefix `/`.

When you create a service instance with multiple clusters, be sure to include the path prefix in the URI patterns for `protectedRootContexts`. For example:

```
"protectedRootContexts": "/cluster1/store/cart/*,/cluster2/
marketplace/*"
```

Update Protected Application Contexts on an Existing Service Instance

This topic does not apply to Oracle Cloud at Customer.

After creating an Oracle Java Cloud Service service instance, you can use the REST API to replace the current list of protected context roots for your service instance. This feature is not available in the web console or CLI.

See Update the Web Tier Policy in *REST API for Oracle Java Cloud Service*. In the request body, provide a list of `resourceFilters`, where each `filter` is a regular expression that begins with the `/` character. Set the `method` of your custom filters to the value `oauth`.

For example, to protect all resources in the `marketplace` application, and also those resources in `/store/cart`, use the following regular expressions in the REST API request body:

```
...
"resourceFilters":[
  {
    "type":"regex",
    "filter":"/marketplace/.*",
    "method":"oauth"
  },
  {
    "type":"regex",
    "filter":"/store/cart/.*",
    "method":"oauth"
  },
  ...
```

Secure an Application Using Deployment Descriptors

Use Oracle Identity Cloud Service to protect a Java application running on Oracle Java Cloud Service by updating the application’s context path, security constraints, and role assignments.

In order to test your application’s updated security configuration, create application users and groups in Oracle Identity Cloud Service See these topics in *Administering Oracle Identity Cloud Service*:

- Creating User Accounts
- Creating Groups

Oracle WebLogic Server supports the Java Enterprise Edition declarative model for securing web applications with XML deployment descriptors.

1. Update the value of `context-root` in the application’s `weblogic.xml` file. Prefix the current value with one of the protected context roots that you configured for your Oracle Java Cloud Service instance.

   By default, the only protected context root is `/__protected` (two underscore characters followed by the word “protected”). For example:

   ```
   <context-root>/__protected/store</context-root>
   ```

If you created a service instance with multiple WebLogic Server clusters, you must specify the path prefix of the cluster to which the application is targeted. For example:

```
<context-root>/cluster1/__protected/store</context-root>
```

2. Create one or more `security-role` elements in the application's `web.xml` file.

   Simply list the user roles for your application. For example:

```
<security-role>
  <role-name>HRAdmin</role-name>
</security-role>
```

3. Create one or more `security-constraint` elements in the application's `web.xml` file.

   Each security constraint grants access to one or more URL patterns in your application, and to specific roles. For example:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>AdminPages</web-resource-name>
    <url-pattern>/admin/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>HRAdmin</role-name>
  </auth-constraint>
</security-constraint>
```

   Do not include the context root path in the URL patterns.

4. Create one or more `security-role-assignment` elements in the application's `weblogic.xml` file.

   Map your application roles to specific users and/or groups found in Oracle Identity Cloud Service. For example:

```
<security-role-assigment>
  <role-name>HRAdmin</role-name>
  <principal-name>HRManagersGroup</principal-name>
</security-role-assigment>
```

5. Redeploy your application for these changes to take effect. For example, use the WebLogic Server Administration Console.

For more information on configuring web application security, see these topics in *Developing Applications with the WebLogic Security Service*:

- Securing Web Applications (12.2.1.3)
- Securing Web Applications (11.1.1.7)

As an alternative to editing deployment descriptors, you can create a deployment plan file to override the application's configuration settings.

Oracle Identity Cloud Service Cloudgate Patching on Oracle Java Cloud Service

Learn how to upgrade Oracle Identity Cloud Service Cloudgate on Oracle Java Cloud Service, revert to previous Cloudgate versions, and clean up the old installation.

Topics

- Upgrade Cloudgate on Oracle Java Cloud Service
- Change to Previous Cloudgate Version
- Remove the Old Installations

Upgrade Cloudgate on Oracle Java Cloud Service

When you create an Oracle Java Cloud Service instance with Oracle Identity Cloud Service, you must upgrade Cloudgate to the latest available version after you update the Oracle Java Cloud Service tools on the OCI Java Cloud Service virtual machine. The upgrade is required only when a new Cloudgate version is released.

To upgrade Cloudgate, you should have enabled authentication with Oracle Identity Cloud Service when creating an Oracle Java Cloud Service instance in Oracle Cloud Infrastructure region.

To upgrade Cloudgate to a new version:

1. Download the latest available version on the virtual machine of the Oracle Java Cloud Service instance.

   The Cloudgate key for JCS blueprint is updated to point to the new Cloudgate location.

   Example of JCS Blueprint code for Cloudgate key:

   ```
   "cloudgateZipLocation" : "JAAS/19.2.1-1903110523/cloudgate.zip"
   ```

   After Oracle Java Cloud Service upgrades to the new version of JCS, the `cloudgate.zip` is downloaded to the `/u01/zips/cloudgate/<version>/` folder on the Java Cloud Service virtual machine, where `<version>` is the current Cloudgate version that is always greater than any previously downloaded Cloudgate versions.

2. To install the new version on the VM:

   > **Note:**
   >
   > You must perform these steps on each VM in the Oracle Java Cloud Service instance.

   a. SSH to the VM in your Oracle Java Cloud Service instance.

      ```
      ssh -i ~/<private_key opc@<ip address of the VM server>
      ```

   b. Switch to the `oracle` user.

```
sudo u oracle
```

c. Stop the Clougate service.

```
export CLOUDGATE_HOME=/u01/data/domains/cloudgate/idcs-cloudgate/
```

```
export CLOUDGATE_NGINX_OVERRIDE=/u01/data/domains/cloudgate/nginx
```

```
$CLOUDGATE_HOME/bin/cg-stop -p $CLOUDGATE_NGINX_OVERRIDE
```

d. Create a backup of the old Cloudgate installation.

```
mv $CLOUDGATE_HOME /u01/data/domains/cloudgate/idcs-
cloudgate_original/
```

e. Extract `cloudgate.zip` into `CLOUDGATE_HOME` directory.

```
unzip /u01/zips/cloudgate/<version>/cloudgate.zip -d /u01/data/
domains/cloudgate/
```

Make sure you select the zip file that has the highest version number.

f. Copy the new `nginx` library.

```
cp /u01/data/domains/cloudgate/idcs-cloudgate/bin/nginx /u01/data/
domains/cloudgate/nginx/sbin
```

This step is required when you upgrade Cloudgate versions 19.2.1 to 19.3.3 or 20.4.3, and 19.3.3 to 20.4.3.

g. After upgrade is complete, to update the configuration files for the Cloudgate version, copy the existing Cloudgate config and policy files based on instructions in the `README` file located in `CLOUDGATE_HOME/cfg/examples`.

> **Note:**
>
> If a new `nginx` entry is added in the configuration file:
>
> - Run a diff of each file under `/u01/data/domains/cloudgate/idcs-cloudgate/cfg` folder (updated version folder) and `/u01/data/domains/cloudgate/idcs-cloudgate-original/cfg folder` (previous installation folder). If the diff shows changes, replace the configuration element in the updated installation with the previous installation.
> - Run a diff of each file under `/u01/data/domains/cloudgate/idcs-cloudgate/cfg/nginx` folder (updated version folder) and `/u01/data/domains/cloudgate/idcs-cloudgate-original/nginx folder` (previous installation folder). If the diff shows changes, replace the configuration element in the updated installation with the previous installation

h. Start the Cloudgate Service.

```
export CLOUDGATE_HOME=/u01/data/domains/cloudgate/idcs-cloudgate/
```

```
export CLOUDGATE_NGINX_OVERRIDE=/u01/data/domains/cloudgate/nginx
```

```
$CLOUDGATE_HOME/bin/cg-start -p $CLOUDGATE_NGINX_OVERRIDE
```

3. Verify the upgrade.

```
cd /u01/data/domains/cloudgate/idcs-cloudgate/bin
```

```
./cg-env
```

This prints the Cloudgate version.

After upgrading Cloudgate to a new version, you can change to the previous Cloudgate Version in case of issue during upgrade. See Change to Previous Cloudgate Version.

Change to Previous Cloudgate Version

If you run into issues during upgrade of Cloudgate, you can revert to the previous Cloudgate version.

> **Note:**
>
> You must perform these steps on each VM in the Oracle Java Cloud Service instance.

1. Stop the Cloudgate Service.

   ```
   export CLOUDGATE_HOME=/u01/data/domains/cloudgate/idcs-cloudgate/
   ```

   ```
   export CLOUDGATE_NGINX_OVERRIDE=/u01/data/domains/cloudgate/nginx
   ```

   ```
   $CLOUDGATE_HOME/bin/cg-stop -p $CLOUDGATE_NGINX_OVERRIDE
   ```

2. Remove the current installation.

   ```
   rm -f $CLOUDGATE_HOME
   ```

3. Change to the previous installation.

   ```
   mv /u01/data/domains/cloudgate/idcs-cloudgate_original/ $CLOUDGATE_HOME
   ```

4. Start the Cloudgate Service.

   ```
   export CLOUDGATE_HOME=/u01/data/domains/cloudgate/idcs-cloudgate/
   ```

   ```
   export CLOUDGATE_NGINX_OVERRIDE=/u01/data/domains/cloudgate/nginx
   ```

   ```
   $CLOUDGATE_HOME/bin/cg-start -p $CLOUDGATE_NGINX_OVERRIDE
   ```

5. Verify the upgrade.

   ```
   cd /u01/data/domains/cloudgate/idcs-cloudgate/bin
   ```

   ```
   ./cg-env
   ```

   This prints the Cloudgate version.

Remove the Old Installations

You can remove the older Cloudgate versions by cleaning the old installation and the downloaded binaries. But, after removing the old installation, you cannot revert to the previous Cloudgate version.

1. Delete the backed up cloudgate installation.

   ```
   rm -rf /u01/data/domains/cloudgate/idcs-cloudgate_original/
   ```

2. (Optional) Delete the downloaded binary.

```
rm -rf /u01/zips/cloudgate/<version>
```

where `<version>` corresponds to the old cloudgate version.

Configure Network Security

By default, an Oracle Java Cloud Service instance is accessible only through secure protocols like SSL and SSH, and only using specific ports. But you can customize the default security configuration to support different access rules and security policies.

To provide the highest level of network security, Oracle Java Cloud Service implements an “access by exception” architecture. You must explicitly grant network access to your service instance for administrators, application users or other cloud services. Similarly, if you want your service instance to be accessible over a non-secure protocol like HTTP, you must change the default configuration.

Topics

- About the Default Access Ports
- Create an Ingress Rule
- Create an Access Rule
- Enable or Disable an Access Rule
- Delete an Access Rule
- Enable Console Access for a Service Instance
- Enable HTTP Access to a Service Instance
- Enable Communication Between Service Instances
- Enable Communication Between a Compute Instance and a Service Instance
- Configure SSL for a Service Instance

About the Default Access Ports

To use the resources available within your Oracle Java Cloud Service instance, access them through the default ports.

See the diagram in About the Deployment Topology of Virtual Machines for an illustration of the default port allocation in a service instance.

Topics

- Ports Open to Traffic from Outside the Oracle Cloud Network
- Ports Open to Traffic from Within the Oracle Cloud Network
- Administration Server Deployment Port

Ports Open to Traffic from Outside the Oracle Cloud Network

If the nodes in your service instance are assigned public IP addresses, then by default the following ports are accessible from the Internet.

If they are not assigned public IP addresses, then these ports are accessible only from within your private IP network, or from your on-premises data center over a VPN network.

Resource	Protocol	Default Port
Secure Shell (SSH) server	SSH	22
Oracle WebLogic Server Administration Console	HTTPS	7002
Oracle Fusion Middleware Control	HTTPS	7002
Oracle Traffic Director Administration Console	HTTPS	8989
End user applications when the load balancer is present	HTTP HTTPS	80 443
End user applications when the load balancer is not present and there are *multiple* Managed Servers	HTTP HTTPS	8001 8002
End user applications when the load balancer is not present and there is *only one* Managed Server	HTTP HTTPS	80 443

By default, if you created your service instance in an Oracle Cloud Infrastructure Classic region, external access to these administration consoles is disabled for security purposes. If you did not enable console access while provisioning your service instance, see Enabling Console Access in an Oracle Java Cloud Service Instance. If you created your service instance in an Oracle Cloud Infrastructure region, this procedure is not necessary. Access to the administration consoles is enabled by default in these regions.

For end user applications, the default ports depend on how the service instance was created.

- If the service instance was created by using the **web console**, the default ports are as follows:
  - If a load balancer is enabled, the HTTP port is disabled by default and the HTTPS port is 443 by default.
  - If a load balancer is not present and the service instance contains more than one Managed Server, the HTTP port is disabled and the HTTPS port is 8002.
  - If a load balancer is not present and the service instance contains only one Managed Server, the HTTP port is disabled and the HTTPS port is 443.
  - You can enable the HTTP port manually after you have created the service instance. See Enabling HTTP Access to an Oracle Java Cloud Service Instance.
- If the service instance was created by using the **REST API** or **CLI**, the default ports are as follows:
  - If a load balancer is present, the default ports for applications are 80 for HTTP and 443 for HTTPS. You can reconfigure these ports.
  - If a load balancer is not present and the service instance contains more than one Managed Server, the default ports are 8001 for HTTP and 8002 for HTTPS.
  - If a load balancer is not present and the service instance contains only one Managed Server, the default ports are 80 for HTTP and 443 for HTTPS.

Ports Open to Traffic from Within the Oracle Cloud Network

Some ports in your service instance are used for private communication between software components.

Resource	Protocol	Default Port
Oracle WebLogic Server Administration Console	HTTP	7001
Oracle Fusion Middleware Control	HTTP	7001
Managed Servers	HTTP HTTPS	8001 8002
Database	SQL Net	1521

Administration Server Deployment Port

The Administration Server node in your service instance has an additional port **9001** that supports the WebLogic-specific T3 protocol.

This port can be used with the WebLogic Scripting Tool (WLST), Integrated Development Environments (IDEs) or other WebLogic Server deployment tools. However, for security reasons the deployment port is not directly accessible from outside of this single node. You can create an SSH tunnel to make this port available to programs that are not running on the Administration Server node. See Create an SSH Tunnel to a Node with OpenSSH or Create an SSH Tunnel to a Node with PuTTY.

Create an Ingress Rule

To secure network access to Oracle Java Cloud Service instances provisioned in Oracle Cloud Infrastructure regions, you can define security ingress rules.

> **Note:**
>
> If you provisioned an Oracle Java Cloud Service instance without explicitly specifying a named subnet, the instance is assigned to the predefined Virtual Cloud Network (VCN) named `svc-vcn`, which is found in the `ManagedCompartmentForPaaS` compartment. You cannot modify resources in `svc-vcn`, such as assign security lists or add ingress rules.
>
> If your Oracle Java Cloud Service instance is assigned to `svc-vcn`, submit a Service Request (SR) with Oracle Support Services to obtain access for updating ports and ingress rules in `svc-vcn`.

When you create an Oracle Java Cloud Service instance, the WebLogic administration server and managed servers are configured with network channels that support HTTP and HTTPS traffic only. The administration server channels (7001 and 7002) and managed server channels (8001 and 8002) do not support the T3 and T3 over SSL (T3S) protocols, and they do not support tunneling. Internal T3 and T3S communication is done via ports 9071 and 9072 (administration server) and 9073 and 9074 (managed servers).

Before you can take advantage of features like Java Message Service or perform certain tasks such as deploying applications via Oracle JDeveloper, you'll need to set up security ingress rules to control access to ports 9071 and 9072 (administration server) or 9073 and 9074 (managed servers).

In the Oracle Cloud Infrastructure Console, add ingress rules to the appropriate ports (9071-9074) using a fixed set of IPs or a restricted CIDR that matches your organization's network addresses. This ensures only known IP addresses have access to the ports.

See To Create a Security List in the Oracle Cloud Infrastructure documentation.

Create an Access Rule

This topic does not apply to Oracle Cloud Infrastructure. Identify the Cloud Infrastructure Used by a Service Instance.

To control network access to the nodes in your Oracle Java Cloud Service instance, you can define access rules.

If you provisioned this service instance in an Oracle Cloud Infrastructure region, instead you must use the Oracle Cloud Infrastructure Console to create security lists instead of access rules. See Security Lists in the Oracle Cloud Infrastructure Services documentation.

For example, you can create rules that:

- Enable an Oracle Database node to access a specific port on your WebLogic Server nodes
- Enable public internet access to a specific port on the WebLogic Administration Server node

Oracle Java Cloud Service creates several *default* rules on a new service instance, such as public access to the WebLogic Administration Server node on port 22 for Secure Shell (SSH). Some of these are *system* rules, which cannot be disabled.

Access to the WebLogic Administration Console, Fusion Middleware Control Console, and Load Balancer Console is disabled by default on a new service instance. To use these consoles, you must enable the corresponding access rules.

> **Caution:**
>
> Make sure you consider the possible security implications before you open ports to external access.

Prior to creating an access rule, ensure that the destination node is configured to listen on the chosen ports. For example, on nodes running Oracle WebLogic Server you can configure network channels to control the listen ports for your Administration Server and Managed Servers. Refer to these topics in *Administering Server Environments for Oracle WebLogic Server*:

- Configuring Network Resources (12.2.1.3)
- Configuring Network Resources (11.1.1.7)

To create an access rule for a service instance:

1. Access your service console.

2. Beside the service that you want to modify, click **Manage this instance** ☰, and then select **Manage Access Rules**.
3. On the Access Rules page, click **Create Rule**.
4. In the **Rule Name** field, enter a name for the access rule.
5. Optional: In the **Description** field, enter a description for the access rule.
6. In the **Source** field, select a source for the rule. The available source options depend on the topology of your service instance, and may include:
   - **PUBLIC-INTERNET**: Any host on the internet
   - **WLS_ADMIN**: The WebLogic Administration Server node
   - **WLS_ADMIN_HOST**: The WebLogic Administration Server node
   - **WLS_MS**: All WebLogic Managed Server nodes
   - **OTD_ADMIN_HOST**: The Oracle Traffic Director (OTD) Administration Server node
   - **OTD_OTD_SERVER**: All Oracle Traffic Director (OTD) Managed Server nodes
   - `DBaaS:`*`Name`*`:DB`: The database service named *Name*
   - **<custom>** : A custom list of addresses from which traffic should be allowed. In the field that appears below this one, enter a comma-separated list of the subnets (in CIDR format, such as `192.0.2.11/24`) or IPv4 addresses for which you want to permit access.

   > **Note:**
   >
   > The first node in your service instance runs an Administration Server and a Managed Server.

7. In the **Destination** field, select the destination node within this service instance. The available source options depend on the topology of your service instance, and may include:
   - **WLS_ADMIN**: The WebLogic Administration Server node
   - **WLS_ADMIN_HOST**: The WebLogic Administration Server node
   - **WLS_MS**: All WebLogic Server nodes
   - **OTD_ADMIN_HOST**: The Oracle Traffic Director (OTD) Administration Server node
   - **OTD_OTD_SERVER**: All Oracle Traffic Director (OTD) Managed Server nodes
8. In the **Destination Port(s)** field, enter the port or range of ports through which the source will be granted access to the destination.
9. In the **Protocol** field, select the TCP or UDP transport for this rule.
10. Click **Create**.

To return to either the Instances page or the Overview page for the selected service instance, click the locator links at the top of the page.

Enable or Disable an Access Rule

This topic does not apply to Oracle Cloud Infrastructure. Identify the Cloud Infrastructure Used by a Service Instance.

You can dynamically enable or disable existing access rules for an Oracle Java Cloud Service instance.

Access rules control the network access to the nodes in your service instance, and to external access from the internet. When a service instance is provisioned, Java Cloud Service defines several default access rules. You can enable or disable these rules to control access to specific port numbers on specific nodes. Make sure you consider the possible security implications before you open ports to external access.

1. Access your service console.
2. Beside the service that you want to modify, click **Manage this instance** ☰, and then select **Manage Access Rules**.
3. On the Access Rules page, beside the rule, click **Actions** ☰, and then select **Enable** or **Disable**.

   You can enable or disable `USER` and `DEFAULT` type rules. You cannot disable `SYSTEM` type rules.
4. When prompted for confirmation, click **Enable** or **Disable**.

To return to either the Instances page or the Overview page for the selected service instance, click the locator links at the top of the page.

Delete an Access Rule

This topic does not apply to Oracle Cloud Infrastructure. Identify the Cloud Infrastructure Used by a Service Instance.

You can delete an access rule for an Oracle Java Cloud Service instance.

Access rules control the network access to the nodes in your service instance, and to external access from the internet. Deleting a rule disables access to specific port numbers on specific nodes.

You can delete only user-created access rules. You cannot delete system-generated access rules.

You cannot modify the configuration of an existing access rule. You must delete the rule and recreate it.

1. Access your service console.
2. Beside the service that you want to modify, click **Manage this instance** ☰, and then select **Manage Access Rules**.
3. On the Access Rules page, beside the rule, click **Actions** ☰, and then select **Delete**.

   You can delete `USER` type rules. You cannot delete `SYSTEM` or `DEFAULT` type rules.

4. When prompted for confirmation, click **Delete**.

To return to either the Instances page or the Overview page for the selected service instance, click the locator links at the top of the page.

Enable Console Access for a Service Instance

This topic does not apply to Oracle Cloud Infrastructure. Identify the Cloud Infrastructure Used by a Service Instance.

You can access a Oracle Java Cloud Service instance through the Weblogic Server Administration Console, the Load Balancer Console or Fusion Middleware Control. By default, access to the administration consoles is disabled for security purposes.

If you provisioned this service instance in an Oracle Cloud Infrastructure region, this procedure is not necessary. Access to the administration consoles is enabled by default in these regions.

1. Access the Oracle Java Cloud Service console.
2. Beside the service that you want to modify, click **Manage this instance** ☰, and then select **Access Rules**.
3. Click **Actions** ☰ beside the `ora_p2admin_ahttps` access rule.
4. Click **Enable**.
5. Click **Actions** ☰ beside the `ora_p2otd_ahttps` access rule.

   This access rule is shown only if you have configured a user-managed load balancer running Oracle Traffic Director (OTD) for this service instance.
6. Click **Enable**.
7. When you have finished using the Access Rules page, click the links at the top of the page to return to the Instances page or the Instance Overview page.

Enable HTTP Access to a Service Instance

If you create an Oracle Java Cloud Service instance by using the web console rather than the REST API or CLI, HTTPS access is enabled by default but HTTP access is disabled. You can enable HTTP access on the load balancer after you have created the service instance.

> **Note:**
>
> This procedure applies only to service instances that include a user-managed load balancer (Oracle Traffic Director). It does not apply to service instances that have an Oracle-managed load balancer (Oracle Cloud Infrastructure Load Balancing or Oracle Cloud Infrastructure Load Balancing Classic).
>
> If there is no load balancer in your service instance, you must instead create an HTTP network channel on all Managed Servers in your Oracle WebLogic Server domain. Refer to these topics in *Administering Server Environments for Oracle WebLogic Server*:
>
> - Configuring Network Resources (12.2.1.3)
> - Configuring Network Resources (11.1.1.7)

By default the load balancer in your service instance listens for HTTP traffic on port 8080. However, the load balancer node automatically redirects incoming traffic on port 80 to port 8080.

Tasks:

- Enable the HTTP Port on Oracle Traffic Director
- Create an Access Rule for the Oracle Traffic Director HTTP Port

Enable the HTTP Port on Oracle Traffic Director

You must enable a port on the load balancer (Oracle Traffic Director) to accept HTTP traffic from the public Internet to your Oracle Java Cloud Service instance.

By default, if you created your service instance in an Oracle Cloud Infrastructure Classic region, external access to the load balancer console is disabled for security purposes. If you did not enable console access while provisioning your service instance, see Enabling Console Access in an Oracle Java Cloud Service Instance.

1. Access the Oracle Java Cloud Service console.
2. Click ☰ for the desired service instance and select **Open Load Balancer Console**.
3. Log in to console using the credentials defined when provisioning your service instance.

   If you created your service instance using the Oracle Java Cloud Service console, the user name and password default to the Oracle WebLogic Server Administration Console user name and password.
4. Access the load balancer configuration (for example, `opc-config`):

   - If your service instance is running Oracle Traffic Director 12*c*, click the ☰ Target Navigation icon. Expand the **Traffic Director** folder and click the name of the Traffic Director configuration.

   - If your service instance is running Oracle Traffic Director 11*g*, click **Configurations** and then click the name of the Traffic Director configuration.

5. Navigate to the Listeners in this configuration:
   - If your service instance is running Oracle Traffic Director 12*c*, click **Traffic Director Configuration** and select **Administration > Listeners**.
   - If your service instance is running Oracle Traffic Director 11*g*, click **Listeners** in the navigation pane.
6. Click **http-listener-1**.
7. Select the **Enabled** checkbox.
8. Activate your changes:
   - If your service instance is running Oracle Traffic Director 12*c*, click **Apply**.
   - If your service instance is running Oracle Traffic Director 11*g*, click **Deploy Changes**.

Create an Access Rule for the Oracle Traffic Director HTTP Port

You must create an access rule to allow public access to the load balancer (Oracle Traffic Director) through the HTTP port.

If you provisioned this service instance in an Oracle Cloud Infrastructure region, you must create security lists instead of access rules. See Security Lists in the Oracle Cloud Infrastructure documentation.

1. Access the Oracle Java Cloud Service console.
2. Click the ☰ Menu icon adjacent to the service instance name and select **Access Rules**.

   The Access Rules page is displayed, showing the list of all access rules.
3. Click **Create Rule**.

   The **Create Access Rule** dialog is displayed.
4. Specify a unique name for the access rule.

   The name must begin with a letter, and can contain numbers, hyphens, or underscores. The length cannot exceed 50 characters. You cannot use prefixes `ora_` or `sys_`.
5. Enter `Permit public http to OTD server` for the description.
6. Select **PUBLIC-INTERNET** for the source.
7. Select **OTD** for the destination.
8. Enter `80` as the port and accept the default protocol (TCP).
9. Click **Create**.
10. Refresh the page periodically. The access rule will appear on the Access Rules table after it is created.

You can now access your application by using the default HTTP port:

```
http://<IP_of_load_balancer>/<context_root>
```

Enable Communication Between Service Instances

This topic does not apply to Oracle Cloud Infrastructure. Identify the Cloud Infrastructure Used by a Service Instance.

The default access rules in an Oracle Java Cloud Service instance only permit communication between Managed Server nodes and the database, and between Managed Server nodes and the load balancer (if enabled). Use custom access rules to enable communication between the Managed Servers of different service instances.

If you provisioned this service instance in an Oracle Cloud Infrastructure region, instead you must use the Oracle Cloud Infrastructure Console to create security lists instead of access rules. See Security Lists in the Oracle Cloud Infrastructure Services documentation.

The architecture of a business application may span multiple tiers, where each application tier is a separate Oracle Java Cloud Service instance. Similarly, certain integration features of Oracle WebLogic Server enable applications to easily communicate across multiple domains, such as Foreign JNDI Providers and Foreign JMS Servers. In these scenarios, you must use access rules to explicitly permit network communication between service instances.

You must identify the host names of the nodes in your *first* service instance. The host names typically use the format `domainName-wls-number`.

For example, if your domain name is `myjcs1` and this domain consists of 3 nodes, the host names would typically be:

- `myjcs1-wls-1`
- `myjcs1-wls-2`
- `myjcs1-wls-3`

You can also identify these host names using the Instance Overview page in the Oracle Java Cloud Service Console. Locate the **Host Name** property of each node.

Before you begin, use a secure shell (SSH) client to connect to the Administration Server node of the *first* service instance.

1. From your SSH session on the Administration Server node, use the `nslookup` command to identify the corresponding private IP address of each host name.

   For example:

   ```
   nslookup myjcs1-wls-2

