Oracle® Public Cloud Machine

Using Oracle Java Cloud Service

Release 17.1

E66979-05

April 2017

Documentation for account administrators and service administrators that explains how to provision Oracle Java Cloud Service instances, and ensure reliable functioning of provisioned service instances. This documentation explains how to perform these tasks by using the Oracle Java Cloud Service web interface.

ORACLE

Oracle Public Cloud Machine Using Oracle Java Cloud Service, Release 17.1

E66979-05

Contents

Preface

Using 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

Using 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 Machine Documentation on Oracle Help Center
- *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.

1

Getting Started with Oracle Java Cloud Service on Oracle Public Cloud Machine

This section describes how to get started with Oracle Java Cloud Service for Oracle Cloud Administrators and Tenant Administrators, and Java EE application developers. A reasonable level of familiarity with Oracle WebLogic Server is assumed.

Topics:

About Oracle Java Cloud Service

By using Oracle Java Cloud Service, you can quickly create, configure and manage your Java EE application environment, including an Oracle WebLogic Server domain, in a fraction of the time it would normally take.

Topics:

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 Compute Cloud Service. The service instance includes Oracle WebLogic Server as the application container, and Oracle Traffic Director as the software load balancer. Optionally, you can enable 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 grow your environment when

your application requirements grow. To learn about Oracle Coherence in Oracle Java Cloud Service, see About Oracle Coherence in Oracle Java Cloud Service.

All Oracle Java Cloud Service instances that you create are also preconfigured to an existing Oracle database, which can either be a database deployment in Oracle Database Cloud Service or some other on-premises database.

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

When creating an Oracle Java Cloud Service instance, you have your choice of an environment that's designed for higher availability needs, such as user acceptance testing, staging and production, or an environment that's designed for development and testing.

Depending on your needs, you can choose between two Oracle Java Cloud Service offerings:

- Oracle Java Cloud Service

  Oracle WebLogic Server Standard Edition, Enterprise Edition, and Enterprise Edition with Coherence are supported at this service level.

  Use the Oracle Java Cloud Service offering if you want to provision your application environment and use advanced tooling that allows you to quickly manage the life cycle of your environment in the cloud. The Oracle Java Cloud Service offering provides one-click tooling for all major lifecyle operations such as scaling (except with Standard Edition), backup and restoration, patching, and stopping and starting the service instance.

- Oracle Java Cloud Service—Virtual Image

  Oracle WebLogic Server Standard Edition, Enterprise Edition, and Enterprise Edition with Coherence are supported at this service level.

  The Oracle Java Cloud Service—Virtual Image offering allows you to provision your application environment and use the built-in tools of the underlying product Oracle WebLogic Server (and Oracle Traffic Director, if load balancing is enabled) to manage the life cycle of your environment in the cloud.

  Do not choose the Oracle Java Cloud Service—Virtual Image offering if you intend to use Oracle WebLogic Server 12*c* (12.2.1).

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

- Oracle WebLogic Server 12*c* (12.2.1) with Java Required Files 12*c* (12.2.1) This is the foundation for Oracle Fusion Middleware 12*c* (12.2.1). WebLogic Server 12*c* (12.2.1) is Java EE 7 compatible.

  Note: You must create a full Oracle Java Cloud Service instance, not an Oracle Java Cloud Service — Virtual Image instance, if you want to create a domain partition for your instance.

- Oracle WebLogic Server 12*c* (12.1.3) with Java Required Files 12*c* (12.1.3). This is the foundation for Oracle Fusion Middleware 12*c* (12.1.3). WebLogic Server 12*c* (12.1.3) is Java EE 6 compatible.

- 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). WebLogic Server 11*g* (10.3.6) is Java EE 5 compatible.

You can enable Oracle Coherence in Oracle Java Cloud Service when you provision an environment to run Oracle WebLogic Server 12*c* (12.1.3) or Oracle WebLogic Server 12*c* (12.2.1), and Enterprise Edition with Coherence. When enabled, 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 will have to configure it yourself once the provisioning is complete.

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 for the 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.

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 When you select this edition when creating a service instance, the service will have 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. For Oracle Java Cloud Service— Virtual Image instances based on any edition, this advanced tooling is not supported. See Oracle WebLogic Server Standard Edition.

WebLogic Server Edition	Description
Enterprise Edition	Includes all features and benefits of WebLogic Server Standard Edition, plus: • 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. For Oracle Java Cloud Service — Virtual Image instances based on any edition, this advanced tooling is not 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.
Enterprise Edition with Coherence (Suite)	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. For Oracle Java Cloud Service — Virtual Image instances based on any edition, this advanced tooling is not 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 Enterprise Edition with Coherence if you want to use Oracle Coherence in your Oracle Java Cloud Service instance.

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

For further assistance in making your selection, see Oracle WebLogic Server product page.

About Oracle Java Cloud Service and Oracle Compute Cloud Service

Oracle Java Cloud Service is built on top of infrastructure and functionality that are provided by Oracle Compute Cloud Service.

When you provision an Oracle Java Cloud Service instance as described in Creating an Oracle Java Cloud Service Instance, all the Oracle Compute virtual machines (VMs) required to support your service are provisioned and configured for you.

See About the Deployment Topology of Virtual Machines for information about the virtual machine (VM) deployment topology that is set up and configured in an Oracle Java Cloud Service instance.

When Oracle Coherence is enabled for a service instance, the VM deployment topology includes a Coherence data tier cluster of storage-enabled Managed Servers. See About Oracle Coherence and the Compute Environment in Oracle Java Cloud Service for information about how the VMs and Managed Servers are deployed in a WebLogic Server environment when Coherence is configured.

About Authentication and Secure Application Environments 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. In addition, Oracle Cloud provides a reliable and flexible network security infrastructure to further control how clients and other cloud services access your applications.

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 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.

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.

Oracle Java Cloud Service

You use the Oracle Java Cloud Service Console to create Oracle Java Cloud Service instances, and to access and manage the life cycle of service instances. If Oracle Traffic Director is enabled as a load balancer for a service instance, you can use the Oracle Java Cloud Service Console to disable or enable the load balancer as the need arises. If Oracle Coherence is enabled for a service instance, you use the same console to perform all major lifecycle operations on the Oracle Java Cloud Service—Coherence instance, such as patching and scaling the Coherence data tier.

See Exploring the Oracle Java Cloud Service Console.

Oracle Database Cloud Service

Each Oracle Java Cloud Service instance must be associated with an Oracle database, which can be provisioned as an Oracle Database Cloud Service deployment. Oracle Java Cloud Service relies on this database to host the required Oracle schema.

See Before You Begin with Oracle Java Cloud Service for details about what you must create in Oracle Database Cloud Service before you can provision Oracle Java Cloud Service instances.

See About Oracle Database Cloud Service.

Oracle Compute Cloud Service

Oracle Java Cloud Service is built on top of infrastructure provided by Oracle Compute Cloud Service. Oracle Java Cloud Service instances are hosted on Oracle Compute virtual machines (VMs) that are based on the Oracle Linux 6 (OEL6) 60-GB disk image.

If load balancing is enabled for a service instance, an Oracle Traffic Director instance is hosted on its own virtual machine (VM). For information about the VM deployment topology that is set up and configured for you when you provision an Oracle Java Cloud Service instance.

See About the Deployment Topology of Virtual Machines.

If Oracle Coherence is enabled for a service instance, a VM on the Coherence data tier can have one or more Managed Servers. For information about the VM deployment topology that is set up and configured for you when you provision Oracle Coherence for a service instance.

See About Oracle Coherence and the Compute Environment in Oracle Java Cloud Service.

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. You can enable Oracle Coherence for an Oracle Java Cloud Service instance to use in-memory data grid and caching for Coherence applications deployed to Oracle WebLogic Server environments in the cloud. When Oracle Coherence is enabled for a service instance, Oracle Coherence is configured as a container subsystem within Oracle WebLogic Server in your environment. See Using Oracle Coherence in Oracle Java Cloud Service to learn about creating and managing Oracle Java Cloud Service–Coherence instances.

Oracle Traffic Director

If load balancing is enabled for an Oracle Java Cloud Service instance, Oracle Traffic Director software is used as the load balancer for the service instance. Oracle Traffic Director implements the mod_wl_oh plug-in, which serves as the proxy to route client requests to Oracle WebLogic Server.

If you add a load balancer to a service instance while creating the service instance, Oracle Java Cloud Service sets the WebLogic Plug-in Enabled control in Oracle WebLogic Server. If you add a load balancer to an Oracle Java Cloud Service instance after the service instance was created, you must set the WebLogic Plug-in Enabled control manually.

The Oracle Traffic Director Administration Console can be used for advanced configuration, such as SSL certificate management and routing policy management.

See Configuring a Load Balancer for an Oracle Java Cloud Service Instance.

About the Interfaces to Oracle Java Cloud Service

The entire Oracle Java Cloud Service environment, including the WebLogic domain and cluster, and the virtual machine (VM) 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.	Accessing Oracle Java Cloud Service Exploring 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.	Accessing the Administrative Consoles Used by Oracle Java Cloud Service Oracle WebLogic Server 12*c* (12.2.1) Administration Console Online Help Oracle WebLogic Server 12*c* (12.1.3) Administration Console Online Help Oracle WebLogic Server 11*g* (10.3.6) Administration Console Online Help
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).	Accessing the Administrative Consoles Used by Oracle Java Cloud Service Getting Started Using Oracle Enterprise Manager Fusion Middleware Control in *Adminstering Oracle Fusion Middleware* 12*c* (12.2.1) Getting Started Using Oracle Enterprise Manager Fusion Middleware Control in *Adminstering Oracle Fusion Middleware* 12*c* (12.1.3) Getting Started Using Oracle Enterprise Manager Fusion Middleware Control in *Adminstering Oracle Fusion Middleware* 11*g* (11.1.1.7)

Type of Access	Description	More Information
Load Balancer Console	If load balancing is enabled for an Oracle Java Cloud Service instance, Oracle Traffic Director software is used as the load balancer for the service instance. Although you open the Load Balancer Console in Oracle Java Cloud Service, you are actually using the Oracle Traffic Director (OTD) Administration Console, which can be used to handle traffic routing (for example, you can configure OTD as the SSL termination point for HTTPS requests). To disable or enable the load balancer for a service instance, use the Oracle Java Cloud Service Console.	Accessing the Administrative Consoles Used by Oracle Java Cloud Service Configuring a Load Balancer for an Oracle Java Cloud Service Instance on OPCM Disabling or Enabling the Load Balancer for an Oracle Java Cloud Service Instance
REST API	Use REST endpoints to manage Oracle Java Cloud Service from a terminal, script, or custom program.	*REST API for Oracle Java Cloud Service*
Secure Shell (SSH)	Access the VM of an Oracle Java Cloud Service instance locally or remotely through SSH. After you use SSH to access a VM, you can run WLST and other command-line applications within the VM.	Accessing a VM Through a Secure Shell (SSH)
Virtual Network Computing (VNC)	Remotely access the graphical desktop of a VM in an Oracle Java Cloud Service instance with a combination of VNC client and SSH tunnel.	Accessing a VM Through Virtual Network Computing (VNC)
WebLogic Scripting Tool (WLST)	Use WLST commands locally or remotely, in online or offline mode. • To use WLST commands locally, you have to use SSH to access the VM on which the Administration Server is running, then run the WLST commands from within the VM. • To use WLST commands remotely, you have to create an SSH tunnel to the VM, then run the WLST commands remotely from your computer against the service instance.	Using WLST to Administer an Oracle Java Cloud 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.	Using an IDE to Deploy and Undeploy an Application

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 Load Balancer Console.

Before You Begin with Oracle Java Cloud Service

Before you begin using Oracle Java Cloud Service to create service instances, you must provision an Oracle database. You may also need to satisfy additional prerequisites depending on your requirements.

Topics

Creating a Database

Oracle Java Cloud Service requires access to an existing relational database that contains the standard Oracle Infrastructure schemas. Oracle Java Cloud Service uses Java Database Connectivity (JDBC) to access this database.

Select one of these options:

- Create a database with Oracle Database Cloud Service. See Creating an Oracle Database Cloud Service Database Deployment
- Use an on-premises Oracle database.

When you create an Oracle Java Cloud Service instance, you have to provide the following information about the database:

- One of the following:
  - Name of an Oracle Database Cloud Service database deployment
  - A connection string to an on-premises database in one of these formats: `host:port:SID` or `host:port/service_name`
- User name for the database administrator
- Password for the database administrator, or the password that was specified when the Oracle Database Cloud Service database deployment was created

.

Creating an Oracle Database Cloud Service Database Deployment

Oracle Java Cloud Service can use Oracle Database Cloud Service to host the Oracle schemas required for Oracle Java Cloud Service.

Prior to creating an Oracle Java Cloud Service instance, use your Oracle Database Cloud Service subscription to create a database deployment. As part of the Oracle Java Cloud Service instance creation process, Oracle Java Cloud Service provisions this database deployment with the Oracle schemas.

For information about subscribing to Oracle Database Cloud Service, provisioning database deployments, and using Oracle RAC database deployments, see *Using Oracle Database Cloud Service*.

When you create an Oracle Java Cloud Service instance, you are prompted for the following information about your database deployments:

- Name of the database deployment that is up and running
- Pluggable database name (for Oracle Database 12*c* only)
- Database administrator user name and password

Note: To ensure that you can restore the database for an Oracle Java Cloud Service instance without risking data loss for other service instances, do **not** use the same Oracle Database Cloud Service database deployment with multiple Oracle Java Cloud Service instances. Backups of an Oracle Database Cloud Service database deployment that is used with multiple Oracle Java Cloud Service instances contain data for all the Oracle Java Cloud Service instances. If you restore the database while restoring an Oracle Java Cloud Service instance, data for all the Oracle Java Cloud Service instances is restored.

Creating 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 private key when you create a service instance. See Generating a Secure Shell (SSH) Public/Private Key Pair.

Creating a Remote Backup Destination

Oracle Java Cloud Service uses a remote Network File System (NFS) to store Oracle Java Cloud Service instance backups.

The remote NFS share is not permanently attached to the service instance's WebLogic Server Administration Server virtual machine.

A remote backup destination is not required for service instances created at the Oracle Java Cloud Service—Virtual Image service level.

When creating an Oracle Java Cloud Service instance, you specify the location of the NFS share in one of the following formats:

- *nfs_host_name*:*/share/path/to/backup/folder*
- *nfs_host_ip*:*/share/path/to/backup/folder*

For example: `172.17.0.5:/export/jcs/backup`

> **Note:** The backup folder must be owned by `oracle:oracle` or `uid=1101` and `gid=1000`, and the permissions `rwxr-xr-x` or `rwx------` must be given.

Creating Networks

When creating an Oracle Java Cloud Service instance, you must select a public access network and a database network.

Contact your Oracle Cloud Administrator for the names of two networks:

- Public Access Network: The network that will be used by clients to access applications running on the service instance. It typically uses the format `/`*`tenant_name`*`/public/`*`EoIB_vNet_name`*. For example, `/mytenant/public/vnet-EoIB-vlan3072`.
- Database Network: The network through which the service instance will access the specified Oracle database. For example, `/mytenant/public/IPoIB-dbaccess`.

Accessing Oracle Java Cloud Service on OPCM

You access Oracle Java Cloud Service through a web console.

You can access the console by:

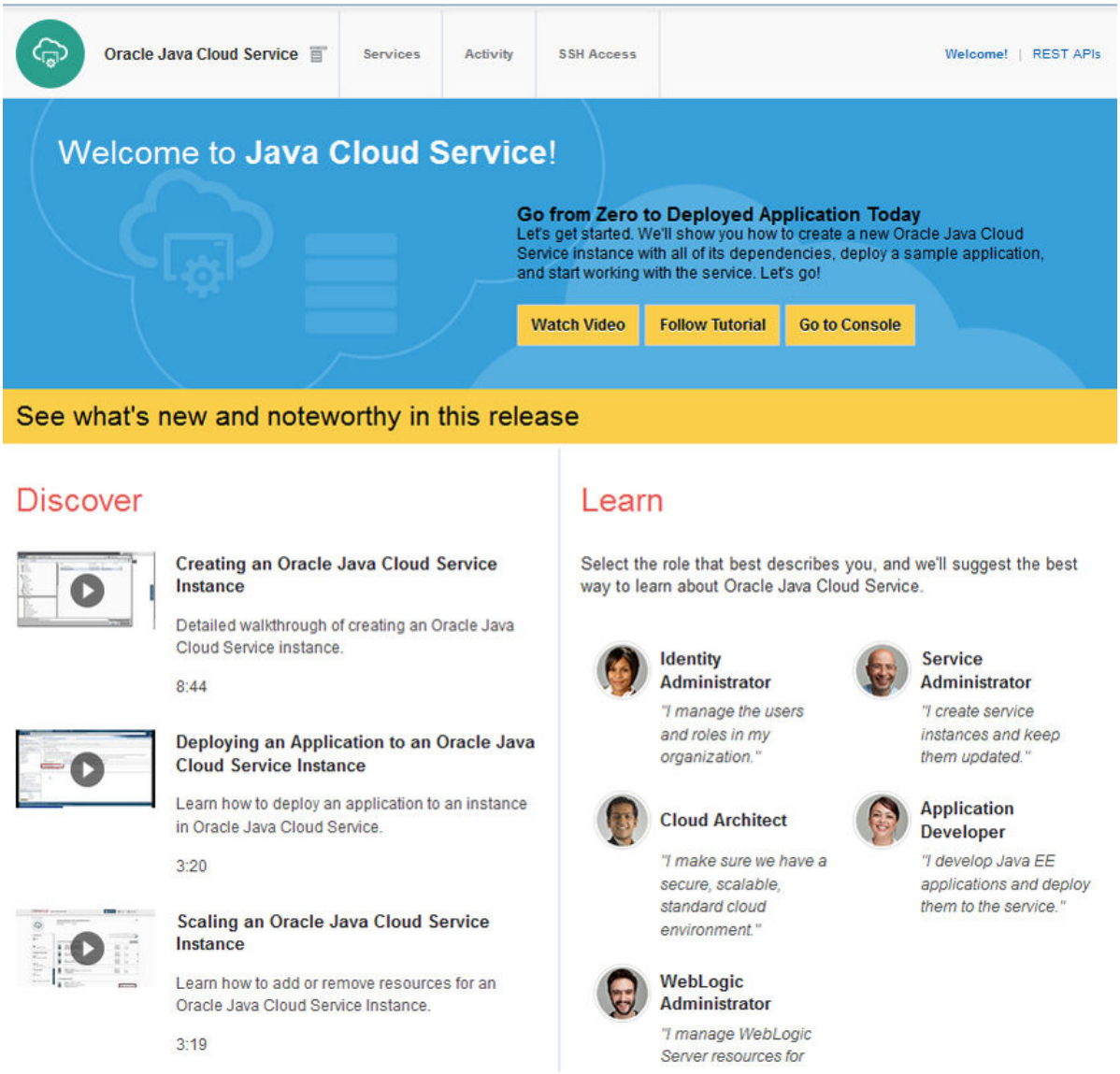
- Using the service URL given to you either in an email or by your administrator.

You must provide an identity domain, user name, and password to sign in. Check your email or contact your administrator for your sign-in credentials.

You can also use the REST API instead of the console. See *REST API for Oracle Java Cloud Service*

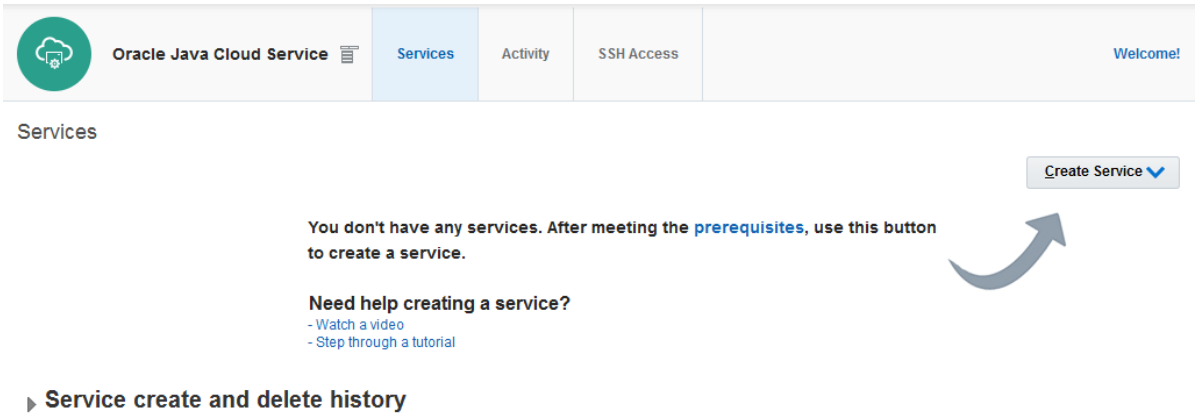
To access Oracle Java Cloud Service:

1. Open your web browser and go to the URL that was provided by your tenant administrator. Sign in with your Oracle Java Cloud Service credentials.
2. When you access Oracle Java Cloud Service the first time for an account, you will see the Welcome page.