   Name:    myjcs1-wls-2.compute-myaccount.oraclecloud.internal
   Address: 203.0.113.13
   ```

2. Access the Oracle Java Cloud Service Console.
3. Beside your *second* service instance, click **Manage this instance** ☰, and then select **Access Rules**.
4. On the Access Rules page, click **Create Rule**.
5. Enter a **Rule Name**, such as `myjcs1-to-myjcs2`.

6. For **Source**, select **<custom>**. Enter the private IP addresses for the *first* service instance as a comma-separated list.

   For example: `203.0.113.13,203.0.113.14,203.0.113.15`

   You can also specify multiple IP addresses in CIDR format, such as `203.0.113.1/24`.

7. For **Destination**, select **WLS_MS**
8. For **Destination Port(s)**, enter `8001`.

   > **Note:**
   >
   > If you configured your Managed Servers to listen on additional ports, you can specify them as a comma-separated list such as `8001,9001`.

9. Accept the default **Protocol** (TCP).
10. Click **Create**.

To return to either the Instances page or the Overview page for the selected service instance, click the locator links at the top of the page.

If you restart a node in the *first* service instance, the private IP address of the node might change. In order to keep communication open between the restarted node and the *second* service instance, take one of the following actions:

- (Not available on Oracle Cloud at Customer) If your service instance is attached to an IP network, use the REST API to restart the node and assign the same private IP address. See Stop and Start a Service Instance and Individual VMs in *REST API for Oracle Java Cloud Service*.
- Create a new access rule with the latest private IP address.

Enable Communication Between a Compute Instance and a Service Instance

This topic does not apply to Oracle Cloud Infrastructure. Identify the Cloud Infrastructure Used by a Service Instance.

The default access rules in an Oracle Java Cloud Service instance only permit communication between Managed Server nodes and the database, and between Managed Server nodes and the load balancer (if enabled). Use custom access rules to enable communication between an Oracle Cloud Infrastructure Compute Classic instance and your Managed Servers.

If you provisioned the service instance in an Oracle Cloud Infrastructure region, instead you must use the Oracle Cloud Infrastructure Console to create security lists instead of access rules. See Security Lists in the Oracle Cloud Infrastructure Services documentation.

If the compute instance is not on the shared network, and the compute instance and the Oracle Java Cloud Service instance were created on different IP networks, then you must also connect the two IP networks to the same IP network exchange.

1. Access the Oracle Cloud Infrastructure Compute Classic console.
2. From the Instances page, identify the IP address of the compute instance that will communicate with your Oracle Java Cloud Service instance.
   - If your Oracle Java Cloud Service instance was created with public IP addresses, then use the *public* IP address of the compute instance.
   - If your Oracle Java Cloud Service instance was created without public IP addresses, then use the *private* IP address of the compute instance.
   - If your compute instance and Oracle Java Cloud Service instance are using custom IP networks, then be sure to use the IP address for the appropriate network.
3. Access the Oracle Java Cloud Service Console.
4. Beside your service instance, click **Manage this instance** ☰, and then select **Access Rules**.
5. On the Access Rules page, click **Create Rule**.
6. Enter a **Rule Name**, such as `compute1-to-myjcs1`.
7. For **Source**, select **<custom>**. Enter the IP address of the compute instance.

> **Note:**
>
> You can specify the IP addresses of multiple compute instances as a comma-separated list (`203.0.113.13,203.0.113.14,203.0.113.15`), or using CIDR format (`203.0.113.1/24`).

8. For **Destination**, select **WLS_MS**
9. For **Destination Port(s)**, enter `8001`.

> **Note:**
>
> If you configured your Managed Servers to listen on additional ports, you can specify them as a comma-separated list such as `8001,9001`.

10. Accept the default **Protocol** (TCP).
11. Click **Create**.

To return to either the Instances page or the Overview page for the selected service instance, click the locator links at the top of the page.

Configure SSL for a Service Instance

Secure Socket Layer (SSL) is the most commonly-used method of securing data sent across the internet, and assures visitors that transactions with your application are secure. You can configure SSL between clients and the nodes in your Oracle Java Cloud Service instance in order to ensure that applications are accessed securely.

> **Note:**
>
> To ensure a successful SSL handshake among the Administration Server, Managed Servers and Node Manager, you must configure Node Manager to use the custom keystores and the SSL certificate. See Configure Node Manager to Use the SSL Certificate.

Topics:

- About SSL in Oracle Java Cloud Service
- Configure SSL for Oracle Traffic Director
- Configure SSL for WebLogic Server
- Configure SSL for Oracle Cloud Infrastructure Load Balancing

About SSL in Oracle Java Cloud Service

By default, SSL is already enabled within the software components of a service instance, including Oracle WebLogic Server and the load balancer.

Oracle Traffic Director and Oracle WebLogic Server are configured to use a self-signed SSL certificate that was generated by Oracle Java Cloud Service. Clients will typically receive a message indicating that the signing certificate authority (CA) for this certificate is unknown and not trusted. You can update the load balancers and/or the WebLogic Servers to use a custom SSL certificate, or a certificate that you've obtained from a CA. For production Oracle Java Cloud Service environments, Oracle recommends that you use a CA-issued SSL certificate, which reduces the chances of experiencing a man-in-the-middle attack.

If your service instance includes an Oracle-managed load balancer instead of Oracle Traffic Director, the load balancer is already configured with a CA-issued SSL certificate.

There are multiple CA vendors in the marketplace today, each offering different levels of service at varying price points. Research and choose a CA vendor that meets your service-level and budget requirements. For a CA vendor to issue you a CA-issued SSL certificate, you typically need to provide the following information:

- The host name of the node or a custom domain name.
- Public information associated with the domain confirming you as the owner.
- Email address associated with the domain for verification.

This information is found in a Certificate Signing Request (CSR) file. Your CA vendor uses the CSR to validate the domain and then provides you with a valid SSL certificate, typically via email. For more information about submitting the CSR, refer to your CA vendor documentation.

Configure SSL for Oracle Traffic Director

You can update the Oracle Traffic Director load balancer in Oracle Java Cloud Service to use a generated, self-signed certificate, or a certificate that has been issued by a CA.

Before you begin, ensure that you have enabled Oracle Traffic Director in your service instance, and also registered your custom domain name, as described in Configure a Vanity Domain Name for a Service Instance.

By default, if you created your service instance in an Oracle Cloud Infrastructure Classic region, external access to the load balancer console is disabled for security purposes. If you did not enable console access while provisioning your service instance, see Enabling Console Access in an Oracle Java Cloud Service Instance.

Tasks:

- Create a Self-Signed SSL Certificate in Oracle Traffic Director
- Import a CA-Issued SSL Certificate to Oracle Traffic Director
- Associate the SSL Certificate with Oracle Traffic Director

Create a Self-Signed SSL Certificate in Oracle Traffic Director

If you are not using a CA-issued certificate, then create a self-signed certificate by using the Load Balancer Console.

1. Access the Oracle Java Cloud Service console.
2. Click ☰ for the desired service instance and select **Open Load Balancer Console**.
3. Log in to console using the credentials defined when provisioning your service instance.

   If you created your service instance using the Oracle Java Cloud Service console, the user name and password default to the Oracle WebLogic Server Administration Console user name and password.
4. Access the load balancer configuration (for example, `opc-config`):
   - If your service instance is running Oracle Traffic Director 12*c*, click the ☰ Target Navigation icon. Expand the **Traffic Director** folder and click the name of the Traffic Director configuration.
   - If your service instance is running Oracle Traffic Director 11*g*, click **Configurations** and then click the name of the Traffic Director configuration.
5. If your service instance is running Oracle Traffic Director 12*c*, perform these steps to create a self-signed certificate:
   a. Click **Traffic Director Configuration** and select **Security > Manage Certificates**.
   b. Click **Generate Keypair**.
   c. Enter an **Alias** for the new certificate.
   d. Set the **Common Name** to your custom domain name. For example, `example.com`.
   e. Complete the remaining fields and click **OK**.
6. If your service instance is running Oracle Traffic Director 11*g*, perform these steps to create a self-signed certificate:

   a. Expand **SSL** in the navigation pane and click **Server Certificates**.
   b. Click **New Self Signed Certificate**.
   c. Set the **Server Name** to your custom domain name. For example, `example.com`.
   d. Complete the remaining fields and click **Next**.
   e. On the Certificate Options page, enter a **Nickname** (alias) for the certificate. Click **Next**.
   f. Click **Create Certificate**.

Import a CA-Issued SSL Certificate to Oracle Traffic Director

Use the Load Balancer Console to create a Certificate Signing Request (CSR). After receiving the CA-issued certificate, import it into the load balancer configuration.

1. Access the Oracle Java Cloud Service console.
2. Click ☰ for the desired service instance and select **Open Load Balancer Console**.
3. Log in to console using the credentials defined when provisioning your service instance.

   If you created your service instance using the Oracle Java Cloud Service console, the user name and password default to the Oracle WebLogic Server Administration Console user name and password.
4. Access the load balancer configuration (for example, `opc-config`):
   - If your service instance is running Oracle Traffic Director 12*c*, click the ☰ Target Navigation icon. Expand the **Traffic Director** folder and click the name of the Traffic Director configuration.
   - If your service instance is running Oracle Traffic Director 11*g*, click **Configurations** and then click the name of the Traffic Director configuration.
5. If your service instance is running Oracle Traffic Director 12*c*, perform these steps to generate a CSR:
   a. Click **Traffic Director Configuration** and select **Security > Manage Certificates**.
   b. Click **Generate Keypair**.
   c. Enter an **Alias** for the new certificate.
   d. Set the **Common Name** to your custom domain name. For example, `example.com`.
   e. Complete the remaining fields and click **OK**.
   f. Select your new certificate and click **Generate CSR**.
6. If your service instance is running Oracle Traffic Director 11*g*, perform these steps to generate a CSR:
   a. Expand **SSL** in the navigation pane and click **Server Certificates**.
   b. Click **Create Certificate Request**.

   c. Set the **Server Name** to your custom domain name. For example, `example.com`.
   d. Complete the remaining fields and click **Next**.
   e. On the Certificate Options page, click **Next** to accept the defaults.
   f. Click **Create CSR**.

7. Save the generated CSR text, including the header line `-----BEGIN NEW CERTIFICATE REQUEST-----` and footer line `-----END NEW CERTIFICATE REQUEST-----`.
8. Submit the CSR to your CA vendor to request a new CA-issued SSL certificate.
9. Return to the Load Balancer Console for your service instance.
10. If your service instance is running Oracle Traffic Director 12*c*, perform these steps to import the CA-issued certificate:
    a. Click **Traffic Director Configuration** and select **Security > Manage Certificates**.
    b. Click **Import**.
    c. Verify that **Certificate Type** is set to Certificate.
    d. Select the **Alias** of the certificate you generated earlier.
    e. You can paste the certificate text directly in the **Paste Certificate String Here** field, or click **Choose File** and select the certificate on your local file system. If you opt to paste the certificate text, be sure to include the headers `BEGIN CERTIFICATE` and `END CERTIFICATE`, including the beginning and ending hyphens.
    f. Click **OK**.
11. If your service instance is running Oracle Traffic Director 11*g*, perform these steps to import the CA-issued certificate:
    a. Expand **SSL** in the navigation pane and click **Server Certificates**.
    b. Click **Install Certificate**.
    c. Enter a **Nickname** (alias) for the certificate.
    d. You can paste the certificate text directly in the **Certificate Data** field, or provide the path to the certificate file in the **Certificate File** field. If you opt to paste the certificate text, be sure to include the headers `BEGIN CERTIFICATE` and `END CERTIFICATE`, including the beginning and ending hyphens.
    e. Click **Next**.
    f. Click **Install Certificate**.

For more information about managing load balancer certificates, see:

- Managing Certificates in *Administering Oracle Traffic Director* (12.2.1)
- Managing Certificates in *Oracle Traffic Director Administrator's Guide* (11.1.1.7)

Associate the SSL Certificate with Oracle Traffic Director

After installing a CA-issued or self-signed SSL certificate to the load balancer, you must associate it with the HTTPS listeners in the load balancer’s configuration. After the association is made, the load balancer will present the SSL certificate while processing any new HTTPS requests.

1. Access the Oracle Java Cloud Service console.

2. Click ☰ for the desired service instance and select **Open Load Balancer Console**.
3. Log in to console using the credentials defined when provisioning your service instance.

   If you created your service instance using the Oracle Java Cloud Service console, the user name and password default to the Oracle WebLogic Server Administration Console user name and password.
4. Access the load balancer configuration (for example, `opc-config`):
   - If your service instance is running Oracle Traffic Director 12*c*, click the ☰ Target Navigation icon. Expand the **Traffic Director** folder and click the name of the Traffic Director configuration.
   - If your service instance is running Oracle Traffic Director 11*g*, click **Configurations** and then click the name of the Traffic Director configuration.
5. Navigate to the Listeners in this configuration:
   - If your service instance is running Oracle Traffic Director 12*c*, click **Traffic Director Configuration** and select **Administration > Listeners**.
   - If your service instance is running Oracle Traffic Director 11*g*, click **Listeners** in the navigation pane.
6. Click **https-listener-1**.
7. In the **SSL/TLS Settings** section select your new certificate in the **RSA Certificate** field.
8. Activate your changes:
   - If your service instance is running Oracle Traffic Director 12*c*, click **OK**.
   - If your service instance is running Oracle Traffic Director 11*g*, click **Deploy Changes**.
9. Repeat from step 3 to update the certificate of any additional HTTPS listeners in this configuration.

   Alternatively, you can configure **SSL/TLS Settings** for an entire Virtual Server in the load balancer configuration.
10. Restart the load balancer node(s) in your service instance for the change to take effect.

    a. Return to the Oracle Java Cloud Service console.

    b. Beside the load balancer node, click **Manage this node** ☰, and then select **Restart**.

    c. When prompted for confirmation, click **OK**.

For more information about the SSL settings of the load balancer, see:

- Configuring SSL/TLS Between Oracle Traffic Director and Clients in *Administering Oracle Traffic Director* (12.2.1)
- Configuring SSL/TLS Between Oracle Traffic Director and Clients in *Oracle Traffic Director Administrator's Guide* (11.1.1.7)

Configure SSL for WebLogic Server

You can update the Oracle WebLogic Server domain in Oracle Java Cloud Service to use a generated, self-signed certificate, or a certificate that has been issued by a Certifying Authority (CA).

If your service instance does not include a load balancer, and you want to use a different SSL certificate for communication between clients and your Java applications, update the configuration for the Managed Servers in your domain.

After scaling out your service instance, you will also need to update the SSL configuration for the new server.

> **Note:**
>
> Oracle recommends that you back up your service instance prior to updating the SSL configuration. If the SSL configuration fails, you will be able to restore the service instance to a known working state.

By default, if you created your service instance in an Oracle Cloud Infrastructure Classic region, external access to the WebLogic Server administration console is disabled for security purposes. If you did not enable console access while provisioning your service instance, see Enabling Console Access in an Oracle Java Cloud Service Instance.

Tasks:

- Create Keystores and Certificates for WebLogic Server
- Add the Oracle Identity Cloud Service Certificate to the Trust Keystore
- Associate Keystores and SSL Certificate with WebLogic Server
- Configure Node Manager to Use the SSL Certificate (Important: To ensure a successful SSL handshake)
- Configure SSL for New Servers After Scaling Out

Create Keystores and Certificates for WebLogic Server

Use keytool to create your own public/private key pairs and self-signed certificates. Optionally, create a Certificate Signing Request (CSR) for each generated certificate and submit it to a CA to obtain a trusted certificate.

1. Connect to the Administration Server node in your service instance with a secure shell (SSH) client, and then switch to the `oracle` user.

   ```
   sudo su - oracle
   ```

2. Create a directory `/u01/data/keystores` to hold the keystore files.

   ```
   cd /u01/data
   mkdir keystores
   cd keystores
   ```

> **Caution:**
>
> Do not place your keystore and certificate files in the Middleware Home (`MIDDLEWARE_HOME`) or Java Home (`JAVA_HOME`) directories. Any modifications you make to these locations might be lost when you apply a patch.

> **Caution:**
>
> Do not place your keystore and certificate files in the Domain Home (`DOMAIN_HOME`) or `/u01/data/domains` directories because they are included in backups. A restore operation might include an expired certificate and result in errors during a server restart.

3. Use the `keytool` command to create a new identity keystore file, and to add a self-signed certificate to the keystore named `server_cert`.

   ```
   keytool -genkeypair -alias alias -keyalg keyalg -sigalg sigalg -
   keysize size -dname dn -keystore keystore_file
   ```

   For example:

   ```
   keytool -genkeypair -alias server_cert -keyalg RSA -sigalg
   SHA256withRSA -keysize 2048 -dname
   "CN=example.com,OU=Support,O=Example,L=Reading,ST=Berkshire,C=GB" -
   keystore identity.jks
   ```

   Note that The X.500 Distinguished Name, which consists of the WebLogic Server host and DNS domain name, is *example.com*.

4. When prompted, enter a password for the keystore.
5. When prompted, enter a password for the private key, `server_cert`, or press **Enter** to use the same password as the keystore.
6. If you are using a self-signed certificate to configure SSL, then create a custom trust keystore file.

   a. Use `keytool` to export the self-signed certificate, `server_cert`, from the identity store to a file named `server_cert.cer`.

      ```
      keytool -exportcert -alias server_cert -file server_cert.cer -
      keystore keystore_file
      ```

      When prompted, enter the password for the keystore.

   b. Use `keytool` to create a trust keystore file, and to import `server_cert.cer` into this new keystore. Use the same alias, `server_cert`.

      ```
      keytool -importcert -alias server_cert -file server_cert.cer -
      keystore trust_keystore_file
      ```

For example:

```
keytool -importcert -alias server_cert -file server_cert.cer -
keystore trust.jks
```

c. When prompted, enter a password for the new keystore.

d. When prompted to trust this certificate, enter **yes**.

7. If you are using a CA-issued certificate to configure SSL, then create a CSR file from the identity keystore.

   a. Use `keytool` to create a CSR file for the `server_cert` private key.

   ```
   keytool -certreq -alias alias -file certreq_file -keystore keystore
   ```

   For example:

   ```
   keytool -certreq -alias server_cert -file server_cert.csr -keystore
   identity.jks
   ```

   b. When prompted, enter the password for the keystore and the private key.

   c. Submit the CSR to a Certificate Authority of your choice in order to obtain a trusted certificate.

   d. Import the CA-issued certificate into the identity keystore.

8. Copy the keystore files to all the other nodes in your service instance.

   For example:

   ```
   ssh myinstance-wls-2
   mkdir /u01/data/keystores
   scp myinstance-wls-1:/u01/data/keystores/identity.jks /u01/data/keystores
   scp myinstance-wls-1:/u01/data/keystores/trust.jks /u01/data/keystores
   ```

Add the Oracle Identity Cloud Service Certificate to the Trust Keystore

If your Oracle Java Cloud Service instance is configured to use Oracle Identity Cloud Service for authentication, you must add the Oracle Identity Cloud Service certificate to your custom trust keystore.

1. Access the Oracle Java Cloud Service console.
2. Click **Manage this service** ☰ for your service instance, and then select **Open Fusion Middleware Control Console**.
3. Click **WebLogic Domain**, select **Security**, and then select **Keystore**.
4. Expand the **system** folder.
5. Click **trust**, and then click **Manage**.
6. Click **idcs_root_ca**, and then click **Export**.
7. Click **Export Certificate**, and then click **Close**.

8. SSH to the Administration Server node and switch to the `oracle` user.

   ```
   sudo su - oracle
   ```

9. Navigate to the `/u01/data/keystores` folder.
10. Create a new file named `idcs_root_ca.cer`. Paste the contents of the exported `idcs_root_ca` certificate into this file.
11. Use `keytool` to import `idcs_root_ca.cer` into your custom trust keystore.

    ```
    keytool -import -alias idcs_root_ca -file idcs_root_ca.cer -
    keystore trust_keystore_file
    ```

    For example:

    ```
    keytool -import -alias idcs_root_ca -file idcs_root_ca.cer -
    keystore trust.jks
    ```

12. When prompted, enter the password for the keystore.
13. When prompted to trust this certificate, enter **yes**.
14. Copy the updated trust keystore file to all the other nodes in your service instance.

    For example:

    ```
    ssh myinstance-wls-2
    scp myinstance-wls-1:/u01/data/keystores/trust.jks /u01/data/
    keystores
    ```

Associate Keystores and SSL Certificate with WebLogic Server

Use the WebLogic Server Administration Console to update the location of each server’s identity and trust keystore files, and the name of the certificate in the identity keystore that the server uses for SSL communication.

By default, the servers in an Oracle Java Cloud Service instance are configured to use a demo identity keystore and a demo trust keystore. Oracle recommends that you use these demo keystores for development purposes only.

1. Access the Oracle Java Cloud Service console.
2. Click the name of your service instance.
3. From the Overview page, identify the host names of all the nodes in your service instance, and the names of all servers in your domain.
4. Click **Manage this service** ☰, and select **Open WebLogic Server Administration Console**.
5. Log in to the console using the credentials that you specified when provisioning your service instance.
6. Within the **Change Center** panel, click **Lock and Edit**.
7. Within the **Domain Structure** panel, expand **Environment**, and then click **Servers**.

8. Click the name of the server for which you want to configure SSL.
9. Verify that the **Configuration** tab is selected. Under **Configuration**, click the **Keystores** tab.
   1. For **Keystores**, click **Change**. Select **Custom Identity and Custom Trust**, and then click **Save**.
   2. For **Custom Identity Keystore**, enter the full path to your identity keystore.

      For example, `/u01/data/keystores/identity.jks`
   3. For **Custom Identity Keystore Type**, enter `JKS`.
   4. For **Custom Identity Keystore Passphrase**, enter your keystore password. Enter the same value for **Confirm Custom Identity Keystore Passphrase**.
   5. For **Custom Trust Keystore**, enter the full path to your trust keystore.

      For example, `/u01/data/keystores/trust.jks`
   6. For **Custom Trust Keystore Type**, enter `JKS`.
   7. For **Custom Trust Keystore Passphrase**, enter your keystore password. Enter the same value for **Confirm Custom Trust Keystore Passphrase**.
   8. Click **Save**.
10. Under **Configuration**, click the **SSL** tab.
    1. For **Private Key Alias**, enter the name of the certificate (private key) in the identity keystore, `server_cert`.
    2. For **Private Key Passphrase**, enter the password for this certificate in the keystore. Enter the same value for **Confirm Private Key Passphrase**.

       By default, the password for the certificate is the same as the identity keystore password.
    3. Click **Save**.
11. Under **Change Center**, click **Activate Changes**.
12. Click the **Control** tab.
13. Click **Restart SSL**. When prompted for confirmation, click **Yes**.
14. Repeat from **step 6** to update each server in your domain for which you want to configure SSL.

    After you have configured SSL for the WebLogic Server to use the keystore `CustomIdentityAndCustomTrust`, go to the `boot.properties` file located in `DOMAIN_HOME/servers/AdminServer/security` and `DOMAIN_HOME/servers/<server_name>/data/nodemanager` and remove the line

    `TrustKeyStore=DemoTrust`.

For more information, refer to *Overview of Configuring SSL* in Administering Security for Oracle WebLogic Server (12.2.1).

Configure Node Manager to Use the SSL Certificate

To ensure a successful SSL handshake among the Administration Server, Managed Servers and Node Manager, you must configure Node Manager to use the custom keystores and the SSL certificate.

1. Connect to the Administration Server node with a secure shell (SSH) client, and then switch to the `oracle` user.

   ```
   sudo su - oracle
   ```

2. Edit the `nodemanager.properties` file located under your Domain Home directory.

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

3. Add the following lines to the end of the file.

   ```
   KeyStores=CustomIdentityAndCustomTrust
   CustomIdentityKeystoreType=jks
   CustomIdentityKeyStoreFileName=path_to_identity_keystore
   CustomIdentityKeyStorePassPhrase=keystore_password
   CustomIdentityPrivateKeyPassPhrase=server_cert_password
   CustomIdentityAlias=server_cert
   CustomTrustKeystoreType=jks
   CustomTrustKeyStoreFileName=path_to_trust_keystore
   CustomTrustKeyStorePassPhrase=keystore_password
   ```

   For example:

   ```
   KeyStores=CustomIdentityAndCustomTrust
   CustomIdentityKeystoreType=jks
   CustomIdentityKeyStoreFileName=/u01/data/keystores/identity.jks
   CustomIdentityKeyStorePassPhrase=keystore_password
   CustomIdentityPrivateKeyPassPhrase=server_cert_password
   CustomIdentityAlias=server_cert
   CustomTrustKeystoreType=jks
   CustomTrustKeyStoreFileName=/u01/data/keystores/trust.jks
   CustomTrustKeyStorePassPhrase=keystore_password
   ```

4. Regenerate the Node Manager startup files.

   a. Launch the WebLogic Scripting Tool (WLST).

      ```
      $MIDDLEWARE_HOME/oracle_common/common/bin/wlst.sh
      ```

   b. Connect to the Administration Server.

      ```
      connect('admin_user','password','t3://admin_server_host:9071')
      ```

      For example:

      ```
      connect('weblogic','password','t3://myinstance-wls-1:9071')
      ```

   c. Generate the `boot.properties` and `startup.properties` files for the server(s) on this node.

      ```
      nmGenBootStartupProps('server_name')
      ```

Both the Administration Server and the first Managed Server run on the first node in your service instance. For example:

```
nmGenBootStartupProps('myinstance_adminserver')
nmGenBootStartupProps('myinstance_server_1')
```

d. Exit WLST.

```
exit()
```

5. Edit the `setDomainEnv.sh` file located under your Domain Home directory.

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

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

```
export WLST_PROPERTIES="${WLST_PROPERTIES} -
Dweblogic.ssl.JSSEEnabled=true -Dweblogic.security.SSL.enableJSSE=true -
Dweblogic.security.SSL.ignoreHostnameVerification=true -
Dweblogic.security.TrustKeyStore=CustomTrust -
Dweblogic.security.CustomTrustKeyStoreFileName=path_to_trust_keystore -
Dweblogic.security.CustomTrustKeyStoreType=JKS"
```

For example:

```
export WLST_PROPERTIES="${WLST_PROPERTIES} -
Dweblogic.ssl.JSSEEnabled=true -Dweblogic.security.SSL.enableJSSE=true -
Dweblogic.security.SSL.ignoreHostnameVerification=true -
Dweblogic.security.TrustKeyStore=CustomTrust -
Dweblogic.security.CustomTrustKeyStoreFileName=/u01/data/keystores/
trust.jks -Dweblogic.security.CustomTrustKeyStoreType=JKS"
```

7. Run `source $DOMAIN_HOME/bin/setDomainEnv.sh`.

8. Navigate to the `$DOMAIN_HOME/bin` directory.

9. If your service instance is running WebLogic Server 12c, then run the following commands to restart the Node Manager.

```
./stopNodeManager.sh
./startNodeManager.sh &
```

10. If your service instance is running WebLogic Server 11g, then perform the following steps to restart the Node Manager.

a. Identity the process ID for the Node Manager.

```
ps -ef | grep weblogic.NodeManager
```

b. Kill the Node Manager process.

```
kill -9 process_id
```

c. Run the following commands.

```
export JAVA_OPTIONS="${JAVA_OPTIONS} -
Dweblogic.ssl.JSSEEnabled=true -
Dweblogic.security.SSL.enableJSSE=true"
$MIDDLEWARE_HOME/wlserver_10.3/server/bin/startNodeManager.sh &
```

11. Restart the servers using the Node Manager.

a. Launch the WebLogic Scripting Tool (WLST).

```
$MIDDLEWARE_HOME/oracle_common/common/bin/wlst.sh
```

b. Connect to the Node Manager.

```
nmConnect(username="weblogic", password="password",
domainName="domain_name", domainDir="/u01/data/domains/
domain_name", nmType="ssl", host="host_name", port="5556",
verbose="false")
```

For example:

```
nmConnect(username="weblogic", password="password",
domainName="myinstance_domain", domainDir="/u01/data/domains/
myinstance_domain", nmType="ssl", host="myinstance-wls-1",
port="5556", verbose="false")
```

c. Restart the server(s) on this node.

```
nmKill('server_name')
nmStart('server_name')
nmServerStatus('server_name')
```

Both the Administration Server and the first Managed Server run on the first node in your service instance. For example:

```
nmKill('myinstance_server_1')
nmKill('myinstance_adminserver')
nmStart('myinstance_adminserver')
nmServerStatus('myinstance_adminserver')
nmStart('myinstance_server_1')
nmServerStatus('myinstance_server_1')
```

d. Exit WLST.

```
exit()
```

12. Repeat from **step 1** for any other nodes in your service instance for which you want to configure SSL.

Configure SSL for New Servers After Scaling Out

After scaling out a cluster in your Oracle Java Cloud Service instance, you must modify the new server's SSL configuration if you want the server to use your custom keystores.

Use the WebLogic Server Administration Console to update the new server. See Associate Keystores and SSL Certificate with WebLogic Server.

Oracle Java Cloud Service automatically performs the following tasks during a scale-out operation:

- Copy the custom keystore files to the new node.
- Copy the Node Manager configuration files to the new node.
- Update the `setDomainEnv.sh` file on the new node.

Configure SSL for Oracle Cloud Infrastructure Load Balancing

If you need the ability to customize the SSL parameters for the Oracle-managed load balancer that you're using with Oracle Java Cloud Service in Oracle Cloud Infrastructure, then you must create and configure an instance of Oracle Cloud Infrastructure Load Balancing manually.

You can't update the default configuration for an Oracle-managed load balancer that is provisioned automatically during the creation of the Oracle Java Cloud Service instance. See Create and Configure an Instance of Oracle Cloud Infrastructure Load Balancing.

If your Oracle Java Cloud Service instance has Oracle Identity Cloud Service enabled, you can remove the Oracle-managed load balancer and use your own instance of Oracle Cloud Infrastructure Load Balancing. See Use an Oracle Cloud Infrastructure Load Balancer.

See also Managing SSL Certificates in the Oracle Cloud Infrastructure documentation.

Authenticate Users

Oracle Java Cloud Service is comprised of multiple components, each with its own identity stores, authentication options and administrative tools.

Topics

- About Users
- About Authentication
- Manage Passwords
- Relocate Oracle Java Cloud Service to a Different Identity Domain

About Users

There are multiple types of users associated with Oracle Java Cloud Service. Each has its own purpose and is found in a specific identity store.

Topics

- Cloud Users and Service Administrators

- WebLogic Server Administrators
- Application Users
- Database Users
- Load Balancer Administrators
- Operating System Users

Cloud Users and Service Administrators

When an Oracle Cloud account is created that includes a subscription to Oracle Java Cloud Service, the default administrator is given the `Java Administrator` role.

Only Oracle Cloud users with this role can create and manage Oracle Java Cloud Service instances with either the console, CLI or REST API. Users in your account who have the `Identity Domain Administrator` role can create additional cloud users and grant them the `Java Administrator` role. Similar roles exist for the other services available in Oracle Cloud. See Add Users, Assign Policies and Roles in *Getting Started with Oracle Cloud*.

Oracle Identity Cloud Service provides a secure, centralized cloud service to manage the relationships that your users have with your applications, including with other Oracle Cloud services like Oracle Java Cloud Service. With Oracle Identity Cloud Service you can create custom password policies and email notifications, onboard new users, assign users and groups to applications, and run security reports. See Use Oracle Identity Cloud Service with Oracle Java Cloud Service.

Oracle Java Cloud Service can optionally store backups of service instances in cloud storage (either Oracle Cloud Infrastructure Object Storage or Oracle Cloud Infrastructure Object Storage Classic). Configuring a service instance for backups includes specifying the credentials for an Oracle Cloud user who has read/write access to cloud storage. See About Backup and Restoration in Oracle Java Cloud Service.

WebLogic Server Administrators

An Oracle Java Cloud Service instance includes an Oracle WebLogic Server domain, which is comprised of an Administration Server and one or more Managed Servers.

A domain also defines a security realm that controls authentication, authorization, role mapping, credential mapping and security auditing across all of the servers in the domain. When you create a service instance you provide the credentials for the initial user in this WebLogic security realm. This user has the `Admin` role and can perform all WebLogic Server administrative operations through either the WebLogic Server Administration Console, Fusion Middleware Control, WebLogic Scripting Tool (WLST) or WebLogic REST API. You can also use the default WebLogic administrator to create additional WebLogic administrators and assign them specific roles and privileges. For example, users with the `Deployer` role can deploy Java applications to the domain.

By default, the domain in an Oracle Java Cloud Service instance is configured to use the embedded LDAP identity store for WebLogic Server roles, users and policies. This embedded LDAP is hosted in the Administration Server and is replicated to all Managed Servers in the domain. If the default security configuration does not meet your requirements, you can modify the default security realm or create a new one with any combination of WebLogic and custom security providers.

If your cloud account includes Oracle Identity Cloud Service, Oracle Java Cloud Service can provision your service instance so that WebLogic Server uses Oracle Identity Cloud Service for authentication in addition to the default embedded LDAP. As a result, when WebLogic administrators access tools like the Administration Console they are authenticated against the users, groups, roles and policies defined in Oracle Identity Cloud Service. See Use Oracle Identity Cloud Service with Oracle Java Cloud Service.

To learn more about WebLogic security see:

- *Understanding Security for Oracle WebLogic Server* (12.2.1.3)
- *Understanding Security for Oracle WebLogic Server* (11.1.1.7)

Application Users

Java applications deployed to the WebLogic Server domain in your Oracle Java Cloud Service instance can have security policies that protect the applications against unauthorized access.

WebLogic Server supports various security providers that assign an identity to the requesting user or software entity. For example, WebLogic Server can determine the identity of an application user by validating a user name and password. By default, the domain in an Oracle Java Cloud Service instance is configured to use the embedded LDAP identity store for both WebLogic administrators and application users. You can use standard WebLogic tools like the WebLogic Server Administration Console to manage users, groups, roles and policies in the embedded LDAP.

If your cloud account includes Oracle Identity Cloud Service, Oracle Java Cloud Service can provision your service instance so that WebLogic Server uses Oracle Identity Cloud Service for authentication in addition to the default embedded LDAP. As a result, users that access your Java applications are authenticated against the users, groups, roles and policies defined in Oracle Identity Cloud Service. See Use Oracle Identity Cloud Service with Oracle Java Cloud Service.

If this security configuration does not meet your requirements, you can modify the default security realm or create a new one with any combination of WebLogic and custom security providers. For large production applications, Oracle recommends that you use a proper identity management system such as Oracle Identity Cloud Service instead of the embedded LDAP.

Database Users

An Oracle Java Cloud Service instance requires access to at least one Oracle Database.

Oracle Java Cloud Service provisions your chosen database with the Oracle Fusion Middleware (FMW) schema and also connects the WebLogic Server domain in your service instance to this database. When you create a service instance you provide appropriate credentials to access and update this FMW database.

You can also connect your service instance to additional relational databases by using standard WebLogic tools like the WebLogic Server Administration Console. Just as with the FMW database, you must provide the necessary credentials to connect to these application databases.

If your database is running Oracle Database 12c, users can be scoped to the container database (CDB) or a pluggable database (PDB). To connect to a specific PDB from WebLogic Server, be sure to specify user credentials in the target PDB and not the CDB.

To learn more about database connectivity in WebLogic Server see:

- *Administering JDBC Data Sources for Oracle WebLogic Server* (12.2.1.3)
- *Configuring and Managing JDBC Data Sources for Oracle WebLogic Server* (11.1.1.7)

A component of your WebLogic Server domain is Oracle Platform Security Services (OPSS), which requires a connection to your service instance's FMW database. The credentials for this database connection are stored in a separate file named `jps-config.xml`.

Load Balancer Administrators

Your Oracle Java Cloud Service instance can optionally include a user-managed load balancer, an Oracle-managed load balancer, or your own instance of Oracle Cloud Infrastructure Load Balancing. The load balancer distributes application traffic to the servers in the WebLogic Server domain.

A user-managed load balancer is Oracle Traffic Director, which has an Administration/Managed server architecture similar to WebLogic Server, along with its own identity store. When you create a service instance, the same WebLogic Server administrator credentials that you provide are also used as the default Traffic Director credentials. This user has full administrative access to the Load Balancer Console and other Traffic Director tools. You can also use the Load Balancer Console to create additional Traffic Director administrators.

An Oracle-managed load balancer runs on Oracle Cloud Infrastructure Load Balancing or Oracle Cloud Infrastructure Load Balancing Classic, depending on the region where the service instance was created. Cloud users must be granted access to these services in order to view or modify the generated configuration for a load balancer.

See Configure a Load Balancer for a Service Instance.

Operating System Users

Each Oracle Java Cloud Service instance is associated with at least one Secure Shell (SSH) public key. Using the matching private key, you can SSH to the underlying nodes running WebLogic Server and the load balancer.

SSH to a node as the `opc` OS user and then switch to the `oracle` OS user in order to manage Oracle Java Cloud Service software like WebLogic Server, or to install additional Oracle software. The `opc` user has root privileges to the OS if you need to modify the OS configuration, create additional OS users, or install additional OS packages. See Access a Node with a Secure Shell (SSH).

About Authentication

Get an overview of the different ways in which you can determine the identity of a user or system that is accessing an application running in Oracle Java Cloud Service. Clients can authenticate against an external LDAP or database, or their identities can be validated with different token technologies like SAML.

By default, cloud users and application users are managed by different security frameworks and are located in different identity stores. Consequently, these users support different authentication options.

Single Sign-On (SSO) is the ability for a user to authenticate once and then gain access to many different application components, even though these components may have their own authentication schemes. SSO enables users to login securely to all their applications, web sites and mainframe sessions with just one identity.

Topics

- Cloud Authentication
- WebLogic Server Authentication

Cloud Authentication

In order to create and manage cloud services such as Oracle Java Cloud Service instances, service administrators in Oracle Cloud are authenticated against a specific identity domain and with a username and password.

If your Oracle Cloud account includes Oracle Identity Cloud Service, then service administrators are authenticated against its identity store. See Add Users, Assign Policies and Roles in *Getting Started with Oracle Cloud*.

WebLogic Server Authentication

An Oracle WebLogic Server domain defines a security realm that controls authentication, authorization, role mapping, credential mapping and security auditing across all of the servers in the domain.

These services are implemented as *security providers*. WebLogic Server includes many types of built-in providers and you can also build your own. Authentication providers in particular establish trust for a user by validating credentials or tokens. They can also identify any groups to which the user belongs, in order to make access decisions.

You can also configure multiple authentication providers in a single security realm. For example, consider a scenario in which the WebLogic Server administration users are located in one LDAP server while application users are found in a different LDAP server.

This table describes some of the authentication options available in a WebLogic Server security realm.

Authentication Option	Description
Embedded LDAP (default)	Each user’s credentials and group memberships are maintained in a Lightweight Directory Access Protocol (LDAP) server that is hosted in the domain’s Administration Server and replicated to all Managed Servers in the domain. Oracle does not recommend using the embedded LDAP for large production applications. See: • Managing the Embedded LDAP Server in *Administering Security for Oracle WebLogic Server (12.2.1.3)* • Managing the Embedded LDAP Server in *Securing Oracle WebLogic Server (11.1.1.7)*

Authentication Option	Description
Oracle Identity Cloud Service	If your cloud account includes Oracle Identity Cloud Service, Oracle Java Cloud Service can provision your service instance so that WebLogic Server is configured to use Oracle Identity Cloud Service for authentication. As a result, when users access your Java applications or tools like the Administration Console they are authenticated against the users, groups, roles and policies defined in Oracle Identity Cloud Service. See Use Oracle Identity Cloud Service with Oracle Java Cloud Service.
External LDAP	WebLogic Server includes authentication providers that are compatible with Oracle Internet Directory, Microsoft Active Directory, iPlanet, Open LDAP or any other LDAP-compliant server. These providers differ primarily in how they are configured by default to match typical directory schemas for their corresponding LDAP server. If this LDAP server is hosted outside of the nodes in your Oracle Java Cloud Service instance, you may need to enable network communication between your nodes and the LDAP server. See Create an Access Rule. See: • Configuring LDAP Authentication Providers in *Administering Security for Oracle WebLogic Server (12.2.1.3)* • Configuring LDAP Authentication Providers in *Securing Oracle WebLogic Server (11.1.1.7)*
Relational Database	WebLogic Server includes authentication providers that use a relational database as a data store for users, passwords and groups. These providers are configured by default with a typical SQL database schema to support these entities, but you can also customize this default configuration to match your database's existing schema. In order to use the database authentication providers you must create a data source in the domain to establish connectivity to the database. If you selected this database when you created your Oracle Java Cloud Service instance, a data source already exists. If this database is hosted outside of the nodes in your Oracle Java Cloud Service instance, you may need to enable network communication between your nodes and the database. See Create an Access Rule. See: • Configuring RDBMS Authentication Providers in *Administering Security for Oracle WebLogic Server (12.2.1.3)* • Configuring RDBMS Authentication Providers in *Securing Oracle WebLogic Server (11.1.1.7)* Tutorial
SAML	In perimeter authentication, a system outside of WebLogic Server establishes trust through tokens. WebLogic Server can generate and consume Security Assertion Markup Language (SAML) tokens (assertions), and supports both SAML 1.1 and SAML 2.0. See: • Configuring Identity Assertion Providers and Configuring Single Sign-On with Web Browsers and HTTP Clients Using SAML in *Administering Security for Oracle WebLogic Server (12.2.1.3)* • Configuring Identity Assertion Providers and Configuring Single Sign-On with Web Browsers and HTTP Clients Using SAML in *Securing Oracle WebLogic Server (11.1.1.7)*

Manage Passwords

You may need to update the various credentials used to run an Oracle Java Cloud Service instance in order to meet Oracle security policies, corporate security policies or government regulations, or in response to a perceived security threat.

The specific tools and procedures you use to modify passwords depends on the type of user and where it is stored in the environment. In addition, there are consequences to changing certain system users because other resources in the environment use these credentials as well.

For general information about users in Oracle Java Cloud Service, see About Users.

Topics

- Cloud User Password
- WebLogic Server Administrator Password
- WebLogic Node Manager Password
- Database Password
- Oracle Traffic Director Password
- Application User Password

Cloud User Password

Learn about updating the password for your Java Administrator and related cloud users.

To update your Oracle Cloud password, see Changing and Managing Your Own Passwords in *Getting Started with Oracle Cloud*.

If you are an Identity Domain Administrator, you can reset other users’ passwords. See Resetting Another User’s Password in *Getting Started with Oracle Cloud*.

When you create an Oracle Java Cloud Service instance you provide the location of an object storage container along with credentials to access and update backup files in this storage container. If you change the password for this cloud user, you will also need to update the backup configuration of your service instance. Otherwise, both automated and manual backups will fail. See Configure Scheduled Backups for an Oracle Java Cloud Service Instance.

WebLogic Server Administrator Password

By default your Oracle WebLogic Server domain is configured to use the embedded LDAP security provider as the identity store for users, passwords and groups. This includes the WebLogic Server administrator user whose credentials you initialize when you create the Oracle Java Cloud Service instance.

You can use any available WebLogic Server tools to modify user credentials in the embedded LDAP, including the Administration Console, WLST and REST API. To use the Administration Console, see *Modify Users* in one of these publications:

- Administration Console Online Help (12.2.1.3)
- Administration Console Online Help (11.1.1.7)

You can optionally create a service instance that uses Oracle Identity Cloud Service for authentication. As a result, you can access the WebLogic Server Administration Console and other WebLogic tools for your service instance as Oracle Cloud users. See Use Oracle Identity Cloud Service with Oracle Java Cloud Service.

Administrative credentials are required in order to boot the servers in your domain. A boot identity file is a text file that contains encrypted user credentials for starting and stopping an instance of WebLogic Server. If you change the password for this user, you must also update any boot identity files that use the same credentials. These files are located on the node’s file system. Replace the current encrypted password with your new password. Otherwise, servers may fail to boot if you attempt to restart them.

See *Boot Identity Files* in one of these publications:

- Administering Server Startup and Shutdown for Oracle WebLogic Server (12.2.1.3)
- Managing Server Startup and Shutdown for Oracle WebLogic Server (11.1.1.7)

For information on using SSH to access Oracle Java Cloud Service nodes, see Access a Node with a Secure Shell (SSH).

WebLogic Node Manager Password

In WebLogic Server, the Node Manager process is used to remotely start and stop servers. When you create or scale out an Oracle Java Cloud Service instance, all Node Managers are configured with a generated user name and password.

These credentials are used to authenticate connections between a client (for example, the Administration Server or Oracle Java Cloud Service) and the Node Manager.

For Oracle Java Cloud Service instances, you cannot modify the Node Manager password by manually editing the `nm_password.properties` file on a node. This will cause lifecycle and other administrative operations to fail. Instead, you must use the Oracle Java Cloud Service REST API. See Change the Node Manager Credentials in *REST API for Oracle Java Cloud Service*.

Database Password

The Oracle WebLogic Server domain in an Oracle Java Cloud Service instance is automatically configured with several JDBC data sources. Each data source connects to a database in Oracle Cloud. You specify the database name and credentials for these data sources when you create the service instance.

The **Infrastructure Schema Database** in a service instance is provisioned with the required Oracle Fusion Middleware schema. To change the password for this database schema and also update the WebLogic domain configuration, see Change the Database Schema Password for an Oracle Java Cloud Service Instance.

When you create a service instance, you can also associate it with one or more **Application Schema Databases**. If you change the password for one of these databases, the corresponding data source in the WebLogic domain will fail to connect to the database. Use one of the standard WebLogic administrative interfaces to modify the connection properties of the existing data source. See *Configuring JDBC Data Sources* in one of the following publications:

- Administering JDBC Data Sources for Oracle WebLogic Server (12.2.1.3)

- Configuring and Managing JDBC Data Sources for Oracle WebLogic Server (10.3.6)

For more information about data sources in Oracle Java Cloud Service, see About Data Sources.

Oracle Traffic Director Password

If you add a user-managed load balancer to your Oracle Java Cloud Service instance when you initially create it, the load balancer is configured with the same credentials as the WebLogic Server administrator.

If you add a user-managed balancer to an existing service instance, you have the option to provide different credentials. In either case you can use the Load Balancer Console to change this user’s password.

- For service instances running Oracle Traffic Director 12c, see Configure WebLogic Server Users in *Administering Oracle WebLogic Server with Fusion Middleware Control*. Be sure to access the console for the load balancer, and not for the WebLogic Server domain.
- For service instances running Oracle Traffic Director 11g, see Securing Access to the Administration Server in *Oracle Traffic Director Administrator’s Guide*.

Application User Password

By default your Oracle WebLogic Server domain is configured to use the embedded LDAP security provider as the identity store for users, passwords and groups. This includes any custom application users you’ve defined.

You can use any available WebLogic Server tools to modify user credentials in the embedded LDAP, including the Administration Console, WLST and REST API. To use the Administration Console, see *Modify Users* in one of these publications:

- Administration Console Online Help (12.2.1.3)
- Administration Console Online Help (11.1.1.7)

Alternatively, you can customize your WebLogic domain to use other security providers for users and passwords, such as a database, an LDAP server, or Oracle Identity Cloud Service. In general, you do not use WebLogic Server to directly modify user credentials in these external identity stores. Instead use the native administrative tools offered by these resources. For more information about security providers, see About Authentication.

Relocate Oracle Java Cloud Service to a Different Identity Domain

This topic does not apply to Oracle Cloud at Customer.

An Oracle Cloud account administrator has the ability to move your Oracle Java Cloud Service entitlement to another identity domain in the same account.

When you activate an order in Oracle Cloud, services in the order are typically activated in a default identity domain within the account. If necessary you can relocate Oracle Java Cloud Service from one identity domain to another. However, you must delete any existing service instances prior to relocating the service.

See Relocating a Service Entitlement to Another Identity Domain in *Managing and Monitoring Oracle Cloud*.

During the relocation process, the service administrator will be added to the target identity domain but other Oracle Cloud users and administrators will not. The identity domain administrator will need to create any other users and administrators in the target identity domain, and to assign them the appropriate roles. If applicable, the bulk user import and role assignment features can be used for this task. See Managing Users and Roles in *Getting Started with Oracle Cloud*.

12 Use Oracle Coherence in Oracle Java Cloud Service

Enable Oracle Coherence in Oracle Java Cloud Service to provision an in-memory data grid and caching infrastructure for your Java Enterprise Edition applications.

Topics:

- Overview of Coherence Tasks for Oracle Java Cloud Service
- About Oracle Coherence in Oracle Java Cloud Service
- About Cache Capacity for a Service Instance
- Add a Coherence Data Grid
- Scale Out a Coherence Data Grid
- Scale In a Coherence Data Grid
- Delete a Coherence Data Grid

Overview of Coherence Tasks for Oracle Java Cloud Service

Use Oracle Java Cloud Service to create, manage and scale an Oracle Coherence in-memory data grid.

> **Note:**
>
> An Oracle Java Cloud Service instance that has been provisioned with Oracle Coherence is also referred to as an Oracle Java Cloud Service—Coherence instance.

Task	Description	More Information
Create an Oracle Java Cloud Service—Coherence instance	When using Oracle Java Cloud Service to create a service instance, enable and configure a Coherence data tier.	Configure the Coherence Data Tier
Add a Coherence data tier to an existing service instance	If you already provisioned a service instance without configuring a Coherence data tier, you can add a data grid cluster by using the REST API.	Add a Coherence Data Grid
Scale the Coherence data tier	You can increase or decrease the total cache capacity in preparation for an increased or reduced load on a service instance.	Scale Out a Coherence Data Grid Scale In a Coherence Data Grid

Task	Description	More Information
Delete a Coherence data tier	Remove the entire Coherence data tier from an existing service instance by using the REST API.	Delete a Coherence Data Grid

About Oracle Coherence in Oracle Java Cloud Service

Use Oracle Coherence in your Oracle Java Cloud Service instances in order to provide your applications with an in-memory data grid and caching solution.

> **Note:**
>
> An Oracle Java Cloud Service instance that has been provisioned with Oracle Coherence is also referred to as an Oracle Java Cloud Service—Coherence instance.

Oracle Coherence is a fault-tolerant, in-memory data management solution that enables Java EE applications to predictably scale by providing fast, reliable, and scalable access to frequently used data. Coherence applications are packaged and deployed to Oracle WebLogic Server 12*c* as Grid Archive (GAR) files, in much the same way as other Java EE modules.

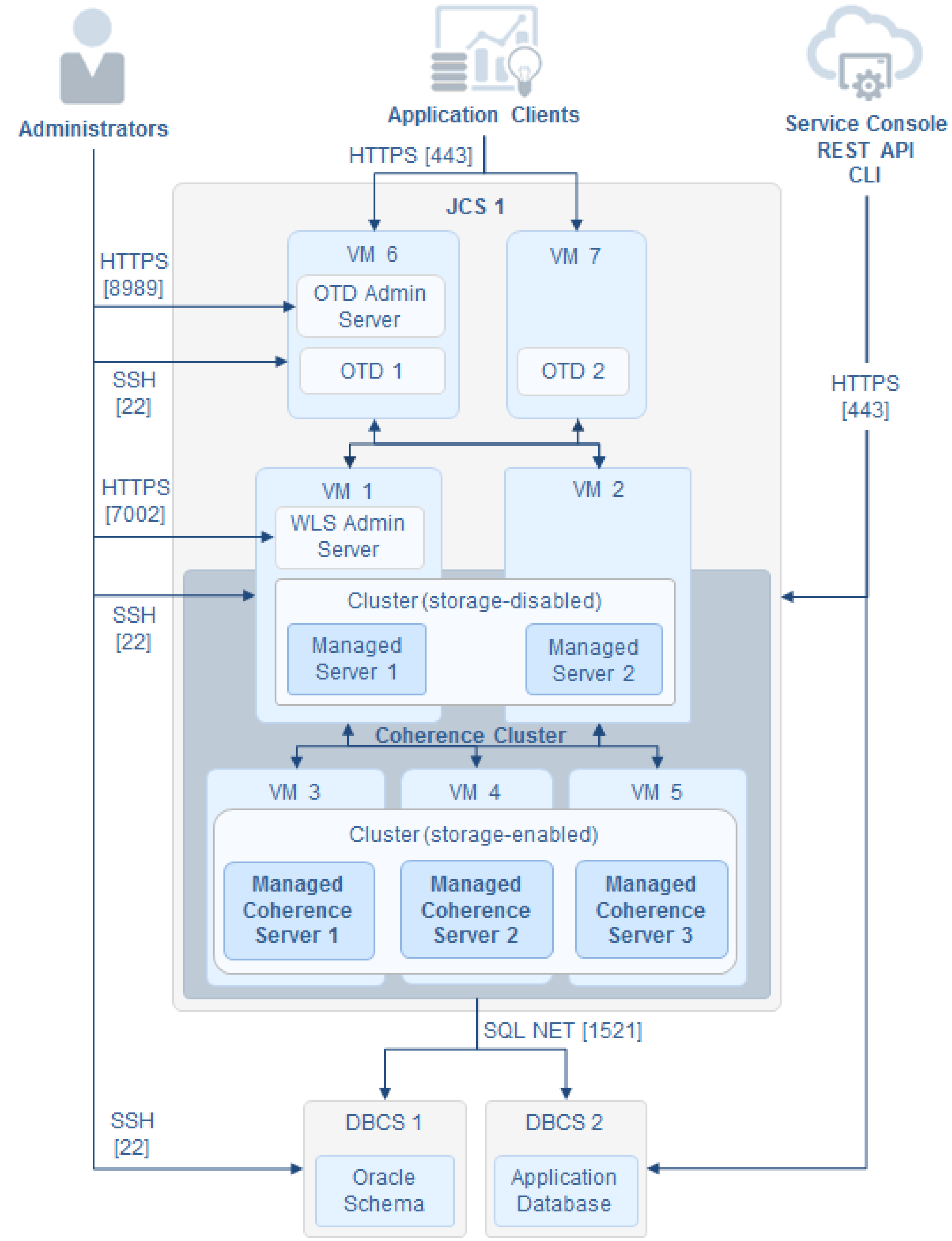
A Coherence data grid is only available to Oracle Java Cloud Service instances that are running **High Performance Edition**. The setup and administration tasks for this data grid also depend on the software release on which the service instance is running:

- Oracle WebLogic Server 12*c* (12.2.1.2 or later) - Use Oracle Java Cloud Service to automatically provision, scale and manage your Coherence data grid. Oracle WebLogic Server 12c (12.2.1.2) is supported on Oracle Cloud at Customer only.
- Oracle WebLogic Server 11*g* - You must manually configure and start Coherence processes on your service instance after creating it, by accessing the nodes that comprise your service instance.

When you use Oracle Java Cloud Service to provision a data grid in a 12*c* service instance, the following infrastructure is added to the service instance:

- A second WebLogic Server cluster is provisioned and configured as a Coherence data grid. This cluster consists of one or more storage-enabled Managed Servers, and is provisioned on one or more nodes. The data tier cluster is created in the same WebLogic Server domain as the application tier cluster that's used for running your Java EE applications. By default, the data tier cluster name is generated from the first eight characters of the service instance name using the following format: `*first8charsOfServiceInstanceName*_DGCluster`.
- A Coherence Cluster is configured in the WebLogic Server domain. The default name is `DataGridConfig`.
- Both the storage-enabled Coherence data tier cluster and the storage-disabled application tier cluster are associated with the Coherence Cluster. These two clusters are scaled independently of one another.

The following illustration shows a typical deployment topology for an Oracle Java Cloud Service—Coherence instance. The example uses a cluster of two Managed Servers for the application tier (storage-disabled), and a cluster of three Managed Servers for the Coherence data tier (storage-enabled):



To learn more about Coherence, see:

- Developing Oracle Coherence Applications for Oracle WebLogic Server (12.2.1.3)

- Coherence Getting Started Guide (10.3.6)

About Cache Capacity for a Service Instance

There are multiple parameters that affect the caching capacity of an Oracle Coherence data tier within an Oracle Java Cloud Service instance.

When you use Oracle Java Cloud Service to provision a service instance with a Coherence data grid cluster, you control the initial cache capacity with these configuration settings:

- **Compute Shape** - The number of OCPUs and the amount of memory for each node in the Oracle WebLogic Server cluster. For example, the shape `oc3` has 1 OCPU and 7.5 GB of memory. All nodes in the data grid cluster have the same compute shape.
- **Cluster Size** - The number of Managed Servers in the data grid cluster. Each Managed Server is a Java Virtual Machine (JVM) running Coherence.
- **Managed Servers Per Node** - The number of Managed Servers that run on each node in the data grid cluster. All nodes in the data grid cluster have the same number of Managed Servers.

When you create an Oracle Java Cloud Service—Coherence instance, the number of nodes that are provisioned for the data grid cluster is determined by the following formula:

```
Nodes = Cluster Size / Managed Servers Per Node
```

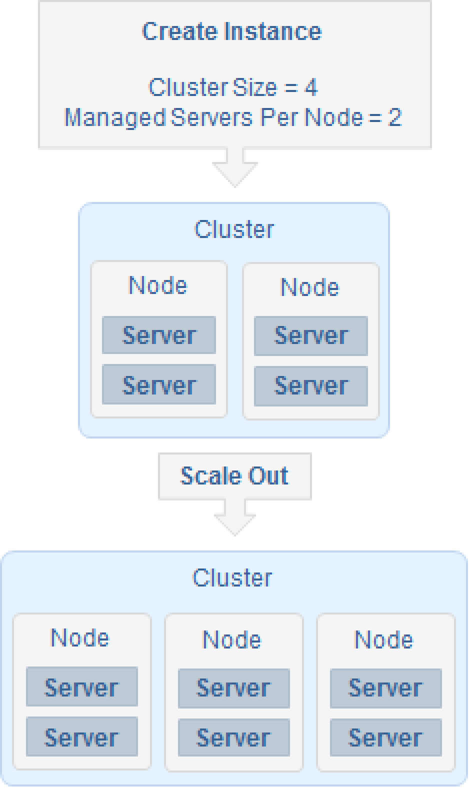
For example, if you set **Cluster Size** to 4 and **Managed Servers Per Node** to 2, Oracle Java Cloud Service will create 2 nodes, each running 2 servers. If the quotient is not a whole number, then it is rounded up to the nearest whole number. For example, if **Cluster Size** is 4 and **Managed Servers Per Node** is 3, your data grid cluster will consist of 2 nodes, each running 3 servers (a total of 6 servers).

Running multiple Coherence servers on each node can improve concurrency and memory management, but it generally also requires more processors and memory (larger shapes).

> **Note:**
>
> You cannot change these data tier configuration parameters (cluster size, compute shape, managed servers per node) after creating a service instance.
>
> If you require maximum availability for the data in the data grid cluster, it must contain **at least three nodes**.

As your application workload increases and your Coherence data tier requires more capacity, you can use Oracle Java Cloud Service to scale out the data grid cluster. Each time you perform a scale-out operation, Oracle Java Cloud Service adds a single node to the data grid cluster. This node runs the same number of servers as the other existing nodes in the cluster. Similarly, a scale-in operation removes a single node from the data grid cluster. The application tier cluster and the data grid cluster in your service instance can be scaled independently of one another.

Consider the previous example in which a service instance's Coherence data tier is configured with a **Cluster Size** of 4 and **Managed Servers Per Node** is set to 2. Scaling out a data grid with this configuration adds one node with 2 servers, as illustrated in this figure:



Each server on a Coherence node is a JVM process that is configured with a default heap size. The memory available for all Coherence servers on a node is 75% of the remaining memory after reserving 1.5 GB for the operating system. The available memory is then divided evenly amongst the servers on the node.

For example, consider a data tier whose compute shape has 7.5 GB of memory per node. The memory available for server heap is: `0.75 x (7 - 1.5) = 4.5 GB`. If the data tier is configured for two servers per node, the heap size for each server is: `4.5 / 2 = 2.25 GB`.

As a general rule for most Coherence applications, approximately one third (1/3) of a server's heap is used for primary cache storage. The other two thirds of the heap is used for backup storage and scratch space. Therefore, if the total heap across all nodes in your data grid is 100 GB, the total cache size for your applications is approximately 33 GB. An exception to this rule is a data grid that's comprised of a single node with a single server. In this scenario there is no high availability, and therefore backup storage does not consume any of the available heap.

Add a Coherence Data Grid

Use the REST API to add an Oracle Coherence data tier to an existing Oracle Java Cloud Service instance.

If you already provisioned a service instance without configuring a Coherence data tier, you can add a data grid cluster by using the REST API. This functionality is not available in the Oracle Java Cloud Service console.

You can add a Coherence data tier to a service instance if it meets the following requirements:

- The **Software Edition** is **High Performance Edition**.
- The service instance does not already include a Coherence data grid cluster.

See Scale Out a Service Instance in *REST API for Oracle Java Cloud Service*.

Scale Out a Coherence Data Grid

Scale out the Oracle Coherence data tier in an Oracle Java Cloud Service instance to increase its total cache capacity.

Tutorial

As your application workload increases and your Coherence data tier requires more capacity, you can use Oracle Java Cloud Service to scale out the data grid cluster. Each time you perform a scale-out operation, Oracle Java Cloud Service adds a single node to the data grid cluster. This node runs the same number of Managed Servers as the other existing nodes in the data grid cluster.

For example, if you created a service instance and set its Coherence **Cluster Size** to 4 and **Managed Servers Per Node** to 2, the initial data grid cluster in this service instance contains 2 nodes, each running 2 servers. Scaling out a data grid cluster with this configuration adds one node with 2 servers, for a total of 3 nodes and 6 servers.

The Coherence data tier cluster within a service instance is scaled independently of its application tier cluster.

You cannot scale a service instance if it is under maintenance, such as during a patching or backup operation.

If backups are configured for the service instance, Oracle Java Cloud Service will attempt to create a backup before scaling the instance.

1. Access the Oracle Java Cloud Service console.
2. Click the name of the service instance to which you want to add a Coherence node.

   The Overview page is displayed.
3. Click the **Menu** ☰ for the Data Grid Cluster, and then select **Scale Out**.
4. When prompted for confirmation, click **Scale Out**.
5. Click **Refresh** ↻ until a new node is added to the Data Grid Cluster section of the Overview page.

The status of the cluster indicates that the scaling operation is in progress.

6. Periodically click **Refresh** until the scaling operation is complete.

   You cannot perform any other management operations on the service instance while the scaling operation is in progress.

   You can also monitor the progress of the scaling operation from the **Activity** page.

Scale In a Coherence Data Grid

Scale in the Oracle Coherence data tier in an Oracle Java Cloud Service instance to decrease its total cache capacity.

Tutorial

Each scale-in operation removes a single node from the data grid cluster in the service instance. If you remove the last node in the data grid cluster, your applications will lose all cached data. Alternatively, you can delete the entire data grid cluster.

The Coherence data tier cluster within a service instance is scaled independently of its application tier cluster.

You cannot scale a service instance if it is under maintenance, such as during a patching or backup operation.

If backups are configured for the service instance, Oracle Java Cloud Service will attempt to create a backup before scaling the instance.

1. Access the Oracle Java Cloud Service console.
2. Click the name of the service instance from which you want to remove a Coherence node.

   The Overview page is displayed.
3. Click the **Menu** for the Data Grid Cluster, and then select **Remove Node**.
4. When prompted for confirmation, click **Remove Node**.
5. Click **Refresh** until the node is removed from the Data Grid Cluster section of the Overview page.

   You cannot perform any other management operations on the service instance while the scaling operation is in progress.

   You can also monitor the progress of the scaling operation from the **Activity** page.

Delete a Coherence Data Grid

Use the REST API to remove the entire Oracle Coherence data grid from an existing Oracle Java Cloud Service instance.

Only resources and entities related to Coherence are deleted from the service instance. A scale-in operation is added to the Activity page. This functionality is not available in the Oracle Java Cloud Service console. See Scale In a Service Instance in *REST API for Oracle Java Cloud Service*.

13 Administer the Load Balancer for an Oracle Java Cloud Service Instance

Configure, manage, and monitor the load balancer for an Oracle Java Cloud Service instance.

Topics:

- About the Load Balancer in Oracle Java Cloud Service
- Overview of Load Balancer Administration Tasks
- Disable or Enable the Load Balancer for an Oracle Java Cloud Service Instance (does not apply to Oracle Cloud Infrastructure Load Balancing)
- Add a Load Balancer to a Service Instance (Oracle Traffic Director only)
- Add a Second Load Balancer Node to a Service Instance (Oracle Traffic Director only)
- Remove a Load Balancer Node from a Service Instance (Oracle Traffic Director only)
- Configure a Load Balancer for a Service Instance
- Set Up an Oracle Cloud Infrastructure Load Balancer (for instances created without a load balancer and without Oracle Identity Cloud Service enabled)
- Use an Oracle Cloud Infrastructure Load Balancer (for instances created with an instance of Oracle Cloud Infrastructure Load Balancing with or without Oracle Identity Cloud Service)
- About the Storage Volumes Attached to the Load Balancer Nodes

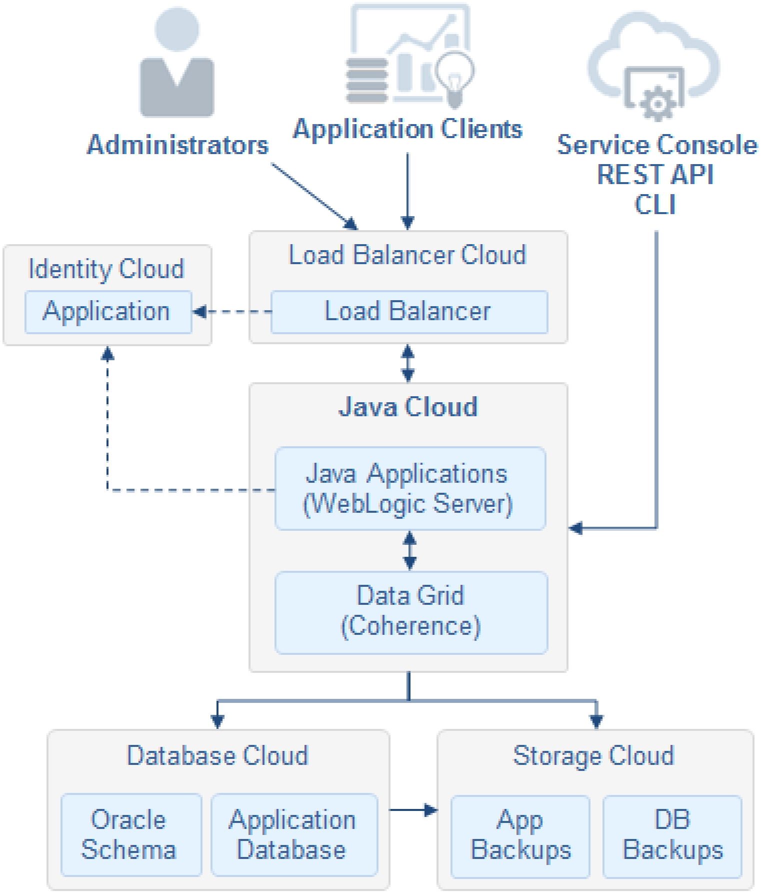
About the Load Balancer in Oracle Java Cloud Service

You can set up a load balancer for your Oracle Java Cloud Service instance, to intercept client requests to the applications deployed on the instance and to distribute the requests to the WebLogic managed servers.

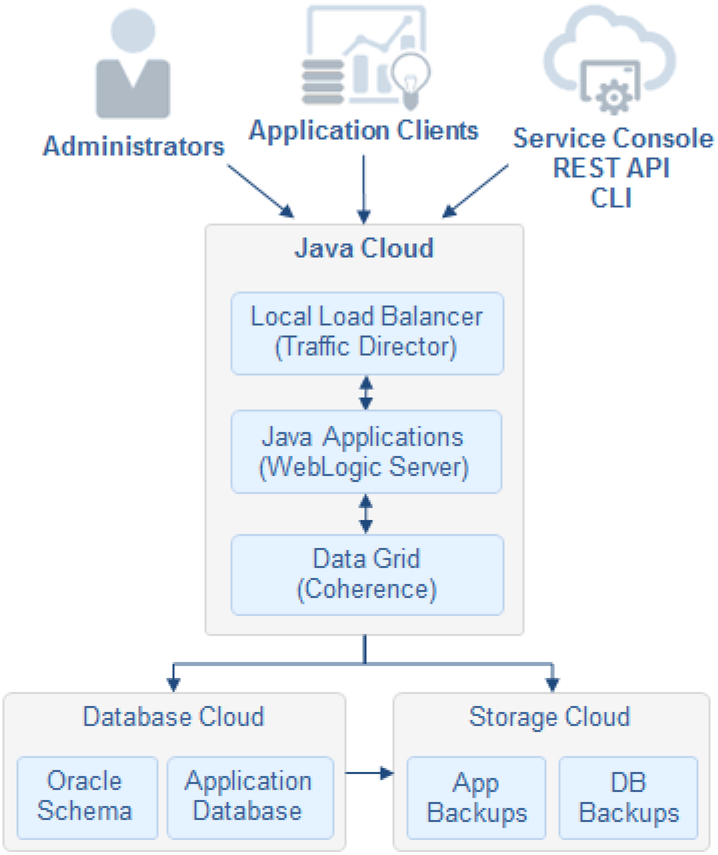
Using a load balancer is recommended if your Oracle Java Cloud Service instance contains more than one WebLogic managed server. You can set up the load balancer to handle encrypting and decrypting TLS traffic, relieving the WebLogic managed servers of the overhead for processing HTTPS requests. You can also suspend access to the applications deployed on your Oracle Java Cloud Service instance, by disabling the load balancer or changing the offline state of the load balancer.

Oracle Java Cloud Service supports two load balancer options:

- **Oracle-managed**: Multiple nodes running in Oracle Cloud Infrastructure Load Balancing or Oracle Cloud Infrastructure Load Balancing Classic, depending on the region where the service instance is created.



- **User-managed (Oracle Traffic Director)**: One or two load-balancer nodes running within a service instance



The following are the differences between the load-balancer options:

Feature	Oracle-managed	User-managed (Oracle Traffic Director)
Creating the load balancer	On Oracle Cloud Infrastructure regions, you can provision the load balancer while creating a service instance, and you can specify the load-balancing policy. You cannot use the Oracle Java Cloud Service Console to add a load balancer to an existing service instance. If you manually create and configure an instance of Oracle Cloud Infrastructure Load Balancing to an existing service instance, this load balancer is not considered an Oracle-managed load balancer. On Oracle Cloud Infrastructure Classic regions, in order to provision a load balancer, you must also configure the service instance to use Oracle Identity Cloud Service as the identity provider.	You can provision the load balancer while creating a service instance, or add the load balancer to an existing service instance. You can specify the load-balancing policy, the number of load-balancer nodes (one or two), and the compute shape for the nodes.
Supported load-balancing policies	• Round robin • Least connections • IP hash	• Round robin • Least connections • Least response time

Feature	Oracle-managed	User-managed (Oracle Traffic Director)
High availability (HA)	Oracle Java Cloud Service provisions a load balancer with two nodes that can be accessed using a single IP address.	You can create a service instance with two load balancer nodes, or add a second node to an existing service instance. Each node is accessed using a separate IP address. **Note**: Oracle Java Cloud Service *does not* fail-over application requests between load-balancer nodes. If one of the load-balancer nodes is unavailable, you are responsible for ensuring that requests fail over to another node. The failover-group feature of Oracle Traffic Director is not supported.
Network topology	When you create a service instance in an Oracle Cloud Infrastructure region, you can assign a regional subnet that is shared by both load balancer nodes. You can assign different non-regional subnets to the Oracle WebLogic Server nodes and the Oracle Cloud Infrastructure Load Balancing nodes. The ability to select non-regional subnets in different availability domains is not supported for Oracle WebLogic Server nodes, only for Oracle Cloud Infrastructure Load Balancing nodes. If the selected Oracle Cloud Infrastructure region has only one availability domain, you can only specify one subnet for the load balancer, which is assigned to both load balancer nodes. You can also create a service instance in which the Oracle WebLogic Server nodes are assigned to a private subnet while the load balancer nodes are assigned to a public subnet. See Create an Oracle Java Cloud Service Instance Attached to a Private Subnet on Oracle Cloud Infrastructure.	All the Oracle WebLogic Server and Oracle Traffic Director nodes in a service instance are assigned to the same IP network or subnet.

Feature	Oracle-managed	User-managed (Oracle Traffic Director)
Configuring the load balancer	You can do the following for instances in Oracle Cloud Infrastructure Classic regions: • Disable and re-enable the load balancer at any time from the Oracle Java Cloud Service interfaces. • Configure the load-balancer settings by using the Oracle Cloud Infrastructure Load Balancing Classic interface. For instances in Oracle Cloud Infrastructure regions, you cannot update the Oracle Cloud Infrastructure Load Balancing configuration if the load balancer was provisioned automatically during service instance creation.	You can do the following: • Disable and re-enable the load balancer at any time from the Oracle Java Cloud Service interfaces. • Add a second load-balancer node, and remove it when it is not required. • Connect to the nodes using a secure shell (SSH). • Configure the load-balancer settings by using the Oracle Traffic Director administration console. **Note**: See the restrictions described in Administration Best Practices.
Patching the nodes	Oracle manages the patching of the Oracle-managed load balancer.	You must patch the Oracle Traffic Director nodes.
Accessing the load balancer	The load balancer is assigned a public IP address, which is associated with a default domain name. You can use Oracle Java Cloud Service to add custom vanity URLs to the load balancer for a service instance.	Each load balancer node is accessed using a public IP address. You can manually update the Oracle Traffic Director configuration to use a custom domain name.
More information	• Overview of Load Balancing in the Oracle Cloud Infrastructure documentation • About the Components of Oracle Cloud Infrastructure Load Balancing Classic in *Using Oracle Cloud Infrastructure Load Balancing Classic* • Use Oracle Identity Cloud Service with Oracle Java Cloud Service	Features of Oracle Traffic Director in *Administering Oracle Traffic Director*

Overview of Load Balancer Administration Tasks

Administer the load balancer for an Oracle Java Cloud Service instance, to control how application requests are sent to the managed servers in your WebLogic Server clusters.

Task	More Information
Set up Oracle Cloud Infrastructure Load Balancing for an instance that was provisioned without any load balancer and without Oracle Identity Cloud Service enabled.	Set Up an Oracle Cloud Infrastructure Load Balancer

Task	More Information
Remove the Oracle-managed instance of Oracle Cloud Infrastructure Load Balancing from an instance in order to use your own load balancer created in Oracle Cloud Infrastructure Load Balancing.	Use an Oracle Cloud Infrastructure Load Balancer
Add Oracle Traffic Director to an instance.	Add a Load Balancer to a Service Instance
Add a second Oracle Traffic Director node.	Add a Second Load Balancer Node to a Service Instance
Remove an Oracle Traffic Director node.	Remove a Load Balancer Node from a Service Instance
Scale an Oracle Traffic Director node up or down.	About Scaling an Oracle Java Cloud Service Node
Disable or enable Oracle Traffic Director or Oracle Cloud Infrastructure Load Balancing Classic.	Disable or Enable the Load Balancer for an Oracle Java Cloud Service Instance
Configure the load balancer parameters.	Configure a Load Balancer for a Service Instance
Configure a custom "vanity" domain name (such as `example.com`).	Configure a Vanity Domain Name for a Service Instance
Configure a custom "vanity" URL (such as `/apps/mystore`) in Oracle Traffic Director.	Configure a Custom URL for an Application Deployed to a Service Instance
Configure SSL between the client and the load balancer.	Configure SSL for a Service Instance

Disable or Enable the Load Balancer for an Oracle Java Cloud Service Instance

You can disable the load balancer for an Oracle Java Cloud Service instance to block access to the service instance during maintenance. You can then enable the load balancer again to resume access.

When enabled, the load balancer forwards the requests it receives from clients to the Oracle WebLogic Server managed servers in your service instance. When it is disabled, the load balancer stops forwarding requests, and responds with a maintenance message and the HTTP status code 503.

Note:

- You cannot enable or disable the load balancer for a service instance while the instance is being backed up.
- This procedure is for enabling and disabling Oracle Traffic Director or Oracle Cloud Infrastructure Load Balancing Classic load balancer. It does not apply to Oracle Cloud Infrastructure Load Balancing.

Topics:

- Disable and Enable Oracle Traffic Director
- Disable and Enable an Oracle-Managed Load Balancer

Disable and Enable Oracle Traffic Director

Disable and enable the user-managed load balancer in an Oracle Java Cloud Service instance.

1. Navigate to the Overview page for the instance for which you want to enable or disable the load balancer.
2. Click **Manage this instance** ☰ in the instance name bar at the top of the page.
3. Click **Disable Load Balancer** or **Enable Load Balancer**, as required.
4. Click **Yes, Disable Load Balancer** or **Yes, Enable Load Balancer**.

The instance is in maintenance mode until the operation is completed. After the operation is completed, the **State** field in the **Oracle Load Balancer** section changes to **Traffic Disabled** or **Traffic Enabled**, as appropriate.

Disable and Enable an Oracle-Managed Load Balancer

This topic does not apply to Oracle Cloud Infrastructure. Identify the Cloud Infrastructure Used by a Service Instance.

Disable and enable the Oracle-managed load balancer in an Oracle Java Cloud Service instance.

1. Navigate to the Overview page for the instance for which you want to enable or disable the load balancer.
2. Locate and expand the Load Balancer section of the page.

   The load balancer endpoint is displayed.
3. From the **Actions** ☰ menu, select **Enable** or **Disable**, as required.

   Within the Load Balancer section, if you click **Expand** ▸ at the left edge of the row, the web console shows the details of the listener configured for the load balancer. Instead of disabling the load balancer endpoint, you can choose to disable just the listener. But the effect of either choice is the same; that is, client requests to the load balancer are not forwarded to the WebLogic Servers.
4. At the confirmation prompt, click **OK**.

The instance is in maintenance mode until the operation is completed. After the operation is completed, the ⚠ icon is displayed if you disabled the load balancer, and the ✓ icon is displayed if you enabled it.

Add a Load Balancer to a Service Instance

You can add a load balancer to an existing Oracle Java Cloud Service instance.

> **Note:**
>
> This procedure applies only to adding a user-managed load balancer (Oracle Traffic Director) that's hosted on nodes within your service instance. You can't use Oracle Java Cloud Service to add an Oracle-managed load balancer to an existing service instance that was provisioned without any load balancer. Instead you must manually provision and configure the load balancer by using either Oracle Cloud Infrastructure Load Balancing or Oracle Cloud Infrastructure Load Balancing Classic, depending on the region where the service instance was created. Refer to these topics:
>
> - Getting Started with Load Balancing in the Oracle Cloud Infrastructure documentation
> - Use an Oracle Cloud Infrastructure Load Balancer (with Oracle Identity Cloud Service)
> - Create and Configure an Instance of Oracle Cloud Infrastructure Load Balancing (without Oracle Identity Cloud Service)
> - Typical Workflow for Creating a Load Balancer in *Using Oracle Cloud Infrastructure Load Balancing Classic*

To add a user-managed load balancer (Oracle Traffic Director) to a service instance:

1. Navigate to the Overview page for the instance to which you want to add the load balancer.
2. Click **Manage this instance** ☰ in the instance name bar at the top.
3. Select **Add Load Balancer**.
4. In the Add Load Balancer dialog box, define the routing policy and compute shape for the load balancer, and the user name and password for the Oracle Traffic Director administrator.

   The user name and password are used to access the Load Balancer Console as described in Accessing the Administrative Consoles Used by Oracle Java Cloud Service.

   > **Note:**
   >
   > If you add a load balancer to an Oracle Java Cloud Service instance after the service instance was created, you must define the user name and password for the Oracle Traffic Director administrator explicitly. The user name and password are **not** set by default to the user name of the WebLogic Server administrator. This behavior differs from the behavior when a load balancer is added to a service instance while the service instance is being created.

Option	Description
Load Balancer Policy	Select the policy to use for routing requests to the load balancer. Valid policies include: • **Least Connection Count**—Passes each new request to the Managed Server with the least number of connections. This policy helps prevent a Managed Server from getting overloaded. Managed Servers with greater processing power to handle requests will receive more connections over time. • **Least Response Time**—Passes each new request to the Managed Server with the fastest response time. This policy is useful when Managed Servers are distributed across networks. • **Round Robin**—Passes each new request to the next Managed Server in line, evenly distributing requests across all Managed Servers regardless of the number of connections or response time.
Compute Shape	Select the compute shape to use for all the load balancer nodes in the service instance. The compute shape is the number of Oracle Compute Units (OCPUs) and amount of memory (RAM) that you want to allocate to these nodes. The list of available shapes varies depending on whether you selected an Oracle Cloud Infrastructure Classic or Oracle Cloud Infrastructure region. You are billed for Oracle Traffic Director nodes at the same price that you are billed for Oracle WebLogic Server nodes in your Oracle Java Cloud Service subscription.
Reserved IPs	If this service instance is in a region, you can assign each load balancer node a public IP address that you had previously reserved. See Creating an IP Reservation.
Add Another Active OTD Node	Choose whether to add a second Oracle Traffic Director (OTD) node to this service instance. **Note:** A configuration with two active load balancer nodes can be used to provide high availability and higher load-balancing capacity. But Oracle Java Cloud Service *does not* fail-over application requests between load-balancer nodes. The failover-group feature of Oracle Traffic Director is not supported. If one of the load-balancer nodes is unavailable, you are responsible for ensuring that requests fail over to another node.
User Name	Enter a user name for the Oracle Traffic Director administrator.
Admin Password	Define the password for the Oracle Traffic Director administrator.
Confirm Admin Password	Re-enter the password for the Oracle Traffic Director administrator.

5. Click **Add**.

The Overview page is updated to show that the load balancer is being added. Click the ⟳ (Refresh) icon to check the latest status.

While the load balancer is being added, the service instance is in maintenance status and you cannot start any other management operation on the service instance.

If you require the WebLogic Plug-in Enabled control to be set in Oracle WebLogic Server, you must set this control manually. If you add a load balancer to an Oracle Java Cloud Service instance after the service instance was created, Oracle Java Cloud Service does **not** set the WebLogic Plug-in Enabled control in Oracle WebLogic Server for you. This behavior differs from the behavior when a load balancer is added to a service instance while the service instance is being created. See *Understanding the use of “WebLogic Plugin Enabled”*.

Add a Second Load Balancer Node to a Service Instance

If your Oracle Java Cloud Service instance has a load balancer node, you can add a second active load balancer node to the same instance.

> **Note:**
>
> This procedure applies only to service instances that include a user-managed load balancer (Oracle Traffic Director). You can’t use Oracle Java Cloud Service to add an Oracle-managed load balancer to an existing service instance that was provisioned without any load balancer. Instead you must manually provision and configure the load balancer by using either Oracle Cloud Infrastructure Load Balancing or Oracle Cloud Infrastructure Load Balancing Classic, depending on the region where the service instance was created.

A service instance can include zero, one, or two Oracle Traffic Director (OTD) nodes. Each node is assigned a separate public IP address. A configuration with two active load balancer nodes can be used to provide high availability and higher load-balancing capacity.

> **Note:**
>
> Oracle Java Cloud Service *does not* fail-over application requests between load-balancer nodes. If one of the load-balancer nodes is unavailable, you are responsible for ensuring that requests fail over to another node. The failover-group feature of Oracle Traffic Director is not supported.

To add a second Oracle Traffic Director node to a service instance, complete the following steps:

1. Navigate to the Overview page for the instance to which you want to add a node.
2. Click **Manage this instance** ☰ next to the instance name and select **Scale Out**.
3. In the Scale Out dialog box, select **OTD**.
4. If the first load balancer node uses a reserved IP address, you must select an IP reservation for the second node in the **Reserved IPs** field.

   To reserve an IP address for use by the second load balancer node, click the gear icon next to this field. See Creating an IP Reservation.
5. Click **Scale Out**.

The Overview page is updated to show that the load balancer is being added. The service instance is in maintenance status and you cannot start any other management operation on the service instance. Click the ⟳ (Refresh) icon to check the latest status.

Both the load balancer nodes are active and can distribute requests to the Managed Servers in your service instance. Each load balancer node has a unique public IP address.

If you require the WebLogic Plug-in Enabled control to be set in Oracle WebLogic Server, you must set this control manually. If you add a load balancer to an Oracle Java Cloud Service instance after the service instance was created, Oracle Java Cloud Service does **not** set the WebLogic Plug-in Enabled control in Oracle WebLogic Server for you. This behavior differs from the behavior when a load balancer is added to a service instance while the service instance is being created. For details, see *Understanding the use of "WebLogic Plugin Enabled"*.

Remove a Load Balancer Node from a Service Instance

An Oracle Java Cloud Service instance can include zero, one, or two load balancer nodes. When you no longer need two load balancer nodes, you can remove the second node.

> **Note:**
>
> This procedure applies only to service instances that include a user-managed load balancer (Oracle Traffic Director). You can't use Oracle Java Cloud Service to remove an Oracle-managed instance of Oracle Cloud Infrastructure Load Balancing Classic from an existing service instance. To remove an Oracle-managed instance of Oracle Cloud Infrastructure Load Balancing, see Remove the Oracle-Managed Load Balancer.

To remove the second load balancer node:

1. Navigate to the Overview page for the instance for which you want to remove a load balancer node.
2. Expand the **Oracle Load Balancer** section of the Overview page.
3. Click **Manage this node** ☰ next to the second load balancer node, and select **Remove Node**.

   The Remove Node dialog box is displayed.
4. Click **Remove Node**.

The Overview page is updated to show that the load balancer node is being removed.

While the node is being removed, the service instance is in maintenance status, and you cannot start any other management operation on the instance.

Configure a Load Balancer for a Service Instance

After creating an Oracle Java Cloud Service instance with a load balancer, or after adding a load balancer to an existing service instance, you can modify the load balancer configuration to meet your requirements.

Topics

- Configure Oracle Traffic Director
- Configure an Oracle Cloud Infrastructure Load Balancing Instance
- Configure an Oracle Cloud Infrastructure Load Balancing Classic Instance

Configure Oracle Traffic Director

For a service instance that has a user-managed load balancer, use the Load Balancer Console to access Oracle Traffic Director (OTD).

If your service instance does not have a load balancer, see Add a Load Balancer to a Service Instance.

> **Note:**
>
> Prior to modifying the default load balancer configuration, read the administration best practices for Oracle Traffic Director in Administration Best Practices.

1. Access the Oracle Java Cloud Service console.
2. Click **Manage this instance** ☰ for the desired service instance and select **Open Load Balancer Console**.

   (Optional) Enter the result of the step here.
3. Log in to Oracle Traffic Director Administration Console using the credentials that you defined when provisioning your Oracle Java Cloud Service instance.
4. For service instances running Oracle Traffic Director 12*c* refer to these topics in *Administering Oracle Traffic Director*:
   - Features of Oracle Traffic Director
   - Overview of Administration Tasks

   For service instances running Oracle Traffic Director 11*g* refer to these topics in *Oracle Traffic Director Administrator's Guide*:
   - Features of Oracle Traffic Director
   - Overview of Administration Tasks

Configure an Oracle Cloud Infrastructure Load Balancing Instance

You can't update the configuration of an Oracle Cloud Infrastructure Load Balancing instance that was provisioned automatically during the creation of an Oracle Java

Cloud Service instance. But if you created and configured an Oracle Cloud Infrastructure Load Balancing instance manually, then you can update its configuration at any time.

See Managing a Load Balancer in the Oracle Cloud Infrastructure documentation.

Configure an Oracle Cloud Infrastructure Load Balancing Classic Instance

Configure the Oracle-managed load balancer for a service instance that was created in an Oracle Cloud Infrastructure Classic region.

See Viewing and Monitoring Your Load Balancers in *Using Oracle Cloud Infrastructure Load Balancing Classic*.

Set Up an Oracle Cloud Infrastructure Load Balancer

If you need the ability to update the Oracle Cloud Infrastructure Load Balancing configuration for an Oracle Java Cloud Service instance, then you must create the load balancer manually. You can't update the Oracle Cloud Infrastructure Load Balancing configuration if the load balancer is provisioned automatically during the creation of the Oracle Java Cloud Service instance.

This topic is applicable to Oracle Java Cloud Service instances that are provisioned without any load balancer and without Oracle Identity Cloud Service enabled.

Topics:

- Prepare to Set Up an Oracle Cloud Infrastructure Load Balancer
- Create and Configure an Instance of Oracle Cloud Infrastructure Load Balancing

Prepare to Set Up an Oracle Cloud Infrastructure Load Balancer

Before you begin setting up an instance of Oracle Cloud Infrastructure Load Balancing for your Oracle Java Cloud Service instance, understand the advantages and disadvantages of using a manually configured load balancer. In addition, gather the required information about the Oracle WebLogic Server managed servers in the Oracle Java Cloud Service instance.

This topic is not applicable to Oracle Java Cloud Service instances with Oracle Identity Cloud Service enabled.

1. Understand the advantages of using a manually configured instance of Oracle Cloud Infrastructure Load Balancing, when compared with a load balancer that's provisioned automatically while creating an Oracle Java Cloud Service instance.

   A manually configured Oracle Cloud Infrastructure Load Balancing instance gives you greater flexibility and control.

   - You can choose the bandwidth shape while creating the load balancer.
     An Oracle Cloud Infrastructure Load Balancing instance that's provisioned automatically during the creation of an Oracle Java Cloud Service instance is configured to use the 100-Mbps shape; you can't choose the shape.

   - You can configure the parameters of the load balancer. For example, you can add your own SSL/TLS certificates, configure listeners, add multiple backend sets, configure routing rules, and so on.

2. Be aware of the disadvantages of using a manually configured instance of Oracle Cloud Infrastructure Load Balancing.

A manually configured load balancer imposes certain administrative responsibilities:

- When you scale-out or scale-in your Oracle Java Cloud Service instance, the backend set of a manually configured Oracle Cloud Infrastructure Load Balancing instance is not updated automatically. You must update the backend set manually to add or remove the Oracle WebLogic Server nodes.
- When you delete the Oracle Java Cloud Service instance, the load balancer instance is not removed automatically; you must delete it separately.

3. Obtain an SSL/TLS certificate.

   You can use a certificate that's issued by a third-party Certificate Authority (CA), or a self-signed certificate that you generate by using tools such as Open SSL.

   For more information about obtaining a CA certificate, see the documentation provided by your CA. For the steps to generate self-signed certificates, see the documentation for the certificate-generation tool that you want to use.

4. Identify the listen ports of the Oracle WebLogic Server managed servers in your Oracle Java Cloud Service instance.

   The default listen ports of the managed servers in an Oracle Java Cloud Service instance are 8001 for HTTP and 8002 HTTPS. The listen ports are defined in the network channel configuration of each managed server.

   - If you want to terminate SSL at the load balancer, then use the HTTP port number of the managed server.
   - If you want the load balancer to route requests to the backend using HTTPS, then use the HTTPS port number.

   You can find the listen ports of the managed servers in the Oracle WebLogic Server administration console.

   a. Sign in the Oracle WebLogic Server Administration Console of your Oracle Java Cloud Service instance.

      See Access an Administration Console for a Service Instance.

   b. In the **Domain Structure** pane, expand **Environment**, and click **Servers**.

      Complete the steps that follow for each managed server to which you want the load balancer to route requests.

   c. Click the name of the managed server.

   d. Click the **Protocols** tab.

   e. Click the **Channels** subtab.

   f. Note the HTTP or HTTPS port number (as required) that's displayed in the **Public Port** field.

Create and Configure an Instance of Oracle Cloud Infrastructure Load Balancing

Using the Oracle Cloud Infrastructure web console, create a load balancer and configure its backend set.

This topic is not applicable to Oracle Java Cloud Service instances with Oracle Identity Cloud Service enabled.

1. Sign in to the Oracle Cloud Infrastructure web console.
2. In the **Regions** list near the upper-right corner, select the region in which you created your Oracle Java Cloud Service instance.
3. Create an instance of Oracle Cloud Infrastructure Load Balancing.
   a. From the navigation menu, under the Core Infrastructure group, select **Networking**, and then select **Load Balancers**.
   b. In the **Compartment** field, select the compartment that you want to create the load balancer in.
   c. Click **Create Load Balancer**.
   d. Enter a name for the load balancer.
   e. Specify whether the load balancer must be public or private.
   f. Select the bandwidth shape.

      Note that the shape you choose here affects the billing for the load balancer.
   g. Select the virtual cloud network (VCN) and the subnets to which you want to attach the load balancer.

      In a region that has more than one availability domain (AD), you can select either a single regional subnet (recommended) or two AD-specific subnets.
   h. Click **Next Step**.
   i. Select a load balancing policy.
   j. Click **Add Backends**.
   k. Click **Change Compartment**, and then select **ManagedCompartmentForPaaS**.
   l. Locate and select the nodes of your Oracle Java Cloud Service instance.

      The names of the compute nodes are in the format, `*subscriptionID*|JaaS|*jcsInstanceName*|wls|vm-*n*`.
      For example: `599949999|JaaS|myJCSinstance|wls|vm-1`

      Look for the compute nodes where *`jcsInstanceName`* matches the name of your Oracle Java Cloud Service instance.
   m. Click **Add Selected Backends**.
   n. Change the **Port** of each server to the port at which the managed server node listens for requests (for example, `8001`).
   o. Optional: Click **Advanced Options**, and then change the default name of the backend set.
   p. Click **Next Step**.
   q. Optional: Change the default listener name and port.
   r. Select or paste your SSL certificate.
   s. If you selected a self-signed certificate, then select or paste the corresponding private key, and enter the private key passphrase.
   t. Click **Show Advanced Options**.

      On the **Session Persistence** tab, by default, persistent sessions inserted by the load balancer is enabled with the **Enable load balancer cookie persistence** option, irrespective of the cookie used by the application running on the WebLogic cluster.

    See Managing Backend Sets.

  u. Click **Create Load Balancer**.

4. Configure the load balancer to include the `WL-Proxy-SSL` header in the requests that it forwards to the Oracle WebLogic Server nodes in the backend set.

   Oracle WebLogic Server uses this header to determine that the requests came to the load balancer over SSL/TLS.

   a. From the load balancer details page, under Resources in left navigation pane, click **Rule Sets**.

   b. Click **Create Rule Set**.

   c. Enter a name for the rule set.

   d. Select **Specify Request Header Rules**.

   e. Configure the rule:

      - **Action**: Select **Add Request Header**.
      - **Header**: Enter `WL-Proxy-SSL`
      - **Value**: Enter `true`

   f. Click **Create**.

   g. Click **Close**.

      Wait for the rule to be created.

   h. Under Resources in the left navigation pane, click **Listeners**.

   i. Locate the listener that you created earlier, click ⋮, and then select **Edit**.

   j. In the Edit Listener dialog box, in the Rule Sets section, click **Additional Rule Set**.

   k. Select the rule set that you created, and then click **Save Changes**.

   l. Click **Close**.

5. Ensure that the security list of the load balancer's subnet has the required security rules to allow TCP traffic from the Internet to the listener port that you created.

   If your region has more than one AD and if you specified two AD-specific subnets for the load balancer, then complete the following steps for each of the two subnets.

   a. On the Load Balancer Details page, locate the **Subnet** field, and then click the subnet.

      The VCN that contains the subnet is displayed.

   b. Click your load balancer's subnet.

   c. Click the first security list for the subnet.

   d. Under **Ingress Rules**, check whether a rule with the following properties exists:

```
Source: 0.0.0.0/0
IP Protocol: TCP
Source Port Range: All
Destination Port Range: yourListenerPort
```

If the rule exists, then skip the remainder of this substep and proceed to the "Verify access" step.

e. If the rule doesn't exist, then click **Add Ingress Rules**.

f. For **Source CIDR**, enter `0.0.0.0/0`.

g. For **Destination Port Range**, enter the port number of your listener.

h. Click **Add Ingress Rules**.

6. Verify access to the applications deployed to your Oracle Java Cloud Service.

   a. Return the Load Balancer Details page.

   b. Note the public IP address displayed in the **IP Address** field.

   c. Construct the load balancer URL for the application that you want to access.

      URL format: `https://`*`yourLBAddress`*`:`*`yourListenerPort/yourAppContextRoot`*
      URL example: `https://203.0.113.100:4343/my-app`

      If you didn't change the listener port, the default is `443`.

      The context root of the sample application that's included with Oracle Java Cloud Service is `/sample-app`. You can find the context root of your application from the application settings in the Oracle WebLogic Server Administration Console.

   d. In your web browser, go the URL that you constructed.

For more information, see Managing a Load Balancer in the Oracle Cloud Infrastructure documentation.

Use an Oracle Cloud Infrastructure Load Balancer

You can remove the Oracle-managed load balancer that's provisioned automatically during the creation of an Oracle Java Cloud Service instance in an Oracle Cloud Infrastructure region, and reconfigure the service instance to use an existing instance of Oracle Cloud Infrastructure Load Balancing.

This topic is applicable to Oracle Java Cloud Service instances that you've originally created by enabling the use of Oracle Identity Cloud Service or by using Oracle Cloud Infrastructure Load Balancing without enabling Oracle Identity Cloud Service.

After you remove the Oracle-managed load balancer that's provisioned automatically, if applicable, you'll need to modify the Open Sample Application URL on the Oracle Java Cloud Service Console.

Topics:

- Remove the Oracle-Managed Load Balancer
- Configure an Oracle Cloud Infrastructure Load Balancer
- Verify Access to the Deployed Sample Application

Remove the Oracle-Managed Load Balancer

You can remove the Oracle-managed instance of Oracle Cloud Infrastructure Load Balancing that's provisioned automatically during the creation of an Oracle Java Cloud Service instance in an Oracle Cloud Infrastructure region.

You can use only the REST API to remove the Oracle-managed instance of Oracle Cloud Infrastructure Load Balancing from a service instance. This remove option is not available from the Oracle Java Cloud Service Console.

The REST API operation can be executed on an Oracle Java Cloud Service instance that has or does not have Oracle Identity Cloud Service enabled.

See Delete the Oracle-Managed Load Balancer in *REST API for Oracle Java Cloud Service*.

> **Note:**
>
> If the sample application `sample-app` is deployed to the original Oracle-managed load balancer, the sample application URL that's displayed on the Oracle Java Cloud Service Console no longer works after you remove or replace the load balancer. To change the sample application URL, see Update a Service Instance Configuration in *REST API for Oracle Java Cloud Service*.

Configure an Oracle Cloud Infrastructure Load Balancer

For the Oracle Cloud Infrastructure Load Balancing instance you plan to use with your existing Oracle Java Cloud Service instance, you have to configure the backend set and listener manually to one or more nodes of the Oracle Java Cloud Service instance.

See Overview of Load Balancing and Managing Backend Servers in the Oracle Cloud Infrastructure documentation.

When you create or configure an instance of Oracle Cloud Infrastructure Load Balancing, be sure to follow these design recommendations:

- For a service instance that uses Oracle Identity Cloud Service to authenticate Oracle WebLogic Server administrators and application users, the load balancer backend servers must use port 9075.
- For a service instance that does not have Oracle Identity Cloud Service enabled, the load balancer backend servers use port 8001.
- The health check policy of the load balancer backend set have the following properties:
  - Protocol: `HTTP`
  - Port: `0`
  - Interval in MS: `30000`
  - Timeout in MS: `15000`
  - Number of retries: `2`
  - Status code: `404`
  - URL Path (URI): `/`
  - Response body regex: `.*`
- The listener has the following properties:

  - Protocol: `HTTP`
  - Port: `443`
  - Use SSL: `Yes`
  - Certificate: Use an SSL/TLS certificate that's issued by a third-party Certificate Authority (CA), or a self-signed certificate that you generate by using tools such as Open SSL. For more information about obtaining a CA certificate, see the documentation provided by your CA. For the steps to generate self-signed certificates, see the documentation for the certificate-generation tool that you want to use.
  - Path Route Set: If the URL matches (Prefix Match) /, direct traffic to the backend set.
  - Rule Set: The Add Request HTTP header `WL-PROXY-SSL` is set to `true`.
- There's communication between the Oracle Java Cloud Service instance nodes and the Oracle Cloud Infrastructure Load Balancing instance if the service instance nodes and the load balancer instance are provisioned in different virtual cloud networks.
- If the sample application `sample-app` was deployed to the original Oracle-managed load balancer, use the REST API to change the sample application URL that's displayed on the Oracle Java Cloud Service Console. See Update a Service Instance Configuration in *REST API for Oracle Java Cloud Service*.
- When you scale out or scale in an Oracle Java Cloud Service instance that uses a manually configured instance of Oracle Cloud Infrastructure Load Balancing, you must update the backend set manually to add or remove the Oracle WebLogic Server nodes.
- When you delete an Oracle Java Cloud Service instance that uses a manually configured instance of Oracle Cloud Infrastructure Load Balancing, the load balancer is not removed automatically so you must delete it separately.

Verify Access to the Deployed Sample Application

If the sample application is originally deployed to the Oracle-managed load balancer that you've removed from an Oracle Java Cloud Service instance, make sure you can access the sample application after you've manually configured your own Oracle Cloud Infrastructure Load Balancing instance to use the nodes on the Oracle Java Cloud Service instance.

Use a web browser

To access the sample application in a web browser, enter the following URL:

```
https://load_balancer_IP_address/sample-app
```

If Oracle Identity Cloud Service is enabled on your Oracle Java Cloud Service instance, use the following URL:

```
https://load_balancer_IP_address/__protected/sample-app
```

Access from the Oracle Java Cloud Service console

After you remove the original Oracle-managed load balancer, the **Open Sample Application** URL that's displayed on the Oracle Java Cloud Service Console will no longer work. You have to change the URL to use the public IP address of your own Oracle Cloud Infrastructure Load Balancing load balancer that's now configured to the Oracle Java Cloud Service instance nodes. You can make this change by using the REST API to update the `SAMPLE_ROOT` parameter as described in Update a Service Instance Configuration.

About the Storage Volumes Attached to the Load Balancer Nodes

If Oracle Traffic Director is enabled for an Oracle Java Cloud Service instance, the Oracle Traffic Director administration server is hosted on one compute node. The second Oracle Traffic Director node, if it exists, is hosted on another node.

> **Note:**
>
> This topic applies only to service instances that include a user-managed load balancer (Oracle Traffic Director). It does not apply to service instances that use an Oracle-managed instance of Oracle Cloud Infrastructure Load Balancing.

The following table lists the mount points of the storage volumes that are attached to a load balancer node:

Mount Point	Content
`/u01/jdk`	Java installation (`JAVA_HOME`) (treat as read-only)
`/u01/app/oracle/middleware/otd`	Oracle Traffic Director installation (`ORACLE_HOME`) (treat as read-only)
`/u01/data/otd-instance/otd_domain` (only for Oracle Traffic Director 12c)	Oracle WebLogic Server domain (`DOMAIN_HOME`) that is used to manage and monitor Oracle Traffic Director
`/u01/data/otd-instance` (only for Oracle Traffic Director 11g)	Scripts and configuration data for Oracle Traffic Director (`INSTANCE_HOME`)

14

About the Infrastructure Resources Used by Oracle Java Cloud Service

When you create an Oracle Java Cloud Service instance, the required virtual machines (VMs), block storage volumes, and most of the network settings are provisioned and configured for you.

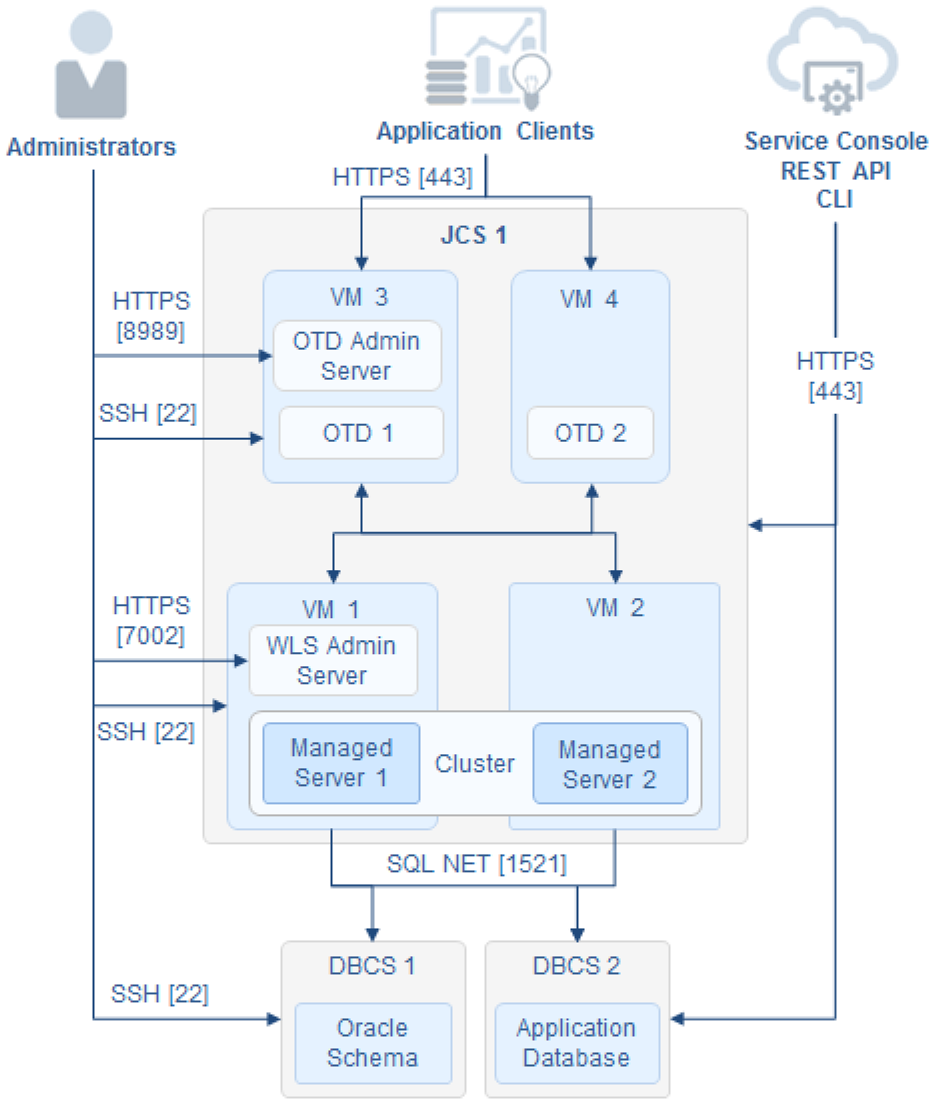
Topics:

- About the Deployment Topology
- Compute Topology for Oracle Java Cloud Service Instances
- About the Storage Volumes Attached to the WebLogic Server Nodes

About the Deployment Topology

Oracle Java Cloud Service enables you to quickly deploy an enterprise-grade application server topology, with caching and load balancing.

The following diagram illustrates an Oracle Java Cloud Service instance that has an Oracle WebLogic Server cluster connected to an Oracle Database Cloud Service deployment. A load balancer receives requests from clients and forwards them to the WebLogic Server nodes.



In this example, the Oracle Java Cloud Service instance has a single Oracle WebLogic Server domain that contains an Administration Server and a cluster of two Managed Servers for hosting applications. The load balancer is a dual-node Oracle Traffic Director setup.

> **Note:**
>
> For information about the network protocols and default ports that can be used to access an Oracle Java Cloud Service instance, see Understanding the Default Access Ports. The HTTP port is disabled if you created the instance by using the instance-creation wizard in the web console.

When Oracle Coherence is enabled for a service instance, there is a second WebLogic Server cluster of Managed Servers for storing Coherence data. Both the WebLogic Server clusters are associated with the single Coherence cluster in the domain. For information about the deployment topology when Oracle Coherence is enabled for an Oracle Java Cloud Service instance, see About Oracle Coherence in Oracle Java Cloud Service.

Compute Topology for Oracle Java Cloud Service Instances

Each instance of Oracle Java Cloud Service is hosted on one or more Oracle Linux compute nodes. The number of compute nodes that an instance has depends on the number of WebLogic Servers and the load-balancer configuration.

The first node contains the WebLogic Administration Server and the first Managed Server. Each of the other Managed Servers runs on a separate node.

If Oracle Traffic Director is used as the load balancer, then the Oracle Traffic Director administration server and the first load-balancer node are on one compute node. If the instance has a second load-balancer node, then that node is on a separate compute node.

The following table summarizes the number of Managed Servers you can have in the WebLogic Server cluster, and the corresponding nodes:

Compute Node	1–Node Cluster	2–Node Cluster	4–Node Cluster
1st node	Contains WebLogic Administration Server and Managed Server 1	Contains WebLogic Administration Server and Managed Server 1	Contains WebLogic Administration Server and Managed Server 1
2nd node		Contains Managed Server 2	Contains Managed Server 2
3rd node			Contains Managed Server 3
4th node			Contains Managed Server 4
5th node	If present, this node contains the load balancer’s administration server	If present, this node contains the load balancer’s administration server	If present, this node contains the load balancer’s administration server
6th node	If present, this node contains a second load balancer	If present, this node contains a second load balancer	If present, this node contains a second load balancer

Note:

By default a load balancer is not enabled for an instance that has a single-node WebLogic cluster, so the Oracle Traffic Director node won't be present. When you create a service instance that consists of a multinode cluster in the domain, Oracle recommends that you enable a load balancer for the service instance. If enabled, the Oracle Traffic Director node would be present.

When Oracle Coherence is enabled for a service instance, a node on the Coherence data tier can have one or more storage-enabled Managed Servers. You configure the initial number of Coherence nodes and the number of Managed Servers per node when you create the service instance.

The following table summarizes the number of nodes on the application tier and Coherence data tier, and the corresponding Managed Servers contained in the nodes for an Oracle Java Cloud Service—Coherence instance. The example in the table shows a configuration consisting of a 2-node application tier cluster (storage-disabled), and a 3-node Coherence data tier cluster (storage-enabled) in which one Managed Server is running on each node:

Compute Node	Contains	WebLogic Server Cluster
1st node	WebLogic Administration Server, Managed Server 1(storage-disabled)	Application Tier
2nd node	Managed Server 2 (storage-disabled)	Application Tier
3rd node	Managed Server 3_DG (storage-enabled)	Coherence Data Tier
4th node	Managed Server 4_DG (storage-enabled)	Coherence Data Tier
5th node	Managed Server 5_DG (storage-enabled)	Coherence Data Tier

Appropriate security rules are configured on the Oracle Java Cloud Service nodes to enable communication among the different nodes hosting the WebLogic managed servers, and also with the Oracle Traffic Director nodes and the Oracle Database Cloud Service nodes.

You have access to all the compute nodes, including the node on which the WebLogic Administration Server is running. You can use a Secure Shell (SSH) client to log into a node, as described in Access a Node with a Secure Shell (SSH).

About the Storage Volumes Attached to the WebLogic Server Nodes

You can connect using `ssh` to all the compute nodes of an instance and access the attached storage volumes. To ensure that the instance remains manageable, you must treat some volumes as read-only disks.

The following table maps the volumes that are attached to a WebLogic Server node and the corresponding mount points:

Mount Point	Content	Volume
`/` and `/boot`	Operating system binaries	`boot`
`/dev/shm`	Swap space	`boot`
`/u01/data/backup`	Backups, if configured (writable by the `oracle` user; the `opc` user has read-only access)	`backup`
`/u01/data/domains`	WebLogic Server domain data (`DOMAIN_HOME`) Deployed applications and configuration files (`APPLICATION_HOME`)	`domain`
`/u01/app/oracle/middleware`	Oracle WebLogic Server binaries and Oracle Traffic Director binaries (`MW_HOME`) (treat as read-only)	`common`

Mount Point	Content	Volume
/u01/app/oracle/tools **Caution**: Don't modify any scripts in this directory.	Binaries and related metadata required for service management (JCS_RESERVED) (treat as read-only)	common
/u01/jdk	JDK binaries (JDK_HOME) (treat as read-only)	common

About the Storage Volumes Attached to the Load Balancer Nodes

If Oracle Traffic Director is enabled for an Oracle Java Cloud Service instance, the Oracle Traffic Director administration server is hosted on one compute node. The second Oracle Traffic Director node, if it exists, is hosted on another node.

Note:

This topic applies only to service instances that include a user-managed load balancer (Oracle Traffic Director). It does not apply to service instances that use an Oracle-managed instance of Oracle Cloud Infrastructure Load Balancing.

The following table lists the mount points of the storage volumes that are attached to a load balancer node:

Mount Point	Content
/u01/jdk	Java installation (JAVA_HOME) (treat as read-only)
/u01/app/oracle/middleware/otd	Oracle Traffic Director installation (ORACLE_HOME) (treat as read-only)
/u01/data/otd-instance/otd_domain (only for Oracle Traffic Director 12c)	Oracle WebLogic Server domain (DOMAIN_HOME) that is used to manage and monitor Oracle Traffic Director
/u01/data/otd-instance (only for Oracle Traffic Director 11g)	Scripts and configuration data for Oracle Traffic Director (INSTANCE_HOME)

15
Troubleshoot Oracle Java Cloud Service

This section describes common problems that you might encounter when using Oracle Java Cloud Service and explains how to solve them.

Topics:

- Before You Begin Troubleshooting
- Find Diagnostic Information to Help with Troubleshooting
- Problems with Failure of a Running Service when the Schema User Password Expires
- Problems with Creating Service Instances
  - I cannot create an Oracle Java Cloud Service — Virtual Image instance when I choose a Oracle Database Cloud Service — Virtual Image database deployment
  - I receive a database connectivity error message
  - I cannot create a service when I have many service instances
  - I cannot create a service instance, even after waiting for an hour
  - I cannot create a service instance when the service instance name is not unique
  - I can create a service instance but the Coherence Data Tier failed to create
  - I receive an error message stating that no database service is available
  - I encounter Intermittent provisioning failures for clustered instances based on WebLogic Server 12.2.1
  - I encounter a database connection error when creating an Oracle Java Cloud Service instance
  - I cannot select my Oracle Autonomous Database (Oracle Autonomous Transaction Processing) or Oracle Cloud Infrastructure Database from the web console
- Problems with Deploying and Accessing Applications
  - I can’t deploy an application to an Oracle Java Cloud Service instance based on WebLogic Server 11*g*
  - I can’t access an application using the URL from the WebLogic Server Administration Console Testing tab
  - I can’t access an application through the HTTP port
- Problems with Scaling
  - My scale-out operation does not start
  - My scale-in operation is not allowed
  - My service is too busy to allow scaling
  - My scaling operation failed when the storage space threshold was exceeded
- Problems with Patching and Rollback

  - My identity key store and trust store are missing after a patching, rollback, or restoration operation
  - I receive a message stating that the virtual machines are unhealthy
  - I receive a message stating that the service is busy with another operation
- Problems with Backup and Restoration
  - Backups fail after changing my Oracle Cloud password
  - The Oracle Traffic Director is not backed up
  - There is not enough space for my backup
  - The restoration operation fails and generates an error about pre-check failure
  - One of my backups is showing a warning icon
- Problems with Performance of Oracle Java Cloud Service—Coherence Service Instances
- Problems with Restart
  - Restart fails after a scale down operation intended to remedy a quota breach
  - Monitor a VM’s boot log
  - My custom storage volumes have become detached
  - My content changes on the Boot/OS volume are gone
- Problems with Connectivity
  - My private key is lost or corrupted
  - My connection to a VM is refused
  - I received a hostname verification error when attempting to connect to Node Manager
- Problems with Database Connectivity When Upgrading the Infrastructure Schema Database
- Problems with the Node Manager
- Problems with a Database Deployment
- Problems with Connection validation when provisioning an Oracle Java Cloud Service-Virtual Image instance with Oracle Database Cloud Service 12.1 VI
- Problems with Transparent Data Encryption Wallet Error when Provisioning an Oracle Java Cloud Service-Virtual Image Instance with Oracle Database Cloud Service 11g Virtual Image
- Problems Opening the WebLogic Server Administration Console from Fusion Middleware Control
- Problems Accessing Applications Via a Custom Web Server
- Security Checkup Tool Warnings
- How can I Fine-Tune Performance?

Before You Begin Troubleshooting

Before you begin troubleshooting, perform the following tasks.

1. Ensure that you are using best practices. See Keeping Your Instances Manageable by Oracle Java Cloud Service.
2. Check *Known Issues with Oracle Java Cloud Service* for known problems and solutions that could help you address your issues with Oracle Java Cloud Service.

Find Diagnostic Information to Help with Troubleshooting

You can use the WebLogic Administration Console and other tools to find more information about problems with Oracle Java Cloud Service and help you troubleshoot them.

Topics:

- Use the WebLogic Server Administration Console to Find Diagnostic Information
- Use the WebLogic Server Administration Console to Find Log Files
- Find Status Messages for Oracle Java Cloud Service Instances
- Find VM Boot Log Messages

Use the WebLogic Server Administration Console to Find Diagnostic Information

You can find diagnostic information easily by using the WebLogic Server Administration Console.

To find diagnostic information:

1. Navigate to the Oracle Java Cloud Service Console.
2. Click the name of your service instance.
3. On the Instance Overview page, open the WebLogic Server Administration Console.
   a. Click the ☰ **Manage this instance** icon and select **Open WebLogic Server Administration Console**.

      A new browser opens and you are redirected to the login page.

      If the server is protected with a self-signed certificate, you will be warned that this certificate is not trusted. This warning only appears if you have not added the self-signed certificated to the browser’s exception list previously.

   b. Accept the certificate.
   c. When the WebLogic Server Console appears, enter the user name and password your provided when you created the service instance.

   The WebLogic Server Administration Console is displayed.
4. In the Domains area, expand **Diagnostics**.
5. Click on the diagnostics that interests you.

   For information on the diagnostic choices, click on **Diagnostics**.

Use the WebLogic Server Administration Console to Find Log Files

You can find log files easily by using the WebLogic Server Administration Console.

To find the log files:

1. Navigate to the Oracle Java Cloud Service Console.
2. Click the name of your service instance.
3. On the Instance Overview page, open the WebLogic Administration Console.

    a. Click the ☰ **Manage this instance** icon and select **Open WebLogic Server Administration Console**.

    A new browser opens and you are redirected to the login page.

    If the server is protected with a self-signed certificate, you will be warned that this certificate is not trusted. This warning only appears if you have not added the self-signed certificated to the browser’s exception list previously.

    b. Accept the certificate.

    c. When the WebLogic Server Console appears, enter the user name and password your provided when you created the service instance.

    The WebLogic Server Administration Console is displayed.

4. In the Domains area, expand **Diagnostics**.
5. Click **Log Files**.
6. The Log Files table is displayed.
7. Click the option to the left of the log file you want to view.
8. Click **View**.

    The log file you selected is displayed in the table.

9. (Optional) If you do not find the information you are looking for, customize the table to select the time interval you want to view.

    a. View the log file.

    b. Click the **Customize this table** link above the log file.

    c. From the Time Interval drop-down menu, select the time interval for filtering the information the information in the table.

    You can choose an interval ranging from the last five minutes to the last one week. You can also view all log entries or customize the time interval.

Find Status Messages for Oracle Java Cloud Service Instances

From the Oracle Java Cloud Service Console, you can view status messages to determine why an attempt to create a service instance failed.

Messages for operations such as backup, restoration, scaling, and patching appear on the Activity page. See Monitor Activity.

To find status messages for a failed attempt to create a service instance:

1. Navigate to the Oracle Java Cloud Service Console.
2. Expand the arrow next to **Instance Create and Delete History**.
3. Click on the name of the service instance you created or deleted, or click on **Details**.

    A list of status messages is displayed. The messages trace the process for creating the service instance from the beginning to the point of failure. Success messages are displayed in addition to error messages.

Find VM Boot Log Messages

You can find diagnostic information in the boot logs for the VMs that make up an Oracle Java Cloud Service instance.

Monitor or troubleshoot the boot progress of individual VMs by using Oracle Cloud Infrastructure Compute Classic. See Viewing the Boot Log of an Instance in *Using Oracle Cloud Infrastructure Compute Classic*. Ignore information in this topic about the Compute API.

Problems with Failure of a Running Service when the Schema User Password Expires

An Oracle Java Cloud Service instance can fail suddenly and issue password expiry error messages.

This failure occurs because the user password for the infrastructure repository schemas is set to expire 180 days after an Oracle Java Cloud Service instance is created. WebLogic Server generates the following error messages:

```
Received exception while creating connection for pool X: ORA-28001:
the password has expired
```

```
java.sql.SQLException: ORA-01017: invalid username/password; logon denied
```

Another symptom of this problem is that a patch precheck, restoration, or scale out operation may fail.

> **Note:**
>
> By default the schema password is set to Weblogic Administrator password during the provisioning of the Oracle Java Cloud Service instance.

See Change the Database Schema Password for an Oracle Java Cloud Service Instance.

Problems with Creating Service Instances

You might experience problems when creating services.

In the process of creating a service, the operation failure becomes visible in the following way:

1. The service instance appears in the Services list in the Oracle Java Cloud Service Console.
2. An **In progress...** message appears in the service instance details.
3. When the creation process fails, a **Failed** message is displayed and a red exclamation mark appears on the service instance's icon.

4. The service instance is listed in the **Instance Create and Delete History** section.
5. Click on **Details** to view progress and error messages.

   You can locate additional error messages using the procedure in Find Status Messages for Oracle Java Cloud Service Instances.

The most common sources of failure when creating a service instance include:

- Timeout errors
- SSH connection isssues
- Incorrect credentials
- Database listener down

The following solutions apply to problems creating service instances for Oracle Java Cloud Service.

I cannot create an Oracle Java Cloud Service — Virtual Image instance when I choose a Oracle Database Cloud Service — Virtual Image database deployment

A failure will occur when you attempt to create an Oracle Java Cloud Service instance with a Oracle Database Cloud Service — Virtual Image database deployment.

To prevent this failure, you must first configure the Oracle Database Cloud Service — Virtual Image environment.

I receive a database connectivity error message

You might not be able to create an Oracle Java Cloud Service instance because the `oracle` user does not have a password on Oracle Database Cloud Service instances. To modify the properties of the `oracle` users so that the password does not expire, see Problems Creating Deployments in *Administering Oracle Database Cloud Service*.

I cannot create a service when I have many service instances

Your account might not have enough compute quota to create the service instance.

If you have instances you do not need, delete them. If you need all your service instances, contact Oracle Sales and Services to buy more quota for your account.

I cannot create a service instance, even after waiting for an hour

If service creation fails after one hour, the system might be experiencing a heavy load, and resources are not yet available.

Wait before you try again to create the service. If the problem persists, contact Oracle Support Services.

I cannot create a service instance when the service instance name is not unique

Oracle Java Cloud Service instance creation can fail if the name you choose for the new service instance is identical to the name of another service instance, including a failed service instance. Also, the Oracle Java Cloud Service instance name cannot be the same as the name of an Database Cloud Service instance.

After an attempt to create an Oracle Java Cloud Service instance fails, Oracle Java Cloud Service may require some time to remove items that were created during the attempt. If the new and failed service instance names are identical, a naming conflict may occur and the attempt to create the new service instance may fail.

> **Note:**
>
> As a best practice, always ensure that your Oracle Java Cloud Service instance names are unique.

I can create a service instance but the Coherence Data Tier failed to create

Delete the service instance. Wait before you try again to create the service instance. If the problem persists, contact Oracle Support Services.

I receive an error message stating that no database service is available

If you attempt to create an Oracle Java Cloud Service using an Oracle Database Cloud Service database deployment that does not have backups enabled (Destination=None), then provisioning will fail and issue the following error message:

```
There are no Oracle Database Cloud Service instances available for Service
Level:
Oracle Java Cloud Service
```

Create a new database deployment with backups enabled and specify this database deployment when you create a new Oracle Java Cloud Service instance.

I encounter Intermittent provisioning failures for clustered instances based on WebLogic Server 12.2.1

Your attempt to create an Oracle Java Cloud Service instance based on WebLogic Server release 12.2.1 can fail if you create an instance containing a large number of cluster members.

The cause of the problem is that an exclusive configuration lock acquired by one process is released by another process that successfully acquires another exclusive lock.

This problem is intermittent, so try again to provision a service instance. Alternatively, provision a smaller cluster and then scale out your nodes. See Scale Out a Cluster.

I encounter a database connection error when creating an Oracle Java Cloud Service instance

In the process of creating an Oracle Java Cloud Service Instance while using the Service Instance Creation Wizard, you may receive the following error message: `Failed to connect to DBaaS Service`.

To help identify the problem, confirm that the user name and password are correct by connecting to the database via sqlplus. Also, confirm that you have the correct privileges. For more information, see About Database Cloud Service Roles and Users in *Administering Oracle Database Cloud Service*.

If you have done these checks and do not see any issues, the problem might be that the `oracle` password has expired on the database node.

You can change the properties of the `oracle` user so that the password does not expire. See Problems Administering Deployments in *Administering Oracle Database Cloud Service*.

Provisioning fails when infrastructure database creation times out

Infrastructure database creation times out and instance provisioning fails if the recycle bin size exceeds its threshold or if the infrastructure database size approaches the maximum database size.

Find the following information in the job file:

- Database recycle bin objects count
- Database recycle bin size
- Database maximum DB size
- Database current DB size

Look for the following messages:

- `Recycle bin size has exeeded 50% of current DB Size. Consider purging the recycle bin.`
- `DB usage size has exeeded 75% of Max DB Size.`

Oracle recommends that you purge the database recycle bin if the size exceed its limit. Clean up the database if the database usage size approaches the maximum database size.

I cannot select my Oracle Autonomous Database (Oracle Autonomous Transaction Processing) or Oracle Cloud Infrastructure Database from the web console

Your Oracle Autonomous Database (Oracle Autonomous Transaction Processing) or Oracle Cloud Infrastructure Database will not appear on the **Database Instance Name** dropdown field as an infrastructure schema database choice unless you specify the following policies:

For an Oracle Autonomous Database:

```
Allow service PSM to inspect autonomous-database in compartment
compartment_name
```

For an Oracle Cloud Infrastructure Database:

```
Allow service PSM to inspect database-family in compartment
compartment_name
```

For information on creating policies, see Creating the Infrastructure Resources Required for Oracle Platform Services.

Problems with Deploying and Accessing Applications

Problems might occur when you attempt to deploy or access an application.

I can’t deploy an application to an Oracle Java Cloud Service instance based on WebLogic Server 11*g*

You can deploy an application that relies on Java EE 6 or Java EE 7 component jars such as JSF 2.0 to an Oracle Java Cloud Service instance based on WebLogic Server 11*g* only if you manually package the relevant libraries for your application. Java EE 6 or Java EE 7 component jars such as JSF 2.0 are not packaged by default.

The recommended versions for deploying this type of application is WebLogic Server 12*c* (12.2.1), which supports Java EE 7.

I can’t access an application using the URL from the WebLogic Server Administration Console Testing tab

You cannot access a deployed application from the public internet if you use the URL displayed on the Testing tab of the WebLogic Sever Administration Console. The URLs shown on this tab are internal to Oracle Java Cloud Service. Instead, use the procedure in Access an Application Deployed to an Oracle Java Cloud Service Instance.

I can’t access an application through the HTTP port

By default, you cannot access an application running on an instance through the HTTP port if the instance was created by using the provisioning wizard available from the Oracle Java Cloud Service Console. You must enable the HTTP port after you create the service instance. The instance is accessible, however, via HTTPS without manual intervention.

Both the HTTP and HTTPS ports are enabled by default if you created the Oracle Java Cloud Service instance by using the REST API.

To enable the HTTP for a service instance created with the wizard, you must enable a listener port on the load balancer, then create an access rule. If your service instance has a load balancer, see Enable HTTP Access to a Service Instance.

If your service instance does not have a load balancer, you must enable a network channel on all Managed Servers to ensure that they are listening on the port you are opening, then create the access rule. See Create and Assign the Network Channel in *Administering Server Environments for Oracle WebLogic Server*.

See:

- About the Default Access Ports
- Updating the Default Access Ports When Creating a Service Instance in *REST API for Oracle Java Cloud Service*.

Problems with Scaling

Problems might occur when you attempt a scaling operation.

The following solutions apply to problems with scale-in and scale-out with Oracle Java Cloud Service.

My scale-out operation does not start

Your scale-out operation has been placed in the request queue, and it might be a few minutes before the operation is performed. Check status on the Activity tab of the Oracle Java Cloud Service Console.

Wait before you try to scale out again. If the problem persists, contact Oracle Support Services.

My scale-in operation is not allowed

The managed server you selected for scale-in is on the same virtual machine as the administration server. Removing this virtual machine is not allowed.

Select another virtual machine to scale in.

My service is too busy to allow scaling

Your service has a pending maintenance operation such as backup or patching.

Wait until maintenance has completed before you try scaling again.

My scaling operation failed when the storage space threshold was exceeded

A scaling operation fails when local disk storage usage exceeds a certain threshold. For a scale-out operation, the threshold is 90 percent. For a scale-in operation the threshold is 98 percent. A scaling precheck operation performs a disk usage check and issues an error message if the threshold has been exceeded, and then scaling fails. If you receive this error message, free up space on local disk storage.

A scale-in operation attempts to create a backup before scaling in. If you initiate frequent backups, local storage can fill up because backups are retained for seven days.

If you create frequent backups, delete backups before scaling in to avoid this problem. See Delete a Backup.

Problems with Patching and Rollback

This section identifies some potential issues you may face after patching and rollback operations.

The following recommendations ensure that patching and rollback operations enable you to continue running your applications.

My identity key store and trust store are missing after a patching, rollback, or restoration operation

If you have identity key stores and trust stores, they can disappear after you apply a patch, roll back a patch, or restore a backup. You may have configured one of the following:

- Custom identity key store and custom trust store
- Custom identity key store and Java standard trust store
- WebLogic Server identity key store and WebLogic Server trust store

Patching , rollback, and restoration operations replace the directories you may have used to keep the custom key store and trust store, so they are essentially emptied.

To protect your key store and trust store, create the key stores and trust stores by using the OPSS KeyStoreService (KSS). See Configuring the OPSS Keystore Service for Demo Identity and Trust in *Administering Security for Oracle WebLogic Server*.

If you don't want to use the OPSS KeyStoreService, you can put the key store and trust store in the WebLogic domain created by Oracle Java Cloud Service.

It's particularly important to protect your key store and trust store for JDK patching. Each JDK patch replaces the previous version.

Before you apply a WebLogic Server patch:

- Do not put CA certificates in the existing demo keystores
- Do not put custom key stores and trust stores in the `<MW_HOME>/wlserver/lib` directory
- Do not put CA certificates anywhere on the system except in key stores

I receive a message stating that the virtual machines are unhealthy

You cannot apply a patch if the service's virtual machines are not in a healthy state.

Restore the service using a backup and try patching again.

I receive a message stating that the service is busy with another operation

You cannot apply a patch when the service is under maintenance, for example, scaling or backup.

Wait until the service is no longer under maintenance and try patching again.

Problems with Backup and Restoration

Problems might occur when you attempt backup or restoration of an Oracle Java Cloud Service instance.

The following solutions apply to problems with backup and restoration operations for Oracle Java Cloud Service.

Backups are disabled

Oracle Java Cloud Service automatically disables backups for a service instance after **three consecutive failures** of scheduled backups. If you provided a notification email address when you created the service instance, you will receive an email message after each backup failure, and also after backups are disabled. The email message specifies the cause of the backup failure. To identify the cause of the backup failure by using the REST API, see View the Backup Configuration and View the Status of an Operation by Job Id in *REST API for Oracle Java Cloud Service*.

The most common cause for a backup failure are invalid storage credentials (see below). After correcting the cause of the backup failure, you can enable backups on the service instance. See Enable or Disable Backups.

Backups fail after changing my Oracle Cloud password

To prevent backup failure after you change your Oracle Cloud password, update the storage credentials in the following locations:

- Configure Backups dialog box, which you invoke from the Backup page in the Oracle Java Cloud Service user interface

  See Configure Scheduled Backups for an Oracle Java Cloud Service Instance.

- Oracle Database Cloud Service deployment

  See Updating the Password for Backing Up to the Storage Cloud in *Administering Oracle Database Cloud Service*.

The Oracle Traffic Director is not backed up

Typically, this occurs when the traffic director is currently busy serving other requests.

Verify that Oracle Traffic Director is running and in a healthy state, and try the backup operation again.

To check the health of the Oracle Traffic Director:

1. Navigate to the Oracle Java Cloud Service Console.
2. Open the Load Balancer Console for your service.

   a. Click the ☰ menu icon for your service instance and choose **Open Load Balancer Console**.

      A new browser opens and you are redirected to the Load Balancer Console's log-in page.

      If the server is protected by a self-signed certificate and you have not specified the certificate previously, you will be warned that your connection is not secure.

   b. Accept the certificate.

   c. When the log-in page appears, enter the username and password you provided when you created the service instance.

      The Oracle Traffic Director Administration Console is displayed.

3. On the left panel, select **Services**.

   The Services page is displayed on the right.

4. Notice the Status, State, and Health of your load balancer. The load balance is up and in good health if:

   - The arrow under Status is green and pointing up
   - The State is Running
   - The Health is OK

   Otherwise, the load balancer is down and the health of the load balancer is not okay.

There is not enough space for my backup

The backup storage area does not have enough space for the backup operation to create the archive. You will receive an error message.

1. Check the Backup page for the amount of space needed. This information is also returned in GET from the `/backupconfig` endpoint.
2. Log in to the Administration Server VM. See Access a Node with a Secure Shell (SSH).

3. Check the size of the backup mounted directory under `/u01/data/backup`:

   ```
   df -k
   ```

4. If there is not enough space for the backup, add more storage.

   See Add Storage to a Node.

The restoration operation fails and generates an error about pre-check failure

Either one or more servers are currently unreachable, or there is not enough space on one of the storage volumes.

To find the reason for the restoration failure:

1. Navigate to the Backup page.
   a. Click the name of the service instance for which you want to find the restoration status information.

      The Oracle Java Cloud Service Instance page is displayed with the Overview tile in focus, displaying detailed information about the service instance.
   b. Click the Administration tile.

      The Oracle Java Cloud Service Instance page is refreshed with the Administration tile in focus.
   c. Click the Backup tab.

      The Backup page is displayed.
2. Locate the icon for the restoration that failed.
3. Click on the date to the right of the icon.

   A pop-up containing the status details is displayed.

If the problem is that a server is unreachable, restart the VM or perform a scale-in operation if it is no longer needed. Try restoring the service again.

If there is not enough space for the backup, you will receive an error message telling you to scale-in the VMs, which were either not backed up or added after the backup operation, or use the **force** option for restoration.

Do one of the following:

- Scale-in the VMs that were either not backed up or added after the backup operation.
- Try to restore the backup by using the **force** option.
- Delete any unwanted backups. See Delete a Backup.
- Archive one or more backups to an Oracle Cloud Infrastructure Object Storage Classic container. See Move a Backup (Download or Archive) in *REST API for Oracle Java Cloud Service*.

One of my backups is showing a warning icon

When a scheduled backup is completed, Oracle Java Cloud Service tries to move older backups from block storage and delete older backups from the Oracle Cloud Infrastructure Object Storage Classic container.

If Oracle Java Cloud Service cannot move or delete the older backups, the **newly completed** backup shows a warning icon, thus:

This problem does not affect the newly completed backup. However, the presence of the older backups may cause future backups to fail because of insufficient space.

To prevent such failures, ensure that Oracle Java Cloud Service can remove the older backups when the next scheduled backup is completed:

1. To find out why Oracle Java Cloud Service could not move or remove the backups, place the cursor over the icon.

   A text rollover appears that contains detailed information about why Oracle Java Cloud Service could not move or remove the backups.

2. Correct the problem that prevented Oracle Java Cloud Service from moving or removing the backups.

   For example, to correct an access permission problem, ensure that the user name and password for the administrator of the Oracle Cloud Infrastructure Object Storage Classic container are correct. If necessary, change them as explained in Configure Scheduled Backups for an Oracle Java Cloud Service Instance.

3. When the next scheduled backup is completed, determine whether it shows the icon for a successful backup, thus:

   - If so, no further action is required.
   - If the next scheduled backup also shows the warning icon, contact Oracle Support Services.

Problems with Performance of Oracle Java Cloud Service—Coherence Service Instances

Depending on access patterns and memory usage, an Oracle Java Cloud Service — Coherence instance might show performance problems under a heavy load. You might see long, multi-second garbage collection times on Managed Servers of the Coherence data tier. To address this issue, you can change the garbage collection scheme to CMS.

To change the garbage collection scheme on a Managed Server on the Coherence data tier:

1. Navigate to the Oracle Java Cloud Service Console.
2. Open the Administration Console for your service by selecting it from the drop-down menu.

3. On the Administration Console, under Domain Structure, expand Environment and select Servers.

   The Summary of Servers page is displayed.

4. On the Configuration page, in the Servers table, click the name of a Managed Server of the Coherence data tier.

   The Settings page is displayed.

5. Click the Configuration tab, then the Server Start tab.
6. Click **Lock & Edit**.
7. In the Arguments pane, add the `–XX:+UseConcMarkSweepGC` flag to the end of the list.
8. Click **Activate Changes**.
9. Restart the Managed Server.

   See Use the WebLogic Server Administration Console to Shut Down Servers and Use the WebLogic Server Administration Console to Start Managed Servers.

10. Repeat this process for all Managed Servers of the Coherence data tier.

Problems with Restart

You might experience unexpected side-effects after restarting an Oracle Java Cloud Service instance or individual VMs. These effects can also occur after patching, which restarts VMs.

Restart fails after a scale down operation intended to remedy a quota breach

You can scale down a Oracle Database Cloud Service database deployment or Oracle Java Cloud Service instance if you have a quota breach in your account. Scaling down reduces compute resources. However, the automatic restart action can fail after scale-down.

For example, you could scale down a node from shape oc5 to oc3. Oracle Java Cloud Service puts the service instance into Maintenance mode, changes the state of the node to Configuring, and stops any servers running on the node. After applying the changes, Oracle Java Cloud Service is supposed to start the servers automatically. If the quota breach is not cleared by the time the orchestration is restarted with the smaller shape, the automatic server restart action could fail.

If the restart action fails, wait one hour for the quota breach to clear, then restart the service instance by using the Oracle Java Cloud Service Console.

Monitor a VM’s boot log

You can monitor the boot progress of individual VMs by using Oracle Cloud Infrastructure Compute Classic. See Viewing the Boot Log of an Instance in *Using Oracle Cloud Infrastructure Compute Classic*. Ignore information in this topic about the Compute API.

My custom storage volumes have become detached

Custom storage volumes you have added after creating an Oracle Java Cloud Service instance will become detached after restart operations.

Do not attach custom storage volumes to a service instance’s VMs. Any custom storage volumes are detached if the service instance is restarted.

If a service instance requires additional storage, add storage by scaling the service instance’s nodes as explained in Scale an Oracle Java Cloud Service Node.

My content changes on the Boot/OS volume are gone

The Boot/OS volume of any service instance provisioned before the mid-August 2015 update to Oracle Java Cloud Service is an ephemeral disk volume. Content added to an ephemeral Boot/OS volume does not persist if the service instance is restarted. To avoid the risk of data loss, do not add content to an ephemeral Boot/OS volume.

This restriction does not apply to more recent service instances. The Boot/OS volume of any service instance provisioned after the mid-August 2015 update to Oracle Java Cloud Service is persistent. Content added to a persistent Boot/OS volume is retained if the service instance is restarted.

For details of this volume, see About the Storage Volumes Attached to the WebLogic Server Nodes.

Problems with Connectivity

Problems might occur when you attempt to connect to an Oracle Java Cloud Service instance.

The following solutions apply to problems with connectivity to an Oracle Java Cloud Service instance.

Topics:

- My private key is lost or corrupted
- My connection to a VM is refused
- I received a hostname verification error when attempting to connect to Node Manager

My private key is lost or corrupted

Learn how to connect when your private key is lost or corrupted.

When you create an Oracle Java Cloud Service instance you must provide an SSH public key. You will be unable to establish an SSH connection to the VMs that comprise the service instance unless you provide the matching SSH private key, as described in Access a Node with a Secure Shell (SSH).

Perform the following steps:

1. Create a new pair of SSH keys.
2. Add the new SSH public key to your existing service instance. See Add an SSH Public Key.
3. SSH to the VMs in your service instance by using the new SSH private key.

My connection to a VM is refused

Be sure you are connecting to the VM as the `opc` user. Other OS users such as `oracle` and `root` cannot be used to establish a remote connection to a VM.

After successfully connecting to a VM as `opc`, you can switch to a different user. See Access a Node with a Secure Shell (SSH).

I received a hostname verification error when attempting to connect to Node Manager

When attempting to connect to the Node Manager using WLST, a hostname verification error is returned, similar to the following.

```
WLSTException: Error occurred while performing nmConnect : Cannot connect to
Node Manager. : Hostname verification failed:
@HostnameVerifier=weblogic.security.utils.SSLWSHostnameVerifier,
hostname=myjcs1-wls-1.
```

To disable hostname verification, use the following `-D` flag when invoking WLST:

```
java -Dweblogic.SSL.ignoreHostnameVerification=true weblogic.wlst
```

Problems with Database Connectivity When Upgrading the Infrastructure Schema Database

When you use the Upgrade Assistant to upgrade the infrastructure schema database during service instance upgrade, you might receive an error message and the upgrade fails. The problem is intermittent, so you might not receive this error message.

If there is a database connectivity problem, the following error message is generated:

```
Failed to establish a connection to data source <data_source_name> because
of the error:
UPGAST-00214: Unable to connect to database as the schema user <db_user>
```

If you receive this message, complete the following steps:

1. Verify that the host name and port you specified in the Upgrade Assistant are correct.
2. Ensure that the database is up and running.
3. Ensure that the database is configured for network access.
4. Retry the operation.

Problems with the Node Manager

Problems may occur if you are trying to restart the Administration Server through the Node Manager.

When you check to see whether the Node Manager is running, you could find that it is not running.

When I try to restart the Administration Server, I discover that the Node Manager is not running

For information about restarting the Administration Server through the Node Manager, see Use WLST Commands to Start the Administration Server.

To restart the Node Manager:

1. Use an SSH client of your choice to access the VM of the Administration Server. If you do not have an SSH client on Windows, you can use PuTTY to access the VM by establishing an SSH tunnel.

   If you are not automatically logged in as user `opc`, log in accordingly.
2. In the command window, change to user `oracle`.

   ```
   sudo su - oracle
   ```
3. Change directories to where `startNodeManager.sh` exists.

   If you are using Oracle Fusion Middleware 11.1.1.7, the location is:

   ```
   /u01/app/oracle/middleware/wlserver_10.3/server/bin
   ```

   If you are using Oracle Fusion Middleware 12.2.1, the location is:

   ```
   /u01/data/domains/<domain_name>/bin
   ```

   For example:

   ```
   cd /u01/data/domains/OurServi_domain/bin
   ```
4. Start the Node Manager:

   ```
   nohup startNodeManager.sh
   ```
5. Check to see that the Node Manager is running:

   ```
   ps -ef | grep NodeManager
   ```

   You should receive messages showing that the Node Manager is running.
6. (Optional) If you have more than one host in your Oracle Java Cloud Service instance, you must restart the Node Manager on each host.

   a. SSH to the second host:

      ```
      ssh <hostname>
      ```

      For example:

      ```
      ssh ourserviceinstance-wls-2
      ```

      You can find the hostname on the Oracle Java Cloud Service Instance page in the Oracle Java Cloud Service Console.

   b. Change directories to where `startNodeManager.sh` exists.

      If you are using Oracle Fusion Middleware 11.1.1.7, the location is:

      ```
      /u01/app/oracle/middleware/wlserver_10.3/server/bin
      ```

      If you are using Oracle Fusion Middleware 12.2.1, the location is:

      ```
      /u01/data/domains/<domain_name>/bin
      ```

For example:

```
cd /u01/data/domains/OurServi_domain/bin
```

c. Start Node Manager:

```
nohup startNodeManager.sh
```

d. Check to see whether the Node Manager is running:

```
ps -ef | grep NodeManager
```

You should receive messages showing that the Node Manager is running.

e. Exit the second host:

```
exit
```

7. Exit the `oracle` session:

```
exit
```

8. Exit out of the command window:

```
exit
```

Problems with a Database Deployment

Problem related to the database deployment used by Oracle Java Cloud Service can occur.

Creating an opss datasource fails

An attempt to create an opss datasource can fail because the database deployment's opss user account is locked.

To unlock the opss user account:

1. Log in to the database deployment's VM by using the private key.

```
ssh -i private-key opc@ip-address-of-db-vm
```

2. Change to user `oracle`.

```
cd $ORACLE_HOME/bin
```

3. Start sqlplus.

```
./sqlplus
```

4. Log in using the `system` user, and enter the password.

```
Enter user-name: system
Enter password: system_user_password
```

5. Unlock the account.