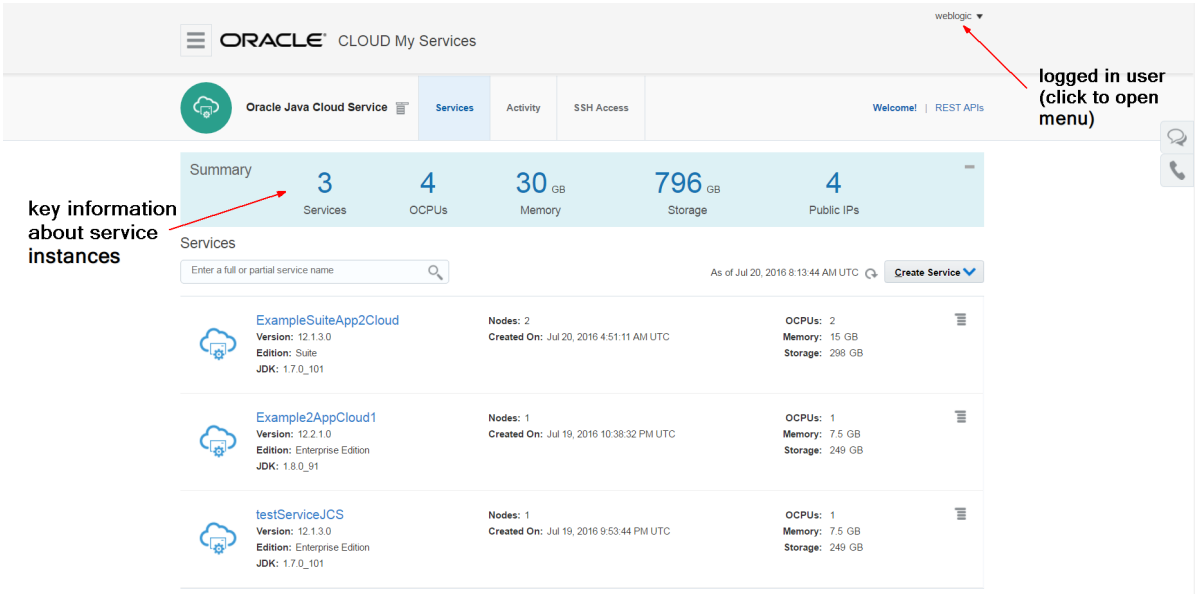


3. Click **Services** or **Go to Console** to display the Services page.

   When no service instances have been created, the Services page will look similar to the following illustration:



If on the other hand, one or more service instances already exist, the Services page will look similar to the following illustration:



For information about the tasks you can perform from the Oracle Java Cloud Service Console, see Exploring the Oracle Java Cloud Service Console.

4. To manage an existing service instance, click the name to open the Oracle Java Cloud Service Instance Overview page.

   For information about the tasks you can perform from the service instance page, see Exploring the Oracle Java Cloud Service Instance Overview Page.

5. To view help for a page, click your user name at the top right corner of that page, select **Help** and then **Help for This Page**.

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.

The following table summarizes the responsibilities of each cloud role in the context of Oracle Java Cloud Service.

Role	Responsibilities
Tenant User	• Use the Oracle Java Cloud Service Console to create and manage service instances within a specific tenant. See the next table for more details. • Consume the network, compute, and storage resources allocated to a specific tenant. **Note:** Tenant Users should contact their Oracle Cloud Administrator or Tenant Administrator for their user sign-in credentials and the URL to access Oracle Java Cloud Service.
Tenant Administrator	• Has permissions similar to a Tenant User. • Create and manage Tenant Users within a specific tenant.
Oracle Cloud Administrator	• Create cloud tenants and configure the tenants for Oracle Java Cloud Service. • Create users and assign them roles. • Create cloud networks and assign them to tenants. • Create remote file systems to use for Oracle Java Cloud Service backups.

The following table summarizes the privileges given to the Tenant User role in Oracle Java Cloud Service.

Description of Privilege	More Information
Can create and delete service instances	Managing the Life Cycle of Oracle Java Cloud Service Instances
Can stop and start service instances, and virtual machines	Stopping and Starting an Oracle Java Cloud Service Instance and Individual VMs
Can suspend and enable service instances by disabling and enabling the load balancer	Suspending an Oracle Java Cloud Service Instance
Can scale, patch, and back up or restore service instances	Scaling an Oracle Java Cloud Service Instance Patching an Oracle Java Cloud Service Instance Backing Up and Restoring an Oracle Java Cloud Service Instance
Can administer load balancers for service instances	Administering the Load Balancer for an Oracle Java Cloud Service Instance

Description of Privilege	More Information
When Oracle Coherence is enabled for a service instance: In addition, the Java Administrator role can: • Remove a Coherence data tier from a service instance (REST API only) • Add a Coherence data tier to an existing service instance (REST API only)	When Oracle Coherence is enabled for a service instance: • Creating an Oracle Java Cloud Service—Coherence Instance • Scaling the Coherence Data Tier of an Oracle Java Cloud Service Instance • Patching an Oracle Java Cloud Service—Coherence Instance • Deleting the Coherence Data Tier of an Oracle Java Cloud Service Instance • Adding a Coherence Data Tier to an Oracle Java Cloud Service Instance

When you create an Oracle Java Cloud Service instance, the following operating system and Oracle WebLogic Server administrative user accounts are created:

User	Description	More Information
VM OS User	The `opc` user has root privileges on the OS running on a VM: • Can connect to a VM through SSH for direct VM-level access to an Oracle Java Cloud Service instance • Can create other OS accounts on a VM using the appropriate OS tool through the SSH interface The `oracle` user cannot be used to log into a machine: • Only has regular user permissions to start and stop Oracle products that have been installed on the machine Note that there are no default passwords for either the `opc` or `oracle` user. SSH access to the VM by the `opc` user is based on the public key provided at the time the Oracle Java Cloud Service instance was provisioned. You provide the private key when you log in to the VM as `opc`. Once logged in, as a root user you can switch to the `oracle` user with: `sudo su - oracle`	Accessing a VM Through a Secure Shell (SSH)

User	Description	More Information
WebLogic Administrator	Can manage Oracle WebLogic Server in Oracle Java Cloud Service	Accessing the Administrative Consoles Used by Oracle Java Cloud Service
	Can access and use the WebLogic Server Administration Console	Using the WebLogic Server Administration Console to Deploy and Undeploy an Application
	Can manage users and groups in the embedded LDAP	Oracle WebLogic Server 12*c* (12.2.1) Administration Console Online Help
	Can configure other identity providers	Oracle WebLogic Server 12*c* (12.1.3) Administration Console Online Help
	Can deploy and undeploy applications using the WebLogic Server Administration Console	Oracle WebLogic Server 11*g* (10.3.6) Administration Console Online Help

Note:

The WebLogic Administrator account and VM OS User accounts are not stored or managed in Oracle Cloud.

You provide the user name and password for the WebLogic Administrator when you create an Oracle Java Cloud Service instance.

The credentials and permissions for the WebLogic Administrator and all end user accounts that the administrator creates are stored and managed in Oracle WebLogic Server.

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
Access the service console	Access the service console and other interfaces to your service.	Accessing Oracle Java Cloud Service Administering Oracle Java Cloud Service Software
Create a service instance	Use the wizard to create a service instance by selecting a level of service, and an Oracle WebLogic Server software release and edition to run on the service instance. See Before You Begin with Oracle Java Cloud Service for the Oracle Cloud services and service-specific information that you must have before you can create an Oracle Java Cloud Service instance.	Creating an Oracle Java Cloud Service Instance About Oracle Java Cloud Service—Coherence Instances

Task	Description	More Information
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.	Deploying and Undeploying Applications for an Oracle Java Cloud Service Instance
Use a load balancer for the service instance	Configure the load balancer (for example, select the load balancing policy). Add a load balancer if one was not enabled during initial service instance creation. Not supported by Oracle Java Cloud Service—Virtual Image instances.	Administering the Load Balancer for an Oracle Java Cloud Service Instance
Manage access to the service instance	Suspend a service instance (if load balancer is enabled). Stop and start a service instance, and virtual machines.	Suspending an Oracle Java Cloud Service Instance Stopping and Starting an Oracle Java Cloud Service Instance and Individual VMs
View heap usage metrics and response times for a service instance	Access the service metrics graph to display heap usage metrics or request response times (if a load balancer is present).	Viewing Service Metrics in Oracle Java Cloud Service
Patch the service instance	Apply a patch or roll back a patch. Not supported by Oracle Java Cloud Service—Virtual Image instances.	Patching an Oracle Java Cloud Service Instance Rolling Back a Patching Operation
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). Not supported by Oracle Java Cloud Service—Virtual Image instances.	Backing Up and Restoring an Oracle Java Cloud Service Instance

Task	Description	More Information
Scale the service instance	Add or remove nodes in preparation for increased or reduced load on a service instance (not supported by Oracle Java Cloud Service—Virtual Image instances). Change the shape of a node or add storage to a node (supported by Oracle Java Cloud Service—Virtual Image instances). When Oracle Coherence is enabled for a service instance: The Coherence data tier is scaled independently of the application tier. You add or remove Coherence data tier nodes by adding or removing one capacity unit at a time.	Scaling Out an Oracle Java Cloud Service Cluster Scaling In an Oracle Java Cloud Service Cluster on OPCM Scaling an Oracle Java Cloud Service Node Scaling the Coherence Data Tier of an Oracle Java Cloud Service Instance
Add a Coherence data tier	Add a Coherence data tier cluster of storage-enabled Managed Servers to an existing service instance (REST API only).	Add a Coherence Data Tier Cluster in *REST API for Oracle Java Cloud Service*
Remove a Coherence data tier	Delete a Coherence data tier cluster from a service instance (REST API only).	Delete a Coherence Data Tier Cluster in *REST API for Oracle Java Cloud Service*
Remove a service instance	Delete a service instance.	Deleting an Oracle Java Cloud Service Instance

2

Managing the Life Cycle of Oracle Java Cloud Service Instances on Oracle Public Cloud Machine

This section describes how to manage the life cycle of Oracle Java Cloud Service instances.

Topics:

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, and optimized for high performance and reliability.

When you create an Oracle 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.
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 (VM). By default, the Managed Servers are named as follows: *first8charsOfDomainName*_server_*n* (where *n* starts with 1 and is incremented by 1 for each additional Managed Server to guarantee unique names).
Cluster	Consists of multiple Oracle WebLogic Server instances 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	Employs Oracle Traffic Director for load balancing to manage routing requests across all Managed Servers and provide failover and replication. It is recommended that you enable the 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 (VMs) 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 VMs 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.
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 12.1.3: 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 12.1.3) Configuring and Managing Coherence Clusters in

Administering Clusters for Oracle WebLogic Server or (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.

Using the Oracle Java Cloud Service Console or REST API, you can stop, start, or restart a service instance or individual VMs. For example, you can stop service instances or individual server VMs to stop metering for these resources, or restart the Administration Server or individual server VMs 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.

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. For information about using 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*.

Task	Description	More Information
Create an Oracle Java Cloud Service instance	Create a new Oracle Java Cloud Service instance by stepping through the Provision New Java Cloud Service wizard.	Creating an Oracle Java Cloud Service Instance
View all Oracle Java Cloud Service instances	View status, resource allocation, and other details for all Oracle Java Cloud Service instances.	Viewing 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.	Viewing 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.	Suspending an Oracle Java Cloud Service Instance

Task	Description	More Information
Stop, start, or restart a service instance or individual server VMs	Stop service instances or individual server VMs to stop metering for these resources. Restart the Administration Server or individual server VMs if reboot is needed.	Stopping and Starting an Oracle Java Cloud Service Instance and Individual VMs
Delete an Oracle Java as Service instance	Manage access to an Oracle Java Cloud Service instance by deleting the service instance.	Deleting an Oracle Java Cloud Service Instance

When Oracle Coherence is enabled for a service instance: See also Workflow for Using Oracle Coherence in Oracle Java Cloud Service.

Design Considerations for an Oracle Java Cloud Service Instance

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

Take the following into consideration when designing a service instance:

Details	Description
Service Level	You can select one of these service levels: • Oracle Java Cloud Service—Virtual Image Developer-level service. Supports Oracle Java Cloud Service instance creation and monitoring. This service level does not support backup and restoration; patching; or scaling. You cannot provision a domain partition if you specify this service level. • Oracle Java Cloud Service Production-level service. Supports Oracle Java Cloud Service instance creation and monitoring; domain partitions; backup and restoration; patching; and scaling. **Note:** The Oracle Java Cloud Service for Fusion Middleware service level is not supported.
Software Release	You can select one of three Oracle WebLogic Server releases: • For Oracle WebLogic Server 11*g* (11.1.1.7) — See *Introducing Oracle WebLogic Server* • For Oracle WebLogic Server 12*c* (12.1.3) — See *Understanding Oracle WebLogic Server* • For Oracle WebLogic Server 12c (12.2.1.2.0) — See *Understanding Oracle WebLogic Server* See About Oracle Java Cloud Service Offerings and Oracle WebLogic Server Software Releases. With Oracle Java Cloud Service you can easily apply patches to an existing service instance, but it does not provide automated tooling to upgrade an existing service instance to a newer release of Oracle WebLogic Server.

Details	Description
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 • Oracle WebLogic Server product page
Compute Shape	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. The larger the compute shape, the greater the processing power and the more memory that is available. See About Default Heap Size Settings for Oracle Java Cloud Service Instances.
Cluster	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 this cluster after you create the service instance. For more information about clusters see: • WebLogic Server Clustering in *Understanding Oracle WebLogic Server (12.2.1)* • WebLogic Server Clustering in *Understanding Oracle WebLogic Server (12.1.3)* • Understanding WebLogic Server Clustering in *Using Clusters for Oracle WebLogic Server (10.3.6)*
Domain Partition	A WebLogic Server 12*c* 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 service level • The Standard Edition of Oracle WebLogic Server • The 11*g* release of WebLogic Server

Details	Description
Client Access	By default, access to an Oracle Java Cloud Service instance can only be done 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. See: • Enabling HTTP Access to an Oracle Java Cloud Service Instance • Understanding the Default Access Ports 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.
Coherence Data Tier	If you choose to provision an Oracle Coherence data tier in your service instance, Oracle Java Cloud Service creates a second 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 selecting a predefined capacity unit or by defining your own custom one. The capacity unit controls the size of the cache in GB, the number of VMs in the cluster, and the compute shape of these VMs. See About Capacity Units for Oracle Java Cloud Service—Coherence Instances. After a service instance is created, you can increase cache capacity by adding one capacity unit at a time. See Scaling the Coherence Data Tier of an Oracle Java Cloud Service Instance. Oracle Java Cloud Service can only provision a Coherence data tier in your service instance if you select Enterprise Edition with Coherence (Suite).
Database	Oracle Java Cloud Service requires access to an existing relational database that contains the standard Oracle Infrastructure schema. Oracle Java Cloud Service uses Java Database Connectivity (JDBC) to access this database. Create an Oracle Database Cloud Service database deployment or use an existing on-premises Oracle database. During the service creation process, Oracle Java Cloud Service will provision Oracle Database Cloud Service with the required schema. If you select some other Oracle database, you must manually create this schema prior to creating the Oracle Java Cloud Service instance. See Creating a Database Deployment in *Using Oracle Database Cloud Service*.

Details	Description
Load Balancer	The load balancer is an instance of Oracle Traffic Director (OTD). The load balancer routes requests it receives from clients to the WebLogic Servers configured in an Oracle Java Cloud Service instance. Using a load balancer is recommended if you are configuring more than one managed server. A load balancer also gives you the ability to suspend an Oracle Java Cloud Service instance temporarily to perform routine maintenance. You can configure a second load balancer while provisioning the service instance, or add one later after the service instance is provisioned. Each load balancer is assigned a separate public IP address. A configuration with two active load balancers provides these advantages: • High availability if a load balancer becomes unavailable. • Multiple load balancers for greater application throughput. However, in this configuration the client is responsible for utilizing both load balancer nodes, and for failing over to another node should one become unavailable. You cannot remove the first load balancer after adding it to your Oracle Java Cloud Service instance. You can disable it, but you will be charged for having it in your service instance. See: • Suspending an Oracle Java Cloud Service Instance • Disabling or Enabling the Load Balancer for an Oracle Java Cloud Service Instance • Adding a Load Balancer to an Oracle Java Cloud Service Instance • Adding a Second Load Balancer to an Oracle Java Cloud Service Instance

Creating an Oracle Java Cloud Service Instance on OPCM

To create a new Oracle Java Cloud Service instance, use the Provision New Java Cloud Service wizard as described in the following sections.

- Invoking the Provision New Java Cloud Service Wizard
- Specifying Basic Service Information
- Configuring Size and Shape
- Configuring WebLogic Server Access
- Configuring the Coherence Data Tier
- Configuring Advanced Settings
- Configuring the Databases
- Configuring the Load Balancer
- Configuring Networks for OPCM
- Configuring Backups for OPCM
- Confirming Your Oracle Java Cloud Service Instance Creation

Video

Tutorial

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

Invoking the Provision New Java Cloud Service Wizard

To create an Oracle Java Cloud Service instance from the Oracle Java Cloud Service Console, you use the Provision New Java Cloud Service wizard.

To invoke the Provision New Java Cloud Service wizard:

1. Navigate to the Oracle Java Cloud Service Console.
2. Click **Create Service** .

The Provision New Oracle Java Cloud Service wizard is invoked.

Specifying Basic Service Instance Information

Enter basic information for your service instance: service name, service level, metering frequency, software release, and software edition.

Note: You cannot change the following choices after you have created the service instance:

- Service Name
- Description
- Service Level
- Metering Frequency
- Software Release
- Service Edition

Complete the following fields:

Field	Description
Service Name	Specify the name for the Oracle Java Cloud Service instance. The service instance name: • Must contain one or more characters. • Must not exceed 50 characters. • Must start with an ASCII letter: a to z , or A to Z. • Must contain only ASCII letters or numbers, or hyphens. • 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.
Service Level	Select a service level from the drop-down menu. • Oracle Java Cloud Service—Virtual Image Developer-level service. Supports Oracle Java Cloud Service instance creation and monitoring. This service level does not support backup and restoration; patching; or scaling. You cannot provision a domain partition if you specify this service level. • Oracle Java Cloud Service Production-level service. Supports Oracle Java Cloud Service instance creation and monitoring; domain partitions; backup and restoration; patching; and scaling. The Oracle Java Cloud Service for Fusion Middleware service level is not supported.
Metering Frequency	Select a metering frequency from the drop-down menu. • Hourly—Pay only for the number of hours used during your billing period. • Monthly—Pay one price for the full month irrespective of the number of hours used. **Note:** 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.

Field	Description
Service Release	Select the software release from the drop-down menu. • **Oracle WebLogic Server 11g (11.1.1.7)** Includes Oracle WebLogic Server 11g (10.3.6) and Oracle Fusion Middleware 11g (11.1.1.7) . • **Oracle WebLogic Server 12c (12.1.3.0)** • **Oracle WebLogic Server 12c (12.2.1.2.0)** You must select either version of Oracle WebLogic Server 12*c* if you want to use Oracle Java Cloud Service to create and manage an Oracle Coherence data tier in your service instance. If you select Oracle WebLogic Server 11*g* and want to use Oracle Coherence you must configure it manually after creating the service instance. The WebLogic Server software runs on the Oracle Linux 6.6 platform.
Service Edition	Select the service edition you want to run on your service instance. You can select one of the following editions of WebLogic Server: • **Standard Edition** • **Enterprise Edition** • **Enterprise Edition with Coherence**

Specifying the Service Instance Details

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

Topics

Configuring Size and Shape

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

Complete the following fields:

Size and Shape Details	Description
Compute Shape	Select the compute shape. The compute shape is the number of Oracle Compute Units (OCPUs) and amount of memory (RAM) that you want to allocate to the VMs that will contain the WebLogic Server Administration Server and Managed Servers. All-purpose compute shapes 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 Memory-intensive compute shapes 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 For more information about compute shapes, contact your Oracle Cloud Administrator.
SSH Public Key	Specify the public key for the secure shell (SSH). This key will be used for authentication when connecting to the Oracle Java Cloud Service instance using an SSH client. Click **Edit** to display the Public key input for VM access dialog and specify the public key using one of the following methods: • Select **Key file name** and click **Browse** to select a file that contains the public key for the secure shell (SSH). • Select **Key value** and paste in the value of the public key. If you paste in a value, make sure the value does not contain line breaks or end with a line break. • Select **Create a New Key** to generate a public/private key pair immediately. Use this method if you did not create a key pair before you started provisioning the service instance.
Cluster Size	Select the cluster size — the number of Managed Servers that you want to configure in a cluster. The valid values are: `1`, `2`, `4`. • This field is not valid if you selected WebLogic Server Standard Edition. In this case, one Managed Server is configured, by default. • If you configure more than one Managed Server in the cluster, Oracle recommends that you enable the load balancer. You can also scale out the cluster after provisioning the service instance.

Configuring WebLogic Server Access

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

Complete the following fields:

Access Details	Description
Local Administrative Username	Specify the username for the WebLogic Administrator. 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 change the user name through the WebLogic Server Administration Console after the service instance is provisioned.
Password	Specify a password for the WebLogic administrator and confirm the password. If you have specified an Oracle Database Exadata Cloud Service database deployment, you must specify a strong password conforming to the following requirements: The 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 use the previous requirements and you do not specify an Oracle Database Exadata Cloud Service database deployment, provisioning will not fail. However, as a best practice, use the strong password requirements. Previous requirements are: • 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.

Configuring the Coherence Data Tier

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
Capacity Unit	Select the default capacity unit that's right for your environment: • **Basic**—Provides 1.5 GB of primary cache capacity. One VM with shape OC3 (1 OCPU total). • **Small**—Provides 4.5 GB of primary cache capacity. Three VMs with shape OC3 (3 OCPUs total). • **Medium**—Provides 10 GB of primary cache capacity. Three VMs with shape OC4 (6 OCPUs total). • **Large**—Provides 22 GB of primary cache capacity. Three VMs with shape OC5 (12 OCPUs total). To create a service instance without enabling Coherence, select **None**.
Cache Size	You can configure a default capacity unit or select **Advanced**. This selection displays the amount of primary cache storage to allocate for Coherence by one capacity unit, based on the default capacity unit configuration you select or the custom (Advanced) configuration you define, and the general rule of using 1/3 of Managed Server heap for primary cache storage. Only one capacity unit of VMs and Managed Servers is created and configured upon initial provisioning of a service instance. After a service instance is created, you can increase cache capacity by adding one capacity unit at a time.
Shape	For Advanced configuration only: Select the number of Oracle Compute Units (OCPUs) and amount of memory (RAM) that you want for the VMs in your custom capacity unit.
Nodes	For Advanced configuration only: Select the number of VMs you want to be configured by one custom capacity unit. Valid values are 1 to 3. Select 3 for Coherence high availability. This is because Coherence high availability is maintained when three or more VMs are configured.
Managed Servers Per Node	For Advanced configuration only: Select the number of Managed Servers to be started on each VM by your custom capacity unit. Up to 4 Managed Servers are allowed if you use the OC3 shape. Using more servers can improve concurrency and memory management but can require more processors. The maximum heap size that will be configured is 16 GB.