```
ALTER USER schema_prefix_opss ACCOUNT UNLOCK;
```

6. Change the password.

   ```
   ALTER USER schema_prefix_opss INDENTIFIED BY new_password;
   ```

7. Exit sqlplus.

   ```
   exit
   ```

Problems with Connection validation when provisioning an Oracle Java Cloud Service-Virtual Image instance with Oracle Database Cloud Service 12.1 VI

A connection validation error may occur when you attempt to provision an Oracle Java Cloud Service —Virtual Image instance with Oracle Database Cloud Service—Virtual Image 12.1.

Provisioning fails and gives this error message:

```
Could not connect to Database Cloud Service. Error Details: [No match
found]
```

This error occurs because the database deployment has been created without a domain name.

To work around this issue:

1. Log on to the database deployment's VM:

   ```
   ssh -i <private-key> opc@ip-address-of-db-vm
   ```

2. Change to user `oracle`.

   ```
   sudo -s -u oracle
   ```

3. Connect to the PDB as `sysdba`.

   ```
   $ORACLE_HOME/bin/sqlplus sys/<password>@"<host>:<port>/<pdb name> as
   sysdba
   ```

4. Create a service using a domain name.

   ```
   exec dbms_service.create_service('<pdb name>.<domain name>', '<pdb
   name>.<domain name>');
   ```

   For example, the domain name could be:

   ```
   <identity domain>.oraclecloud.internal
   ```

5. Start the service.

   ```
   exec dbms_service.start_service('<pdb name>.<domain name>');
   ```

6. Using the Oracle Java Cloud Service provisioning wizard to provision an instance, enter a connect string.

   For example:

   ```
   1521/<pdb name>.<domain name>
   ```

7. Continue to provision the Oracle Java Cloud Service instance.

Problems with Transparent Data Encryption Wallet Error when Provisioning an Oracle Java Cloud Service-Virtual Image Instance with Oracle Database Cloud Service 11g Virtual Image

A Transparent Data Encryption wallet error can occur when you attempt to provision an Oracle Java Cloud Service VI instance with Oracle Database Cloud Service 11g VI.

The error message you receive is:

```
Transparent Data Encryption wallet is not open. Please open the wallet and try
again.
```

To configure the Transparent Data Encryption wallet:

1. Create a keystore.
   a. Edit the `ORACLE_HOME/network/admin/sqlnet.ora` file, and add the following entry.

      ```
      ENCRYPTION_WALLET_LOCATION =
      (SOURCE =(METHOD = FILE)(METHOD_DATA) =
      (DIRECTORY = <wallet directory>/$ORACLE_SID/encryption_keystore/)
      ```

   b. Create the directory to hold the keystore.

      ```
      $mkdir -p <wallet directory>/$ORACLE_SID/encryption_keystore
      ```

   c. Run `sql` and enter the following command:

      ```
      SQL > alter system set encryption key identified by "<password>";
      ```

2. Use `orapki` to create an autologin wallet.

   ```
   orapki wallet create -wallet <wallet directory>/$ORACLE_SID/
   encryption_keystore/  -auto_login
   ```

3. Continue to provision the Oracle Java Cloud Service VI service.

Problems Opening the WebLogic Server Administration Console from Fusion Middleware Control

You can experience problems opening the WebLogic Server Administration Console from Fusion Middleware Control.

You can use the WebLogic Server Administration Console and Fusion Middleware Control to administer Oracle Java Cloud Service instances. If you attempt to open the WebLogic Server Administration Console from the Fusion Middleware Control Console, the console will not open and you will receive an error message:

```
The Host is not resolvable. Most commonly this is due to mistyping the URL
in
```

```
the browser bar. Please verify the spelling and that the site exists
and hit refresh.
```

The problem occurs three ways.

From the Deployments tile:

1. Click on the Deployments tile.
2. Click the name of your deployed application.
3. From the Domain Application Deployment drop-down menu, select Administration — General Settings.
4. Select the Instrumentation tab.
5. In “To configure Instrumentation, use the WebLogic Server Administration Console,” click **Weblogic Server Administration Console**.

   The error message appears in a new browser tab.

From the WebLogic Domain drop-down menu:

- From the WebLogic Domain drop-down menu, select WebLogic Server Administration Console.

  The error message does not appear, but neither does the WebLogic Service Administration Console.

When administering a security realm from the WebLogic Domain drop-down menu:

1. From the WebLogic Domain drop-down menu, select Security — Security Realms.
2. Select **myrealm**.
3. Select Settings for Security Realm.
4. Click **WebLogic Server Administration Console**.

   The error messaged appears in a new browser tab.

By design, Fusion Middleware Control has a URL composed of the hostname and HTTP port 7001 for the console. In the Oracle Java Cloud Service environment, only HTTPS port 7002 is enabled and accessible because it is a secure port. Additionally, the Administration Server VM host is not DNS resolvable to its IP address because the IP address is a public NAT IP address.

Use the HTTPS protocol, NAT IP address instead of host name, and port 7002 to access the console, for example:

```
https://198.51.100.1:7002/console
```

Security Checkup Tool Warnings

Learn about the security check warnings that are displayed in the Oracle WebLogic Server Administration console and how to troubleshoot them.

At the top of the WebLogic Server Administration console, the message `Security warnings detected. Click here to view the report and recommended remedies` is displayed for Oracle Java Cloud Service instances created after July 20, 2021, or the instances on which the July 2021 PSUs are applied.

When you click the message, a list of security warnings are displayed as listed in the following table.

Note:

The SSL host name verification and the umask warnings are displayed for existing Oracle Java Cloud Service instances created before release 21.3.2 (August 26, 2021).

The warning messages listed in the table are examples.

Security Warnings

Warning Message	Resolution
`Tunneling is enabled on server channel channel-dep. Allowing T3 or IIOP to be tunneled on a server channel may allow deserialization of specially crafted, malicious serialized objects that can potentially cause denial of service.` **Note:** This warning is displayed only for existing Oracle Java Cloud Service instances created before release 22.1.1 (January 31, 2022) on which the October 2021 PSUs are applied.	Disable tunneling on `channel-dep` server channel. See Disable Tunneling on Server Channel.

Warning Message	Resolution
`SSL hostname verification is disabled by the SSL configuration.`	Review your applications before you make any changes to address these SSL host name security warnings. For applications that connect to SSL endpoints with a host name in the certificate, which does not match the local machine's host name, the connection fails if you configure the BEA host name verifier in Oracle WebLogic Server. See Using the BEA Host Name Verifier in Administering Security for Oracle WebLogic Server. For applications that connect to Oracle provided endpoints such as Oracle Identity Cloud Service (for example,`*.identity.oraclecloud.com`), the connection fails if you did not configure the wildcard host name verifier or a custom host name verifier that accepts wildcard host names. If you are not sure of the SSL configuration settings you should configure to address the warning, Oracle recommends that you configure the wildcard host name verifier. See Configure the Wildcard Host Name Verifier. For existing Oracle Java Cloud Service instances (created before July 20, 2021), to address this SSL host name verification warning, in addition to configuring the host name verifier, you must edit the `startup.properties` file for administration server instances and restart the managed server instances. See Configure the Wildcard Host Name Verifier, Update Administration Server Startup Properties, and Restart Managed Server Using Node Manager.
`Production mode is enabled but the file or directory /u01/data/domains/<`*`domain_name`*`>/servers/<`*`domainname`*`>_adminserver/security/boot.properties is insecure`	Run the following command in the administration server as `oracle` user: `chmod -R 750 /u01/data/domains/<`*`domain_name`*`>/servers/<`*`adminserver_name`*`>/security/` **Note**: This permission setting is applicable only for existing Oracle Java Cloud Service instances created before release 21.3.2 (August 26, 2021) on which the July 2021 PSUs are applied.
`Remote Anonymous RMI T3 or IIOP requests are enabled. Set the RemoteAnonymousRMIT3Enabled and RemoteAnonymousRMIIIOPEnabled attributes to false.`	Disable the anonymous RMI T3 and IIOP requests in the WebLogic Server Administration Console as soon as possible unless your deployment requires anonymous T3 or IIOP (not typical). See Disable Remote Anonymous RMI T3 and IIOP Requests. **Note**: These attribute settings are also applicable to Oracle Traffic Director, but only for service instances running Oracle Traffic Director 12.2.1.4.

After you address the warnings, you must click **Refresh Warnings** to see the warnings removed in the console.

For Oracle Java Cloud Service instances created after July 20, 2021, though the java properties to disable anonymous requests for preventing anonymous RMI access are configured, the warnings still appear. This is a known issue in Oracle WebLogic Server.

If you want to perform anonymous RMI requests, you must set the java properties for anonymous RMI T3 and IIOP requests. See Set the Java Properties.

Problems Accessing Applications Via a Custom Web Server

You might have trouble accessing applications through an Oracle HTTP Server or similar software.

If you installed Oracle HTTP Server or similar software on a node in your service instance, you might not be able to access this software using ports 80 or 443. The operating system on a node is typically configured to intercept and redirect incoming traffic on these ports to the WebLogic Server listen ports, such as ports 9073 and 9074. To view these network policies, run the `iptables` command as the root user:

```
sudo iptables --numeric -t nat -L
```

Use the `-D` option in `iptables` to delete the redirect policy that is causing your access issue.

How can I Fine-Tune Performance?

Performance issues might occur when you are using the Oracle Java Cloud Service.

If you are experiencing issues with the performance of your Oracle Java Cloud Service instance, or if you simply want to fine-tune performance, you can find information in the following documentation:

- If you selected Oracle WebLogic Server version 12.2.1 when you provisioned your service, see Top Tuning Recommendations in *Tuning Performance of Oracle WebLogic Server*.
- If you selected Oracle WebLogic Server version 11.1.1.7 when you provisioned your service, see Top Tuning Recommendations in *Performance and Tuning for Oracle WebLogic Server*.

A

Oracle Fusion Middleware Products Certified on Oracle Java Cloud Service

Oracle Fusion Middleware is a portfolio of products that enable you to build and deploy enterprise applications for web collaboration, content management, data integration, and portals. Certified Oracle Fusion Middleware products can be provisioned on your Oracle Java Cloud Service instances.

> **Note:**
>
> Only the Oracle Fusion Middleware products that are listed in this section are certified on Oracle Java Cloud Service. The products that are not listed here are not certified.
>
> If the installation of any product on your Oracle Java Cloud Service instance modifies the `MW_HOME` directory, then the patching capability within Oracle Java Cloud Service can't be used to patch the instance. If you try to patch the instance, the precheck operation fails.

Contents

- Certified Oracle Fusion Middleware Products
- #GUID-B4A53E27-B580-492D-80B0-F909F470C7D4/CERTIFIEDCLOUDSKUS-B0AEC881
- Processor License Ratio to Oracle Compute Unit (OCPU)
- Deployment Tutorials

Certified Oracle Fusion Middleware Products

The following table lists the Oracle Fusion Middleware product components and versions that are certified on Oracle Java Cloud Service. You can quickly deploy your existing on-premises licenses (perpetual or term) for these products. There is no formal license migration process. For detailed installation and setup instructions, see Deployment Tutorials.

> **Note:**
>
> Some of these products are certified on only the Oracle Java Cloud Service Virtual Image service level, which is not available for universal credits subscriptions.

Certified Oracle Fusion Middleware Product	Certified Service Level
Oracle WebCenter Portal 12.2.1.2.0	Oracle Java Cloud Service

Certified Oracle Fusion Middleware Product	Certified Service Level
Oracle WebCenter Content 12.2.1.2.0	Oracle Java Cloud Service
Oracle WebCenter Sites 12.2.1.2.0	Oracle Java Cloud Service

Certified Oracle Database Cloud Service Editions

Oracle Java Cloud Service–Virtual Image requires either Oracle Database Cloud Service or Oracle Database Cloud Service–Virtual Image.

For Oracle Data Integrator Enterprise Edition, select one of the following Oracle Database Cloud Service editions:

- Oracle Database Cloud Service Enterprise Edition
- Oracle Database Cloud Servicee Enterprise Edition - High Performance
- Oracle Database Cloud Service Enterprise Edition - Extreme Performance

Processor License Ratio to Oracle Compute Unit (OCPU)

When installing and deploying perpetual or term on-premises licenses in the Oracle Cloud, you must have a sufficient number of licenses to cover your use. For information on the ratio of Processor licenses to Oracle Compute Units (OCPU), see the document titled Oracle Processor Core Factor Table.

> **Note:**
>
> For Oracle Data Integrator, only the processor(s) where the data transformation processes are executed must be counted for the purposes of determining the number of licenses required.

Deployment Tutorials

Refer to the following tutorials for information about provisioning a certified Oracle Fusion Middleware product on Oracle Java Cloud Service.

- Provisioning Oracle Data Integrator Cloud Service
- Provisioning Oracle Data Integrator on Oracle Java Cloud Service
- Provisioning Oracle WebCenter Portal Cloud Service in the Development Topology
- Provisioning Oracle WebCenter Portal Cloud Service in the EDG Topology
- Provisioning Oracle WebCenter Sites on Oracle Java Cloud Service
- Provisioning Oracle SOA Suite on Oracle Java Cloud Service
- Provisioning Oracle BPM Suite on Oracle Java Cloud Service
- Provisioning Oracle Service Bus on Oracle Java Cloud Service
- Provisioning Oracle Enterprise Data Quality on Oracle Java Cloud Service
- Provisioning Oracle Business Intelligence Publisher 12*c* on Oracle Java Cloud Service
- Provisioning Oracle Business Intelligence Publisher 11*g* on Oracle Java Cloud Service

B

Patches Included in Oracle Java Cloud Service

Apply the patches listed in the following section to your Oracle Java Cloud Service instance.

See About Patching and Rollback for the different kinds of patches that are available.

> **Tip:**
>
> For a list of new features and enhancements that were added recently to improve your Oracle Java Cloud Service experience, see What's New for Oracle Java Cloud Service.

The following table shows the patches that are found in each release of Oracle Java Cloud Service. The patches are displayed on the Patching page of the Oracle Java Cloud Service console. See *When to Apply Patches* in About Patching and Rollback.

Java Cloud Service Version	Patches	Oracle Java Cloud Service Release	Patch List
12.2.1.4.220207	opatch: • 33618954 - OWSM bundle patch 12.2.1.4.211129 • 32772437 - FMW Platform 12.2.1.4.0 SPU for April 2021 CPU • 33723124 - FMW third-party bundle patch 12.2.1.4.220104 • 30385564 - Oracle XML Developers Kit patch • 32784652 - OPSS patch for April 2021 • 31544353 - ADR for WebLogic Server 12.2.1.4.0 JULY CPU 2020 • 33591019 - Coherence 12.2.1.4 Cumulative Patch 12 (12.2.1.4.12) • 33697227 - ADF bundle patch 12.2.1.4.211221 • 33727616 - WLS Patch Set update 12.2.1.4.220105 • 33791665 - log4j v1 for CVE-2021-4104, CVE-2022-23302, CVE-2022-23305, CVE-2022-23307 • 33735326 - log4j v2 for CVE-2021-44832 • 33275353 - WSM template application fails with free tier ATP database	22.1.1	January 2022 PSUs

Java Cloud Service Version	Patches	Oracle Java Cloud Service Release	Patch List
12.2.1.3.220207	opatch: • 32772477 - FMW Platform 12.2.1.3.0 SPU For April CPU 2021 • 32651962 - FMW common third-party SPU 12.2.1.3.0 for April 2021 CPU • 26394536 - Managed server going into failed state due to deadlock • 31544340 - ADR for WebLogic Server 12.2.1.3.0 JULY CPU 2020 • 30186876 - 12.2.1.4.0 OIM: SOA composer deployment fails during SOA_Server startup (intermittent) • 30252137 - Xpath-functions current-datetime should consider DST • 27263211 - EM configuration fails in FOH (12.2.1.3.1) • 31464643 - Merge request on top of 12.2.1.3.0 for bugs 29011959 and 30385564 • 33591009 - Coherence 12.2.1.3 Cumulative Patch 17 (12.2.1.3.17) • 33590225 - ADF bundle patch 12.2.1.3.211119 • 33699205 - WLS patch set update 12.2.1.3.211222 • 29840258 - RCU 12.2.1.3 fails due to bad password generation for FMW registry user • 28659321 - New exception showing up in managed server logs • 32397127 - OPSS patch for April 2021 • 26045997 - Enabling driver fan without running ONS daemons causes connect request error • 27608287 - Upgrade assistant readiness received error when schema prefix is mixed case • 24738720 - 12.2.1.2 EM template can't be added with JRF Cloud template added • 33618953 - OWSM bundle patch 12.2.1.3.211129 • 33791665 - log4j v1 for CVE-2021-4104, CVE-2022-23302, CVE-2022-23305, CVE-2022-23307 • 33735326 - log4j v2 for CVE-2021-44832 • 18345580 - XSLT grouping using muenchian method does not work at runtime • 26355633 - PoolDataSource fails connect to 12c non container DB using service name • 26287183 - PSR:PERF:WLS 12.2.1.3 : ~49% regression in JDBC benchmarks • 26261906 - Merge request on top of 12.2.0.1.0 for bugs 24811916, 25232931, and 25559137	22.1.1	January 2022 PSUs

Java Cloud Service Version	Patches	Oracle Java Cloud Service Release	Patch List
	• 26051289 - Invalid arguments while using Preparestatement (string, string[]) with WE8ISO8859		
12.1.3.0.220207	opatch: • 33494824 - WLS patch set update 12.1.3.0.220118 • 22513703 - PasteConfig failed while updating domain template for CoherenceCluster props • 31544363 - ADR for WebLogic Server 12.1.3.0.0 JULY CPU 2020 • 29676263 - OWSM bundle patch 12.1.3.0.190503 • 30100252 - ADF bundle patch 12.1.3.0.191015 • 28172680 - OPSS bundle patch 12.1.3.0.180814 • 24963946 - Tracking bug for diagnostic patch of IDCS integration • 21680406 - Log files rotated by node manager are not getting closed by the file handler • 20960881 - Integrity enabled, sql fails, get IO error instead of ORA error • 20741228 - 12.1.3 WLS upgrade to 12.1.0.2 JDBC release • 19030178 - Add TLSV1.1 and TLSV1.2 in JDBC Thin • 29879150 - Merge request on top of 12.1.3.0.0 for bugs 20629366 25237184 • 33385024 - OPatch Patch for WLS 12.1.3 PSU driver updates • 33791665 - log4j v1 for CVE-2021-4104, CVE-2022-23302, CVE-2022-23305, CVE-2022-23307	21.4.2 22.1.1	January 2022 PSUs

The following table lists the older patches for the Oracle Java Cloud Service releases.

Java Cloud Service Version	Patches
12.2.1.4	opatch: • 33671996 - WebLogic overlay patch for October 2021 PSU for CVE-2021-44228 and CVE-2021-45046 • 31544353 - ADR for WebLogic Server • 33416868 - WLS patch set update 12.2.1.4.210930 • 33313802 - ADF bundle patch 12.2.1.4.210903 • 33286160 - Coherence 12.2.1.4 Cumulative Patch 11 (12.2.1.4.11) • 32880070 - FMW common third-party SPU 12.2.1.4.0 for April 2021 CPU • 32905339 - OWSM bundle patch 12.2.1.4.210520 • 33059296 - WLS patch set update 12.2.1.4.210629 • 33084721 - ADF bundle patch 12.2.1.4.210706 • 32973297 - Coherence 12.2.1.4 Cumulative Patch 10 (12.2.1.4.10) • 122148 (32581859) - WebLogic Server Coherence patch for April 2021 • 32684757 - ADF/JDev bundle patch for April 2021 • 32698246 - WebLogic Server CPU patch April 2021
12.2.1.3	opatch: • 33313934 - ADF bundle patch 12.2.1.3.210903 • 33671996 - WebLogic overlay patch for October 2021 PSU for CVE-2021-44228 and CVE-2021-45046 • 33313934 - ADF bundle patch 12.2.1.3.210903 • 32917014 - OWSM bundle patch 12.2.1.3.210524 • 31544340 - ADR for WebLogic Server 12.2.1.3.0 July CPU 2020 • 33286132 - Coherence 12.2.1.3 Cumulative Patch 16 (12.2.1.3.16) • 33412599 - WLS patch set update 12.2.1.3.210929 • 32973279 - Coherence 12.2.1.3 Cumulative Patch 15 (12.2.1.3.15) • 33064699 - WLS patch set update 12.2.1.3.210630 • 32997257 - ADF bundle patch 12.2.1.3.210614 • 32656495 - ADF/JDEV patch for April 2021 • 31944067 - OWSM bundle patch 12.2.1.3.200928 • 1221313 (32581838) - Coherence patch for April 2021 • 32697734 - WLS CPU patch for April 2021
12.1.3	opatch: • 33172866 - WLS patch set update 12.1.3.0.211019 • 31544363 - ADR for WebLogic Server 12.1.3.0.0 July CPU 2020 • 32832660 - WLS patch set update 12.1.3.0.210720

Java Cloud Service Version	Patches
12.2.1.2	opatch: • 26910516 - createStatement() creates new threads. Hundreds remain live in WLS • 19795066 - ORA-904 calling Oracledatabasemetadata.getcolumns against old DB • 19154304 - JDBC: Retry_count does not retry when service down as required • 21663638 - Fusion Apps functionality failure due to type mismatch Enqueue/Dequeue message • 22754279 - Make authenticationservice compatible for JDK9 environment • 19030178 - Add TLSV1.1 and TLSV1.2 in JDBC Thin • 19632480 - Oracledatabasemetadata.gettables cursor connection leak • 18459080 - TNS-12599 in alert.log after enabling network encryption with JDBC Thin • 18905788 - Update subscriber creation to be specific to each ONS object • 19002423 - Java.lang.arrayindexoutofboundsexception: -1 loading data using executebatch
10.3.6	bsu: • 21Y4 - WLS patch set update 10.3.6.0.211019 • CW7X - ADR for WebLogic Server • I1EV - WebLogic Server patch • SY38 - WebLogic Server patch • TTGM - Update certgen and related artifacts to support new demo certs • 3NVW - WLS patch set update 10.3.6.0.210720 opatch: • 27214515 - ADF patch • 27846936 - OPSS bundle patch • 17617649 - FMW bug fixes • 22577934 - FMW Control patch • 22852289 - FMW bug fixes • 33172858 [21Y4] - WLS patch set update 10.3.6.0.211019 • 13964737 [TTGM] - Update certgen and related artifacts to support new demo certs • 32832785 [3NVW] - WLS patch set update 10.3.6.0.210720

C

Effect of Lifecycle and Administration Operations on Billing

This topic does not apply to Oracle Cloud at Customer.

Some of the administration and lifecycle operations that you run for an Oracle Java Cloud Service instance affect the billing for the infrastructure resources that the instance uses.

The following table summarizes the effect that each administration and lifecycle operation has on billing. The first column lists the operations. The other columns show the effect of a given operation on the billing for a specific infrastructure resource.

- **Down**: After the operation is completed, billing for the resource will decrease or stop.
- **Up**: Billing for the resource will start, resume, or increase after the operation is completed.
- **No effect**: The operation has no effect on billing for that resource.

Operation	OCPUs	Block Storage	Object Storage
Create an instance	Up	Up	No effect
Delete the instance	Down	Down	Down[1]
Stop the instance	Down	No effect	No effect
Start the instance	Up	No effect	No effect
Restart the instance	No effect	No effect	No effect
Stop a node	Down	No effect	No effect
Scale out a cluster	Up	Up	No effect
Scale in a cluster	Down	Down	No effect
Scale up a node	Up	No effect	No effect
Scale down a node	Down	No effect	No effect
Add block storage	No effect	Up	No effect
Take a backup	No effect	No effect	Up
Take a colocated snapshot	No effect	Up	No effect
Add Oracle Traffic Director as the load balancer	Up	Up	No effect
Add an Oracle Traffic Director node	Up	Up	No effect
Remove an Oracle Traffic Director node	Down	Down	No effect

[1] When using the REST API, if you opt for automatic backup of the instance before deletion, then that final backup will be retained in object storage.

D

Migrate Applications to Oracle Java Cloud Service with AppToCloud

This topic applies only to Oracle Cloud at Customer.

Oracle’s AppToCloud tools allow you to export an existing domain configuration and Java applications, and to provision a new Oracle Java Cloud Service instance with the same domain resources and applications.

AppToCloud saves you the tedious and error-prone task of manually migrating all of your existing Oracle WebLogic Server environments to Oracle Cloud. The AppToCloud tools quickly capture your on-premises domain configurations and applications. You then can direct Oracle Java Cloud Service to create one or more service instances based on these exported artifacts.

Topics:

- Typical Workflow for Migrating Applications to Oracle Java Cloud Service with AppToCloud
- Prerequisites for Using AppToCloud
- AppToCloud Considerations and Limitations
- Migrate an Oracle Database to Oracle Cloud for Oracle Java Cloud Service
- Install the On-Premises AppToCloud Tools
- Check the Health on an On-Premises WebLogic Domain
- Export an On-Premises WebLogic Domain
- Create a Service Instance with AppToCloud
- Import Applications into a Service Instance
- Recreate On-Premises Domain Resources
- AppToCloud Command Reference

Typical Workflow for Migrating Applications to Oracle Java Cloud Service with AppToCloud

This topic applies only to Oracle Cloud at Customer.

Oracle’s AppToCloud tools enable you to quickly migrate existing Java applications and their supporting Oracle WebLogic Server resources to Oracle Java Cloud Service. The process consists of several tasks.

These migration tasks fall into two main categories, *On-Premises* and *Cloud*:

- The on-premises tasks involve archiving your existing Oracle WebLogic Server environment and applications and uploading the files into Oracle Cloud.
- The cloud tasks involve creating an Oracle Java Cloud Service service instance and automatically provisioning it with the same resources and applications as your on-premises environment.

Topics:

- On-Premises Tasks
- Cloud Tasks

On-Premises Tasks

This topic applies only to Oracle Cloud at Customer.

Learn about the on-premises phase of the migration process.

Task	Description	More Information
Verify the prerequisites	Ensure that your existing Oracle WebLogic Server domain meets the requirements of the AppToCloud tools. Understand the consequences and limitations of the migration process.	Prerequisites for Using AppToCloud AppToCloud Considerations and Limitations
Install the tools	Download and install the AppToCloud tools on the on-premises machine hosting your domain's Administration Server.	Install the On-Premises AppToCloud Tools
Perform a health check	Use the AppToCloud Client or command line tools to validate your on-premises Oracle WebLogic Server domain and applications. This process ensures that your domain and its applications are in a healthy state. These tools also identify any WebLogic Server features in your domain that the AppToCloud framework cannot automatically migrate to Oracle Java Cloud Service. This step is mandatory. It cannot be skipped.	Check the Health on an On-Premises WebLogic Domain
Export the domain to Oracle Cloud	Use the AppToCloud Client or command line tools to capture your on-premises WebLogic Server domain and applications as a collection of files. These files are uploaded by the tools to a storage container that you have previously created in Oracle Cloud Infrastructure Object Storage Classic.	Export an On-Premises WebLogic Domain

Cloud Tasks

This topic applies only to Oracle Cloud at Customer.

Learn about the cloud phase of the migration process.

Task	Description	More Information
Verify the prerequisites	Ensure that your Oracle Cloud account meets the requirements for the AppToCloud infrastructure, including storage and database requirements.	Prerequisites for Using AppToCloud
Migrate the databases to Oracle Cloud	Use standard Oracle Database tools to move existing relational schemas to one or more databases in Oracle Cloud.	Migrate an Oracle Database to Oracle Cloud for Oracle Java Cloud Service
Create an Oracle Java Cloud Service service instance.	Create a service instance and select the AppToCloud option. As part of the creation process, you provide the location of the AppToCloud artifacts on cloud storage.	Create a Service Instance with AppToCloud
Import your applications into the service instance.	After the Oracle Java Cloud Service service instance is running, import the AppToCloud artifacts. Oracle Java Cloud Service updates the service instance with the same resources and applications as your exported source environment. The import operation can only be performed on a new and unmodified service instance. Do not perform any scaling operations, modify the domain configuration or otherwise change the service instance prior to this step.	Import Applications into a Service Instance
Recreate resources if necessary	Some Oracle WebLogic Server features are not currently supported by the AppToCloud tools. These features must be configured manually after provisioning your Oracle Java Cloud Service instance. Use the same Oracle tools to perform these modifications that you originally used to configure the source environment: • WebLogic Server Administration Console • Fusion Middleware Control • WebLogic Scripting Tool (WLST)	Recreate On-Premises Domain Resources

Prerequisites for Using AppToCloud

This topic applies only to Oracle Cloud at Customer.

Your on-premises WebLogic Server environment and your Oracle Java Cloud Service environment must meet certain prerequisites in order to use the Oracle AppToCloud tools.

Topics

- Source WebLogic Server Domain
- Destination Oracle Java Cloud Service

Source WebLogic Server Domain

This topic applies only to Oracle Cloud at Customer.

Verify that your source domain meets the following requirements.

- The domain must be Oracle WebLogic Server version 10.3.3 or later.

  > **Note:**
  >
  > An Oracle Java Cloud Service created with AppToCloud will always be provisioned with Oracle WebLogic Server 12*c*, even if your source domain is running 11*g*.

- All servers in the domain should be running and in a healthy state.

  The AppToCloud Export tool will fail if the Health Check tool is unable to connect to a server.
- The domain configuration must not be locked or have an active edit session.
- The domain cannot be based on a template that uses Java Required Files (JRF). This includes domains running Oracle Fusion Middleware products. The Restricted JRF template is supported, however.

  > **Note:**
  >
  > While the source domain cannot be JRF-enabled, all Oracle Java Cloud Service instances are JRF-enabled.

- The domain must not include any domain partitions.
- If the domain contains multiple managed servers and no clusters, the same resources and applications must be targeted to all of the managed servers. The Export tool prompts the user to add the managed servers to a new cluster in the exported domain configuration.

- Any Java EE applications to export must be in the active deployment state. They cannot be in the admin state.
- All files and directories to export must be accessible from the file system of the Administration Server, including:
  - Java EE applications
  - Deployment plans
  - Additions to the server `CLASSPATH`
  - Contents of *`DOMAIN_HOME`*`/lib`

Destination Oracle Java Cloud Service

This topic applies only to Oracle Cloud at Customer.

Perform prerequisite tasks in Oracle Cloud to support migration.

Perform all required tasks described in the topic Before You Begin with Oracle Java Cloud Service. You do not need to create an Oracle Java Cloud Service instance prior to performing the migration process.

The AppToCloud tools can create a storage container for you in Oracle Cloud Infrastructure Object Storage Classic in which to store the generated AppToCloud artifacts. You can use the same storage container for both AppToCloud and service instance backups, or create separate containers. Alternatively, if you prefer to manually create a storage container, see Creating Containers in *Using Oracle Cloud Infrastructure Object Storage Classic*.

> **Note:**
>
> AppToCloud does not currently support Oracle Cloud Infrastructure Object Storage buckets. You must use an Oracle Cloud Infrastructure Object Storage Classic container.

If your source domain contains multiple clusters, AppToCloud can add all of these clusters to a single Oracle Java Cloud Service instance, but only if both of these conditions are true:

- The service instance is created in an Oracle Cloud Infrastructure Classic region.
- Your Oracle Cloud account includes Oracle Identity Cloud Service.

Alternatively, AppToCloud can export a specific cluster from your source domain.

AppToCloud Considerations and Limitations

This topic applies only to Oracle Cloud at Customer.

Consider the following scenarios before using AppToCloud to migrate your on-premises Oracle WebLogic Server domain to Oracle Java Cloud Service.

Topics:

- Database Services

- Multiple Clusters
- 11g Applications
- File System Locations
- Server Classpath
- Exploded Archive Deployments

Database Services

This topic applies only to Oracle Cloud at Customer.

When you create a service instance with AppToCloud, you associate it with one or more existing relational databases in Oracle Cloud.

You must select one infrastructure schema database. Oracle Java Cloud Service provisions this database with the required schemas for Oracle Java Cloud Service. For a list of supported database services, see Database.

You must also associate each data source in your original WebLogic Server domain with an existing Oracle Database Cloud Service deployment. You cannot assign a data source to other database services in Oracle Cloud while creating a service instance with AppToCloud. However, after creating the service instance, you can manually update the generated data sources and configure them to use any database.

Multiple Clusters

This topic applies only to Oracle Cloud at Customer.

If your source domain contains multiple clusters, choose from one of these options.

- If your Oracle Cloud account includes Oracle Identity Cloud Service and an Oracle Cloud Infrastructure Classic region, AppToCloud can migrate all of the clusters to a single Oracle Java Cloud Service instance.
- Use AppToCloud to export a single cluster from the domain. You can repeat this migration process and select a different cluster to export each time. For every cluster that you export from the source domain, a separate Oracle Java Cloud Service instance and domain are created.

> **Note:**
>
> AppToCloud does not support the migration of a source domain that contains more than **eight** clusters.

11g Applications

This topic applies only to Oracle Cloud at Customer.

An Oracle Java Cloud Service created with AppToCloud is always provisioned with WebLogic Server 12c, even if your source domain is running 11g.

WebLogic Server is generally backwards compatible and supports applications built with earlier versions of the server and Java Enterprise Edition specification, but there may be exceptional cases that require you to modify your 11g application in order for it to function properly in 12c. Oracle recommends that you perform extensive testing of 11g applications after importing them into Oracle Java Cloud Service.

You can also refer to one of these publications to help identify potential compatibility issues:

- WebLogic Server Compatibility with Previous Releases in *Upgrading Oracle WebLogic Server (12.2.1)*
- WebLogic Server Compatibility with Previous Releases in *Upgrading Oracle WebLogic Server (12.1.2)*

File System Locations

This topic applies only to Oracle Cloud at Customer.

The AppToCloud tools locate and export the following files and directories from your source domain.

- Application deployments
- Custom entries in the server's `CLASSPATH`
- Custom files and directories under the domain's home folder (`DOMAIN_HOME`)

After creating an Oracle Java Cloud Service instance with AppToCloud, all of these custom files and directories are placed on the server's file system at the location `DOMAIN_HOME/a2c`. The original file system on your source machine is not preserved. This may impact applications that have a dependency on a specific directory structure.

Additionally, the AppToCloud tools do not preserve other non-default directories used in your source domain configuration, such as:

- **Paging Directory** for a JMS Server or SAF Agent
- **Log File Rotation Directory** for a JMS Server or SAF Agent

These configuration settings are reset to their default file system locations in the Oracle Java Cloud Service instance.

Server Classpath

This topic applies only to Oracle Cloud at Customer.

The AppToCloud tools export all folders and files found in the server's `CLASSPATH`, including any subdirectories at these locations.

Verify that any custom locations in your `CLASSPATH` include only those files that you want to export and upload to Oracle Cloud.

Exploded Archive Deployments

This topic applies only to Oracle Cloud at Customer.

Oracle WebLogic Server supports application and shared library deployments that are packaged as either archive files or exploded archive directories.

For applications and libraries that are packaged as directories, the AppToCloud tools export all files and subdirectories found at these deployment locations. Verify that these directories include only those files that you want to export and upload to Oracle Cloud.

Migrate an Oracle Database to Oracle Cloud for Oracle Java Cloud Service

This topic applies only to Oracle Cloud at Customer.

Migrate your on-premises Oracle Databases to Oracle Cloud using Oracle Database Backup Cloud Service.

An Oracle Java Cloud Service instance requires an existing database to host the Oracle Fusion Middleware schema. This schema is provisioned automatically when you create a new service instance. Supported services for the infrastructure database schema include Oracle Database Cloud Service.

Your Java applications likely use additional on-premises databases. Oracle recommends that you migrate these application databases to Oracle Cloud as well. When you create an Oracle Java Cloud Service instance with AppToCloud you can associate each of your existing Oracle WebLogic Server data sources with a database running in Oracle Database Cloud Service.

To migrate your application databases to Oracle Database Cloud Service:

1. Take backups of your on-premises databases with Oracle Database Backup Cloud Service.
2. Create Oracle Database Cloud Service deployments from these backups.

To get started, see Creating a Database Deployment Using a Cloud Backup in *Using Oracle Database Cloud Service*.

Install the On-Premises AppToCloud Tools

This topic applies only to Oracle Cloud at Customer.

Download and extract the AppToCloud archive to the machine hosting the Administration Server of the domain that you want to export.

1. Access the Oracle Java Cloud Service console.

2. Click your user icon at the top right of the console, select **Help**, and then select **Download Center**.
3. Click the download icon for AppToCloud.
4. Upload the file `a2c-zip-installer.zip` to the machine running the domain's Administration Server.
5. If necessary, create a destination directory for the AppToCloud tools.

   For example:

   Linux: `mkdir /u01/tools`

   Windows: `mkdir c:\u01\tools`
6. Use an archive tool to extract `a2c-zip-installer-.zip` to your destination directory.

   For example:

   Linux: `unzip a2c-zip-installer.zip -d /u01/tools`

   Windows: From Windows Explorer, right-click `a2c-zip-installer.zip`, select **Extract All**, and then enter `c:\u01\tools`.
7. Verify that the file `oracle_jcs_app2cloud/bin/a2c-healthcheck.sh` or `oracle_jcs_app2cloud/bin/a2c-healthcheck.cmd` exists in your destination directory.

   For example: `/u01/tools/oracle_jcs_app2cloud/bin/a2c-healthcheck.sh`

Check the Health on an On-Premises WebLogic Domain

This topic applies only to Oracle Cloud at Customer.

Use the AppToCloud Client to validate your on-premises WebLogic Server domain and applications in preparation to move them to an Oracle Java Cloud Service environment.

You must always run a Health Check on your WebLogic Server domain prior to performing an Export. It performs the following:

- Connects to your running Administration Server
- Verifies that your domain is in a healthy state
- Generates the initial archive file that will be used during the export
- Collects runtime information from all running servers in the domain, including the server's JVM arguments and classpath

Before you perform a Health Check, you must install the AppToCloud tools to the host machine that is running the Administration Server for your domain. You must also ensure that your source domain meets all required prerequisites.

The AppToCloud Client can be used to perform both the Health Check and the Export tasks. Alternatively, you can perform these tasks from the command line or a script.

Decide how you want to specify the administrator credentials for the domain. Choose from one of these options:

- Enter the credentials in the AppToCloud Client user interface.
- Use the AppToCloud Client to store the credentials in a new Oracle Wallet file, or select credentials in an existing file.

- Use the WebLogic Scripting Tool (WLST) `storeUserConfig` command to generate an encrypted file containing the credentials. Provide the locations of this file and the associated key file.

Any warnings or errors that the Health Check tool detects are written to a log, an HTML report, and the archive file. The Export tool checks the validation results in the archive before completing the export of your domain's configuration.

To check the health of a WebLogic Server domain:

1. Access the host that is running the Administration Server for your domain.
2. Identify the top level directory of your WebLogic Server installation on this machine. This location is also referred to as `ORACLE_HOME`.

   For example, `/u01/app/fmw`.
3. Start all servers in the domain if they are not already running.

   Refer to the relevant documentation for your version of WebLogic Server. For example:

   - Starting and Stopping Servers in *Administering Server Startup and Shutdown for Oracle WebLogic Server (12.2.1.3)*
   - Starting and Stopping Servers in *Administering Server Startup and Shutdown for Oracle WebLogic Server (12.2.1.2)*
   - Starting and Stopping Servers in *Managing Server Startup and Shutdown for Oracle WebLogic Server (10.3.6)*
4. Determine the administration URL of the Administration Server: `t3://`*`host`*`:`*`port`*.

   For example, `t3://myserver.mycompany.com:7001` or `t3://192.0.2.10:9001`

   AppToCloud connects to your domain using the T3 protocol. Do not use HTTP.
5. Launch a terminal.
6. If not already configured on your machine, set the `JAVA_HOME` environment variable to the directory where you have installed the Java SE Development Kit (JDK) version 7 or later.

   For example, on Linux:

   ```
   export JAVA_HOME=/u01/jdk
   ```

   On Windows:

   ```
   set JAVA_HOME=c:\u01\jdk
   ```

   > **Note:**
   >
   > Do not run the AppToCloud tools using an older JDK version than the version being used to run your WebLogic Server domain. In addition, if your domain is running JDK 6, you must use a separate JDK 7 installation to run the AppToCloud tools.
7. Go to the `bin` directory of your AppToCloud tools installation.

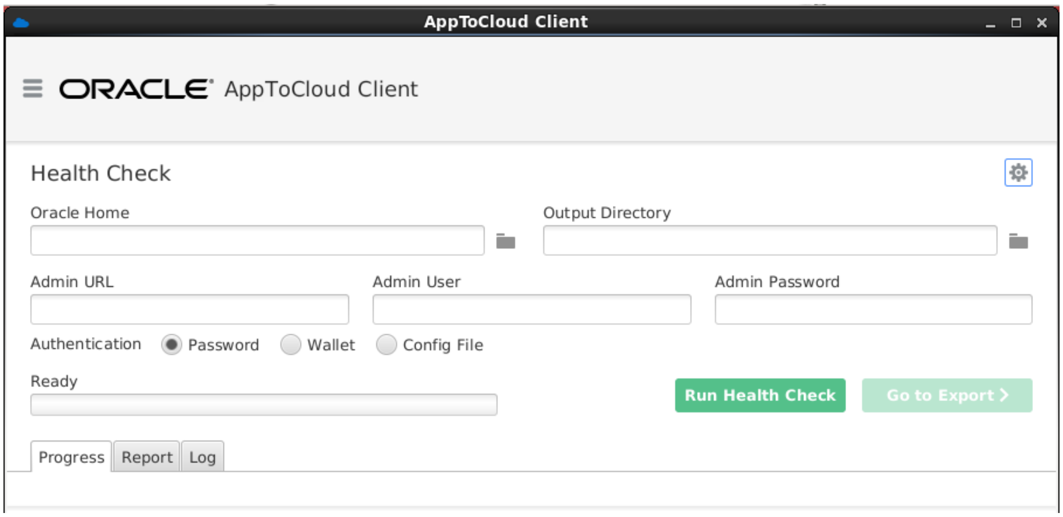
For example: `cd /u01/tools/oracle_jcs_app2cloud/bin`

8. Run the AppToCloud Client tool.

   On Linux: `a2c-client.sh`

   On Windows: `a2c-client.cmd`

   The Health Check page is displayed.



9. Specify the **Oracle Home** directory for your WebLogic Server installation. Also enter an **Output Directory** for the Health Check.

   Optionally use the folder icons to browse your local file system. If the output directory does not exist it will be created.

10. Enter the **Admin URL** for your running administration server.

11. Select an **Authentication** option:

    - **Password**: Enter the **Admin User** and **Admin Password** for your domain.
    - **Wallet**: Proceed to the next step.
    - **Config File**: Use **WLS Config File** and **WLS Key File** to provide the location of your WebLogic credential file and corresponding key file.

12. If you selected the **Wallet** authentication option:

    a. Click the folder icon next to **Wallet Directory**.

       The Change Wallet dialog is displayed.

    b. Choose to use an existing wallet or create a new one. Then click **OK** and follow the instructions:

       - **Select an existing wallet**: Select the folder containing the wallet. Enter a password if one is required.
       - **Create a new wallet**: Select a folder for your new wallet.
       - **Create a new encrypted wallet**: Select a folder for your new wallet and then enter a password.

    c. Click the user icon next to **Admin User Alias**.

       The Select Credential dialog is displayed.

   d. Select an existing credential in the wallet or click **New** to create a new one.

      For a new credential, enter its **Alias**, **User** and **Password**.

      Click **OK** to return to the main Health Check page.

13. Click **Run Health Check**.

14. Use the **Progress**, **Report** and **Log** tabs to verify that the Health Check completed successfully.

    An example Progress output:

    ```
    Initializing...
    Connecting to domain...
    Checking Java configuration...
    Checking applications health...
    Checking datasource health...
    Checking JMS health...
    Finished.
    ```

    An example Log output:

    ```
    Checking Domain Health
    Connecting to domain

    Connected to domain domainName

    Checking Java Configuration

    Checking server runtime : AdminServer

    Checking server runtime: serverName

    Checking server runtime: serverName

    Done checking Java Configuration

    Checking Servers Health
    Done checking Servers Health

    Checking componentName
    Checking componentName
    Done checking Domain Health

    Activity Log for HEALTHCHECK

    Informational Messages:
    Any informational messages

    Warning Messages:
    Any warning messages

    Error Messages:
    Any error messages

    An HTML version of this report can be found at outputDirectory/
    ```

```
reports/reportFile

Output archive saved as outputDirectory/archiveFile. You can use this
archive for the a2c-export tool.

a2c-healthcheck completed successfully (exit code = 0)
```

15. If any problems are described in the Report or Log, address these problems and then click **Run Health Check** again.

    If the AppToCloud Client experiences time-out errors while trying to connect to your servers, click the **Settings** icon and modify the **WLST Timeout** field.

After completing the health check, you are ready to export the domain.

Export an On-Premises WebLogic Domain

This topic applies only to Oracle Cloud at Customer.

Use the AppToCloud Client to capture your on-premises WebLogic Server domain and applications, and move them to a cloud storage container that's accessible to your Oracle Java Cloud Service environment.

The Export process captures a domain's configuration and Java applications. It updates the archive file that was previously generated by the Health Check procedure, generates a JSON file, and uploads these AppToCloud artifacts to a storage container in the Oracle Cloud Infrastructure Object Storage Classic. If the machine hosting the AppToCloud tools does not have access to the Internet, you can manually upload the output files to your storage container after performing the Export.

The AppToCloud Client can be used to perform both the Health Check and Export tasks. Alternatively, you can perform these tasks from the command line or a script.

Determine the credentials used to access Oracle Cloud Infrastructure Object Storage Classic. This is often the same user that you use to access the Oracle Java Cloud Service console. Choose from one of these options:

- Enter the credentials in the AppToCloud Client user interface.
- Use the AppToCloud Client to store the credentials in a new Oracle Wallet file, or select credentials in an existing file.

If the source domain does not contain a cluster, the AppToCloud Client will prompt you to add the managed servers in the domain to a new cluster in the exported domain configuration. The original source domain is not modified. If there are multiple managed servers, the same resources and applications must be targeted to all of them.

If your source domain contains multiple clusters, the AppToCloud Client will prompt you to either export all of the clusters or a specific cluster. You can export all clusters only if your Oracle Cloud account includes Oracle Identity Cloud Service.

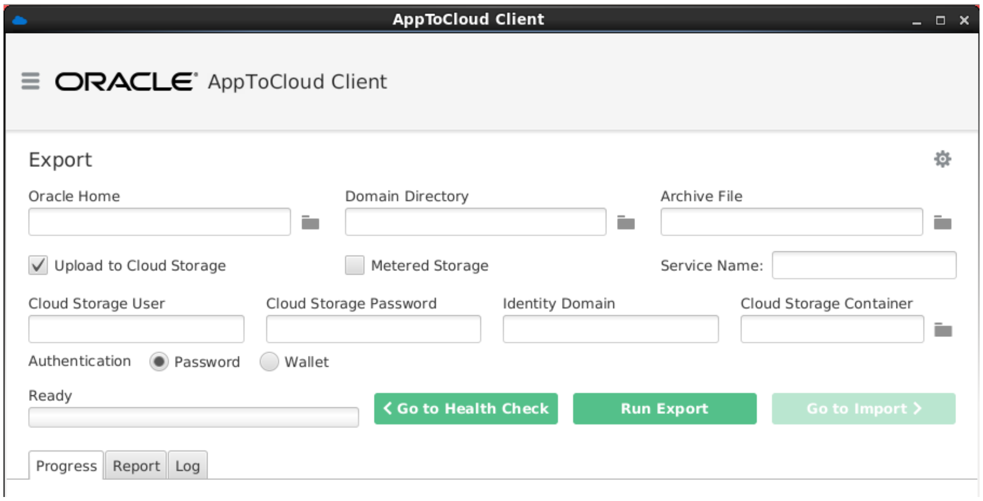
Any warnings or errors that the Export tool detects are written to a log, an HTML report, and the archive file.

To export a WebLogic Server domain:

1. Launch the `a2c-client` and perform a Health Check if not done previously.

2. Click **Go to Export**.

   The Export page is displayed.



   Another way to access the Export page is to click the menu icon in the top left corner and select **Export**.

   The values of the **Oracle Home** and **Archive File** fields are set automatically based on your previous Health Check parameters. Alternatively, use the Health Check output, report or log to identify the path and filename of the generated archive file. For example, `/u01/jcs_a2c_output/domain1.zip`. Use the folder icons to browse your local file system.

3. Specify the top level **Domain Directory**.

   For example, `/u01/domains/domain1`.

4. If you do not want the AppToCloud Client to upload the generated files to Oracle Cloud, clear the check box **Upload to Cloud Storage** and *skip to step 11*.

5. Select **Metered Storage** if you have a metered subscription to Oracle Cloud Infrastructure Object Storage Classic. If you have a non-metered subscription, enter your storage **Service Name**.

6. Provide the name of a cloud user that has access to Oracle Cloud Infrastructure Object Storage Classic. Select from these **Authentication** options:

   - **Password**: Enter your **Cloud Storage User** and **Cloud Storage Password**.
   - **Wallet**: Proceed to the next step.

7. If you selected the **Wallet** authentication option:

   a. Click the folder icon next to **Wallet Directory**.

      The Change Wallet dialog is displayed.

   b. Choose to use an existing wallet or create a new one. Then click **OK** and follow the instructions:

      - **Select an existing wallet**: Select the folder containing the wallet. Enter a password if one is required.

- **Create a new wallet**: Select a folder for your new wallet.
- **Create a new encrypted wallet**: Select a folder for your new wallet and then enter a password.

c. Click the user icon next to **Cloud Storage User Alias**.

   The Select Credential dialog is displayed.

d. Select an existing credential in the wallet or click **New** to create a new one.

   For a new credential, enter its **Alias**, **User** and **Password**.

   Click **OK** to return to the main Export page.

8. Enter your Oracle Cloud **Identity Domain**.
9. Click the folder icon for **Cloud Storage Container**.

   The Select Cloud Storage Container dialog is displayed.
10. Select an existing storage container from the list or click **New** to create a new one. Then click **OK**.
11. Click **Run Export**.
12. If the WebLogic Server domain contains multiple clusters, click **Yes** to export a specific cluster, or click **No** to export all clusters.

    If you clicked **Yes**, then select the cluster to export and click **OK**.

> **Note:**
>
> You can export all clusters only if your Oracle Cloud account includes Oracle Identity Cloud Service.

13. Use the **Progress**, **Report** and **Log** tabs to verify that the Export completed successfully. Also note the name of the generated JSON file.

    An example Progress output:

```
Discovering mail sessions for domain domainName
Discovering Coherence clusters for domain domainName
Discovering WLDF modules for domain domainName
Discovering server classpath and system property settings for domain
domainName
End of the Environment discovery
Adding application to the archive: myapp from /u01/apps/myapp.ear
Sucessfully exported model and artifacts to outputDirectory/domainName.zip
Overrides file written to outputDirectory/domainName.json
Uploading override file to cloud storage from outputDirectory/
domainName.json
Uploading archive file to cloud storage from outputDirectory/
domainName.zip
Successfully exported model and artifacts to cloudStorageURL. Overrides
file written to containerName/domainName.json
```

An example Log output:

```
Activity Log for EXPORT

Informational Messages:
Other informational messages
Uploaded override file to Oracle Cloud Storage container
containerName
Uploaded archive file to Oracle Cloud Storage container
containerName

Warning Messages:
Any warning messages

Error Messages:
Any error messages

Features Not Yet Implemented Messages
Any messages about features not yet supported by the export tool

An HTML version of this report can be found at outputDirectory/
reports/reportFile

Successfully exported model and artifacts to cloudStorageURL.
Overrides file written to containerName/domainName.json