Configuring Advanced Settings

You can specify domain partitions, console access, and sample application deployment.

Complete the following table:

Size and Shape Details	Description
Domain Partitions	Select the number of partitions you want for your service. The choices are `0`, `1`, `2`, or `4`. If you are configuring an Oracle Java Cloud Service — Virtual Image instance and you select WebLogic Server 12.2.1 as the version, this field will not appear. This field will also not appear if you select Standard Edition as the edition. You cannot create a domain partition in these cases.
Deploy Sample Application	(Optional) Deploy the sample application. A sample application, `sample-app.war`, is deployed automatically to the Managed Servers in your instance and started. If you do not want to automatically deploy the application, deselect the checkbox.

Configuring the Databases

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
For Oracle Required Schema — Name	Specify the name of the Oracle Database Cloud Service deployment you want to use for storing the required database schema for your service instance. Alternatively, specify a connection string to some other on-premises Oracle database. The connection string must be in one of the following formats: • *`host:port:SID`* • *`host:port/service_name`* **Note:** To ensure that you can restore the database for an Oracle Java Cloud Service instance without risking data loss for other service instances, do **not** use the same Oracle Database Cloud Service database deployment with multiple Oracle Java Cloud Service instances. Backups of an Oracle Database Cloud Service database deployment that is used with multiple Oracle Java Cloud Service instances contain data for all the Oracle Java Cloud Service instances. If you restore the database while restoring an Oracle Java Cloud Service instance, data for all the Oracle Java Cloud Service instances is restored. Only Oracle Database Cloud Service database deployments that are up and running are available in the list. If your Oracle Database Cloud Service database deployment is In Progress, it will not appear in the list. The database deployment you choose must not be configured with a **Backup Destination** set to None. You must select one of the other available backup options in the database deployment prior to creating this service instance. **Backup Destination** is not applicable to Oracle Database Cloud Service — Virtual Image database deployments.

Database Details	Description
PDB Name	Enter the name of the pluggable database for Oracle Database 12*c*. If not specified, the PDB name provided when the Oracle Database Cloud Service database deployment was created will be used. This value does not apply to Oracle Database Cloud Service database deployments running Oracle Database 11*g*.
Administrator Username	Enter the name of the database administrator. For database deployments associated with Oracle Java Cloud Service instances based on Oracle WebLogic Server 11*g* , this value must be set to a database user with DBA role. You can use the default user `SYSTEM` or any user that has been granted the `DBA` role. For database deployments associated with Oracle Java Cloud Service instances based on Oracle WebLogic Server 12*c* (12.1.3 or 12.2.1), this value must be set to a database user with `SYSDBA` system privileges. You can use the default user `SYS` or any user that has been granted the `SYSDBA` privilege.
Password	Enter the database administrator user password.

Configuring the Load Balancer

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	Select **Yes** or **No** to enable or disable the Oracle Traffic Director (OTD) load balancer. This choice defaults to **Yes** when you configure more than one Managed Server for Oracle Java Cloud Service.
Compute Shape	Select the number of Oracle Compute Units (OCPUs) and amount of RAM memory that you want to allocate for the VM(s) that will run the load balancer. All-purpose compute shapes 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 Memory-intensive compute shapes include: • OC1M: 1 OCPU and 15 GB memory • OC2M: 2 OCPUs 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 For more information about compute shapes, contact your Oracle Cloud Administrator.

Load Balancer Details	Description
Add Another Active OTD Node	(Optional) Add another OTD node during provisioning. The default is **No**. If you don't add an OTD node during provisioning, you can do so after provisioning.
Load Balancer Policy	Select one of the following policies: • **Least Connection Count**—Passes each new request to the Managed Server with the least number of connections. This policy is useful for smoothing distribution when Managed Servers get bogged down. 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.

Configuring Networks for OPCM

You must assign a public access network and a database network to your Oracle Java Cloud Service instance.

Complete the following fields:

Networks	Description
Public Access Network	Select the name of an existing network from the drop-down list. The public access network will be used by clients to access applications running on the service instance.
Database Network	Select the name of an existing network from the drop-down list. The database network will be used by your service instance to access the specified Oracle database.

Configuring Backups for OPCM

You must configure a remote backup destination for your Oracle Java Cloud Service instance.

Oracle Java Cloud Service uses a remote Network File System (NFS) to store Oracle Java Cloud Service instance backups. A remote backup destination is not required for service instances created at the Oracle Java Cloud Service—Virtual Image service level.

Complete the following field:

Backup Location	Description
NFS Remote Backup	Specify the location of an existing NFS share in one of the following formats: • *nfs_host_name*:*/share/path/to/backup/folder* • *nfs_host_ip*:*/share/path/to/backup/folder* For example: `172.17.0.5:/export/jcs/backup`

Confirming Your Oracle Java Cloud Service Instance Creation

On the Confirmation page of the Provision New Java Cloud Service 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**.

> **Note:** When creating a service instance that has Oracle Coherence enabled: Upon creation of the service instance, the initial set of Managed Servers configured for the Coherence data tier is listed as a scale out operation in the Activity section on the Topology page. See Viewing Scaling Requests for information about how to access the Activity section and view details for a specific scaling operation.

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.

After the service instance has been created, you can view the system messages logged during the creation process, including error messages. Click **Instance create or 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.

You can view the sample application that was deployed automatically when the service instance was created (unless you elected not to deploy it). See About the Sample Application Deployed to an Oracle Java Cloud Service Instance.

Discovering an Oracle Java Cloud Service Instance on OPCM

You can use Oracle Enterprise Manager to discover your Oracle Java Cloud Service instance.

Discovery creates a JMX connection to the Administration Server running the Oracle Java Cloud Service instance. The discovery process creates targets in Oracle Enterprise Manager for processes such as the Administration Server, Managed Servers, and applications running on the domain. As a result of discovery, you will be able to monitor the service, and perform operations such as starting and stopping the service.

To discover your Oracle Java Cloud Service instance:

1. Sign in to the Oracle Enterprise Manager Cloud Control Console on your Oracle Public Cloud Machine with the URL and credentials that your administrator gave you.
2. From the Enterprise Menu, choose Cloud and then Self Service Portal.

   The Oracle Public Cloud Machine Self Service Portal page is displayed.
3. Select Manage **Exalogic Private Cloud** in the Self Service Portal.

   Note: Ensure that the Oracle Compute Site service manager endpoint URL is set before the Exalogic Private Cloud page appears when you select Manage **Exalogic Private Cloud**.
4. Click on the Oracle Java Cloud Service **Actions** menu and select **Monitor your Oracle Java Cloud Services in Enterprise Manager**.

   The Oracle Java Cloud Services page is displayed.
5. Click **Discover** in the Targets column of the table that is displayed on the Oracle Java Cloud Services page.

   The Discovery page is displayed, including the Administration Server details of the WebLogic Domain and the Managed Servers in the domain.

   On the Discover page:

   a. Copy and paste the SSH public and private keys for the managed server host machines into the Cloud SSH Public Key fields.

   b. Enter the user name and password of the Administration Server host in the WebLogic Domain Information section.

   c. Click **Submit**, which is located above the table and on the right.

   An Oracle Java Cloud Service Discovery job is submitted and a confirmation message is displayed. You can go to Enterprise — Job — Activity to track the progress of the job. When the job has been completed, you will see a link in the Targets field on the Oracle Java Cloud Services page in Enterprise Manager, to the Oracle Java Cloud Service Console in the cloud.

You can then monitor the service and perform other operations.

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 on the URL under the Additional Information section of the page.

You can also access the application from a web browser as you would any other application. The application context is `sample-app`.

Type a URL in the following format into the browser's address bar:

`https://IP_address:port/sample-app`

For example:

`https://192.0.2.1:443/sample-app`

`IP_address` is the IP address of the VM through which you access the application:

- If your service instance has a load balancer, use the IP address of the VM where the load balancer is running.
- Otherwise, use the IP address of the VM where the Administration Server and first Managed Server is running.

`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.

See:

-

Accessing an Application Deployed to an Oracle Java Cloud Service Instance

Understanding 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.

Viewing All Oracle Java Cloud Service Instances on OPCM

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 service name.

To view all Oracle Java Cloud Service instances:

1. Navigate to the Oracle Java Cloud Service Console.
2. Click on **Services** if this tab is not already selected.

Your Oracle Java Cloud Service instances are listed on this page.

See Exploring the Oracle Java Cloud Service Console.

Viewing Activity for Service Instances in an Identity Domain

Use the Activity page to view activities for Oracle Java Cloud Service instances in your identity domain. You can restrict the list of activities displayed by using search filters.

To view activities for your Oracle Java Cloud Service instances:

1. Navigate to the Oracle Java Cloud Service Console.
2. Click the **Activity** tab.

   The Activity page is displayed.
3. Use the options in the Search Activity Log section to obtain the results that meet your needs.
4. When finished, click the **Services** tab.

See Exploring the Platform Services Activity Page in *Using the Platform Services Console*.

Viewing Detailed Information About an Oracle Java Cloud Service Instance on OPCM

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. Navigate to the Oracle Java Cloud 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.

See Exploring the Oracle Java Cloud Service Instance Overview Page.

Viewing Service Metrics in Oracle Java Cloud Service on OPCM

Oracle Java Cloud Service provides a graph view that enables you to view heap usage metrics and response times for service instances.

Topics

- About Service Metrics for Oracle Java Cloud Service

- Viewing Heap Usage and Load Balancer Response Time for Oracle Java Cloud Service

About Service Metrics for Oracle Java Cloud Service

You can use the Service Metrics graph to view heap usage or response time (if a load balancer is present) for your Oracle Java Cloud Service instance.

Heap usage percentage data appears numerically on the Oracle Java Cloud Service Instance Overview page, but the Service Metrics graph enables you to visualize this information.

Accessing the Service Metrics Graph

1. Select the  Health Check icon in the icon bar.

   Heap usage is displayed for each server and the load balancer.

2. Click the heap usage number.

   The Service Metrics graph is displayed.

Information the Graph Displays

The graph gives you the option of displaying heap usage metrics or request response times over the past 60 minutes.

- Heap Usage

  —You can select the server(s) whose heap usage interests you, and the data are displayed in a graph of percentage of heap used versus time.

- Average Response Time

  —Request response time is displayed if a load balancer is present. The graph displays response time in milliseconds (response time) versus elapsed time.

Refreshing Data

You can refresh graph data by clicking on the  **Refresh** icon on the upper right corner of the graph. The graph provides 5–minute snapshots of the data. The graph only fetches the data the first time you enter the page or when you refresh the page. If you exit the page then access it again, previous data are displayed. You must click **Refresh** to see new data.

Note: The graph takes some time to load the data. If it doesn't appear right away, wait.

Viewing Heap Usage and Request Response Time for Oracle Java Cloud Service

The Service Metrics graph displays either heap usage or response time (if a load balancer is present) for an Oracle Java Cloud Service instance.

To view the Service Metrics graph:

1. Access the Service Metrics graph from the Instance Overview page.
2. On the Service Metrics graph, select one of the following graph types:

- Heap Usage (default)
- Average Response Time (if a load balancer is present)

The graph displays either heap usage or response time data.

3. In the Heap Usage Selection field, choose one or more of the following:
   - All servers (default)
   - Admin Server
   - Overall Managed Servers
   - Individual managed server (*serverName*_server_*n*)

   A graph of your selections is displayed.
4. To return to the Oracle Java Cloud Service Instance Overview page, click the name of the service instance.

Suspending an Oracle Java Cloud Service Instance on OPCM

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.

See Disabling or Enabling the Load Balancer for an Oracle Java Cloud Service Instance.

For more information about enabling and disabling the load balancer, see .

Note:

If a load balancer is not configured, you cannot suspend the Oracle Java Cloud Service instance. See Adding a Load Balancer to an Oracle Java Cloud Service Instance.

Stopping and Starting an Oracle Java Cloud Service Instance and Individual VMs

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 VMs.

Topics

About Stopping and Starting an Oracle Java Cloud Service Instance and Individual VMs

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 VMs.

> **Note:** The stop and restart procedures affect entire VMs. If you want to shut down the WebLogic Administration Server or Managed Server processes running on the VMs, without stopping the VMs, see Shutting Down and Starting the WebLogic Server Managed Servers and Administration Server Processes on VMs. You might want to do this if you have other processes besides the servers running on the VMs and you do not want to shut down these other processes.

This topic describes why you would want to stop or start a service instance, or stop, start, or restart individual server or load balancer VMs. This topic also describes what happens when service instances are stopped and started, and how to monitor these operations.

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 VMs. Metering for those resources stops.

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 VMs 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 VMs individually while the service instance is stopped.
- **Starting the service instance**: All VMs 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 VMs individually. You cannot do the same for Coherence Data Tier VMs individually.

Why Stop, Start, or Restart an Administration Server, Managed Server, or Load Balancer VM

If an Oracle Java Cloud Service instance is running:

- You can restart the VMs 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 VM, then starting it immediately.
- You can stop the VMs 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 VMs except for one, you might want to stop the load balancer VM because it is not needed.
- You can start a Managed Server or load balancer VM 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 WebLogic Server 12c (12.1.3) or 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 to the Coherence Data Tier When a Service Instance is Stopped or Started

All VMs in a Coherence Data Tier, 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 Tier Managed Server VMs 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 Viewing Activity for Service Instances in an Identity Domain.

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.

Stopping and Starting an Oracle Java Cloud Service Instance

You can stop and start an Oracle Java Cloud Service instance through the Oracle Java Cloud Service Instance Overview page or the Topology page.

Note: When you stop an Oracle Java Cloud Service instance, all data in the Coherence cache (if present) will be lost.

To stop or start an Oracle Java Cloud Service instance:

1. Navigate to the Oracle Java Cloud Service Instance Overview page.
2. Complete one of the following actions:
   - Click the  Menu icon adjacent to the service name and select **Start** or **Stop**.
   - Click the  Start or Stop icon.

A confirmation dialog is displayed.

3. Click **OK** in the confirmation dialog.

   A yellow status icon is displayed adjacent to the service instance icon while the service instance is in the process of stopping or starting.

   When the operation completes, the Oracle Java Cloud Service instance is stopped or started. The yellow icon is no longer displayed. A red icon is displayed when the service instance is stopped. On the Activity page, the entry for the service shows that the stop or start operation has ended.

Restarting the Administration Server VM

You can restart the VM on which the Administration Server is running, in an Oracle Java Cloud Service instance that is in a running state.

> **Note:** You can restart the Administration Server only if you specified Oracle WebLogic Server 12c (12.1.3) or 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.

To restart the Administration Server:

1. Navigate to the Oracle Java Cloud Service Instance Overview page.
2. Click the Menu icon adjacent to the Administration Server row and select **Restart**.

   A confirmation dialog is displayed.

3. Click **OK** in the confirmation dialog.

   A yellow status icon is displayed next to the service icon.

   The Administration Server VM starts. The yellow icon is no longer displayed.

Stopping, Starting, and Restarting Managed Server and Load Balancer VMs

You can stop, start, and restart the VMs on which the Managed Servers or the load balancer are running in an Oracle Java Cloud Service instance if the service instance is in a running state. Restarting a Managed Server or load balancer VM is the same as stopping it, then starting it.

> **Note:** You can stop, start, and restart individual Managed Servers and the load balancer only if you specified Oracle WebLogic Server 12c (12.1.3) or 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.

To stop, start, or restart the Managed Server or load balancer VM:

1. Navigate to the Oracle Java Cloud Service Instance Overview page.

2. Click the Menu icon to the right of the Managed Server or load balancer row and select **Stop**, **Start**, or **Restart**.

   A confirmation dialog is displayed.

3. Click **OK** in the confirmation dialog.

   The Managed Server or load balancer VM is stopped, started, or restarted.

Deleting an Oracle Java Cloud Service Instance on OPCM

When you no longer require an Oracle Java Cloud Service instance, you can delete it. Your account is no longer charged for the instance.

The service must be running before you attempt to delete it. See Stopping and Starting an Oracle Java Cloud Service Instance.

When you delete an Oracle Java Cloud Service instance:

- Resources such as IP addresses are removed.
- If you specified an Oracle Database Cloud Service instance when you created the Oracle Java Cloud Service instance, the database instance is not deleted. Only the database repository and schemas are deleted. Your account continues to be charged for the database instance. You might want to retain this database instance for use with other service instances; otherwise, you must delete the database instance manually to avoid being charged for it. See Deleting a Database Deployment in *Using Oracle Database Cloud Service*.
- If you specified the location of an on-premises database when you created the Oracle Java Cloud Service instance, no modifications are made to this database.
- The NFS share where backups are stored is not modified or deleted.

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 Database Cloud Service 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 deployment cannot be reached to delete the database schemas. If enabled, you may need to delete the associated database schemas manually on the database deployment 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 **Retry Delete** 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 **Service creation and deletion 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.

Exploring 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

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 about Oracle Java Cloud Service by selecting your role to customize your learning path.
- 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
Services	Click to navigate to the Oracle Java Cloud Service Console. See Exploring the Oracle Java Cloud Service Console.
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 Exploring the Oracle Java Cloud Service Console.
Welcome!	Click to redisplay this page.
REST API	Click to open the Oracle Java Cloud Service REST API documentation.
Discover	Watch videos that demonstrate how to perform key tasks.
Learn	Click your role to customize your learning path.

Exploring 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

Use Oracle Java Cloud Service Console to perform the tasks described in the following topics:

- Creating an Oracle Java Cloud Service Instance
- Viewing All Oracle Java Cloud Service Instances
- Viewing Detailed Information About an Oracle Java Cloud Service Instance
- Viewing Service Metrics in Oracle Java Cloud Service
- Viewing Activity for Service Instances in an Identity Domain
- Managing SSH Access for an Oracle Java Cloud Service Instance
- Deleting an Oracle Java Cloud Service Instance
- Accessing the Administrative Consoles Used by Oracle Java Cloud Service
- Configuring a Load Balancer for an Oracle Java Cloud Service Instance
- Stopping and Starting an Oracle Java Cloud Service 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 **Services** to view the Oracle Cloud Service Console home page.

There are additional tabs on the Oracle Java Cloud Service Console: **Activity** and **SSH Access**. Refer to these topics in *Using the Platform Services Console* for more information about these tabs:

- Exploring the Platform Services Activity Page
- Exploring the Platform Services SSH Access Page

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 Service** and access information about the prerequisites and steps for creating an instance.

Element	Description
 (Next to the word Oracle)	Click and select a choice from the menu to open the service console for one of the Oracle Cloud Services to which you subscribe.
 (Next to the service name)	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	Click to redisplay this page.
Activity	Click to view all operations performed on your service instances. See Viewing Activity for Service Instances in an Identity Domain.
SSH Access	Click to manage SSH keys for your service instances. See Managing SSH Access for an Oracle Java Cloud Service Instance.
Welcome!	Click to return to the Welcome page.
REST API	Opens the REST API documentation.
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.

Element	Description
Public IPs (Summary panel)	Total number of IP reservations allocated across all Oracle Java Cloud Service instances.
Services (heading)	All Oracle Java Cloud Service instances in the identity domain.
Enter a full or partial service name	Enter a full or partial service instance name to filter the list of service instances to include only the services that contain the string in their service name.
	Click to refresh the page. The date and time the page was last refreshed is displayed adjacent to this button.
Create Service	Create a new Oracle Java Cloud Service Instance. See Creating an Oracle Java Cloud Service Instance.
	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 is failed to be 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. Valid values include: 12.2.1.2, 12.1.3.0 or 11.1.1.7.
Edition	Software edition. Valid values include: Standard, Enterprise, or Suite.
JDK	JDK version used by the Oracle Java Cloud Service instance.
Nodes	Number of nodes allocated for the Oracle Java Cloud Service instance. 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).
Load Balancer	Flag indicating that the load balancer is configured for the Oracle Java Cloud Service instance. If not configured, this field does not appear.

Element	Description
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.
OCPUs	Number of OCPUs allocated for the Oracle Java Cloud Service instance.
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.
 (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 Accessing the Administrative Consoles Used by Oracle Java Cloud Service. • **Open Fusion Middleware Control Console**—Open Fusion Middleware Control to administer your application environment. See Accessing the Administrative Consoles Used by 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 Accessing the Administrative Consoles Used by Oracle Java Cloud Service and Configuring a Load Balancer for an Oracle Java Cloud 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 Viewing Service Metrics in Oracle Java Cloud Service. • **SSH Access**—Add public SSH keys to the VMs that make up this service instance. To manage the keys for all service instances see Managing SSH Access for an Oracle Java Cloud Service Instance. • **Delete**—Deletes the service instance. See Deleting an Oracle Java Cloud Service Instance.

Element	Description
Service create or delete history	Shows details about created or deleted service instances. • **Range**—Specifies a range of days for which you are interested in viewing created and failed 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.

Exploring 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

Use Oracle Java Cloud Service Instance Overview page to perform the tasks described in the following topics:

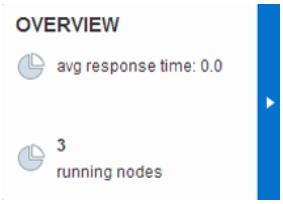
- Viewing Detailed Information About an Oracle Java Cloud Service Instance
- Viewing Service Metrics in Oracle Java Cloud Service. Note that you can click on the service instance name on the View Metrics screen to return to the Oracle Java Cloud Service Instance Overview page.
- Accessing the Administrative Consoles Used by Oracle Java Cloud Service
- Click the Administration tile to backup and restore, and patch an Oracle Java Cloud Service instance; and configure the load balancer. See:
  - Backing Up and Restoring an Oracle Java Cloud Service Instance
  - Patching an Oracle Java Cloud Service Instance
  - Configuring a Load Balancer for an Oracle Java Cloud Service Instance

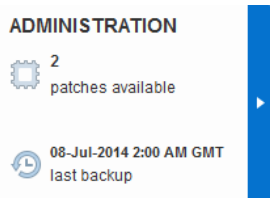
- When Oracle Coherence is enabled for a service instance, see also Scaling the Coherence Data Tier of an Oracle Java Cloud Service Instance for information about adding or removing a capacity unit to scale out or scale in storage-enabled cluster.
- Stop and start a service instance. See Stopping and Starting an Oracle Java Cloud Service Instance.

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	Follow this link to return to the Oracle Java Cloud Service Console.
 (adjacent to the service instance name)	**Menu** icon provides the following options: • **Open WebLogic Server Console**—Open the WebLogic 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 the load balancer has been configured for the service instance. See Configuring a Load Balancer for an Oracle Java Cloud Service Instance. • **Start**—Start the VMs for the Administration Server, Managed Servers, load balancer, and Managed Servers on the Coherence data tier (if provisioned). • **Stop**—Stop the VMs for the Administration Server, Managed Servers, load balancer, and Managed Servers on the Coherence data tier (if provisioned). • **Scale Out**—Scales the Managed Server node. • **Add Load Balancer**—Adds a load balancer to the service instance. If you did not specify a load balancer when you created the service instance, you can add it now. This can also be a second load balancer. • **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. • **SSH Access**—Add public SSH keys to the VMs that make up this service instance. See Managing SSH Access for an Oracle Java Cloud Service Instance. • **View Activity**— Opens the Service Activities page. For a selected service instance: – Stop Service – Start Service – Stop Server – Start Server – Restart Server – Scale Out – Scale In – Scale Up – Scale Down

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. **Note:** This menu is disabled when the service instance is stopped. When the service instance is stopped, you cannot restart the Administration Server VM or scale the Administration Server node.
 (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. **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: 12.2.1.2, 12.1.3.0, or 11.1.1.7.
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.
	Click the start/stop/scale/monitor icons to: • Start the VMs for the Administration Server, Managed Servers, load balancer, and Managed Servers on the Coherence data tier (if provisioned) • Stop the VMs for the Administration Server, Managed Servers, load balancer, and Managed Servers on the Coherence data tier (if provisioned) • Add a node • Display monitoring data: – 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.
	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).

Element	Description
	Click the Administration tile to backup and restore, and patch an Oracle Java Cloud Service instance; and configure the load balancer. See: • Backing Up and Restoring an Oracle Java Cloud Service Instance • Patching an Oracle Java Cloud Service Instance • Configuring a Load Balancer for an Oracle Java Cloud Service Instance The Administration tile displays the number of patches applied and the date of last backup.
Nodes	Number of nodes defined 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).
OCPUs	Total number of Oracle CPUs allocated for the Oracle Java Cloud Service instance.
Memory	Total amount of memory in GBs allocated for the Oracle Java Cloud Service instance.
Storage	Total amount of block storage in GBs allocated for the Oracle Java Cloud Service instance.
Public IPs	Total number of IP reservations allocated for the Oracle Java Cloud Service instance.
Heap Usage	Percent of available heap space that is being used by the Oracle Java Cloud Service instance. When Oracle Coherence is enabled for a service instance, the Managed Servers on the Coherence data tier are not included in the heap usage calculation.
Response Time	Displays the response time to requests, if a load balancer is present. If there is no activity, N/A is displayed.
Service Overview	Provides the following information: • Status—Status of the service instance. Valid values include: In Progress, Maintenance, Terminating, Stopped, and Failed. • Edition—WebLogic Server edition: Standard Edition, Enterprise Edition, Enterprise Edition with Coherence • Cloud Storage Container—The NFS backup destination you specified when you created the service instance. • Version—Version of Oracle WebLogic Server configured for the Oracle Java Cloud Service instance. Valid values include: 12.2.1.2, 12.1.3.0 or 11.1.1.7. Also displays whether domain partitions are enabled. • JDK—Java Development Kit version used by the WebLogic Servers
Resources	Displays the following information about the resources: • Administration Server Domain—Domain name of the resource • Public IP—Public IP address of the resource • Instance—Displays the servers running on the VM. • OCPUs—Number of OCPUs used in the resource • Memory—Amount of memory the resource uses • Storage—Amount of storage used by the resource

Element	Description
	Provides information about the Administration Server or Managed Server: • Administration Server Domain *domainName*—Tag that identifies the node as the Administration Server and lists the domain name. • Admin Server Up Since—Date and time the server started. Appears only after you click the Monitoring icon. • Admin Server 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. If you click the percentage number, the heap usage graph is displayed. • Managed Server—Name of the Managed Server. • Managed Server Up Since—Date and time the server started. Appears only after you click the Monitoring icon. • Managed Server 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. If you click the percentage number, the heap usage graph is displayed. • Public IP—Public IP address of the Administration Server.
	Identifies information about the load balancer on the cluster: • Version • Public IP address • Hostname • Content endpoint, specified as an IP address • Number of Oracle CPUs allocated (OCPU) • The amount of memory allocated to the node • The amount of storage available to the node
Host	Name of the Managed Server or load balancer.
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
Up Since	Date and time each node was last started. **Note**: This information display only after you click the Monitoring icon.
Heap Usage	For the Administration Server and each Managed Server, the percent of heap space used as compared to the total heap space in GBs available. **Note**: This information displays only after you click the Monitoring icon and click  Refresh.

Element	Description
 (available only when Oracle Coherence is enabled for the service instance)	Provides information about the Coherence data tier cluster and nodes. Click the icon to view details about the capacity unit configuration, data tier cluster, and host names for the data tier Managed Servers. • **Capacity Unit** – **Shape**—Compute shape for all virtual machines configured by a capacity unit. For example: `oc5`. – **Node Count**—Number of virtual machines configured by one capacity unit. – **Servers Per Node**—Number of Managed Servers configured per virtual machine by one capacity unit. – **Primary Capacity**—Primary cache storage in GBs allocated for Coherence by one capacity unit. • **Cluster** – **Name**—Data tier (storage-enabled) WebLogic Server cluster name. For example: `service1_DGCluster` – **Node Count**—Total number of virtual machines configured in the cluster. – **Primary Capacity**—Total primary cache storage allocated for Coherence based on the number of capacity units configured in the cluster. – **Capacity Units**—Number of capacity units configured in the cluster. • **Hosts**—Lists the host names for the Managed Servers configured in the cluster.
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: • **Cache Capacity**: Total amount of primary cache storage allocated for Coherence in the domain, based on the capacity unit configuration, and the general rule of splitting the Managed Server heap size into thirds: using 1/3rd for primary cache storage, 1/3rd for backup storage, and 1/3rd for scratch space. • **Capacity Unit**: Primary cache storage allocated for Coherence by one capacity unit. If using a default capacity unit, the name of the configuration is also displayed (for example, `SMALL`). If using your own custom configuration for a capacity unit, `ADVANCED` is displayed. • **Nodes**: Number of virtual machines configured in the Coherence data tier cluster. • **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. 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.

Element	Description
Associations	Information about the Oracle Database Cloud Service database deployment used by the Oracle Java Cloud Service instance. • Service Name: Name of the Database Cloud Service database deployments used by the Oracle Java Cloud Service instance. The name was specified during the process of creating the Oracle Java Cloud Service instance. • Connect Descriptor: Specifies the connection string for the on-premises database. This string was specified during the process of creating the Oracle Java Cloud Service instance. • Type: The only available value is Oracle Database Cloud Service. • Role: Depends on the database selections you made when you created the service instance: – Infrastructure Database—Stores data for the schema only – Application Database—Stores data for the application only – Infrastructure and Application Database—The database is used to store schema and application data.

3

Administering Oracle Java Cloud Service Software on Oracle Public Cloud Machine

From an Oracle Java Cloud Service instance, you can access the administration consoles and individual virtual machines through a secure shell to run Oracle WebLogic Server utilities, such as the WebLogic Scripting Tool (WLST).

Topics:

Accessing the Administrative Consoles

You can use various consoles to administer the software that an Oracle Java Cloud Service instance is running and to also administer related Oracle Cloud services.

Topics:

Accessing an Administration Console for Oracle Java Cloud Service Software

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 Console
- Fusion Middleware Console
- Load Balancer Console

> **Note:** 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. Navigate to the Services page of the Oracle Java Cloud Service Console.
2. Click  for the desired service instance and choose the command to open the console that you want to access:

To access this console	Click this shortcut
WebLogic Server Console	**Open WebLogic Server Console**
Fusion Middleware 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. To update the default SSL configuration of your service instance, see Configuring SSL for an Oracle Java Cloud Service Instance.

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 WebLogic username and password you provided when you created the service instance.

Accessing the Administration Console of a Related Oracle Cloud Service

You can access the administration consoles for related Oracle Cloud Services, such as Oracle Database Cloud Service, from the Oracle Java Cloud Service console.

1. Navigate to the Oracle Java Cloud Service Console.
2. From the  menu at the top left of page, choose the service you want to access.

   For example, choose **Oracle Database Cloud Service**.

   The console for that service is displayed.

Alternatively you can use the  menu next to the text *Oracle Java Cloud Service*.

Accessing a VM Through a Secure Shell (SSH)

You can access the services and resources that an Oracle Java Cloud Service instance's VM provides by logging into the VM 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.

Note: Only the `opc` user can remotely connect to your VMs. You can not use SSH to connect to a VM as the `oracle` user. After successfully connecting to a VM, tasks such as starting and stopping the server and accessing the administrative logs should only be performed by the `oracle` user.

Topics:

Understanding SSH Keys

In order to access an Oracle Java Cloud Service virtual machine (VM) with a secure shell (SSH) client, you must create a public/private key pair and configure the service instance with the public key.

When you create an Oracle Java Cloud Service instance, you are prompted to supply the public key. To connect to a VM in an Oracle Java Cloud Service instance, you supply the paired private key when logging in to the machine using an SSH client.

You can provide an existing public key that you previously created with an external tool, or Oracle Java Cloud Service can create a new key pair for you. You may also use the same SSH public/private key pair that you used for creating an Oracle Database Cloud Service database deployment.

For more information, see:

- Configuring Size and Shape
- Generating a Secure Shell (SSH) Public/Private Key Pair
- The Creating SSH Keys for Use with Oracle Cloud Services tutorial
- Managing SSH Access for an Oracle Java Cloud Service Instance

Generating a Secure Shell (SSH) Public/Private Key Pair

Several tools exist to generate SSH public/private key pairs. The following sections show how to generate an SSH key pair on UNIX, UNIX-like and Windows platforms.

Generating an SSH Key Pair on UNIX and UNIX-Like Platforms Using the ssh-keygen Utility

UNIX and UNIX-like platforms (including Solaris and Linux) include the ssh-keygen utility to generate SSH key pairs.

To generate an SSH key pair on UNIX and UNIX-like platforms using the ssh-keygen utility:

1. Navigate to your home directory:

   ```
   $ cd $HOME
   ```

2. Run the ssh-keygen utility, providing as ***filename*** your choice of file name for the private key:

   ```
   $ ssh-keygen -b 2048 -t rsa -f filename
   ```

   The ssh-keygen utility prompts you for a passphrase for the private key.

3. Enter a passphrase for the private key, or press Enter to create a private key without a passphrase:

   ```
   Enter passphrase (empty for no passphrase): passphrase
   ```

   Note:

   While a passphrase is not required, you should specify 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.

   The ssh-keygen utility prompts you to enter the passphrase again.

4. Enter the passphrase again, or press Enter again to continue creating a private key without a passphrase:

   ```
   Enter the same passphrase again: passphrase
   ```

5. The ssh-keygen utility displays a message indicating that the private key has been saved as *filename* and the public key has been saved as *filename*.pub. It also displays information about the key fingerprint and randomart image.

Generating an SSH Key Pair on Windows Using the PuTTYgen Program

The PuTTYgen program is part of PuTTY, an open source networking client for the Windows platform.

To generate an SSH key pair on Windows using the PuTTYgen program:

1. Download and install PuTTY or PuTTYgen.

   To download PuTTY or PuTTYgen, go to http://www.putty.org/ and click the **You can download PuTTY here** link.

2. Run the PuTTYgen program.

   The PuTTY Key Generator window is displayed.

3. Set the **Type of key to generate** option to **SSH-2 RSA**.
4. In the **Number of bits in a generated key** box, enter **2048**.
5. Click Generate to generate a public/private key pair.

   As the key is being generated, move the mouse around the blank area as directed.
6. (Optional) Enter a passphrase for the private key in the **Key passphrase** box and reenter it in the **Confirm passphrase** box.

   Note:

   While a passphrase is not required, you should specify 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** to save the private key to a file. To adhere to file-naming conventions, you should give the private key file an extension of `.ppk` (PuTTY private key).

   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 as your SSH client. It cannot be used with other SSH client tools. Refer to the PuTTY documentation to convert a private key in this format to a different format.
8. Select all of the characters in the **Public key for pasting into OpenSSH authorized_keys file** box.

   Make sure you select all the characters, not just the ones you can see in the narrow window. If a scroll bar is next to the characters, you aren't seeing all the characters.
9. Right click somewhere in the selected text and select **Copy** from the menu.
10. Open a text editor and paste the characters, just as you copied them. Start at the first character in the text editor, and do not insert any line breaks.
11. Save the text file in the same folder where you saved the private key, using the `.pub` extension to indicate that the file contains a public key.
12. If you or others are going to use an SSH client that requires the OpenSSH format for private keys (such as the `ssh` utility on Linux), export the private key:
    a. On the **Conversions** menu, choose **Export OpenSSH key** .
    b. Save the private key in OpenSSH format in the same folder where you saved the private key in `.ppk` format, using an extension such as `.openssh` to indicate the file's content.

Connecting to an Administration Server or Load Balancer VM

You can access an Administration Server or a Load Balancer VM through a secure shell (SSH) utility.

To access a VM through SSH:

1. Navigate to the Services page of the Oracle Java Cloud Service console.
2. Click the service instance associated with the VM you want to access.

   The Oracle Java Cloud Service Instance page is displayed with the Overview tile in focus, displaying detailed information about the service instance.
3. From the list of virtual machines, note the **Public IP** address of the Administration Server or the Load Balancer, depending on which VM you want to access.

   This address will be specified in the typical octet format (111.111.111.111).

   Note: The console displays public IP addresses only for the Administration Server and the Load Balancer VMs, not for the managed server VMs. For more information, see Connecting to a Managed Server VM.
4. On UNIX and UNIX-like platforms, use the standard OpenSSH command (`ssh`) to connect to the VM as the `opc` user.

   Provide the following:

   - The path to the private key corresponding to the public key used at the time of provisioning.
   - The VM's public IP address.

   in this format:

   ```
   ssh -i path_to_private_key opc@VM_IP_address
   ```

   For example:

   ```
   ssh -i /home/myuser/id_rsa opc@111.111.111.111
   ```
5. On Windows, you can use PuTTY, an open source networking client for the Windows platform, to connect to the VM as the `opc` user.

   To download PuTTY, go to http://www.putty.org/ and click the **You can download PuTTY here** link.

   a. Launch PuTTY.

      The PuTTY Configuration window is displayed, showing the Session panel.

   b. In the **Host Name (or IP address)** field, enter the public IP address of the VM.

   c. In the Category tree, expand **Connection** if necessary and then click **Data**.

   d. In the **Auto-login username** field, enter `opc`.

   e. Confirm that the **When username is not specified** option is set to **Prompt**.

   f. In the Category tree, expand **Connection > SSH**, and then click **Auth**.

   g. Under **Private key file for authentication**, click **Browse**.

   h. Navigate to and select your private key file. Then click **Open**.

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 you have to use a key saved in a different format, see the PuTTY documentation.

i. Click **Open** to open the connection to the VM.

6. If the private key was defined with a passphrase, enter this value when prompted.

When the VM command line appears, you can use any resource accessible from the VM. For example, you can run the WebLogic Scripting Tool on the Administration Server VM.

Connecting to a Managed Server VM

You can access a Managed Server VM through a secure shell (SSH) utility by using the Administration Server VM as a proxy.

Alternatively, you can connect to the Administration Server VM with SSH, and from within this SSH session start another SSH connection to the Managed Server VM.

To connect to a Managed Server VM by using the proxy method:

1. Navigate to the Oracle Java Cloud Service Console.
2. Click the service instance associated with the VM you want to access.

   The Oracle Java Cloud Service Instance page is displayed with the Overview tile in focus, displaying detailed information about the service instance.
3. From the list of virtual machines, identify the following information:
   - The **Public IP** address of the Administration Server VM (used as the proxy).
   - The **Host** name of the Managed Server VM to which you want to connect.
4. On UNIX and UNIX-like platforms, use the standard OpenSSH command (`ssh`) to connect to the VM as the `opc` user.

   Provide the following:
   - The path to the private key corresponding to the public key used at the time of provisioning.
   - The Administration Server VM's public IP address.
   - The Managed Server VM's host name.

   in this format:

   ```
   ssh -i path_to_private_key -o ProxyCommand="ssh -W %h:%p -i path_to_private_key
   opc@admin_server_VM_IP_address" opc@managed_server_host_name
   ```

   For example:

   ```
   ssh -i /home/myuser/id_rsa -o ProxyCommand="ssh -W %h:%p -i /home/myuser/id_rsa
   opc@111.111.111.111" opc@myjcs-wls-2
   ```
5. On Windows, you can use PuTTY, an open source networking client for the Windows platform, to connect to the VM as the `opc` user.

To download PuTTY, go to http://www.putty.org/ and click the **You can download PuTTY here** link.

a. Launch PuTTY. If your private key was defined with a passphrase, then you must use the `pageant` utility to launch PuTTY:

```
pageant "path to private key" -c "path to putty"
```

For example:

```
c:\PuTTY\pageant "c:\oracle\rsa.ppk" -c "c:\PuTTY\putty"
```

b. If you used `pageant` to start PuTTY, enter the passphrase for the private key.

The PuTTY Configuration window is displayed, showing the Session panel.

c. In the **Host Name (or IP address)** field, enter the host name of the Managed Server VM.

d. In the Category tree, expand **Connection** if necessary and then click **Data**.

e. In the **Auto-login username** field, enter `opc`.

f. Confirm that the **When username is not specified** option is set to **Prompt**.

g. In the Category tree, click **Connection > Proxy**.

h. Set **Proxy type** to **Local**.

i. In the **Proxy hostname** field, enter the IP address of the Administration Server VM.

j. Set the **Port** to 22.

k. In the **Telnet command or local proxy command** field, enter the following value:

```
plink -i "path to private key" opc@%proxyhost -nc %host:%port
```

For example:

```
plink -i "c:\\oracle\\rsa.ppk" opc@%proxyhost -nc %host:%port
```

l. In the Category tree, expand **Connection > SSH**, and then click **Auth**.

m. Under **Private key file for authentication**, click **Browse**.

n. Navigate to and select your private key file. Then click **Open**.

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 you have to use a key saved in a different format, see the PuTTY documentation.

---

o. Click **Open** to open the connection to the VM.

Note: You can optionally save this session configuration by navigating to the Session panel and clicking **Save**. When you open PuTTY the next time, you can load this configuration by selecting it and clicking **Load**.

When the VM command line appears, you can use any resource accessible from the VM.

Creating an SSH Tunnel

An SSH tunnel to an Oracle Java Cloud Service VM enables you to connect to other non-public ports on the VM though a port your local machine.

If a resource provided by a VM uses a port that is not directly accessible through the Internet, you can access that resource by creating an SSH tunnel to the port. For example, you can use an SSH tunnel to connect a local Integrated Development Environment (IDE) such as Eclipse to the dedicated deployment port (9001) of the Administration Server.

In general an SSH tunnel may map a remote port to any available port number on your local machine. However, port 9001 on the Administration Server uses JMX/RMI for communication, which requires that the remote and local port numbers be the same value. Therefore, the following instructions configure the tunnel's local port number to the same value as the VM's port number.

To set up an SSH tunnel to a VM:

1. Navigate to the Services page of the Oracle Java Cloud Service console.
2. Click the service instance associated with the VM you want to access.

   The Oracle Java Cloud Service Instance page is displayed with the Overview tile in focus, displaying detailed information about the service instance.
3. From the list of virtual machines, note the **Public IP** address of the Administration Server or the Load Balancer, depending on which VM you want to access.

   This address will be specified in the typical octet format (111.111.111.111).

   Note: The console displays public IP addresses only for the Administration Server and the Load Balancer VMs, not for the managed server VMs. For more information, see Connecting to a Managed Server VM.
4. On UNIX and UNIX-like platforms, use the standard OpenSSH command (`ssh`) to create an SSH tunnel to the VM.

   Provide the following:

   - The path to the private key corresponding to the public key used at the time of provisioning.
   - The VM's public IP address.
   - The port number on the VM to which you want to connect. The SSH tunnel will enable connectivity to this remote port though the same port number on your local machine.

   in this format:

   ```
   ssh -i path_to_private_key -L port:VM_IP_address:port opc@VM_IP_address -N
   ```

   For example, to create an SSH tunnel to port 9001 on the Administration Server VM:

   ```
   ssh -i /home/myuser/id_rsa -L 9001:111.111.111.111:9001 opc@111.111.111.111 -N
   ```

5. On Windows, you can use PuTTY, an open source networking client for the Windows platform, to create an SSH tunnel to the VM.

   To download PuTTY, go to http://www.putty.org/ and click the **You can download PuTTY here** link.

   a. Launch PuTTY.