a2c-export completed successfully (exit code = 0)
```

14. Address any problems described in the *Error Messages* section of the Report or Log. Then click **Run Export** again.

    By default the Export will fail if the previous Health Check resulted in any errors. Alternatively, click the **Settings** icon and select the **Force AppToCloud Export** check box. However, Oracle strongly recommends that you only export domains that have successfully passed the Health Check.

    The tool will output a warning message if any of the servers in your source domain include the current directory (`.`) in its `CLASSPATH`. If this server has been started by using the WebLogic Node Manager, the Node Manager adds the current directory to the server's `CLASSPATH`. Therefore you can ignore this warning message if using the Node Manager.

15. If you did not use the AppToCloud Client to upload the generated files to Oracle Cloud Infrastructure Object Storage Classic, then manually upload these files as objects to an existing storage container:

    - *outputDirectory*/*domainName*.zip
    - *outputDirectory*/*domainName*.json

    See Creating Containers and Creating a Single Object in *Using Oracle Cloud Infrastructure Object Storage Classic*.

After exporting your domain and uploading the files to a storage container, you are ready to create an Oracle Java Cloud Service instance.

If the output from the Export includes messages about features that are not yet implemented, you will need to manually configure these features after creating your service instance.

Create a Service Instance with AppToCloud

This topic applies only to Oracle Cloud at Customer.

In order to import your source domain configuration and applications into Oracle Java Cloud Service, you must associate a new service instance with the files that you previously generated with the AppToCloud tools.

Most of the steps that you use to create a service instance with AppToCloud are the same as those you use to create a standard service instance. However, there are some additional inputs:

- You must provide the location of your AppToCloud JSON file in object storage.
- You must associate each Data Source in your original WebLogic Server domain with an existing Oracle Database Cloud Service deployment. Other database services in Oracle Cloud are not supported.
- You may need to configure properties (such as URL and credentials) for the following types of resources if found in the original WebLogic Server domain:
  - Foreign Java Naming and Directory Interface (JNDI) Providers
  - Java Mail Sessions
  - Foreign Java Message Service (JMS) Servers
  - Remote Store and Forward (SAF) Contexts
  - JMS Messaging Bridge Destinations

Before creating an Oracle Java Cloud Service instance, Oracle recommends that you review the options described in Design Considerations for an Oracle Java Cloud Service Instance.

1. Access the Oracle Java Cloud Service console.
2. Click **Create Instance** and select the **Java — AppToCloud** option.
3. Provide details about the JSON file generated by the Export tool:

Field	Description
Exported .json File	Enter the URL of the JSON file that was uploaded to Oracle Cloud Infrastructure Object Storage Classic. If you used the Export tool to upload this file to cloud storage, the Export tool provided this URL in its output. **Format**: *`rest_endpoint_url`*`/`*`containerName`*`/`*`fileName`*`.json` You can also find the REST endpoint URL of the Oracle Cloud Infrastructure Object Storage Classic service instance in the Oracle Cloud My Services portal. **Example**: `https://foo.storage.oraclecloud.com/v1/MyService-bar/MyContainer/mydomain.json` The corresponding AppToCloud ZIP file must be in the same location as the JSON file in cloud storage. AppToCloud does not currently support the creation of a service instance from a JSON file in an Oracle Cloud Infrastructure Object Storage bucket.
Cloud Storage User Name	Enter the user name of the Oracle Cloud Infrastructure Object Storage Classic service user who created the container you specified earlier.
Cloud Storage Password	Enter the password of the user you specified in the previous field.

4. Click **OK**.
5. Complete the Instance page:

Field	Description
Instance Name	Specify a name for the Oracle Java Cloud Service instance. The service instance name: • Must contain one or more characters. • Must not exceed 30 characters. • Must start with an ASCII letter: `a` to `z` , or `A` to `Z`. • Must contain only ASCII letters or numbers. • Must not contain a hyphen. • Must not contain any other special characters. • Must be unique within the identity domain.
Description	(Optional) Enter a short description of the Oracle Java Cloud Service instance.
Notification Email	(Optional) Specify an email address where you would like to receive a notification of any events occurring with the service instance, including whether provisioning has succeeded or failed.
Region	(Available only if your account has regions) Select a region if you want to create the service instance in a specific region, or if you want to use a custom IP network. You must also select a region if you intend to assign reserved IP addresses to your service instance nodes. A region supports either Oracle Cloud Infrastructure or Oracle Cloud Infrastructure Classic. For a list of available regions, see Data Regions for Platform and Infrastructure Services. The database that you intend to associate with your Oracle Java Cloud Service instance must be in the same region. If you select **No Preference**, Oracle Java Cloud Service will select one of the available Oracle Cloud Infrastructure Classic regions. However, you will not be able to use an IP network or reserved IP addresses for your service instance.
IP Network	(Only if a region is selected) (Not available on Oracle Cloud Infrastructure) Select an IP network if you want to create the service instance in an IP network that you've defined. By default, each node in your instance is auto-assigned a public and a private IP address. The IP addresses might change each time the service instance is restarted. You can reserve and assign fixed public IP addresses. In order to select an IP network if you have selected **Enable Authentication Using Identity Cloud Service**, which automatically configures a managed load balancer, you must first attach an internet-facing load balancer to the IP network. This field is not relevant to Oracle Cloud Infrastructure.
Tags	(Optional) Select existing tags or add tags to associate with the service instance. To select existing tags, select one or more check boxes from the list of tags that are displayed on the pull-down menu. To create tags, click + to display the **Create Tags** dialog box. In the **New Tags** field, enter one or more comma-separated tags that can be a key or a key:value pair. If you do not assign tags during provisioning, you can create and manage tags after the service instance is created.

Field	Description
Bring Your Own License	The **Bring Your Own License** (BYOL) option enables you to bring your on-premises Oracle WebLogic Server licenses to Oracle Cloud. BYOL instances are billed at a lower rate than other instances. See Frequently Asked Questions: Oracle BYOL to PaaS. You must own a Universal Credits or Government subscription in order to use BYOL. BYOL is enabled by default. If you deselect this option, your account will be charged for the new service instance according to your Oracle Java Cloud Service agreement. **Note**: Before you scale up or scale out a BYOL instance, you must have enough WebLogic Server licenses for the additional OCPUs that will be allocated to the instance after it is scaled.
Software Release	Select a WebLogic Server software release for this service instance: • **Oracle WebLogic Server 12c (12.1.3.0)** • **Oracle WebLogic Server 12c (12.2.1.2)** • **Oracle WebLogic Server 12c (12.2.1.3)** Oracle WebLogic Server 12c (12.2.1.4) is not supported. The Oracle WebLogic Server 11*g* software release option is not available. All service instances created with AppToCloud will be provisioned with Oracle WebLogic Server 12*c*, even if your source domain was running 11*g*.
Software Edition	Select a WebLogic Server software edition: • **Enterprise Edition** • **Enterprise Edition with Coherence** The Standard Edition option is not supported for a service instance created with AppToCloud.
Metering Frequency	This option appears only if you have a traditional metered subscription. If you have a Universal Credits subscription, this field is absent. Select a metering frequency to determine how you are billed for this service instance: • **Hourly**—Pay only for the number of hours that this service instance was running during your billing period. • **Monthly**—Pay one price for the full month irrespective of the number of hours that this service instance was running. For services that are started in the middle of a month, the price will be pro-rated; you pay only for the partial month from the day the service instance is created.

6. Click **Next**.

   The Instance Details page is displayed.

7. If you want to configure any of the advanced options, including Backup and Recovery, click **Advanced**.

   You cannot provision a Coherence data tier for service instances created with AppToCloud.

8. Complete the WebLogic Configuration section of the Instance Details page:

Field	Description
Compute Shape	Select the compute shape to use for all Administration Server and Managed Server nodes. The compute shape is the number of Oracle Compute Units (OCPUs) and amount of memory (RAM) that you want to allocate to these nodes. The selected shape is not used for Coherence or Load Balancer nodes. The list of available shapes varies depending on whether you selected an Oracle Cloud Infrastructure Classic or Oracle Cloud Infrastructure region. (Advanced option) When you create multiple WebLogic clusters, you can assign a different compute shape for different clusters. This field displays the compute shape of the selected cluster. If you purchased a Universal Credits subscription for Oracle Java Cloud Service, you will pay at the Pay-As-You-Go rate when you exceed your monthly or annual maximum credit.
SSH Public Key	Specify the public key that will be used for authentication when connecting to a node in your instance by using a Secure Shell (SSH) client. Click **Edit** to display the SSH Public Key for VM Access dialog, and then specify the public key using one of the following methods: • Select **Key file name** and use your web browser to select a file on your machine that contains the public key. • Select **Key value** and paste the value of the public key into the text area. Be sure the value does not contain line breaks or end with a line break. • Select **Create a New Key** if you want Oracle to generate a public/private key pair for you. You will be prompted to download these generated keys. If you choose to create a new key, the generated private key file is in OpenSSH format. Before connecting to a node in this service instance with the PuTTY SSH client, you must first convert the key to PuTTY's proprietary format.
Server Count	You can only edit this field if the source domain contains a single cluster. Select the initial number of Managed Servers that you want to provision in this service instance. The default value is the number of Managed Servers in the source domain. Additional choices are: `1`, `2`, `4`. If you configure more than one Managed Server in the cluster, Oracle recommends that you also enable the Load Balancer. You can also perform scaling operations to increase or decrease the cluster size after provisioning the service instance.
WebLogic Clusters	You can only edit this field if the source domain contains multiple clusters and you selected an Oracle Cloud Infrastructure Classic region. Configure the initial number of Managed Servers that you want to provision in each cluster. Select a cluster from the list, and then click **Edit cluster**. The default value is the number of Managed Servers in the source domain. Additional choices are: `1`, `2`, `4`. You can also perform scaling operations to increase or decrease the cluster size after provisioning the service instance. You cannot remove existing clusters from the service instance, or add new clusters to the service instance.
Enable Access to Administration Consoles	(Advanced option) Select this check box if you want to enable access to the WebLogic Service Administration Console, Fusion Middleware Control, and Load Balancer Console for the service instance. If you do not select this option, these consoles will not be externally accessible, and also will not appear as choices in the service instance's menu ☰. Alternatively, you can enable access to the administration consoles after creating the service instance. If you are creating this service instance in Oracle Cloud Infrastructure, access to the administration consoles is enabled by default; selecting or deselecting this check box has no effect.

Field	Description
Deploy Sample Application	(Advanced option) By default, a sample application, `sample-app.war`, is deployed automatically to the Managed Servers in your instance. If you do not want to automatically deploy the sample application, deselect this check box.
Reserved IPs	(Not available on Oracle Cloud Infrastructure) Select reserved IP addresses for the nodes in your cluster, or leave the default value as **Assign Automatically** if you want Oracle to automatically assign IP addresses to these nodes. The number of IP addresses that you select must equal the number of nodes in the cluster. This option is displayed only if you selected a specific **Region** for this service instance. You create IP reservations by using the **Reserved IPs** tab in the Oracle Java Cloud Service Console. If you do not see this tab on the console, click the gear icon next to this field and follow the instructions to create your first IP reservation. After creating IP reservations, you need to restart the instance creation wizard.

9. Complete the WebLogic Access section of the Instance Details page:

Field	Description
Local Administrative User Name	Enter your choice of user name for the WebLogic Server administrator. The default is `weblogic`. This name is used to access the WebLogic Server Administration Console, Fusion Middleware Control, and Load Balancer Console for the service instance. The name must be between 8 and 128 characters long and **cannot** contain any of the following characters: • Tab • Brackets • Parentheses • These special characters: – Left angle bracket (`<`) – Right angle bracket (`>`) – Ampersand (`&`) – Pound sign (`#`) – Pipe symbol (`|`) – Question mark (`?`) You can also change the user name through the WebLogic Server Administration Console after the service instance is provisioned.

Field	Description
Password	Specify a password for the WebLogic Server administrator and confirm the password. If you selected an Oracle Database Exadata Cloud Service database deployment for **Database Instance Name**, this password must start with a letter, be of 8 to 30 characters in length, and contain at least: • 1 uppercase character • 1 lower case character • 1 digit (0 through 9) • One of the following special characters: _ (underscore), - (hyphen), or # (pound sign or hash) If you did not select an Oracle Database Exadata Cloud Service database deployment, Oracle still recommends following these password requirements as a best practice. However, the following basic password criteria are acceptable: • Starts with a letter • Is between 8 and 30 characters long • Contains letters, at least one number, and, optionally, any number of these special characters: – Dollar sign ($) – Pound sign (#) – Underscore (_) No other special characters are allowed.
Enable Authentication with Oracle Identity Cloud Service	Select this check box if you want WebLogic Server to authenticate application users and administrators against Oracle Identity Cloud Service in addition to the local WebLogic Server identity store. This field appears only if your cloud account includes Oracle Identity Cloud Service. This check box is automatically selected for you if the source domain contains multiple clusters, and you cannot change it. (Not available on Oracle Cloud Infrastructure)

10. Complete the Database Configuration section of the Instance Details page:

Field	Description
Database Instance Name	For Oracle Cloud Infrastructure regions, select an existing database in Oracle Autonomous Database (Oracle Autonomous Transaction Processing), Oracle Cloud Infrastructure Database or Oracle Database Cloud Service. • Database instances in Oracle Database Cloud Service and Oracle Cloud Infrastructure Database must be in the same region and virtual cloud network (VCN) as the Oracle Java Cloud Service instance. The database and service instance do not need to be in the same subnet or availability domain. The database and service instance can be on different VCNs only if you configure VCN peering. • To use a Bare Metal database in Oracle Cloud Infrastructure Database, you must create the service instance with the Oracle Java Cloud Service REST API or CLI. • To use an Oracle Cloud Infrastructure Database running Oracle Database 12.2 or later, the service instance must be running WebLogic Server 12.2.1 or later. • For a 1-node virtual machine DB system in Oracle Cloud Infrastructure Database, you can use a database that is created with the fast provisioning option. Oracle Java Cloud Service supports using Logical Volume Manager as the storage management software for a 1-node virtual machine DB system. • To use an Oracle Autonomous Database, the service instance must be running WebLogic Server 12.2.1.3 or later. You must use an Oracle Autonomous Database that is created with the serverless option. Oracle Java Cloud Service does not yet support using a dedicated deployment autonomous database. For Oracle Cloud Infrastructure Classic regions, select an existing Oracle Database Cloud Service or Oracle Database Exadata Cloud Service deployment. • If you selected an **IP Network** for this service instance, you must also select an Oracle Database Cloud Service deployment that is attached to an IP network. If the service instance and database deployment are attached to different IP networks, the two IP networks must be connected to the same IP network exchange. • The Oracle Database Cloud Service deployment must not configured with a **Backup Destination** set to `None` (not applicable to Oracle Database Cloud Service — Virtual Image deployments). Note the following additional restrictions for Oracle Database Cloud Service: • You cannot use an Oracle Database Cloud Service deployment running Oracle Database 18c as the infrastructure schema database. • You can use an Oracle Database Cloud Service deployment running Oracle Database 12.2, but only for service instances running Oracle WebLogic Server 12.2.1 or later. • Create Oracle Database Cloud Service deployments with a backup option other than `NONE`. This configuration enables Oracle Java Cloud Service to coordinate backups across your service instance and the database. Coordinated backups are not supported for other database services.
PDB Name	Select the pluggable database the service instance will connect to. • For Oracle Cloud Infrastructure databases, the PDB name is populated. If you did not specify a PDB name when you created the Oracle Cloud Infrastructure database, the default PDB name populated in this field is `<dbName>_pdb1`. • For Oracle Database Cloud Service (Classic) databases, if you don't specify a PDB name, Oracle Java Cloud Service uses the default Oracle Database 12c PDB name that was provided when the Oracle Database Cloud Service (Classic) database deployment was originally created.
Administrator User Name	Specify the name of the database administrator that Oracle Java Cloud Service will use to connect to the selected database and to provision the required schemas for this service instance. This value is set automatically for: • Oracle Autonomous Database (Oracle Autonomous Transaction Processing): `ADMIN` • Oracle Cloud Infrastructure Database: `SYS`
Password	Enter the password for the database administrator.

11. Complete the Backup and Recovery Configuration section of the Instance Details page:

    This section is displayed only if you clicked **Advanced**.

Field	Description
Backup Destination	(Advanced option) Select **Both Remote and Disk Storage** if you want to enable automated and on-demand backups for this service instance. Backups will be saved to object storage *and* to block storage volumes that are attached to the nodes of the instance. The default value is **None**, meaning that you cannot use Oracle Java Cloud Service to take backups of this service instance. You can configure backups on a service instance after creating it. This field is not relevant if you selected **Oracle Java Cloud Service—Virtual Image**.
Object Storage Container	This field is displayed only if **Backup Destination** is set to **Both Remote and Disk Storage**. Enter the object storage location where backups of the service instance must be stored. The object storage container field in the instance creation wizard is auto-populated with a default container URL in the format `restEndpointUrl/JaaS`, where `restEndpointUrl` is the REST endpoint URL of the Oracle Cloud Infrastructure Object Storage Classic service in the account, and `JaaS` is the default container name. You can change the container name. Note that if the account doesn't include an Object Storage service entitlement or if the region selected is an Oracle Cloud Infrastructure region, then the container field is not autopopulated. • **Oracle Cloud Infrastructure Classic**: Enter the URL of a container in Oracle Cloud Infrastructure Object Storage Classic. **Format**: `rest_endpoint_url/containerName` You can find the REST endpoint URL of the Oracle Cloud Infrastructure Object Storage Classic service instance in the Infrastructure Classic Console. **Example**: `https://acme.storage.oraclecloud.com/v1/MyService-acme/MyContainer` **Note**: You can select the **Create Object Storage Container** check box to have a new container created automatically. • **Oracle Cloud Infrastructure**: Enter the URL of a bucket in Oracle Cloud Infrastructure Object Storage. See Prerequisites for PaaS Services on Oracle Cloud Infrastructure in the Oracle Cloud Infrastructure documentation. **Format**: `https://swiftobjectstorage.region.oraclecloud.com/v1/namespace/bucket` To find out your `namespace`, sign in to the Oracle Cloud Infrastructure web console, click the tenancy name, and look for the **Object Storage Namespace** field. **Example**: `https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/myCompany/myBucket`
User Name	This field is displayed only if **Backup Destination** is set to **Both Remote and Disk Storage**. In Oracle Cloud Infrastructure Classic regions only, this field is not displayed if you selected **Enable Authentication Using Identity Cloud Service**. • **Oracle Cloud Infrastructure Classic**: Enter the user name of the Oracle Cloud Infrastructure Object Storage Classic service user who created the container you specified earlier. If the container doesn't exist, then enter the user name of a service administrator. • **Oracle Cloud Infrastructure**: Enter the user name of the Oracle Cloud Infrastructure Object Storage user who created the bucket you specified earlier.

Field	Description
Password	This field is displayed only if **Backup Destination** is set to **Both Remote and Disk Storage**. In Oracle Cloud Infrastructure Classic regions only, this field is not displayed if you selected **Enable Authentication Using Identity Cloud Service**. • **Oracle Cloud Infrastructure Classic**: Enter the password of the user you specified. • **Oracle Cloud Infrastructure**: Enter the Auth Token generated in Oracle Cloud Infrastructure for the user you specified. See Prerequisites for PaaS Services on Oracle Cloud Infrastructure in the Oracle Cloud Infrastructure documentation.
Create Object Storage Container	This option is displayed only if **Backup Destination** is set to **Both Remote and Disk Storage**. If the Oracle Cloud Infrastructure Object Storage Classic container that you specified doesn’t exist, or if you aren’t sure whether it exists, then select this check box. If the container doesn’t exist, it will be created automatically. This option is not relevant to Oracle Cloud Infrastructure. The specified Oracle Cloud Infrastructure Object Storage bucket must exist prior to creating a service instance.

12. Complete the Load Balancer section of the Service Details page:

Field	Description
Provision Local Load Balancer	(Advanced option) Select **Yes** to provision a load balancer node running Oracle Traffic Director in this service instance. This user-managed load balancer is configured to distribute client requests to the Managed Servers in the service instance. Provisioning a load balancer is recommended if the cluster size is 2 or more. The default value is **No**. If you selected **Enable Authentication Using Identity Cloud Service**, then you cannot configure a user-managed load balancer. An Oracle-managed load balancer is provisioned for you automatically. You can also add an Oracle Traffic Director load balancer node to a service instance after creating the service instance.

Field	Description
Load Balancer	This option is displayed only if you selected an Oracle Cloud Infrastructure region. Select the type of load balancer that you want to configure for your service instance: • **Oracle-Managed Load Balancer**: A dual-node, Oracle-managed instance of the Oracle Cloud Infrastructure Load Balancing service, providing active-passive high-availability. Failover from the active load-balancer node to the other node occurs automatically. You can't customize the default listeners, certificates, and so on for an Oracle Cloud Infrastructure Load Balancing instance that is provisioned by Oracle Java Cloud Service. If you need the ability to configure Oracle Cloud Infrastructure Load Balancing, then you must create the load balancer manually. See Set Up an Oracle Cloud Infrastructure Load Balancer. • **Oracle Traffic Director**: One or two Oracle Traffic Director nodes within your service instance. The dual-node configuration is in active-active mode, but failover to the second node is not automatic. • **None**: No load balancer will be configured for this instance. Provisioning a load balancer is recommended if the cluster size is 2 or more. The default value is **None**. If you selected **Enable Authentication Using Identity Cloud Service**, then you cannot configure a user-managed load balancer. You must select **Oracle-Managed Load Balancer**. If you select **Oracle Traffic Director** and configure one Oracle Traffic Director node, you can also add a second Oracle Traffic Director node to a service instance after creating the service instance. If you configured two Oracle Traffic Director nodes during provisioning, you cannot add another Oracle Traffic Director node. If you select **None**, then you can add an Oracle Traffic Director load balancer after creating the service instance.
Compute Shape	This option is displayed only if **Provision Local Load Balancer** is set to `Yes` or **Load Balancer** is set to **Oracle Traffic Director**. Select the compute shape to use for all the load balancer nodes in the service instance. The compute shape is the number of Oracle Compute Units (OCPUs) and amount of memory (RAM) that you want to allocate to these nodes. The list of available shapes varies depending on whether you selected an Oracle Cloud Infrastructure Classic or Oracle Cloud Infrastructure region. You are billed for Oracle Traffic Director nodes at the same price that you are billed for Oracle WebLogic Server nodes in your Oracle Java Cloud Service subscription.
Add Another Active OTD Node	This option is displayed only if **Provision Local Load Balancer** is set to `Yes` or **Load Balancer** is set to **Oracle Traffic Director**. Select this check box to provision a second load balancer node running Oracle Traffic Director (OTD) in this service instance. Both load balancer nodes route traffic to the cluster of WebLogic Managed Servers. You can also add a second load balancer node to a service instance after creating the service instance.

Field	Description
Reserved IPs	Select reserved IP addresses for the load balancer nodes in your cluster, or leave the default value as **Assign Automatically** if you want Oracle to automatically assign IP addresses to these nodes. The number of IP addresses that you select must equal the number of load balancer nodes in the service instance. This option is displayed only if these conditions are true: • You selected a specific Oracle Cloud Infrastructure Classic **Region** for this service instance. • **Provision Local Load Balancer** is set to `Yes` You create IP reservations by using the **Reserved IPs** tab in the Oracle Java Cloud Service Console. If you do not see this tab on the console, click the gear icon next to this field and follow the instructions to create your first IP reservation. After creating IP reservations, you need to restart the instance creation wizard.
Load Balancing Policy	This option is displayed if you selected **Enable Authentication Using Identity Cloud Service** or **Provision Local Load Balancer**. If you selected **Provision Local Load Balancer**, choose one of the following policies: • **Least Connection Count** (default)—Passes each new request to the Managed Server with the least number of connections. This policy is useful for smoothing distribution when a Managed Server receives more requests than it can handle efficiently. • **Least Response Time**—Passes each new request to the Managed Server with the fastest response time. • **Round Robin**—Evenly distributes requests across all Managed Servers, regardless of the number of connections or response times. If you selected **Enable Authentication Using Identity Cloud Service**, choose one of the following policies: • **Round Robin**— (default) Same as above. • **IP Hash**—The IP Hash policy uses an incoming request's source IP address as a hashing key to route traffic to the same backend server. The load balancer routes requests from the same client to the same backend server as long as that server is available. • **Least Connection Count**—Same as above. You can also use the Load Balancer console to modify this policy after creating the service instance.

13. Click **Next**.

    The Additional Details page is displayed if your source domain contains resources that may require additional configuration. Otherwise skip to step 18.

14. Expand **Application Data Source** and select the first data source. Update the data source’s configuration:

Field	Description
DataSource Name	The name of the data source you selected. This field is read-only.
Source DataSource Type	The type of this data source in the original source domain (Generic, Multi, or GridLink). This field is read-only.
DBCS Instance	Select an existing Oracle Database Cloud Service database deployment. Oracle Java Cloud Service will configure the selected data source to connect to this database deployment. You cannot select databases in other services such as Oracle Autonomous Database (Oracle Autonomous Transaction Processing), Oracle Cloud Infrastructure Database, or Oracle Database Exadata Cloud Service. However, after creating the service instance, you can manually update the generated data sources and configure them to use any database.

Field	Description
Target DataSource Type	The type of this data source (Generic, Multi, or GridLink) that will be configured in the service instance. This field may be read-only or a select box, depending on the **Software Edition** you chose for this service instance, and whether or not you selected a RAC-enabled database deployment in **DBCS Instance**: • If the database deployment does not use Oracle RAC, the type must be Generic. • If the database deployment uses Oracle RAC and the service instance is running Enterprise Edition, the type must be Multi. • If the database deployment uses Oracle RAC and the service instance is running Enterprise Edition with Coherence, the type can either be Multi or GridLink.
Username	Enter the name of a valid user in the selected Oracle Database Cloud Service database deployment. This data source will connect to the database as this user.
Password	Enter the database user's password.
PDB	Enter the name of the pluggable database for Oracle Database 12c. If not specified, the PDB name provided when the Database Cloud Service database deployment was created will be used. This value does not apply to Database Cloud Service database deployments running Oracle Database 11g.

15. Click **OK** to accept your changes or click **Disable** to remove this data source from the service instance. Repeat the previous step for each additional data source in the **Application Data Source** list.

    A check mark icon indicates a data source that you have already configured.

16. Update the configuration for any other resources found on the Additional Details page:

    - Configure a Foreign JNDI Provider
    - Configure a Java Mail Session
    - Configure a Foreign JMS Server
    - Configure a Remote SAF Context
    - Configure a JMS Messaging Bridge Destination
    - Disable a JMS Messaging Bridge

17. Click **Next**.

    The Confirmation page is displayed.

18. If you are satisfied with your choices click **Create**.

    If you need to change the service instance details, use the navigation bar or **Back** button at the top of the wizard to step back through the pages in the wizard. Similarly, if you disabled a resource on the Additional Details page and now want to include it in the service instance, select the resource and click **Enable**.

    Click **Cancel** to cancel out of the wizard without creating a new service instance.

    You can also review the log messages that were generated by the AppToCloud Export tool for the source domain. Expand **Export/Activity Log Messages**.

19. Expand the **Instance Create or Delete History** section of the page, and then click the service instance name or **Details**.

20. Monitor the progress and status of the creation of your service instance.

After your service instance is provisioned and is running, you are ready to import the AppToCloud artifacts into the service instance.

> **Note:**
>
> The AppToCloud import operation can only be performed on a new and unmodified service instance. Do not perform any scaling operations, modify the domain configuration or otherwise change the service instance prior to completing the import operation.

Configure a Foreign JNDI Provider

This topic applies only to Oracle Cloud at Customer.

When using AppToCloud to create an Oracle Java Cloud Service instance, you must configure the location of any Foreign Java Naming and Directory Interface (JNDI) Providers that were present in your source Oracle WebLogic Server domain.

A foreign JNDI provider is a feature of WebLogic Server that allows applications to access JNDI resources on a remote application server as if they were JNDI resources in the host WebLogic Server. You can choose from these options:

- Select a specific server or cluster in an existing Oracle Java Cloud Service instance. Both service instances must be in the same identity domain.
- Enter the URL of the remote JNDI provider. If this is the location of another Oracle Java Cloud Service instance, you must also enable network communication between this service instance and your new one.

To configure a foreign JNDI provider:

1. From the Additional Service Details page of the AppToCloud instance creation wizard, expand **Foreign JNDI Provider**.
2. Select the name of the foreign JNDI provider.
3. Click **Disable** if you want to remove this foreign JNDI provider from the service instance. Otherwise continue to the next step.
4. Update the configuration for this foreign JNDI provider:

Field	Description
Resource Name	The name of the foreign JNDI provider you selected. This field is read-only.
JCS Instance	Choose one of these options: • Select another Oracle Java Cloud Service instance to which this domain will connect for this foreign JNDI provider. • Select **User Provided URL**.
Provider URL	Enter the URL that the domain will use to connect to this foreign JNDI provider. This field is only applicable if you choose **User Provided URL**.
Cluster	Select a cluster within the target Oracle Java Cloud Service instance. This field is not shown if you choose **User Provided URL**.

Field	Description
Server	Choose one of these options: • Select **All Servers in Cluster**. • Select a specific server within the target Oracle Java Cloud Service instance. This field is not shown if you choose **User Provided URL**.
Protocol	The default protocol that is used for communication between multiple WebLogic Server domains is `t3`. If a different protocol is required, enter it here. This field is not shown if you choose **User Provided URL**.
Username	Enter the name of a user that is authorized to access this foreign JNDI provider, if one is required.
Password	Enter the JNDI user’s password, if a username is required.
Bypass Precheck	By default, Oracle Java Cloud Service will validate the connection to this provider and continue with the AppToCloud import operation only if it succeeds. Select this check box to skip the validation of this provider.

5. Click **OK** to accept your changes. Repeat these steps for each additional foreign JNDI provider in the **Foreign JNDI Provider** list.

   A check mark icon indicates a provider that you have already configured.

6. Continue with the creation of this service instance.

Configure a Java Mail Session

This topic applies only to Oracle Cloud at Customer.

When using AppToCloud to create an Oracle Java Cloud Service instance, you must configure the location of any Java Mail Sessions that were present in your source Oracle WebLogic Server domain.

A Java Mail session provides applications running on WebLogic Server with access to mail servers through either the Internet Message Access Protocol (IMAP) or Simple Mail Transfer Protocol (SMTP) mail protocols. The configuration for a mail session includes the location of the mail server for sending messages and the location of the mail server for receiving messages.

1. From the Additional Service Details page of the AppToCloud instance creation wizard, expand **Java Mail Session**.
2. Select the name of the mail session.
3. Click **Disable** if you want to remove this mail session from the service instance. Otherwise continue to the next step.
4. Update the configuration for this mail session:

Field	Description
Mail Session Name	The name of the mail session you selected. This field is read-only.

Field	Description
Protocol **Host** **Port** **Username** **Password**	Enter the protocol, host name, port and credentials (if required) of the mail server: • For sending messages (Send) • For receiving messages (Receive) • To be used by default for both sending or receiving messages (Default) If a **Host** is not configured in either the Send or Receive sections, you must configure a Default host.
Bypass Precheck	By default, Oracle Java Cloud Service will validate the connection to this mail server and continue with the AppToCloud import operation only if it succeeds. Select this check box to skip the validation of this provider.

> **Note:**
>
> You can also directly update the standard Java Mail properties found in the **Optional Properties** text area.

5. Click **OK** to accept your changes. Repeat these steps for each additional mail session in the **Java Mail Session** list.

   A check mark icon indicates a resource that you have already configured.

6. Continue with the creation of this service instance.

Configure a Foreign JMS Server

This topic applies only to Oracle Cloud at Customer.

When using AppToCloud to create an Oracle Java Cloud Service instance, you must configure the location of any Foreign Java Message Service (JMS) Servers that were present in your source Oracle WebLogic Server domain.

A foreign JMS server is a feature of WebLogic Server that allows applications to access JMS resources (connection factories and destinations) on a remote application server as if they were local JMS resources running in the host WebLogic Server. You can choose from these options:

- Select a specific server or cluster in an existing Oracle Java Cloud Service instance. Both service instances must be in the same identity domain.
- Enter the URL of the remote JNDI provider. If this is the location of another Oracle Java Cloud Service instance, you must also enable network communication between this service instance and your new one.

To configure a foreign JMS server:

1. From the Additional Service Details page of the AppToCloud instance creation wizard, expand **JMS Module**.
2. Expand the JMS module containing the foreign JMS server.
3. Expand **Foreign Server** and select the name of the foreign JMS server.

4. Click **Disable** if you want to remove this foreign JMS server from the service instance. Otherwise continue to the next step.
5. Update the configuration for this foreign JMS server:

Field	Description
Name	The name of the foreign server you selected. This field is read-only.
JCS Instance	Choose one of these options: • Select another Oracle Java Cloud Service instance to which this domain will connect for this foreign JMS server. • Select **User Provided URL**.
JNDI Connection URL	Enter the URL that the domain will use to connect to this foreign server. This field is only applicable if you choose **User Provided URL**.
Cluster	Select a cluster within the target Oracle Java Cloud Service instance. This field is not shown if you choose **User Provided URL**.
Server	Choose one of these options: • Select **All Servers in Cluster**. • Select a specific server within the target Oracle Java Cloud Service instance. This field is not shown if you choose **User Provided URL**.
Protocol	The default protocol that is used for communication between multiple WebLogic Server domains is `t3`. If a different protocol is required, enter it here. This field is not shown if you choose **User Provided URL**.
Username	Enter the name of a user that is authorized to access this foreign server, if one is required.
Password	Enter the JNDI user’s password, if a username is required.
Bypass Precheck	By default, Oracle Java Cloud Service will validate the connection to this foreign server and continue with the AppToCloud import operation only if it succeeds. Select this check box to skip the validation of this resource.

6. Click **OK** to accept your changes. Repeat these steps for each additional foreign server on this page.

   A check mark icon indicates a resource that you have already configured.
7. Continue with the creation of this service instance.

For more information on foreign JMS servers, refer to one of the following topics:

- Configuring Foreign Server Resources to Access Third-Party JMS Providers in *Administering JMS Resources for Oracle WebLogic Server (12.2.1.3)*
- Configuring Foreign Server Resources to Access Third-Party JMS Providers in *Administering JMS Resources for Oracle WebLogic Server (12.2.1.2)*
- Configuring Foreign Server Resources to Access Third-Party JMS Providers in *Administering JMS Resources for Oracle WebLogic Server (12.1.3)*
- Configuring Foreign Server Resources to Access Third-Party JMS Providers in *Configuring and Managing JMS for Oracle WebLogic Server (11.1.1.7)*

Configure a Remote SAF Context

This topic applies only to Oracle Cloud at Customer.

When using AppToCloud to create an Oracle Java Cloud Service instance, you must configure the location of any Remote Store-and-Forward (SAF) Contexts that were present in your source Oracle WebLogic Server domain.

The SAF service enables WebLogic Server to deliver JMS messages reliably between destinations running in different WebLogic Server domains. A remote SAF context defines the location of the remote server or cluster. You can choose from these options:

- Select a specific server or cluster in an existing Oracle Java Cloud Service instance. Both service instances must be in the same identity domain.
- Enter the URL of the remote SAF context. If this is the location of another Oracle Java Cloud Service instance, you must also enable network communication between this service instance and your new one.

To configure a remote SAF context:

1. From the Additional Service Details page of the AppToCloud instance creation wizard, expand **JMS Module**.
2. Expand the JMS module containing the remote SAF context.
3. Expand **Remote SAF Context** and select the name of the remote SAF context.
4. Click **Disable** if you want to remove this remote SAF context from the service instance. Otherwise continue to the next step.
5. Update the configuration for this remote SAF context:

Field	Description
Name	The name of the remote SAF context you selected. This field is read-only.
JCS Instance	Choose one of these options: • Select another Oracle Java Cloud Service instance to which this domain will connect for this remote SAF context. • Select **User Provided URL**.
URL	Enter the URL that this domain will use to connect to the remote server or cluster. This field is only applicable if you choose **User Provided URL**.
Cluster	Select a cluster within the target Oracle Java Cloud Service instance. This field is not shown if you choose **User Provided URL**.
Server	Choose one of these options: • Select **All Servers in Cluster**. • Select a specific server within the target Oracle Java Cloud Service instance. This field is not shown if you choose **User Provided URL**.
Protocol	The default protocol that is used for communication between multiple WebLogic Server domains is `t3`. If a different protocol is required, enter it here. This field is not shown if you choose **User Provided URL**.

Field	Description
Username	Enter the name of a user that is authorized to access JMS destinations on this remote server, if one is required.
Password	Enter the user's password, if a username is required.
Bypass Precheck	By default, Oracle Java Cloud Service will validate the connection to this remote server and continue with the AppToCloud import operation only if it succeeds. Select this check box to skip the validation of this resource.

6. Click **OK** to accept your changes. Repeat these steps for each additional remote SAF context on this page.

   A check mark icon indicates a resource that you have already configured.

7. Continue with the creation of this service instance.

For more information on SAF, refer to one of the following topics:

- Understanding the Store-and-Forward Service in *Administering the Store-and-Forward Service for Oracle WebLogic Server (12.2.1.3)*
- Understanding the Store-and-Forward Service in *Administering the Store-and-Forward Service for Oracle WebLogic Server (12.2.1.2)*
- Understanding the Store-and-Forward Service in *Administering the Store-and-Forward Service for Oracle WebLogic Server (12.1.3)*
- Understanding the Store-and-Forward Service in *Configuring and Managing Store-and-Forward for Oracle WebLogic Server (11.1.1.7)*

Configure a JMS Messaging Bridge Destination

This topic applies only to Oracle Cloud at Customer.

When using AppToCloud to create an Oracle Java Cloud Service instance, you must configure the location of any Java Message Service (JMS) Messaging Bridge Destinations that were present in your source Oracle WebLogic Server domain.

The WebLogic Messaging Bridge is a forwarding mechanism that provides JMS interoperability between different versions of WebLogic Server, and between WebLogic Server and other messaging products. A messaging bridge forwards messages between a pair of bridge destinations. Each bridge destination connects to a JMS destination through a URL. You can choose from these options:

- Select a specific server or cluster in an existing Oracle Java Cloud Service instance. Both service instances must be in the same identity domain.
- Enter the URL of the remote JMS destination. If this is the location of another Oracle Java Cloud Service instance, you must also enable network communication between this service instance and your new one.

To configure a bridge destination:

1. From the Additional Service Details page of the AppToCloud instance creation wizard, expand **JMS Messaging Bridge Destination** and select the name of the bridge destination.
2. Update the configuration for this bridge destination:

Field	Description
Name	The name of the bridge destination you selected. This field is read-only.
JCS Instance	Choose one of these options: • Select another Oracle Java Cloud Service instance to which this domain will connect for this bridge destination. • Select **User Provided URL**.
JNDI Connection URL	Enter the URL that this bridge destination will use to connect to the JMS destination. This field is only applicable if you choose **User Provided URL**.
Cluster	Select a cluster within the target Oracle Java Cloud Service instance. This field is not shown if you choose **User Provided URL**.
Server	Choose one of these options: • Select **All Servers in Cluster**. • Select a specific server within the target Oracle Java Cloud Service instance. This field is not shown if you choose **User Provided URL**.
Protocol	The default protocol that is used for communication between multiple WebLogic Server domains is `t3`. If a different protocol is required, enter it here. This field is not shown if you choose **User Provided URL**.
Username	Enter the name of a user that is authorized to access the JMS destination at this URL, if one is required.
Password	Enter the user's password, if a username is required.
Bypass Precheck	By default, Oracle Java Cloud Service will validate the connection to this URL and continue with the AppToCloud import operation only if it succeeds. Select this check box to skip the validation of this resource.

3. Click **OK** to accept your changes. Repeat these steps for each additional bridge destination on this page.

   A check mark icon indicates a resource that you have already configured.

4. Continue with the creation of this service instance.

For more information on the Messaging Bridge, refer to one of the following topics:

- Understanding the Messaging Bridge in *Administering the WebLogic Messaging Bridge for Oracle WebLogic Server (12.2.1.3)*
- Understanding the Messaging Bridge in *Administering the WebLogic Messaging Bridge for Oracle WebLogic Server (12.2.1.2)*
- Understanding the Messaging Bridge in *Administering the WebLogic Messaging Bridge for Oracle WebLogic Server (12.1.3)*
- Understanding the Messaging Bridge in *Configuring and Managing the Messaging Bridge for Oracle WebLogic Server (11.1.1.7)*

Disable a JMS Messaging Bridge

This topic applies only to Oracle Cloud at Customer.

When using AppToCloud to create an Oracle Java Cloud Service instance, you can optionally disable the migration of a Java Message Service (JMS) Messaging Bridge that was present in your source Oracle WebLogic Server domain.

The WebLogic Messaging Bridge is a forwarding mechanism that provides JMS interoperability between different versions of WebLogic Server, and between WebLogic Server and other messaging products. A messaging bridge forwards messages between a pair of bridge destinations.

1. From the Additional Service Details page of the AppToCloud instance creation wizard, expand **JMS Messaging Bridge**.
2. Select the name of the messaging bridge.

   The bridge's Source Destination and Target Destination are displayed. If the source or target has no value, it is a local JMS destination running within this service instance.
3. Click **Disable** to remove this messaging bridge from the service instance.
4. Continue with the creation of this service instance.

For more information on the Messaging Bridge, refer to one of the following topics:

- Understanding the Messaging Bridge in *Administering the WebLogic Messaging Bridge for Oracle WebLogic Server (12.2.1.3)*
- Understanding the Messaging Bridge in *Administering the WebLogic Messaging Bridge for Oracle WebLogic Server (12.2.1.2)*
- Understanding the Messaging Bridge in *Administering the WebLogic Messaging Bridge for Oracle WebLogic Server (12.1.3)*
- Understanding the Messaging Bridge in *Configuring and Managing the Messaging Bridge for Oracle WebLogic Server (11.1.1.7)*

Import Applications into a Service Instance

This topic applies only to Oracle Cloud at Customer.

After creating an AppToCloud service instance in Oracle Java Cloud Service, perform an import to automatically update the service instance with the applications and other domain resources collected from your on-premises environment.

> **Note:**
>
> It is strongly recommended that you back up your service instance prior to performing an import. If the import fails, you will be able to restore the service instance to a known working state.

1. Access the Oracle Java Cloud Service console.
2. Locate the AppToCloud service instance that you created previously. Click the ☰ Menu icon adjacent to the service instance name and select **AppToCloud Import**.
3. When prompted for confirmation, click **OK**.
4. Click the **Activity** tab.

5. Monitor the progress of the import operation.
6. If the import process fails, you can try running it again:
   a. Return to the **Services** tab.
   b. Click the ☰ Menu icon adjacent to the service instance name and select **Retry AppToCloud Import**.

      The Service Details page is displayed.
   c. Click the **Show Error Details** link for more information on the cause of the failure.
   d. If there was a problem validating a specific domain resource during the precheck phase, the offending resource will be highlighted (an Application Data Source, for example). Select this resource and either modify its configuration or choose the **Bypass Precheck** option.
   e. Click **Submit**.

After a successful import, the applications and other domain resources found in your source domain are deployed to your service instance. You can verify these changes by using one of the administration consoles.

If the output of the Export tool listed any features in your source domain that are not yet implemented in AppToCloud, you can manually configure these features in your service instance.

Recreate On-Premises Domain Resources

This topic applies only to Oracle Cloud at Customer.

Some Oracle WebLogic Server features are not currently supported by the AppToCloud infrastructure. These features must be configured manually after provisioning your Oracle Java Cloud Service instance.

When you run the Export tool to capture an existing domain, the output and activity log from the tool includes messages about features that it detected in your domain but which cannot be automatically provisioned when you create a service instance. The features that are not yet implemented by AppToCloud include:

- Custom users, groups, roles and policies in the security realm
- Keystores
- Coherence clusters
- Custom WebLogic Diagnostics Framework (WLDF) modules and policies

To recreate domain resources in a new service instance:

1. Use the activity log file or report file generated by the Export tool to identify features that you must configure manually:

   ```
   Activity Log for EXPORT
   . . .
   Features Not Yet Implemented Messages
   Any messages about features not yet supported by the export tool
   ```

2. Access the Oracle Java Cloud Service console.

3. Click **Manage this instance** ☰ for the desired service instance and select **Open WebLogic Server Administration Console**.
4. When the console login page appears, enter the WebLogic Server username and password you provided when you created the service instance.
5. Recreate any custom users, groups, roles and policies in the security realm.

   Refer to one of the following topics:

   - Users, Groups, and Security Roles in *Securing Resources Using Roles and Policies for Oracle WebLogic Server (12.2.1.3)*
   - Users, Groups, and Security Roles in *Securing Resources Using Roles and Policies for Oracle WebLogic Server (12.2.1.2)*
   - Users, Groups, and Security Roles in *Securing Resources Using Roles and Policies for Oracle WebLogic Server (12.1.3)*
6. Reconfigure any keystores.

   Refer to one of the following topics:

   - Configuring Keystores in *Administering Security for Oracle WebLogic Server (12.2.1.3)*
   - Configuring Keystores in *Administering Security for Oracle WebLogic Server (12.2.1.2)*
   - Configuring Keystores in *Administering Security for Oracle WebLogic Server (12.1.3)*

   > **Caution:**
   >
   > Do not create or modify keystore files in `MW_HOME`. Any changes you make to this location may be lost when you perform management operations on your Oracle Java Cloud Service instance like applying a patch.
7. Recreate any Coherence clusters.

   Refer to one of the following topics:

   - Configuring and Managing Coherence Clusters in *Administering Clusters for Oracle WebLogic Server (12.2.1.3)*
   - Configuring and Managing Coherence Clusters in *Administering Clusters for Oracle WebLogic Server (12.2.1.2)*
   - Configuring and Managing Coherence Clusters in *Administering Clusters for Oracle WebLogic Server (12.1.3)*
8. Recreate any custom WLDF modules and policies.

   Refer to one of the following topics:

   - Understanding WLDF Configuration in *Configuring and Using the Diagnostics Framework for Oracle WebLogic Server (12.2.1.3)*
   - Understanding WLDF Configuration in *Configuring and Using the Diagnostics Framework for Oracle WebLogic Server (12.2.1.2)*

    - Understanding WLDF Configuration in *Configuring and Using the Diagnostics Framework for Oracle WebLogic Server (12.1.3)*

9. Thoroughly test the applications running on your service instance to ensure they function the same as they did on premises.

You have completed theAppToCloud process and successfully migrated your on-premises applications to Oracle Java Cloud Service.

AppToCloud Command Reference

This topic applies only to Oracle Cloud at Customer.

Oracle provides a collection of graphical and command line tools to automate the process of exporting your on-premises Oracle WebLogic Server environments, so that you can import them to Oracle Java Cloud Service .

Each command line tool is available for both Unix (`.sh`) and Windows (`.cmd`) platforms.

Prior to using any of the AppToCloud tools, set the `JAVA_HOME` environment variable to the directory where you have installed the Java SE Development Kit (JDK).

Do not run the AppToCloud tools using an older JDK version than the version being used to run your WebLogic Server domain. In addition, if your domain is running JDK 6, you must use a separate JDK 7 installation to run the AppToCloud tools.

Depending on the options you specify, the AppToCloud tools may prompt you for various credentials (user names and passwords). When prompted, you can either enter the values for these credentials interactively or pipe them into the standard input stream. For example: `echo "`*`password`*`" | a2c-healthcheck.sh`. However, Oracle does not recommend saving or displaying passwords in plain text.

Topics

- Client
- Health Check
- Export
- Wallet Manager

Client

This topic applies only to Oracle Cloud at Customer.

The Client tool provides a graphical user interface for the Health Check and Export tools.

Usage:

```
a2c-client[.sh | .cmd]
```

By default, the log file for this tool is located in your AppToCloud tools installation at `logs/jcsa2c-client.log`. If this location is not writable, the tool will attempt to write the log file to either the current directory or the temporary directory for this user.

Health Check

This topic applies only to Oracle Cloud at Customer.

The Health Check tool validates a running Oracle WebLogic Server domain to ensure compatibility with the Export tool. It also captures the runtime configuration of the domain and records this information along with the health check results in an archive file.

For detailed instructions, see Check the Health on an On-Premises WebLogic Domain

You must provide a valid set of administrative credentials for the domain. If they are not provided through the available command line options, the program will prompt you for the credentials.

If the connection to the Administration Server requires SSL, set the `LAS_SSL_OPTIONS` environment variable prior to running the Health Check tool. Refer to the contents of the `a2c-healthcheck.sh/cmd` file for an example value.

Usage:

```
a2c-healthcheck[.sh | .cmd] -oh oracle-home -adminUrl admin-url -
outputDir output-dir [-wlstTimeout timeout-millis] [[-adminUser admin-
user] | [-userConfigFile config-file -userKeyFile key-file] | [-
walletDir wallet-dir -adminUserAlias wallet-alias [-walletAutoLogin]]]
```

Example:

```
a2c-healthcheck.sh -oh /u01/app/fmw -adminUrl t3://
myserver.example.com:7001 -adminUser weblogic -outputDir /u01/
jcs_a2c_output
```

Option	Description
`oracle-home`	Top-level installation directory where WebLogic Server is installed. If not set, the value of the `ORACLE_HOME` environment variable is used by default.
`admin-url`	URL to connect to the domain's Administration Server; for example, `t3://myserver.example.com:7001`
`output-dir`	The directory to which the output files should be written; this directory will be created if it does not already exist
`timeout-millis`	The number of milliseconds WebLogic Scripting Tool (WLST) online commands should wait before timing out.
`admin-user` (Optional)	User with administrative rights to the domain. The tool will prompt you for the password.
`config-file` and `key-file` (Optional)	File containing the encrypted credentials for a user with administrative rights to the domain, along with a file containing the encryption key Use the WLST `storeUserConfig` command to generate this file

Option	Description
wallet-dir and wallet-alias (Optional)	The location of an existing Oracle Wallet file and the alias for credentials in this wallet that have administrative rights to the domain. The tool will prompt you for the wallet's password, if it requires one. If you specify both wallet-dir and admin-user, admin-user is ignored. If you specify both wallet-dir and config-file, config-file is ignored.

By default, the log file for this tool is located in your AppToCloud tools installation at logs/jcsa2c-healthcheck.log. If this location is not writable, the tool will attempt to write the log file to either the current directory or the temporary directory for this user.

Export

This topic applies only to Oracle Cloud at Customer.

The Export tool captures an Oracle WebLogic Server domain's configuration and applications and records this information in the archive file created by the Health Check tool.

Optionally, the Export tool can also upload these artifacts to an existing storage container in the Oracle Cloud Infrastructure Object Storage Classic. This tool does not require the domain to be running. For detailed instructions, see Export an On-Premises WebLogic Domain.

You must provide a valid set of Oracle Cloud credentials in order to access the storage container. If they are not provided through the available command line options, the program will prompt you for the credentials.

You can export multiple clusters from a domain only if your Oracle Cloud account includes Oracle Identity Cloud Service . Otherwise, you must specify an individual cluster to export.

Usage:

```
a2c-export[.sh | .cmd] -oh oracle-home -domainDir domain-dir -archiveFile
archive-file [-exportSingleCluster] [-clusterToExport cluster-name] [-
clusterNonClusteredServers new-cluster-name] [-force] [-
cloudStorageContainer storage-container] [[-cloudStorageUser storage-user |
[-walletDir wallet-dir -cloudStorageUserAlias wallet-alias [-
walletAutoLogin]]]
```

Example:

```
a2c-export.sh -oh /u01/app/fmw -domainDir /u01/domains/domain1 -
archiveFile /u01/jcs_a2c_output/domain1.zip -cloudStorageContainer Storage-
MyAccount/MyContainer -cloudStorageUser myuser
```

Option	Description
oracle-home	Top-level installation directory where WebLogic Server is installed. If not set, the value of the ORACLE_HOME environment variable is used by default.

Option	Description
domain-dir	The directory containing the domain to be exported
archive-file	The archive file produced by the Health Check tool
cluster-name (Optional)	If the domain contains multiple clusters, specify the name of an existing cluster to export.
new-cluster-name (Optional)	If the domain does not contain any clusters, specify the name of a new cluster to add to the exported configuration.
storage-container (Optional)	The name of an existing storage container in Oracle Cloud Infrastructure Object Storage Classic to which to upload the generated AppToCloud artifacts • For metered storage subscriptions, use the format Storage-*identitydomain*/*containername* • For non-metered storage subscriptions, use the format *storageservicename*-*identitydomain*/*containername*
storage-user (Optional)	A cloud user that can access your Oracle Cloud Infrastructure Object Storage Classic container. This user is typically the same as those you use to log in to the My Services dashboard. The tool will prompt you for the password.
wallet-dir and wallet-alias (Optional)	The location of an existing Oracle Wallet file and the alias for credentials in this wallet that have access to yourOracle Cloud Infrastructure Object Storage Classic container. The tool will prompt you for the wallet's password, if it requires one. If you specify both wallet-dir and storage-user, storage-user is ignored.

The -force option will attempt a domain export even if the previous health check found errors. It is intended only for expert users.

If the -exportSingleCluster option is used and the source domain contains multiple clusters, the user is prompted to enter the name of the cluster to export. Alternatively, you can specify the cluster name with -clusterToExport.

The -exportSingleCluster option is required if the source domain contains multiple clusters and your Oracle Cloud account does not include Oracle Identity Cloud Service.

If the -clusterNonClusteredServers option is used, and the source domain has both clustered and non-clustered managed servers, all existing clusters will be discarded from the exported domain, and a new cluster will be added.

If the -clusterToExport option is used, and the source domain has both clustered and non-clustered managed servers, all other clusters and non-clustered managed servers will be discarded from the exported domain.

If both the -clusterNonClusteredServers and -clusterToExport options are used, -clusterNonClusteredServers takes precedence.

By default, the log file for this tool is located in your AppToCloud tools installation at `logs/jcsa2c-export.log`. If this location is not writable, the tool will attempt to write the log file to either the current directory or the temporary directory for this user.

Wallet Manager

This topic applies only to Oracle Cloud at Customer.

The Wallet Manager tool is used to create, view, and update Oracle Wallet files. An Oracle Wallet enables you to securely store credentials that you use to run various Oracle commands within a single file for convenience.

By default a password is required to access any of the credentials in the file or to update the credentials in the file.

Usage:

```
a2c-wallet-manager[.sh | .cmd] -walletDir dir-name [-walletAutoLogin]
```

Example:

```
a2c-wallet-manager.sh -walletDir /u01/tools/wallet
Please enter the wallet password: ********
Please re-enter the wallet password: ********
Please specify a command [(l)ist, (a)dd, (u)pdate, (r)emove, (q)uit]: a
Enter the alias for the new credential: wls-domain1-admin
Enter the user for alias wls-domain1-admin: weblogic
Enter the password for alias wls-domain1-admin: ********
Credential with alias wls-domain1-admin has been added to the wallet
Please specify a command [(l)ist, (a)dd, (u)pdate, (r)emove, (q)uit]: l
Wallet Contents:
   1.) User credential: alias = wls-domain1-admin, user = weblogic
   2.) User credential: alias = cloud-account1-storage, user =
bill@example.com
Please specify a command [(l)ist, (a)dd, (u)pdate, (r)emove, (q)uit]: q
```

Option	Description
`dir-name`	Directory in which to create a new wallet or the location of an existing wallet to edit
`walletAutoLogin` (Optional)	Create a wallet that does not require a password to access or edit it. If not set, Wallet Manager prompts you for a password.

After starting a Wallet Manager session, you can issue one or more commands:

Command	Description
`a`	Add a new credential to this wallet. Each credential has an alias, user name and password.
`l`	List the alias and user name for each credential in this wallet

Command	Description
`r`	Remove an existing credential, given its alias.
`u`	Update the user name and password of an existing credential
`q`	Quit Wallet Manager

By default, the log file for this tool is located in your AppToCloud tools installation at `logs/jcsa2c-wallet-manager.log`. If this location is not writable, the tool will attempt to write the log file to either the current directory or the temporary directory for this user.