      The PuTTY Configuration window is displayed, showing the Session panel.

   b. In the **Host Name (or IP address)** field, enter the public IP address of the VM.

   c. In the Category tree, expand **Connection** if necessary and then click **Data**.

   d. In the **Auto-login username** field, enter `opc`.

   e. Confirm that the **When username is not specified** option is set to **Prompt**.

   f. In the Category tree, click **Connection > SSH**.

   g. Under **Protocol options**, select the checkbox **Don't start a shell command at all**.

   h. In the Category tree, expand **Connection > SSH**, and then click **Auth**.

   i. Under **Private key file for authentication**, click **Browse**.

   j. Navigate to and select your private key file. Then click **Open**.

      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 you have to use a key saved in a different format, see the PuTTY documentation.

   k. In the Category tree, click **Connection > SSH > Tunnels**.

   l. In the **Destination** field, enter *`IP:port`*,

      where *IP* is the IP address of the VM and *port* is the port number on the VM to which you want to connect.

   m. In the **Source Port** field, enter the same port number.

   n. Click the **Add** button.

   o. Click **Open** to create the SSH tunnel to the VM.

      Note: You can optionally save this session configuration by navigating to the Session panel and clicking **Save**. When you open PuTTY the next time, you can load this configuration by selecting it and clicking **Load**.

6. If the private key was defined with a passphrase, enter this value when prompted.

Applications running on your local machine can now communicate with the VM by using `localhost:`*`port`*, where *port* is the local port number.

For example, after creating an SSH tunnel to port 9001 on the Administration Server VM, launch a web browser and connect to `http://localhost:9001/console`.

Note: After your work with the SSH tunnel is complete, perform a `<ctrl> C` to shut down the SSH tunnel.

Switching VM Users

You can change users on an Oracle Java Cloud Service VM in order to perform specific administration tasks.

You must SSH to a VM only as the `opc` user. This user has root privileges on the OS running in the VM. For example, `opc` can be used to create other OS users on a VM. Simply prefix root operations with the `sudo` command. For example:

```
sudo useradd myuser
```

Note: There is no default password for the `opc` user.

Switching to Oracle

The `oracle` VM user has regular OS user permissions. It is intended to be used to start and stop Oracle products that have been installed on the VM, or to run other Oracle applications and utilities on the VM.

Type the following to become the `oracle` user:

```
sudo su - oracle
```

Note: There is no default password for the `oracle` user.

Switching to Root

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
```

Note: Avoid using the `root` user except to perform privileged OS administration tasks.

Accessing a VM Through Virtual Network Computing (VNC)

You can access the services and resources that a Oracle Java Cloud Service VM provides by logging into the VM through VNC.

You can use any VNC client utility to access a VM. For example, if you are using Windows, you might use RealVNC or TightVNC; if you are using Linux, you might use the `vncviewer` utility included with your Linux distribution.

By default, the port used by the VNC server on a Oracle Java Cloud Service VM is not directly accessible through the Internet. An SSH tunnel enables access to the VNC server port on your local machine. An SSH tunnel also ensures that VNC communication is using a secure channel.

In order create a VNC session on a VM, you must first identify the public IP address of the VM and connect to it with SSH, as described in Accessing a VM Through a Secure Shell (SSH).

1. SSH to the VM and switch to the `oracle` user:

   ```
   sudo su - oracle
   ```

   Note: The `oracle` VM user has regular OS user permissions. It is intended to be used to start and stop Oracle products that have been installed on the VM, or to run other Oracle applications and utilities on the VM.

2. 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.

3. Start the VNC server on the VM:

   ```
   vncserver :1 -nolisten tcp -localhost -geometry 1680x1050
   ```

   Note:

   The VNC server is not directly accessible from clients outside of this VM. An SSH tunnel will be used to enable external and secure access to the VNC server.

   By default, the listen port for VNC session `:1` is 5901, session `:2` is 5902, and so on.

   If your local machine has a smaller display resolution, use a different geometry setting such as `1024x768`.

4. When prompted, enter a password for this VNC session.

5. Disconnect from the VM.

6. Create an SSH tunnel to `localhost:5901` on the VM.

   ```
   ssh -i path_to_private_key -L 5901:localhost:5901 opc@VM_IP_address -N
   ```

   For example:

   ```
   ssh -i /home/myuser/id_rsa -L 5901:localhost:5901 opc@111.111.111.111 -N
   ```

   For more information, see Creating an SSH Tunnel.

7. Launch your VNC client application and connect to `localhost:5901`.

8. When prompted, enter the password that you previously configured for this VNC session.

You can use VNC to work with any resource accessible from the VM, including graphical applications. For example, you can launch the Fusion Middleware Configuration Wizard application on the Administration Server VM.

Note: After your VNC work is complete, you can perform a `<ctrl> C` to shut down the SSH tunnel.

Note: To terminate the VNC server on the VM, run `vncserver –kill :1`.

Using WLST to Administer an Oracle Java Cloud 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.

To administer a service instance through WLST, use one of these methods:

- You can use a secure shell (SSH) to connect to the virtual machine (VM) that hosts the Administration Server and run WLST commands local to the VM. When running WLST commands local to the VM, you can use both WLST online and offline.

  For more information, see Running WLST Commands on a VM

- You can connect to the Administration Server by using WLST online and run WLST commands remotely, for example, from a command shell in your local environment. When running WLST commands remotely, you can use WLST online only.

  For more information, see Running WLST Commands from a Different Host

Using 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 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 domain or modify security data. You cannot run offline commands from a remotely-attached Oracle Java Cloud Service because the domain configuration files are not local to your system, so you cannot manipulate them.

Running WLST Commands on a VM

You can run WLST commands from within a VM that includes an Oracle WebLogic Server installation.

Prior to running WLST, identify the public IP address of the VM and connect to it with SSH, as described in Accessing a VM Through a Secure Shell (SSH).

1. After connecting to the VM, switch to the `oracle` user:

```
$> sudo su - oracle
```

The `oracle` user is used to run Oracle applications and utilities on VMs, including WLST.

2. At the VM prompt, set the domain environment by using the `setDomainEnv.sh` command, which resides in `/u01/data/domains/`*`domain-name`*`/bin`:

```
$> . ./setDomainEnv.sh
```

You must use the `.` to ensure that the environment variables are set in the current shell.

3. Launch WLST:

```
$>java weblogic.WLST
```

4. Connect to the Administration Server:

```
> connect('loginID', 'password', 'admin-server-host:admin-server-port')
```

For example:

```
> connect('weblogic', 'welcome1', '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)
- WLST Command Reference for WebLogic Server (12.1.3)
- WebLogic Scripting Tool Command Reference (10.3.6)

Running WLST Commands from a Different Host

You can run WLST commands from a host that is not the VM running Oracle WebLogic Server in Oracle Java Cloud Service (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, remote access to the administration console and WLST over port `7002` is disabled for security purposes.

Alternatively, you can create an SSH tunnel to port `9001` on the Administration Server VM, as described in Creating an SSH Tunnel.

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', 'welcome1', '10.11.12.13:7002')
   ```

   If you are accessing the Administration Server VM 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)
- WLST Command Reference for WebLogic Server (12.1.3)
- WebLogic Scripting Tool Command Reference (10.3.6)

Shutting Down and Starting Server Processes

You can shut down and start WebLogic Servers (including Administration Servers) employed in your Oracle Java Cloud Service instance.

The topics in this section describe how to shut down and start server *processes* only. The server VMs remain running. For information about how to shut down Managed Server and Administration Server VMs, see Stopping and Starting an Oracle Java Cloud Service Instance and Individual Nodes.

Topics

- Using the WebLogic Server Administration Console to Shut Down Servers Without Stopping VMs
- Using WLST Commands to Start the Administration Server
- Using the WebLogic Server Administration Console to Start Managed Servers

Using the WebLogic Server Administration Console to Shut Down Servers Without Stopping VMs

Use the WebLogic Server Administration Console to shut down servers employed by your Oracle Java Cloud Service instance: the Administration server and one or more managed servers. Do not attempt to shut down servers through the Oracle Traffic Director.

This topic only presents how to shut down servers without stopping the VMs on which the servers are running. This procedure enables you to shut down servers for any reason, such as periodic shutdown, or to address CPU and memory resource contention. This approach shuts down the server processes only, leaving other processes you might have running on the nodes to continue to run. For more

information about stopping server nodes, see Stopping and Starting an Oracle Java Cloud Service Instance and Individual VMs.

To use the WebLogic Server Console to shut down servers:

1. Navigate to the Oracle Java Cloud Service Console.
2. Click  for the desired service instance and select **Open WebLogic Server Console**.
3. When the console login page appears, enter the WebLogic username and password you provided when you created the service instance.
4. On the Administration Console, under Domain Structure, expand **Environment**.
5. Select **Servers**.
6. On the Configuration tab of the Summary of Servers page, notice that the state of the Administration server and managed servers is RUNNING.
7. Click the Control tab.
8. Click the check box to the left of each server.
9. Select **Shutdown**, then **Force Shutdown Now** or **When Work Completes**.

   The servers are shut down. A **Server Shutdown** message appears. The message you receive will be different based on whether you are shutting down the administration server or the managed servers. When you shut down the administration server, the message warns you that the browser session will end.

You have now shut down the servers.

Using WLST Commands to Start the Administration Server

You start the Administration Server through the Node Manager by using WLST commands.

Prior to running WLST, identify the public IP address of the Administration Server VM and connect to it with SSH, as described in Accessing a VM Through a Secure Shell (SSH).

1. After connecting to the VM, switch to the `oracle` user:

   ```
   sudo su - oracle
   ```

2. Check to see 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 where environment setup is located:

   ```
   cd /u01/data/domains/domain_name/bin
   ```

   For example:

   ```
   cd /u01/data/domains/OurServi_domain/bin
   ```

4. Set up the environment.

```
. ./setDomainEnv.sh
```

5. Start WLST:

```
java weblogic.WLST
```

6. To connect to the Node Manager, use the WLST `nmConnect` command:

```
nmConnect (username,password,host,nmPort,domainName,domainDir,nmType)
```

Parameter	Description	Example
`username`	Username you specified when you created the service instance.	`weblogic`
`password`	Password you specified when you created the service instance.	`welcome1`
`host`	The host name of the Node Manager. Instance name in all lower case, followed by a dash, followed by `wls`, followed by a dash, followed by `1`.	`ourserviceinstance-wls-1`
`nmPort`	Port number of the node manager. .	`5556`
`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','welcome1','ourserviceinstance-
wls-1','5556','OurServi_domain','/u01/data/domains/OurServi_domain','SSL')
```

For more information about `nmConnect` parameters, see:

- For Oracle Fusion Middleware 12.2.1: nmConnect in *WLST Command Reference for WebLogic Server*
- For Oracle Fusion Middleware 12.1.3: nmConnect in *WLST Command Reference for WebLogic Server*
- For Oracle Fusion Middleware 11.1.1.7: nmConnect in *WLST Scripting Tool Command Reference*

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. Open a browser window.

12. Navigate to the Oracle Java Cloud Service Console.

13. Click  for the desired service instance and select **Open WebLogic Server Console**.

14. When the console login page appears, enter the WebLogic username and password you provided when you created the service instance.

    If the Administration Console opens, you know that the Administration Server is running. We will also verify that it is running by using the Administration Console.

15. On the Administration Console, under **Domain Structure**, expand **Environment**.

16. Select **Servers**.

17. On the **Configuration** page, check that the Administration Server state is RUNNING.

You have now started the Administration Server through the Node Manager by using WLST commands.

Using the WebLogic Server Administration Console to Start Managed Servers

After you have started the Administration Server through the Node Manager by using WLST commands, you can start the Managed Servers through theWebLogic Server Administration Console.

To start Managed Servers:

1. Navigate to the Oracle Java Cloud Service Console.

2. Click  for the desired service instance and select **Open WebLogic Server Console**.

3. When the console login page appears, enter the WebLogic username and password you provided when you created the service instance.

4. Under **Domain Structure**, expand **Environment**.

5. Select **Servers**.

6. On the Configuration page, notice that the Administration Server state is RUNNING, and the Managed Server state is SHUTDOWN.

7. Select the **Control** tab.
8. Click the check box to the left of each Managed Server name.
9. Click **Start**.
10. On the Server Life Cycle Assistant, click **Yes**.

    The server state changes to STARTING.
11. Click the **Refresh** icon.

    The server state changes to RUNNING.

You have now started the Managed Servers.

Understanding JVM Heap Size Settings

When you provision an Oracle Java Cloud Service instance and specify a compute shape, the JVM heap size for WebLogic Server and Load Balancer processes is determined automatically.

Default Heap Sizes

The compute shape you select for a WebLogic Server cluster determines the availability of RAM on VMs in this cluster, and the amount of available RAM is used to determine the preset heap size for the JVM processes running on the VMs.

For the RAM and heap settings used in Oracle Java Cloud Service—Coherence instances, see About Default Capacity Units for Oracle Java Cloud Service—Coherence Instances.

The following table shows the Oracle Java Cloud Service JVM heap size settings for each compute shape.

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)
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)

Custom Heap Sizes

If you create an Oracle Java Cloud Service instance by using the REST API, you can specify a custom heap size for the JVMs in the service instance. See Create a Service Instance in *REST API for Oracle Java Cloud Service*. You cannot specify a custom heap size when creating a service instance with the web console.

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)*
- Increasing the heap size for a managed server in *Administration Console Online Help (12.1.3)*
- 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.

Understanding 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

When you create an Oracle Java Cloud Service instance you provide the locations of one or two Oracle Database Cloud Service database deployments. These database deployments are used for different purposes:

- Oracle Required Schema — Oracle Java Cloud Service provisions this Oracle Database or Pluggable Database (PDB) with the required Oracle Fusion Middleware schema and data. By default, this data is automatically deleted when you delete the service instance.
- Application Schema (optional) — You can provision this Oracle Database or PDB with any business data that your applications require, and using any standard Oracle Database tools.

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 database deployments.

Data Source Types

WebLogic Server provides several types of data sources, including:

- *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 Oracle Database Cloud Service database deployment 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
Enterprise	No	Generic
Enterprise	Yes	Multi
Enterprise with Coherence (Suite)	No	Generic
Enterprise with Coherence (Suite)	Yes	GridLink

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)
- Administering JDBC Data Sources for Oracle WebLogic Server (12.1.3)
- Configuring and Managing JDBC Data Sources for Oracle WebLogic Server (10.3.6)

Monitoring Applications with Java Flight Recorder and Java Mission Control

You can use Oracle's commercial profiling tools, Java Flight Recorder and Java Mission Control, to analyze the performance of applications deployed on Oracle Java Cloud Service.

Java Flight Recorder (JFR) and Java Mission Control (JMC) collect detailed runtime information so that you can analyze incidents after they occur. JFR, available in Oracle HotSpot 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. JMC enables you to monitor and manage Java applications without introducing the performance overhead normally associated with these types of tools. It includes the JFR user interface, which allows users who are running a Java Flight Recorder-compliant version of Oracle HotSpot to view JVM recordings, current recording settings, and runtime parameters. The JFR interface includes the Events Type View, which gives you direct access to event information that has been 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 JFR and JMC

Monitoring applications with JFR and JMC comprises these steps:

1. Enable JFR in your WebLogic Server instance.
2. Obtain the flight recording by generating a diagnostic image capture.
3. Analyze the recording witht he Flight Recorder UI.

Enable JFR in Your WebLogic Server Instance

Because it is a commercial feature, if WebLogic Server is configured with Oracle HotSpot, Java Flight Recorder is, by default, disabled. To enable it, use the following JVM commands in the startup script for the WebLogic Server instance in which the JVM runs:

```
-XX:+UnlockCommercialFeatures -XX:+FlightRecorder
```

Note:

The sequence of these commands is critical: `+UnlockCommercialFeatures` command advises the JVM to recognize the command `+FlightRecorder`. The commands must be entered in that order or the JVM will not start.

For example:

```
java -XX:+UnlockCommercialFeatures -XX:+FlightRecorder -
XX:FlightRecorderOptions=defaultrecording=true,maxage=20m MyApp
```

You can also enter the `+UnlockCommercialFeatures` and `+FlightRecorder` commands in the WebLogic Server configuration file's `JAVA_OPTIONS` (or equivalent) variable.

For more detailed information on enabling JFR, see Running Java Flight Recorder in the *Java Platform, Standard Edition Java Flight Recorder Runtime Guide*.

Obtain the Flight Recording by Generating a Diagnostic Image Capture

The diagnostic image capture itself is a single JFR file that contains individual images produced by the different server subsystems. If the JFR file is available, it is included in the diagnostic image as the file `FlightRecording.jfr`.

You can generate a diagnostic image capture on-demand — for example, from the WebLogic Server Administration Console, Fusion Middleware Control, WLST, or a JMX application — or as the result of an image action. To generate a diagnostic image captures and configure the location in which they are created, see Configuring and Capturing Diagnostic Images in *Oracle® Fusion Middleware Configuring and Using the Diagnostics Framework for Oracle WebLogic Server*.

Analyze the Recording with the Flight Recorder UI

Once you've obtained the recording, you can then view and analyze it by using the Flight Recorder user interface, a JMC component. Assuming you are running a Java Flight Recorder-compliant version of Oracle HotSpot, the JFRUI allows you to view JVM recordings, current recording settings, and runtime parameters. The JFR interface includes the Events Type View, which gives you direct access to event information that has been recorded in the JFR file, such as event producers and types, event logging and graphing, event by thread, event stack traces, and event histograms. Some of the activity you can monitor on the JFR UI includes:

- Display Event Data for a Product Subcomponent
- View the Event Log to Display Details
- Track Execution Flow by Analyzing an Operative Set
- Expand the Operative Set and View Correlated Diagnostic Data

Use of the JFR UI for these tasks and more is described in greater detail in Analyzing Java Flight Recorder Data in *Oracle® Fusion Middleware Configuring and Using the Diagnostics Framework for Oracle WebLogic Server*.

Defining a Custom Domain Name for an Oracle Java Cloud Service Instance

By using the load balancer as the front-end, you can quickly and easily associate a custom "vanity" domain name to your application environment. For example, rather than accessing your applications using a public IP address, you can define a custom URL, such as `example.com`.

To identify the public IP address for the load balancer see Viewing Detailed Information About an Oracle Java Cloud Service Instance.

Tasks:

- Registering a Custom Domain Name with a Third-Party Registration Vendor
- Updating the Load Balancer to Use the Custom Domain Name

Registering a Custom Domain Name with a Third-Party Registration Vendor

Third-party vendors enable you to register custom domain names.

To register your custom domain and resolve it to the Oracle Java Cloud Service load balancer:

1. Register your domain name through a third-party domain registration vendor, such as `Register.com`, `Namecheap`, and so on.

   For example, `example.com`.

2. Resolve your domain name to the IP address of the Oracle Java Cloud Service load balancer, using the third-party domain registration vendor console.

   For more information, refer to the third-party domain registration documentation.

Updating the Load Balancer to Use the Custom Domain Name

You must update the load balancer configuration to use the custom domain name.

1. Navigate to the Services page of 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**.

For more information about configuring the load balancer, see Configuring a Load Balancer for an Oracle Java Cloud Service Instance on OPCM.

Defining a Custom URL for an Application Deployed to an Oracle Java Cloud Service Instance

You can 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.

1. Navigate to the Services page of 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. Go to **Routes** and click **default-route**.
7. Edit these fields in the **URI Mapping** section:

   a. Select the **Enabled** checkbox.

   b. For **From URI**, enter the context root for the application. For example, enter `/shopping-cart/widgets`.

   c. For **Target URI**, enter the URI to which the **From URI** should be redirected. For example, enter `/`.
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**.

For more information about configuring the load balancer, see Configuring a Load Balancer for an Oracle Java Cloud Service Instance on OPCM.

Keeping Your Service Instances Manageable

Following best practices will ensure that your Oracle Java Cloud Service instances stay manageable.

To keep your service instances manageable by Oracle Java Cloud Service, follow these guidelines:

- To ensure that you can restore the database for an Oracle Java Cloud Service instance without risking data loss for other service instances, do **not** use the same Oracle Database Cloud Service database deployment with multiple Oracle Java Cloud Service instances.

  Backups of an Oracle Database Cloud Service database deployment that is used with multiple Oracle Java Cloud Service instances contain data for all the Oracle Java Cloud Service instances. If you restore the database while restoring an Oracle Java Cloud Service instance, data for all the Oracle Java Cloud Service instances is restored.

- Apply **only** patches that are available through Oracle Java Cloud Service. Do **not** apply patches from any other source.

  You can get the list of available patches for a service instance from the service instance's Patching page or through the Oracle Java Cloud Service REST API.

  For more information, see:

  – Viewing Details of Approved Patches

  – View Available Patches in *REST API for Oracle Java Cloud Service*

- Use only the default domain that was provisioned when a service instance was created. Do not add any Oracle WebLogic Server domains to the service instance.

- If you plan to integrate multi-domain environments, ensure that the first eight characters of your Oracle Java Cloud Service instance name are unique so that all domains and associated resources have unique names.

  By default, the names of the domain and cluster in the Oracle Java Cloud Service instance will be generated from the first eight characters of the Oracle Java Cloud Service instance name, and will use the following formats, respectively:

  – *`first8charsOfServiceInstanceName`*`_domain`

  – *`first8charsOfServiceInstanceName`*`_cluster`

  – *`first8charsOfServiceInstanceName`*`_DGCluster` (when Oracle Coherence is enabled for a service instance)

  For more information, see one of the following topics:

  - Integration and Multi-Domain Best Practices in *Administering JMS Resources for Oracle WebLogic Server (12.2.1.0)*.
  - Integration and Multi-Domain Best Practices in *Administering JMS Resources for Oracle WebLogic Server (12.1.3.0)*.
  - Integration and Multi-Domain Best Practices in *Configuring and Managing JMS for Oracle WebLogic Server (11.1.1.7)*.

- Add managed servers to a service instance only by scaling out the Oracle WebLogic Server cluster in the service instance. Do not use Oracle WebLogic Server administrative interfaces for this purpose.

  For information about how to scale out the cluster in a service instance, see Scaling Out an Oracle Java Cloud Service Cluster on OPCM.

  When Oracle Coherence is enabled for a service instance: The Coherence data tier is scaled independently of the application tier. See Scaling the Coherence Data Tier of an Oracle Java Cloud Service Instance for information about scaling out the Coherence data tier by adding a capacity unit.

- Add Oracle WebLogic Server clusters to a service instance only by using the Oracle Java Cloud Service REST API for scaling out a service instance. Do not use Oracle WebLogic Server administrative interfaces for this purpose.

  For information about the REST API for scaling out a service instance, see Scale Out in *REST API for Oracle Java Cloud Service*.

  When Oracle Coherence is enabled for a service instance: Note that you can scale out the Coherence data tier but you cannot add a second data tier WebLogic Server (storage-enabled) cluster in the domain. See Scaling the Coherence Data Tier of an Oracle Java Cloud Service Instance for information about scaling out the Coherence data tier.

- When Oracle Coherence is enabled for a service instance: Use only the `DataGridConfig` Coherence cluster that was created when the service instance was initially provisioned. Do not use `defaultCoherenceCluster`.

- Do not attach custom storage volumes to a service instance's VMs.

  Any custom storage volumes that you attach 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 Scaling an Oracle Java Cloud Service Node on OPCM.

- Do not detach, change file access permissions for, or change the mount point of any disk volume that Oracle Java Cloud Service attaches to a service instance's VMs during creation of the service instance.

  For details of these volumes, see About the Disk Volumes.

- Except for the `DOMAIN_HOME` volume, do not change the content of any disk volume that Oracle Java Cloud Service attaches to a service instance's VMs during creation of the service instance.

  For details of these volumes, see About the Disk Volumes.

- Do **not** change the content of the Boot/OS volume of any service instance provisioned before the mid-August 2015 update to Oracle Java Cloud Service.

  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.

  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 Disk Volumes.

- Do not change the egress and ingress network and security settings of any infrastructure resources that the service instance uses.

- Do not close any ports that Oracle Java Cloud Service opened during creation of a service instance.

  You can open new ports, but closing existing ports may impair the functioning of a service instance.

  For more information about the ports that Oracle Java Cloud Service opens during creation of a service instance, see Understanding the Default Access Ports.

- Do not detach NAT IP addresses from any of a service instance's VMs.

- Do not change the Oracle Fusion Middleware component schemas with which a service instance was provisioned.

- Do not change the ports for the Oracle WebLogic Server administration server and the Oracle Traffic Director administration server.

- Do not change OS users and SSH key settings that Oracle Java Cloud Service configured during creation of a service instance.

4

Deploying and Undeploying Applications for an Oracle Java Cloud Service Instance on Oracle Public Cloud Machine

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 Deployment Tasks for an Oracle Java Cloud Service Instance

Consider the typical tasks for deploying and undeploying an application to an Oracle Java Cloud Service instance, as shown in the following table.

Task	Description	More Information
Use Fusion Middleware Control	Deploy and undeploy applications just as you would for an on-premise service instance.	Using Fusion Middleware Control to Deploy an Application
Use the WebLogic Server Administration Console	Deploy and undeploy applications just as you would for an on-premise service instance.	Using the WebLogic Server Administration Console to Deploy and Undeploy an Application
Use WLST commands	Use WLST commands online or offline to deploy an application.	Using WLST Commands to Deploy and Undeploy an Application

Task	Description	More Information
Use an IDE	Use an IDE such as Eclipse to deploy and undeploy an application.	Using an IDE to Deploy and Undeploy an Application
Access a deployed application	Copy the public IP address of the load balancer into the URL for the application.	Accessing an Application Deployed to an Oracle Java Cloud Service Instance
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.	Defining a Custom Domain Name for an Oracle Java Cloud Service Instance Application Environment
Define a custom URL for a deployed application	Define a custom "vanity" URL for an application deployed to an Oracle Java Cloud Service instance.	Defining a Custom URL for an Application Deployed to an Oracle Java Cloud 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.	Configuring SSL for Your Custom Domain in an Oracle Java Cloud Service Instance Application Environment
Use third-party frameworks with Oracle Java Cloud Service	Use third-party frameworks to extend the functionality of Oracle Java Cloud Service.	Using Third-Party Frameworks with Oracle Java Cloud Service

Using 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

Using 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 service instance.

The following documentation is available to help you learn more about using Fusion Middleware Control to deploy an application:

- Oracle Fusion Middleware 12.2.1

  Deploying, Undeploying, and Redeploying Java EE Applications in *Administering Oracle Fusion Middleware*

- Oracle Fusion Middleware 12.1.3

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*

To deploy an application by using Fusion Middleware Control:

1. Navigate to the Oracle Java Cloud Service Console.
2. Open the Fusion Middleware Control console.

   The Fusion Middleware Control Logic page is displayed.
3. Log in to the Fusion Middleware Control console by using the administrator log-in credentials you specified when you created the Oracle Java Cloud Service instance.

   The Fusion Middleware Control console is displayed.
4. From the navigation pane on the Fusion Middleware Control console, expand the domain in which you want to deploy the application.
5. Select the cluster or server to which you want to deploy the application.

   The server home page is displayed.
6. From the WebLogic Cluster menu, choose **Control**, then **Deployments**.

   The Deployments page is displayed.
7. In the Change Center, select **Changes** – **Lock & Edit**.
8. Click **Deploy** to open the Deploy Java EE Application Assistant.

   The Select Archive page is displayed.
9. In the Archive or Exploded Directory section, select one of the following choices and browse to select the application or directory:
   - Archive is on the machine where the browser is running.
   - Archive or exploded directory is on the server where Enterprise Manager is running.
10. Depending on whether you have a deployment plan, and the location of the deployment plan, select one of the following choices:
    - Create a new deployment plan when deployment configuration is done.
    - Deployment plan is on the machine where the Web browser is running.
    - Deployment plan is on the server where Enterprise Manager is running.
11. In the Deployment Type section, select one of the following choices and click **Next**:
    - Deploy this archive or exploded directory.
    - Deploy this archive or exploded directory as a library

    The Select Target page is displayed.

12. Depending on where you want to deploy the application, select the cluster an individual Managed Server and click **Next**.

    The Application Attributes page is displayed.

13. In the Deployment Type section, select one of the following choices and click **Next**:

    - Deploy the archive or exploded directory as an application
    - Deploy the archive or exploded directory as a library

    The Select Target page is displayed.

14. Select the Managed Server or **All managed servers in the cluster** and click **Next**.

    The Application Attributes page is displayed.

15. Under Application Attributes, in the Application Name field, enter the name of the application if the software has not provided the name already.

16. In the Context Root of Web Modules section, enter the application context root.

17. In the Distribution section, select one of the following choices:

    - Install and start application (servicing all requests)
    - Install and start application in administration mode (servicing only administration requests)
    - Install only. Do not start.

18. Keep or change choices in the Other Options section and click **Next**.

    The Deployment Wizard, Deployment Settings page is displayed.

19. Keep or change default settings.

20. Click **Deploy**.

    The Fusion Middleware Control console displays processing messages.

21. When the deployment is completed, click **Close**.

22. In the Change Center, select **Changes – Activate Changes**.

    Note that the application appears on the Deployments table.

The application has been deployed.

Using Fusion Middleware Control to Undeploy 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 undeploy the application on an on-premises service instance.

The following documentation is available to help you learn more about using Fusion Middleware Control to undeploy an application:

- Oracle Fusion Middleware 12.2.1

Deploying, Undeploying, and Redeploying Java EE Applications in *Administering Oracle Fusion Middleware*

- Oracle Fusion Middleware 12.1.3

  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*

To undeploy an application by using Fusion Middleware Control:

1. Navigate to the Oracle Java Cloud Service Console.
2. Open the Fusion Middleware Control console.

   The Fusion Middleware Control console is displayed.
3. From the navigation pane, expand **Application Deployments**.
4. Select the application to undeploy.

   The application home page is displayed.

   Note that the application is deployed in an **Active** state.
5. From the Domain Application Deployment menu, choose **Deployments**.

   The Deployments page is displayed.
6. In the Change Center, click **Changes – Lock & Edit**.
7. Select the application and click **Undeploy.**.

   The Confirmation page is displayed.
8. In the Confirmation page, click **Undeploy**.

   Processing messages are displayed.
9. When the operation completes, click **Close**.
10. In the Change Center, select **Changes – Activate Changes**

    The application no longer appears on the Deployments table.

The application is undeployed.

Using the WebLogic Server Administration Console to Deploy and Undeploy an Application

You can use the WebLogic Server Administration Control graphical user interface to deploy and undeploy an application to an Oracle Java Cloud Service instance, just as you would deploy and undeploy the application to an on-premises service instance.

Topics

- Using the WebLogic Server Administration Console to Deploy an Application

Using the WebLogic Server Administration Console to Deploy an Application

You can use the Administration Console to deploy an application to an Oracle Java Cloud Service instance.

To deploy an application by using the WebLogic Server Administration Console:

1. Navigate to the Oracle Java Cloud Service Console.
2. Open the WebLogic Server Administration Console.

   The WebLogic Server Administration Console is displayed.
3. On the Welcome screen, enter the administrator login credentials you specified when you created the Oracle Java Cloud Service instance.
4. On the WebLogic Server Console, click **Lock & Edit**.
5. Under Domain Structure, select ***domain*** **> Deployments**, where *domain* is the domain into which you want to deploy the application.
6. On the Deployments page, click **Install**.
7. On the Install page, click **upload your file(s)** ..
8. In the Install Application Assistant, click **Browse** to find the application you want to deploy and click **Next**.
9. Under **Locate deployment to install and prepare for deployment**, select the application if it is not already selected, and click **Next**.
10. Under **Choose targeting style**, you can either install the deployment as an application or as a library. Make your selection and click **Next**.
11. Under **Select deployment targets**, select the servers or clusters to which you want to deploy the application and click **Next**.
12. (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.
13. Review the configuration settings you specified and click **Finish**.

    The application shows a status of **distribute Initializing** on the Deployments table.

14. In the Change Center, click **Activate Changes**.

    The changes are activated and no restart is necessary. The WebLogic Server Console shows the application in a **Prepared** state.

The application is now deployed.

Using 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. Navigate to the Oracle Java Cloud Service Console.
2. Open the WebLogic Server Administration Console.

   The WebLogic Server Administration Console is displayed.
3. On the Welcome screen, enter the administrator login credentials you specified when you created the Oracle Java Cloud Service instance.
4. Under Domain Structure, click ***domain*** – **Deployments**, where ***domain*** is the domain into which you want to deploy the application.
5. Click the **Control** tab if not already selected.
6. From the Deployments table select the application.
7. Click **Start**, then **Servicing all requests**.
8. 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.

Using 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. Navigate to the Oracle Java Cloud Service Console.
2. Open the WebLogic Server Administration Console.

   The WebLogic Server Administration Console is displayed.
3. On the Welcome screen, enter the administrator login credentials you specified when you created the Oracle Java Cloud Service instance.
4. In the Change Center of the WebLogic Server Administration Console, click **Lock & Edit**.
5. Under Domain Structure, select ***domain*** –**Deployments**, where ***domain*** is the domain in which the application is deployed.
6. In the Deployments table, select the check box next to the application you want to undeploy.

7. From the **Stop** menu, select one of the following:
   - **When work completes**
   - **Force Stop Now**

   Do not select **Stop, but continue servicing administration requests**.
8. When the application is in a **Stopped** state, ensure that the application is selected and click **Delete**.
9. Click **Yes** to confirm your decision and remove the application.
10. To activate your changes, in the Change Center of the WebLogic Server Administration Console, click **Activate Changes**.

The application has been undeployed.

Using 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 Running WLST Commands on a VM. 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

  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 12.1.3

  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 11.1.1.7

  Using WLST Commands Online to Deploy Applications in *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*

  Using WLST Commands Offline to Deploy Applications in *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*

Using 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.

Topics

Prerequisites to Deployment with an IDE

Complete the following tasks before deploying an application:

- Create an Oracle Java Cloud Service instance.
- Install the IDE on your local machine.

  If you want to use Eclipse, install Oracle Enterprise Pack for Eclipse (OEPE). You can download OEPE from this site:

  `http://www.oracle.com/technetwork/developer-tools/eclipse/downloads/index.html`

Connecting 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.

To connect the Eclipse IDE to a remote 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 on **Workbench**.

   The Workbench is displayed.
4. Select **Window – Show View – Others...**
5. Type `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 Oracle WebLogic Server version you are using and click **Next**.

   You chose the version number when you created your Oracle Java Cloud Service instance.

8. On the New Server dialog, click the browse icon next to **WebLogic home** and select your WebLogic Server home directory.

   Note: Make sure that the local version of Weblogic you're running is the same version as the instance running on the cloud service. If they are not the same version then you will be unable to 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 cloud service Weblogic instance.

    b. Set **Port** to **7002**. (This is the SSL port of your cloud service Weblogic instance.)

    c. Make sure that **Use SSL port** is checked.

    d. Set **User** to the WebLogic administrator credentials you specified when you created the Oracle Java Cloud Service instance.

11. Click **Test Connection**.

    A Success dialog is displayed, informing you that the test connection succeeded.

12. In the Success dialog, click **OK**.

    On 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**.

Using 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 the 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.

   For information, see Connecting an IDE to a Remote WebLogic 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.

Using 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.

Accessing 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.

To access a deployed application:

1. Navigate to the Oracle Java Cloud Service Console.
2. Click the service instance where you deployed the application.

   The Oracle Java Cloud Service Instance Overview page is displayed.

3. Copy the Public IP Address of the load balancer or managed server, depending on whether your Oracle Java Cloud Service instance has a load balancer.
4. Find the context-root of the application.

   The context-root is defined in the service project 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. Navigate to the Oracle Java Cloud Service Console.

   b. From the Oracle Java Cloud Service Console, open the WebLogic Server Administration Console.

   c. Select *domain* > Deployments, where *domain* is the domain where the application is deployed.

   d. In the Deployments table, click on the name of your service.

      The Settings dialog is displayed.

   e. In the Overview tab, locate the context-root.

5. Open a browser.
6. In the address bar, specify the URL of the application:

- `https://<public_IP_of_load_balancer_or_managed_server>/<application_context_root>`
- `http://<public_IP_of_load_balancer_or_managed_server>/<application_context_root>`

a. Specify the SSL or non-SSL protocol.

If you have 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 Enabling HTTP Access to an Oracle Java Cloud Service Instance.

b. Paste the Public IP Address of the load balancer (or the managed server if not using a load balancer) into the URL.

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 must acquire and configure a third-party DNS provider to map the custom domain name. See Defining a Custom Domain Name for an Oracle Java Cloud Service Instance.

Note: Programs should not use IP addresses to access applications running in a production environment. Production services should be referenced using a custom domain name, or the default internal host name if a custom domain name has not been provided.

c. Specify the context-root for the application.

If you want to customize the default URL for your application, see Defining a Custom URL for an Application Deployed to an Oracle Java Cloud Service Instance.

7. If you receive a warning, accept the self-signed certificate.

The application opens in your browser.

Enabling the JVM Debug Port

Opening a JVM Debug port on your Oracle Java Cloud Service instance 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, so make sure that the server itself is not running under a load balancer.

Topics

Setting up the Debug Port in WebLogic Server

In order to set up the debug port on WebLogic Server, you need to make changes to the server's start up parameters. Once the changes have been made you will need to restart the server.

1. Navigate to the Oracle Java Cloud Service console
2. Open the WebLogic Server Administration Console.

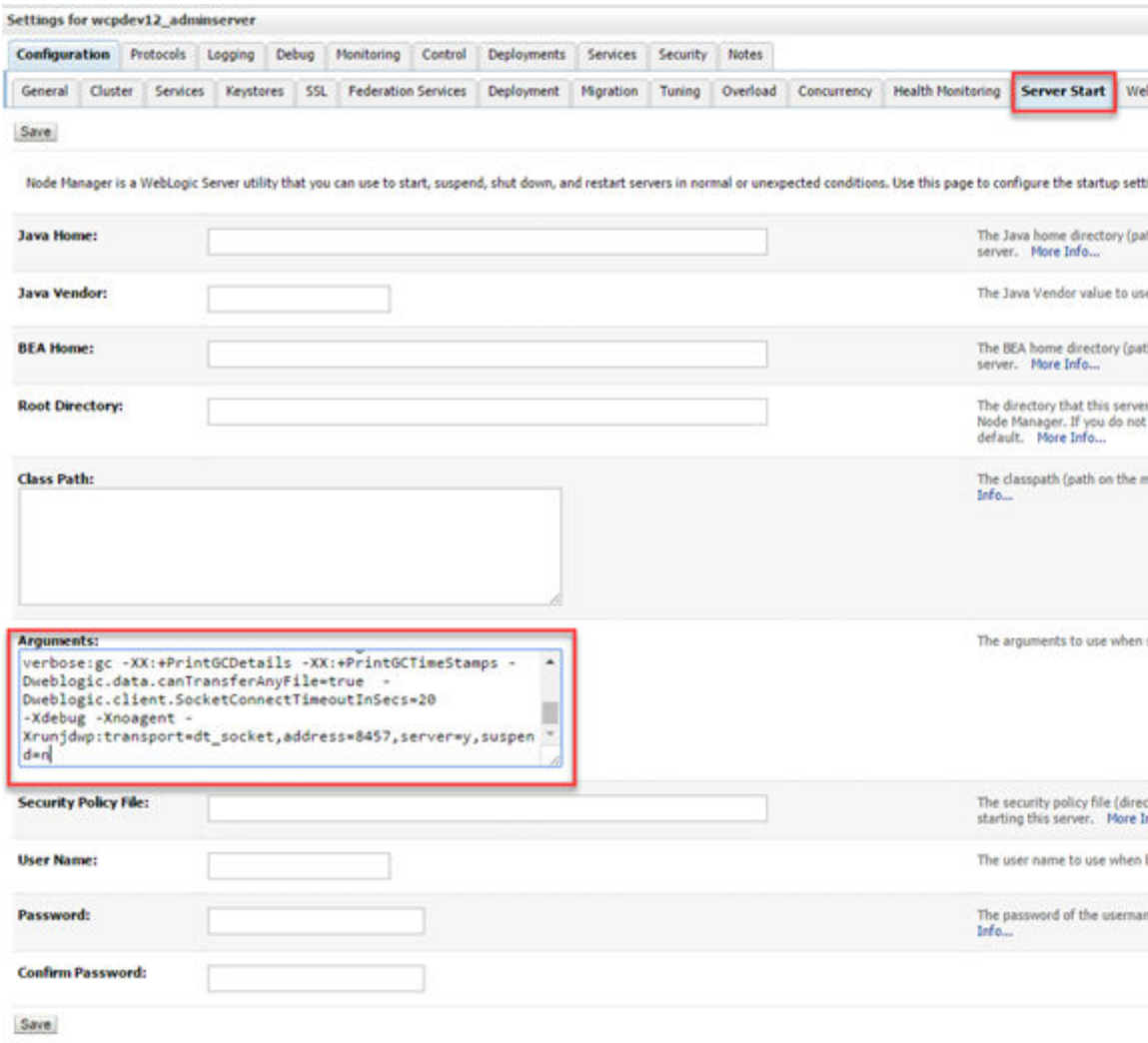
   For more information, see Accessing the Administrative Consoles Used by Oracle Java Cloud Service.

   The WebLogic Server Administration Console is displayed.
3. On the Welcome screen, enter the administrator login credentials.
4. On the WebLogic Server Console, click **Lock & Edit**.
5. Click **Servers** from the **Domain Structure** tile.

   The list of available servers is displayed.

   Note: Make sure that you are on the **Configuration** tab.
6. Click on the **Name** of the server running your application to access its configuration.
7. Navigate to the **Server Start** tab.
8. Add the following entry to the **Arguments**:

```
-Xdebug -Xnoagent -Xrunjdwp:transport=dt_socket,address=8457,server=y,suspend=n
```



9. Save your changes and click on **Activate Changes**.

10. Restart the server.

Using Third-Party Frameworks with Oracle Java Cloud Service on OPCM

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 Deploying and Undeploying 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 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 with Oracle Java Cloud Service.

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.
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.
Apache Tapestry	5.3.7	Simplify the development of dynamic, robust, highly scalable web applications in Java.

Framework	Release Tested	Purpose
Apache Thrift	0.9.0	Build services that work efficiently and seamlessly between languages including, among other langauges: • 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.

Framework	Release Tested	Purpose
JRuby	1.7.2	Provide a 100% Java implementation of the Ruby programming language.
PrimeFaces	4.0	Provide a library of user interface components for JavaServer Faces (JSF) applications.
Quartz Job Scheduler	2.1.5	Create simple or complex schedules for executing jobs whose tasks are defined as standard Java components.
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.

Omitting Checks for Updates to Quartz Job Scheduler on OPCM

By default, Quartz Job Scheduler checks for updates when it starts.

The check for updates involves connecting to a remote server. If the server cannot be reached, the check fails and an exception is written to 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 following Quartz Job Scheduler documentation:

   - Skip Update Check in *Best Practices*
   - Configure Main Scheduler Settings in *Configuration Reference*

   For information about how to set a property, see the Administration Console online help for the release of Oracle WebLogic Server that you are using:

   - For release 12.2.1.0, see Configure startup arguments for Managed Servers in *Oracle WebLogic Server Administration Console Online Help 12c (12.2.1)*.

- For release 12.1.3.0, see Configure startup arguments for Managed Servers in*Oracle WebLogic Server Administration Console Online Help 12c (12.1.3)*.
- For release 11.1.1.7, see Configure startup arguments for Managed Servers in*Oralce WebLogic Server Administration Console Online Help 11g Release (10.3.6)* .

2. Restart each managed server for which you set a property in the previous step.

   For more information, see Using the WebLogic Server Administration Console to Start Managed Servers.

5

Scaling an Oracle Java Cloud Service Instance on Oracle Public Cloud Machine

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

You can scale an Oracle Java Cloud Service instance by scaling a cluster, a node, or the Coherence data tier in the service instance.

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 heap usage is high, consider scaling up the nodes in the cluster. Viewing Service Metrics in Oracle Java Cloud Service on OPCM provides information about metrics that Oracle Java Cloud Service provides for a service instance.

You cannot scale a service instance if the service instance is under maintenance, such as during patching or backing up.

Topics:

About Scaling an Oracle Java Cloud Service Cluster

Scale an Oracle Java Cloud Service cluster to add nodes to or remove nodes from the cluster in response to changes in the load on the cluster. A node is a virtual machine (VM) running a Managed Server instance that is a member of a cluster.

Note: Scaling a cluster is **not** supported by Oracle Java Cloud Service—Virtual Image instances.

About Scaling Out an Oracle Java Cloud Service Cluster

Scaling out an Oracle Java Cloud Service cluster adds one node to the cluster.

Before scaling out an Oracle Java Cloud Service cluster, ensure that all these conditions are met:

- The service instance is **not** under maintenance.

If any of these conditions is not met, the scaling operation fails and Oracle Java Cloud Service logs an error message.

Oracle Java Cloud Service logs a message when scaling out is started or completed, or when a failure is detected. See Viewing Scaling Requests on OPCM.

If an attempt to scale out a cluster fails, Oracle Java Cloud 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 VM that it created to run the additional Managed Server instance.

About Scaling In an Oracle Java Cloud Service Cluster

Scaling in an Oracle Java Cloud Service cluster removes the selected node from the cluster.

Before scaling in an Oracle Java Cloud Service cluster, ensure that the cluster contains at least one Managed Server node in addition to the node for the Administration Server and first Managed Server. You cannot scale in a cluster that contains only the node for the Administration Server and first Managed Server. If you no longer require that node, you must delete the entire service instance. See Deleting an Oracle Java Cloud Service Instance.

By default, Oracle Java Cloud Service scales in a cluster gracefully by shutting down the Managed Server instance before removing the Managed Server instance from the cluster and terminating its VM. To ensure that the node is removed even if the Managed Server instance is unresponsive, you can choose to forcibly scale in a cluster.

If an attempt to scale in a cluster fails, Oracle Java Cloud 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 or you want to add the new managed server to a new cluster. You can add a cluster to an instance by using the REST API to add a managed server and include the `createClusterIfMissing=true` query parameter on the scale out REST endpoint. Once the cluster is added, you can verify that it was added when it will appears in the Topology tile. This will not only add the cluster but will scale it out to include the new node.

About Scaling an Oracle Java Cloud Service Node

You can scale an Oracle Java Cloud Service node 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 can scale only the Administration Server node and Managed Server nodes in a WebLogic Server cluster. Oracle Java Cloud Service does **not** support scaling for other nodes in a service instance, such as the load balancer node or capacity unit nodes in the Coherence data tier.

You must scale each node in a cluster individually. You cannot scale all nodes in a cluster in a single operation.

About Changing the Compute Shape of a Node

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.

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. The larger the compute shape, the greater the processing power. For more information about the defined compute shapes, contact your Oracle Cloud Administrator.

- To meet the demands of heavier workloads, scale up the compute shape of a node by choosing a larger compute shape.

  For example, changing the compute shape from OC3 to OC4 doubles the capacity of the node from one OCPU to two OCPUs and doubles the amount of RAM allocated to the node.

- To save costs if the workload is lightened, scale down the compute shape of a node by choosing a smaller compute shape.

  For example, changing the compute shape from OC4 to OC3 reduces the capacity of the node by half from two OCPUs to one OCPU and reduces the amount of RAM allocated to the node by half.

> **Note:** To optimize performance and balance the load on Managed Sever instances correctly, ensure that the compute shapes of all nodes in a cluster are the same. When routing requests to Managed Server instances, the load balancer treats all Managed Sever instances as being equivalent.

About Adding Block Storage to a Node

You can add block storage to a node that is running out of storage. When you add storage to a node, an Oracle Compute Cloud Service storage volume is created and attached to the node's VM.

> **Note:** You **cannot** remove block storage from a node.

The new storage volume created by scaling remains attached and available to the node's VM 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.

You can add the storage to a new volume or one of the following existing volumes:

- Backup storage volume (Administration Server node only)
- Domain home storage volume
- Middleware home storage volume

For details of these volumes, see About the Disk Volumes.

You can add storage a maximum of five times to a storage volume.

Caution: Before adding storage to the domain home or middleware home storage volume, back up the service instance to avoid the risk of data loss. For instructions, see Initiating an On-Demand Backup of an Oracle Java Cloud Service Instance on OPCM.

Overview of Scaling Tasks for an Oracle Java Cloud Service Instance

You 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. For information about using REST endpoints to perform these tasks, see Scaling a Service Instance and Scaling the Coherence Data Tier of a Service Instance in *REST API for Oracle Java Cloud Service*.

Task	Description	More Information
Scale out an Oracle Java Cloud Service cluster **Not** supported by Oracle Java Cloud Service—Virtual Image instances	Scale out an Oracle Java Cloud Service cluster to add one node to the cluster.	Scaling Out an Oracle Java Cloud Service Cluster on OPCM
Scale in an Oracle Java Cloud Service cluster **Not** supported by Oracle Java Cloud Service—Virtual Image instances	Scale in an Oracle Java Cloud Service cluster to remove a selected node from the cluster.	Scaling In an Oracle Java Cloud Service Cluster on OPCM
Scale an Oracle Java Cloud Service node	Scale an Oracle Java Cloud Service node to change its compute shape in response to changes in workload or to add storage to a node that is running out storage.	Scaling an Oracle Java Cloud Service Node on OPCM
Scale the Coherence data tier **Not** supported by Oracle Java Cloud Service—Virtual Image instances	When Oracle Coherence is enabled for a service instance: Scale the Coherence data tier to increase or decrease the Coherence cache capacity for an Oracle Java Cloud Service—Coherence instance.	Scaling the Coherence Data Tier by Using the Oracle Java Cloud Service Console
View scaling requests	View scaling requests to check the status of ongoing scaling requests, and the success or failure of previous requests.	Viewing Scaling Requests on OPCM

Scaling Out an Oracle Java Cloud Service Cluster on OPCM

You can scale out an Oracle Java Cloud Service cluster to add one node to the cluster. When you scale out, Oracle Java Cloud Service creates a new VM running a WebLogic Server Managed Server instance.

> **Note:** Scaling out a cluster is **not** supported by Oracle Java Cloud Service - Virtual Image instances.

To scale out an Oracle Java Cloud Service cluster:

> **Note:** If this 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, you should delete those servers manually or select **Force Remove** during the Remove Node process.

1. Navigate to the Overview page for the cluster to which you want to add a node.
2. Click  .

   (Alternately, click the  next to the instance name.)

   The Add Node dialog box is displayed.
3. To confirm you want to scale out the cluster, click **Add Node**.
4. After a few moments, click the  (Refresh) icon to update the page.

   You might need to click the icon more than once to see any change to the page.

After a few moments of processing, the new node will appear on the Overview page. The **Overview** tile will show the number of nodes increased by one.

> **Note:** 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-out status in the activity table.

The VM of the new node has the compute shape and the amount of storage with which the service instance was originally created. If necessary, you can change the compute shape and add storage as explained in Scaling an Oracle Java Cloud Service Node on OPCM.

Adding 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` query parameter on the scale out REST endpoint. Once the cluster is added, you can verify that it was added when it

will appears in the Topology tile. This will not only add the cluster but will scale it out to include the new node.

For example, if you wanted to create cluster for the node you are adding by using cURL, you would invoke the API like this:

```
curl -i -X POST -u username:userPassword1! -H "X-ID-TENANT-
NAME:ExampleIdentityDomain" https://rest_server_url/paas/service/jcs/api/v1.1/
instances/ExampleIdentityDomain/ExampleInstance/servers/ExampleI_cluster?
createClusterIfMissing=true
```

`createClusterIfMissing` defaults to false. If not specified, the Managed Server is added to the existing cluster.

Scaling out by using the REST API, including how to use `createClusterIfMissing`, is discussed in greater detail in Scale Out a Cluster in the *REST API for Oracle Java Cloud Service*.

Scaling In an Oracle Java Cloud Service Cluster on OPCM

You can scale in an Oracle Java Cloud Service cluster to remove a selected node from the cluster. When you scale in, Oracle Java Cloud Service removes the selected WebLogic Server Managed Server instance and the VM that it is running on.

If you've created any managed servers outside of the Java Cloud Service provisioning and scaling (for example, from the WebLogic console), you should either manually delete those servers before scaling-in a cluster, or use the force delete feature, as described in step 3, below.

Note: Scaling in a cluster is **not** supported by Oracle Java Cloud Service—Virtual Image instances.

To scale in an Oracle Java Cloud Service cluster:

Note: If this cluster contains any managed servers that were created outside of Oracle Java Cloud Service (for example, by using the WebLogic console), before scaling in the cluster, you should delete those servers manually or select **Force Remove** during the Remove Node process, as described in step 3, below.

---

1. Navigate to the Overview page for the cluster from which you want to remove a node.
2. Click  for the node you want to remove and, from context menu, select **Remove node**.

   The Remove Node dialog box is displayed.
3. In the Remove Node dialog box, confirm how you want to scale in.
   - To scale in gracefully, click **Remove Node**.
   - To forcibly scale in, select **Force Remove** and click **Remove Node**.

     If you forcibly scale in, Oracle Java Cloud Service removes the node even if the node's VM or Managed Server instance is unresponsive.

4. After a few moments, click the (Refresh) icon to update the page.

   You might need to click the icon more than once to see any change to the page.

After a few moments of processing, the node is removed from the Overview page. The **Overview** tile will show the number of nodes decreased by one.

> **Note:** 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.

Scaling an Oracle Java Cloud Service Node on OPCM

You can scale an Oracle Java Cloud Service node 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.

> **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.

To scale an Oracle Java Cloud Service node:

1. Navigate to the Overview page.
2. From the menu for the node, choose **Scale Up/Down**.

   The Scale Up/Down Server dialog box opens.
3. In the Scale Up/Down Server dialog box, specify the new compute shape of the node, how much storage to add to the node, and where to add the storage.

Option	Description
New Compute Shape	From the drop-down list, select the new compute shape for the node. • To scale up the node, choose a larger compute shape than the current compute shape. • To scale down the node, choose a smaller compute shape than the current compute shape.
Additional Storage (GB)	Enter a whole number in the range 1-1,000 to specify the number of Gbytes of block storage you want to add, if any.

Option	Description
Add Storage to	If you are adding block storage, from the drop-down list, select one of the following storage volumes to which the storage will be added: • New Storage Volume • Backup Storage Volume • Domain Home Storage Volume • Middleware Home Storage Volume You have access to all Oracle Compute Cloud Service virtual machine instances created for Oracle Java Cloud Service, including the virtual machine on which the WebLogic Administration Server is running. If you are unfamiliar with how to use these volumes, see About the disk Volumes.

4. Click **Yes, Scale Up/Down Server**.

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.

Note: 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 are equivalent, you must scale the new node to match the other nodes in your cluster.

Adding Block Storage Only

You can scale up a node by adding block storage without changing the instance compute shape.

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.

To add block storage only, follow the instructions in Scaling an Oracle Java Cloud Service Node on OPCM but leave the compute shape as is. Specify the new amount of block storage in **Additional Storage (GB)** and, from the **Add Storage to** drop-down list, select the storage volumes to which you want to add the storage:

- New Storage Volume
- Backup Storage Volume
- Domain Home Storage Volume
- Middleware Home Storage Volume

Be aware that, if you add additional block storage, you cannot remove it later.

Viewing Scaling Requests on OPCM

You can view scaling requests to check the status of ongoing scaling requests, 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.
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 Status column will indicate whether the scaling operation succeeded or failed. To see more details about a specific operation, click its drop-down control.

6

Backing Up and Restoring an Oracle Java Cloud Service Instance on Oracle Public Cloud Machine

You can back up and restore your Oracle Java Cloud Service instances to return their software and data to a particular state.

Note:

This section does **not** apply to Oracle Java Cloud Service—Virtual Image instances. Backup and restoration are **not** supported by Oracle Java Cloud Service—Virtual Image instances.

Topics:

About Backup and Restoration of Oracle Java Cloud Service Instances

By backing up your Oracle Java Cloud Service instances, you can preserve them in a particular state. If you later make configuration changes to a service that you don't want, you can undo them by restoring the service instance's configuration data from a backup. You can also restore the software to its current official patch set update (PSU) level.

Topics:

Contents of a Backup

What a backup contains depends on whether the backup is an **incremental backup** or a **full backup**.

Oracle Java Cloud Service ensures that backups contain only the volumes and other metadata that are needed for a proper restoration of a service instance. For details about the volumes that are backed up, see About the Disk Volumes.

Topics:

Contents of a Full Backup

A full backup contains all the runtime artifacts required to restore the service instance's configuration data.

Specifically, a full backup contains these items:

- The Oracle WebLogic Server domain configuration of the service instance, which consists of these items:
  - The `$DOMAIN_HOME` volume of each virtual machine

    This includes any Managed Server persistent stores that are not stored in the database. WebLogic Server persistent stores are used for transaction logs and Java Message Service (JMS) providers.
  - Oracle WebLogic Server domain configuration files in the `$MW_HOME` volume of the Administration Server virtual machine
- Oracle Traffic Director configuration for the load balancer

Note:

Oracle Java Cloud Service does **not** back up any software, including Oracle Fusion Middleware software installed on the `$MW_HOME` volume. You are responsible for ensuring that you can re-install any software that you have installed on a service instance's VMs if necessary.

For more information about the disk volumes in a service instance see About the Disk Volumes.

Contents of an Incremental Backup

An incremental backup contains changes to configuration data on all virtual machines since the last scheduled full backup.

Note:

All incremental backups are automated scheduled backups. You cannot create an incremental backup on demand.

While creating an incremental backup, Oracle Java Cloud Service promotes the backup to a full backup if any of the following conditions are met:

- The configuration data of the service instance has been restored since the last scheduled full backup.
- The service instance has been scaled out since the last scheduled full backup.
- Oracle Java Cloud Service can now reach a virtual machine that it could not reach during the last scheduled full backup.
- The last full scheduled backup is no longer available.

You can recognize a promoted backup in the list of available backups from its type and creation time. Any full, automated backup created at the scheduled time for an incremental backup has been promoted.

Contents of a Database Backup

A database backup contains Oracle Fusion Middleware component schemas and application data for an Oracle Java Cloud Service instance.

Oracle Java Cloud Service can only back up this database if the service instance is connected to an Oracle Database Cloud Service database deployment. See Items that Are Not Backed Up.

All scheduled automated backups include a database backup. You can choose whether or not to include a database backup in backups that you initiate. For more information, see:

- How Backups Are Initiated
- Initiating an On-Demand Backup of an Oracle Java Cloud Service Instance

When you initiate an on-demand backup, you have the option to keep the backup forever (it will not expire and not be deleted automatically). The extent of a database backup depends on whether the backup is to be kept forever:

- If the backup is to be kept forever, the database backup is a full backup.
- Otherwise, the Recovery Manager (RMAN) software that backs up the database determines whether the backup shall be a full backup or an incremental backup.

> **Note:** To ensure that you can restore the database for an Oracle Java Cloud Service instance without risking data loss for other service instances, do **not** use the same Oracle Database Cloud Service database deployment with multiple Oracle Java Cloud Service instances. Backups of an Oracle Database Cloud Service database deployment that is used with multiple Oracle Java Cloud Service instances contain data for all the Oracle Java Cloud Service instances. If you restore the database while restoring an Oracle Java Cloud Service instance, data for all the Oracle Java Cloud Service instances is restored.

Links Between an Incremental Backup and Its Related Full Backup

Each incremental backup is linked to the last full scheduled backup that was performed before the incremental backup. As a result, each full scheduled backup is linked to all incremental backups that were performed between that full backup and the next scheduled full backup.

You can restore a service instance from an incremental backup without the need to restore the full backup to which the incremental backup is linked. Oracle Java Cloud Service restores all the data from the linked full backup required for a complete restoration of the service instance. For more information, see Restoring an Oracle Java Cloud Service Instance from a Backup.

However, you **cannot** delete or archive a full backup with linked incremental backups without deleting or archiving the incremental backups. For more information, see:

- Deleting a Backup
- Deleting a Backup
- Downloading or Archiving a Backup for a Service Instance in *REST API for Oracle Java Cloud Service*

Items that Are Not Backed Up

Oracle Java Cloud Service ensures that backups contain only the information that is needed for a proper restoration of a service instance.

Therefore, the following items are **not** backed up:

- Users' custom volumes
- The `$JDK_HOME` volume, which contains the JDK software
- Software binary files in the `$MW_HOME` volume

Additionally, Oracle Java Cloud Service does not back up the database unless it is connected to an Oracle Database Cloud Service database deployment. For other types of databases, you can manually coordinate the database backups to occur on the same schedule as Oracle Java Cloud Service. Similarly, if you use Oracle Java Cloud Service to restore a service instance from a backup, you can also separately restore the database from a backup taken at the same time.

How Backups Are Initiated

Backups are initiated in several different ways.

- Oracle Java Cloud Service initiates scheduled automated backups on the following default schedule:
  - Full backups are initiated weekly starting 12 hours after a service instance was created, rounded to the nearest five-minute interval.

    For example, if a service instance is created 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 at 1:01 PM on a Monday, incremental backups are initiated at 1:00 AM every day except Tuesdays.

  You can change the schedule on which automated backups are initiated. See Configuring Automated Backups for an Oracle Java Cloud Service Instance.
- You can initiate a backup immediately without having to wait for the next scheduled backup as explained in Initiating an On-Demand Backup of an Oracle Java Cloud Service Instance.
- Oracle Java Cloud Service initiates a full backup immediately before a service instance is patched, scaled in, or scaled out.

Backups are enabled by default in Oracle Java Cloud Service, but you can disable this feature on new and existing service instances. See:

- Disabling or Enabling Backups for an Oracle Java Cloud Service Instance on OPCM

Where Backups Are Stored

Oracle Java Cloud Service stores all backups on a remote file system. To speed up restorations from recent backups, Oracle Java Cloud Service also keeps a local copy of any backup it has recently created.

> **Note:** Do not attempt to download the backup files generated by Oracle Java Cloud Service. These files are encrypted and not accessible offline. You must use Oracle Java Cloud Service to restore a service instance from a backup.

How Backups on a Remote File System Are Stored

By default, Oracle Java Cloud Service stores backups on the remote file system that was specified when the service instance was created. You can choose to store the backups on a different file system as explained in Configuring Automated Backups for an Oracle Java Cloud Service Instance on OPCM.

> **Note:** If the file system contains backup archives that Oracle Java Cloud Service has moved there, it cannot be changed.

Oracle Java Cloud Service automatically deletes a backup when the retention period for the backup has elapsed.

How Local Copies of Backups Are Stored

Oracle Java Cloud Service stores local copies in a dedicated volume mounted on the block storage attached to the virtual machine where the Administration Server is running. A backup fails if there is insufficient free space on this volume.

How long Oracle Java Cloud Service keeps the local copy of a backup before deleting it depends on the extent of the backup:

- For an incremental backup, Oracle Java Cloud Service keeps the local copy for seven days.
- For a full backup, Oracle Java Cloud Service keeps the local copy for as long as the local copy of its last related incremental backups is kept, or for seven days, whichever is longer.

How Backups and Local Copies Are Deleted Automatically

After completing the day's scheduled backup, Oracle Java Cloud Service deletes any backups or local copies that are due to be deleted that day. If the scheduled backup fails because of insufficient space, backups and local copies that are due to be deleted are still deleted.

Note:

When an Oracle Java Cloud Service instance is deleted, all its backups are deleted.

What Happens During a Backup

During a backup of an Oracle Java Cloud Service instance, the service instance continues to run and all applications deployed to the service instance remain available.

To prevent configuration changes during a backup, Oracle Java Cloud Service locks the Oracle WebLogic Server domain. After locking the domain, Oracle Java Cloud Service backs up files on each node as described in Contents of a Backup.

Note:

Do **not** to attempt to start the administration server while a backup is in progress.

While the backup is in progress, you cannot start any other management operation on the service instance.

When the backup is complete, Oracle Java Cloud Service unlocks the Oracle WebLogic Server domain. If the backup is a scheduled backup, Oracle Java Cloud Service also cleans up aged backups as follows:

- It deletes from local storage all backups old enough to be stored only on a remote file system.
- It deletes from wherever they are stored any remaining copies of backups whose retention period has elapsed.

If the scheduled backup fails because of insufficient space, the aged backups are still cleaned up.

If the backup includes a database backup, Oracle Java Cloud Service uses Recovery Manager (RMAN) to back up the Oracle Database Cloud Service database deployment. The backup of the database is coordinated with the backup of other volumes.

Note:

Oracle Java Cloud Service does not automatically remove transaction records when backing up a service instance. Therefore, you must remove transaction records after you restore a service instance from a backup.

How Restorations Are Initiated

Restorations are initiated in several different ways.

- You can initiate a restoration as explained in Restoring an Oracle Java Cloud Service Instance from a Backup.
- When a patch is rolled back, Oracle Java Cloud Service initiates a restoration to return the software to the official PSU level it was at before the patch was applied. The restoration that Oracle Java Cloud Service initiates appears in the instance's restoration history on the Backup page.

What Happens when a Service Instance Is Restored

When restoration of a service instance is initiated, Oracle Java Cloud Service shuts down the service instance's servers. After the restoration is complete, Oracle Java Cloud Service restarts the service instance's servers.

If the hosts in the backup and the service instance don't match, how Oracle Java Cloud Service handles the mismatch depends on the cause of the mismatch.

The Service Instance Contains Managed Server Hosts Not in the Backup

If the service instance contains any managed server hosts that are not in the backup, Oracle Java Cloud Service warns you that it cannot restore the managed server hosts that are not part of the backup.

Before trying to restore again, you can scale in the service instance to delete the nodes that correspond to these managed server hosts. For more information, see Scaling In an Oracle Java Cloud Service Instance.

If you choose to continue without scaling in the service instance, Oracle Java Cloud Service asks you to confirm that you understand that the service instance will be scaled in automatically.

The Service Instance Contains Hosts on the Coherence Data Tier Not in the Backup

When Oracle Coherence is enabled for a service instance: Managed Servers configured on the Coherence data tier are identified by *managedserver*_server_*n*_DG. If a service instance contains Managed Server hosts on the Coherence data tier that are not in the backup, scale in the service instance by removing the capacity unit that manages those specific nodes and the corresponding managed server hosts before trying to

restore the service instance. For information about removing a specific capacity unit, see Scaling the Coherence Data Tier by Using the Oracle Java Cloud Service Console.

The Backup Contains Hosts Not in the Service Instance

If the backup contains any hosts that are not in the service instance, Oracle Java Cloud Service does **not** attempt to remove the managed servers on these hosts from the administration server configuration. You must use Oracle WebLogic Server to remove the managed servers on these hosts from the administration server configuration.

Typical Workflow for Backing Up and Restoring an Oracle Java Cloud Service Instance

To back up and restore an Oracle Java Cloud Service instance, consider this typical workflow.

Task	Description	More Information
Configure backups for an Oracle Java Cloud Service instance	Customize the following properties of backups for an Oracle Java Cloud Service instance: • When automated backups are performed • Where backups are stored • How long new backups are retained	Configuring Backups for an Oracle Java Cloud Service Instance
Initiate an on-demand backup of an Oracle Java Cloud Service instance	Create a backup immediately without having to wait for the next scheduled backup.	Initiating an On-Demand Backup of an Oracle Java Cloud Service Instance Initiating an On-Demand Backup for a Service Instance in *REST API for Oracle Java Cloud Service*
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.	Deleting a Backup
Restore an Oracle Java Cloud Service instance from a backup	Undo configuration changes you don't want by returning an Oracle Java Cloud Service instance to a particular state.	Restoring an Oracle Java Cloud Service Instance from a Backup Restoring a Service Instance in *REST API for Oracle Java Cloud Service*
Return an Oracle Java Cloud Service instance to service after restoration from a backup	Modify a restored service instance to return it to the state you require and perform steps to return the service instance to service that Oracle Java Cloud Service does not automate.	Returning an Oracle Java Cloud Service Instance to Service After Restoration from a Backup
Download or archive a backup	Use a REST endpoint to move a backup from the remote NFS backup destination to block storage.	Downloading or Archiving a Backup for a Service Instance in *REST API for Oracle Java Cloud Service*

Configuring Automated Backups for an Oracle Java Cloud Service Instance on OPCM

You can configure backups for an Oracle Java Cloud Service instance to customize when the service instance is automatically backed up and how backups are stored.

Specifically, you can customize the following properties of the service instance:

- **When automated backups are initiated.** By default, backups are performed at the following times:
  - Full backups are initiated weekly starting 12 hours after the service instance was created, rounded to the nearest five-minute interval.

    For example, if a service instance is created 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 at 1:01 PM on a Monday, incremental backups are initiated at 1:00 AM every day except Tuesdays.
- **Where backups are stored.** By default, backups are stored in the remote file system that you provided when the service instance was created.

Because the changes affect only one service instance, you can configure different values for these properties for each of your service instances. However, you cannot configure automated backups for a service instance while it is currently being backed up.

Alternatively, you can disable all backups. See Disabling or Enabling Backups for an Oracle Java Cloud Service Instance on OPCM.

To configure automated backups for an Oracle Java Cloud Service instance:

1. Navigate to the Backup page.
   a. Access the Oracle Java Cloud Service console.
   b. Click the name of the service instance for which you want to configure automated backups.

      The Oracle Java Cloud Service Instance page is displayed with the Overview tile in focus, displaying detailed information about the service instance.
   c. Click the **Administration** tile.

      The Oracle Java Cloud Service Instance page is refreshed with the Administration tile in focus.
   d. Click the **Backup** tab.

      The Backup page is displayed.
2. Click **Configure Backups**.

   The Configure Backups dialog box opens.

3. In the Configure Backups dialog box, set the options to configure automated backups for the service instance.

   a. In the Schedule section, set options to configure when automated backups are performed:

   Note:

   All times must be for the Coordinated Universal Time (UTC) time zone, not your local time zone.

Option	Description
Full Backup	From the drop-down lists, select the day of the week and the time of day UTC when you want full backups to be performed.
Incremental Backup	From the drop-down list, select the time of day UTC when you want incremental backups to be performed.

   b. In the **NFS Remote Backup** field, enter the remote file-system directory where you want backups to be stored.

4. Click **Save**.

Initiating an On-Demand Backup of an Oracle Java Cloud Service Instance on OPCM

You can create a full backup immediately without having to wait for the next scheduled backup.

Create a backup when making major changes to your service instance, for example, in these situations:

- Before any configuration changes that you may need to undo
- Before deploying an application
- After deploying an application

All on-demand backups are full backups. For more information, see Contents of a Full Backup.

Note:

Do **not** to attempt to start the administration server while a backup is in progress.

To initiate an on-demand backup of an Oracle Java Cloud Service instance:

1. Navigate to the Backup page.
2. Click **Back Up Now**.

   The Back Up Now dialog box opens.

3. In the Back Up Now dialog box, set the following options:

Option	Description
Include Database	Whether to back up the database: • **Yes**—Back up the database. • **No**—Do not back up the database. **Note:** Oracle Java Cloud Service can only back up the database if the service instance is connected to an Oracle Database Cloud Service database deployment. See Items that Are Not Backed Up.
Keep Forever	Whether to keep the backup indefinitely: • **Yes**—Keep the backup indefinitely or until it is deleted manually. • **No**—Delete the backup at the end of the default retention period for the service instance's backups. **Note:** If you selected the **Include Database** option and also enabled the Oracle Real Application Clusters (RAC) option on your database deployment, then the database backups will not be kept forever.
Notes	Up to 255 characters of free-form text to provide additional information about the backup. This text is displayed in the Notes field for the backup in the list of available backups. Provide information to enable an administrator to determine when to restore from the backup, for example, why the backup was created, or the state of the service instance at the time of the backup.

4. Click **Back Up**.

The Backup page is updated to show that the backup is in progress.

While the backup is in progress, you cannot start any other management operation on the service instance.

When the backup is complete, it is added to the list of available backups on the Backup page.

Deleting a Backup on OPCM

You can 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.

As an alternative to deleting a backup, you can move it from block storage to your NFS backup location. See Downloading or Archiving a Backup for a Service Instance in *REST API for Oracle Java Cloud Service*.

Note:

You can't delete a full backup with linked incremental backups without deleting the incremental backups. If the backup you're deleting has linked incremental backups, Oracle Java Cloud Service asks if you want to delete both the full backup and its linked incremental backups. For more information, see Contents of a Backup.

To delete a backup:

1. Navigate to the Backup page.
   a. Access the Oracle Java Cloud Service console.
   b. Click the name of the service instance for which you want to delete a backup.

      The Oracle Java Cloud Service Instance page is displayed with the **Overview** tile in focus, displaying detailed information about the service instance.
   c. Click the **Administration** tile.

      The Oracle Java Cloud Service Instance page is refreshed with the **Administration** tile in focus.
   d. Click the **Backup** tab.

      The Backup page is displayed.
2. From the  menu for the backup in the list of available backups, choose **Delete**.
3. When prompted, confirm that you want to delete the backup.

   If the backup you're deleting has linked incremental backups, Oracle Java Cloud Service asks if you want to delete both the full backup and its linked incremental backups.

Disabling or Enabling Backups for an Oracle Java Cloud Service Instance on OPCM

Disable both automated and on-demand backups for an Oracle Java Cloud Service instance.

Backups in Oracle Java Cloud Service are enabled by default on a new service instance but you can disable and enable this feature as required. After disabling backups for a service instance:

- Scheduled backups will not run.
- Automated backups will not run prior to patching or scaling operations.
- You cannot take an on-demand backup.

Disabling backups for a service instance does not affect any backups that were taken previously.

1. Navigate to the Backup page.

a. Access the Oracle Java Cloud Service console.

b. Click the name of the service instance for which you want to disable backups.

   The Instance Overview page is displayed.

c. Click the **Administration** tile.

d. Click the **Backup** tab if not already displayed.

2. Click **Disable Backups**.
3. When prompted for confirmation click **Disable Backups**.
4. To enable backups again at a later time, return to the Backup page and click **Enable Backups**.

Restoring an Oracle Java Cloud Service Instance from a Backup on OPCM

You can restore an Oracle Java Cloud Service instance from a backup to undo configuration changes you don't want and to return the service instance to a particular state. The instance's configuration data is always restored, but you can also choose whether to restore the Oracle WebLogic Server and the JDK software to its current official patch set update (PSU) level or to leave it unchanged.

You may choose to restore your service instance to a previous PSU level in order to undo a change to the software that you don't want, for example:

- Some files in the software installation were accidentally deleted.
- A patch from a source other than Oracle Java Cloud Service was installed and the software no longer works as required.

The software is restored from an image maintained internally to Oracle Cloud. If you are restoring from an old backup that contains a copy of the binary files, the copy in the backup is **not** restored. If you backed up software that includes patches from a source other than Oracle Java Cloud Service, those patches are **not** included in the restored software.

In addition, you **cannot** use Oracle Java Cloud Service to restore the database. To restore the database, you must use Oracle Database Cloud Service to restore from the associated database backup as identified by its RMAN tag or timestamp. For instructions, see these topics in *Using Oracle Database Cloud Service*:

- Restoring from a Specific Backup
- Restoring to a Specific Point in Time

If the hosts in the backup and the service instance don't match, how Oracle Java Cloud Service handles the mismatch depends on the cause of the mismatch. For details, see What Happens when a Service Instance Is Restored.

Before restoring a service instance, Oracle recommends that you disable any load balancer for the service instance. See Disabling or Enabling the Load Balancer for an Oracle Java Cloud Service Instance.

To restore an Oracle Java Cloud Service instance from a backup:

1. Navigate to the Backup page.

2. From the menu for the backup in the list of available backups, choose **Restore**.

   You can restore a service instance from an incremental backup without the need to restore the full backup to which the incremental backup is linked. Oracle Java Cloud Service restores all the data from the linked full backup required for a complete restoration of the service instance.

   A dialog box in which to set options for restoring from the backup opens.

3. In the dialog box, set the following options and click **Restore**.

   You don't need to select configuration data because Oracle Java Cloud Service always restores configuration data from a backup.

Option	Description
Include Database	If present, this option reminds you that the selected backup has an associated Oracle Database Cloud Service backup. If you select this option, Oracle Java Cloud Service asks you to confirm that you have restored the database before continuing.
Restore binary files	Select this option to restore the Oracle WebLogic Server and the JDK software to the versions installed at the time the backup was taken.
Notes	Enter up to 255 characters of free-form text to provide additional information about the restoration, for example, why you are restoring the service instance.

   The Backup page is updated to show that the restoration is in progress. While the restoration is in progress, you cannot start any other management operation on the service instance.

   When the restoration is complete, it is added to the restoration history in the Backup page.

If you restore a service instance from a backup whose topology does not match the current service instance, you may need to modify the restored service instance to return it to the state you require. See Returning an Oracle Java Cloud Service Instance to Service After Restoration from a Backup on OPCM.

If you previously disabled the load balancer for the service instance, enable it again. See Disabling or Enabling the Load Balancer for an Oracle Java Cloud Service Instance.

Returning an Oracle Java Cloud Service Instance to Service After Restoration from a Backup on OPCM

After restoring an Oracle Java Cloud Service instance from a backup, you must perform additional steps to return the service instance to service. You may also need to modify the service instance to return it to the state you require.

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. For details, see What Happens when a Service Instance Is Restored.

You must also remove a restored service instance's transaction logs.

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 contained any hosts that are not in the service instance, modify the service instance as follows:

   a. Use Oracle WebLogic Server to remove the managed servers on these hosts from the administration server configuration.

   b. If you require your service instance to contain the number of nodes in the backup, scale out the service instance.

      For more information, see Scaling Out an Oracle Java Cloud Service Instance.

      When Oracle Coherence is enabled for a service instance: See also Scaling the Coherence Data Tier of an Oracle Java Cloud Service Instance.

4. If you scaled in the service instance to delete any nodes and you require your service instance to contain the number of nodes it contained before you restored it, scale out the service instance.

   For more information, see Scaling Out an Oracle Java Cloud Service Instance.

   When Oracle Coherence is enabled for a service instance: See also Scaling the Coherence Data Tier of an Oracle Java Cloud Service Instance.

5. Remove the service instance's transaction records.

   Oracle Java Cloud Service does not provide any tools for removing a service instance's transaction records. Instead, use Oracle WebLogic Server for this purpose.

   For more information, see:

   - Accessing VM Through Secure Shell (SSH)
   - Relevant documentation for the release of Oracle WebLogic Server that your service instance is running:
     - For release 12*c* (12.2.1), see How to Remove Transaction Records in *Developing JTA Applications for Oracle WebLogic Server 12c (12.2.1)*.
     - For release 12*c* (12.1.3), see How to Remove Transaction Records in *Developing JTA Applications for Oracle WebLogic Server 12c (12.1.3)*.
     - For release 11*g* (11.1.1.7), see How to Remove Transaction Records in *Developing JTA Applications for Oracle WebLogic Server 11g (11.1.1.7)*

Exploring 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:

- Configuring Automated Backups for an Oracle Java Cloud Service Instance on OPCM
- Initiating an On-Demand Backup of an Oracle Java Cloud Service Instance on OPCM
- Deleting a Backup on OPCM
- Restoring an Oracle Java Cloud Service Instance from a Backup on OPCM

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	Follow this link to return to the Oracle Java Cloud Service Console.

Element	Description
 (adjacent to the service instance name)	**Menu** icon provides the following options: • **Open WebLogic Server Console**—Open the WebLogic 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 the load balancer has been configured for the service instance. See Configuring a Load Balancer for an Oracle Java Cloud Service Instance. • **Start**—Start the VMs for the Administration Server, Managed Servers, load balancer, and Managed Servers on the Coherence data tier (if provisioned). • **Stop**—Stop the VMs for the Administration Server, Managed Servers, load balancer, and Managed Servers on the Coherence data tier (if provisioned). • **Scale Out**—Scales the Managed Server node. • **Add Load Balancer**—Adds a load balancer to the service instance. If you did not specify a load balancer when you created the service instance, you can add it now. This can also be a second load balancer. • **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. • **SSH Access**—Add public SSH keys to the VMs that make up this service instance. See Managing SSH Access for an Oracle Java Cloud Service Instance. • **View Activity**— Opens the Service Activities page. For a selected service instance: – Stop Service – Start Service – Stop Server – Start Server – Restart Server – Scale Out – Scale In – Scale Up – Scale Down
WebLogic Server Version	Version of Oracle WebLogic Server configured for the Oracle Java Cloud Service instance: 12.2.1.2, 12.1.3.0, or 11.1.1.7.
Description	Description of the Oracle Java Cloud Service instance.
Cloud Storage Volume Used	The total amount of space, in MBytes or GBytes, that backups are occupying in the NFS remote backup location configured for this service instance.

Element	Description
Backup Volume Used	The total amount of space, in MBytes or GBytes, 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 Percent 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 Configuring Automated Backups for an Oracle Java Cloud Service Instance on OPCM.
Full Backups	Indicates the schedule for running full backups. For information about configuring the full backup schedule, see Configuring Automated Backups for an Oracle Java Cloud Service Instance on OPCM.
Last Successful 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.
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.
Backup Now	Click this button to create an on-demand backup of the Oracle Java Cloud Service instance. See Initiating an On-Demand Backup of an Oracle Java Cloud Service Instance on OPCM.
Configure Backups	Click this button to update the backup schedule and where backups are stored. See Configuring Automated Backups for an Oracle Java Cloud Service Instance on OPCM.
Disable Backups or **Enable Backups**	Click this toggle button to disable/enable automated and on-demand backups. See Disabling or Enabling Backups for an Oracle Java Cloud Service Instance on OPCM.
Available Backups (Last 7 Days)	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.

Element	Description
	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.
	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 was created, 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.

Element	Description
	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 Where Backups Are Stored. 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 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.
	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. For more information, see Contents of a 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 How Backups Are Initiated.
Available Until	The date and time until which the backup will be retained.
Location	The location where the backup is stored: • Oracle Storage Cloud Service—The backup is stored in the NFS remote file system for this service instance. • Both Disk and Oracle Storage Cloud Service—The backup is stored in the NFS remote file system for this service instance. A local copy of the backup is stored on the block storage of the virtual machine where the Administration Server is running. For more information, see Where Backups Are Stored.

Element	Description
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** icon provides the following options: • **Restore**—Restore the Oracle Java Cloud Service instance from the backup. See Restoring an Oracle Java Cloud Service Instance from a Backup on OPCM. • **Delete**—Delete the backup. See Deleting a Backup on OPCM.
Restoration 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.
	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.

Element	Description
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.
Contents	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.

7

Patching an Oracle Java Cloud Service Instance on Oracle Public Cloud Machine

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:

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.	Viewing Details of Approved Patches
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.	Performing Prechecks to Help Ensure Patching Success

Task	Description	More Information
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 an Oracle Java Cloud Service Instance Applying a Patch to an Oracle Java Cloud Service Instance About Patching an Oracle Java Cloud Service—Coherence Instance
Roll back a patch	Initiate a roll back operation to return a service to its previous patch level.	Rolling Back a Patching Operation

About Patching an Oracle Java Cloud Service Instance

You can quickly and easily apply patches to an Oracle Java Cloud Service instance, without any command-line intervention.

Once you navigate to the Patching page on the Oracle Java Cloud Service user interface, applying a patch to service instance is a one-click operation. See Exploring the Patching Page.

What kinds of patches are available

You can patch the Oracle WebLogic Servers, JDK, or load balancer of an Oracle Java Cloud Service instance. A WebLogic Server patch is called a Patch Set Update (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: from the Patching page. See Applying a Patch to an Oracle Java Cloud Service Instance.

About Operating System (OS) Patching

Oracle 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 the Oracle Linux Public yum server: `http://public-yum.oracle.com`.

When to Apply Patches

New approved patches are available and displayed on the Patching page roughly on a quarterly basis. Apply the most recent patches promptly. Delaying patches could cause your service to be unsupported for future patching and upgrade.

What Happens When Patching Starts

As patching starts, the patching operation first performs internal prechecks. If the prechecks fail, the patching operation will fail and leave the service instance untouched.

The prechecks check for the following:

- Disk space shortage
- Missing database connectivity
- Servers not running
- Storage access failure

Next, the Node Manager, Administration Server, and Managed Servers are all restarted to ensure that they can be restarted 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 perform prechecks without attempting to patch, and first remedy any problems found. For more information, see Performing Patching Prechecks and Addressing Patching Issues.

An automatic backup to use for rollback and restoration 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 to the Service During Patching

The WebLogic Server and JDK patching operations are rolling operations, so the service functions with very little interruption during the patch process. The patching operation shuts down one VM at a time and applies the patch to the server or servers on the VM. 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 VM at a time until all servers are patched.

For example, if you have a two-node cluster, each node exists on a separate machine, and one node keeps running while the other is being patched.

If a load balancer is provisioned, when you patch the load balancer, the Managed Servers and the Administration Server remain running. No requests will be routed to these servers during load balancer patching. The load balancer is stopped while patching is in progress.

Patching restarts VMs. For information about what happens when a VM is restarted, see About Starting and Stopping an Oracle Java Cloud Service Instance and Individual VMs.

What Happens When Patching is Not Fully Successful or Fails

Patching operations can be partially successful. Patching could succeed for the Administration Server and one or more Managed Servers, but the patching operation might not be able to start the remaining Managed Servers. In this case, you can start the Managed Servers manually. For more information, see Using the WebLogic Server Administration Console to Start Managed Servers.

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 Oracle Java Cloud 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.

What Happens if Coherence is Enabled

When Oracle Coherence is enabled for a service instance: Applying a patch will do 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 may choose to override the default behavior. See About Patching an Oracle Java Cloud Service—Coherence Instance if you need more information.

Viewing Details of Approved Patches on OPCM

You can use the Patching page to view the information about the available patches and service patch history.

Patch Set Update (PSU), Java Development Kit (JDK), and load balancer patches are available roughly quarterly. PSUs are cumulative. Each PSU contains all the previous PSUs bug fixes in addition to any bug fixes released after the previous PSU.

Check the Patching page quarterly to keep informed about available patches.

Note:

Oracle recommends only applying approved patches displayed on the Patching page. If you apply an unapproved or one-off patch, the patching operation overrides the patch unless the patch is included in the PSU.

To view details of approved patches:

1. Navigate to the Patching page.
2. Check the information in the Available Patches section on the Patching page.
3. Check the Administration tile to see the number of available patches, and the date and time of the previous backup.

Performing Patching Prechecks and Addressing 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

- Performing Patching Prechecks
- Addressing Patching Precheck Issues

Performing Patching Prechecks

Patching performs prechecks before applying a patch, but you can perform an optional precheck operation before patching to identify and remedy issues first.

Prechecks occur automatically once you make a selection from the menu, then produce messages letting you know the results.

Prechecks check for the following conditions:

- Disk space shortage
- Missing database connectivity
- Servers not running
- Storage access failure

Prechecks do not check whether another administration task (backup, restoration, or scaling) is in progress, which prevents patching.

If prechecks pass or you address all the issues identified by prechecks, you can run a patching operation that will not encounter these issues.

You can only request prechecks for an approved patch that is listed on the patching page.

1. Click on the  menu icon to the right of the patch you want to check.

   A menu containing **Precheck** and **Patch** is displayed.

2. Select **Precheck**.

   A message is displayed beneath the patch name: `Prechecks in progress`.

3. Click the  Refresh icon to check whether prechecks are done.

   When prechecks are done, a link is displayed: **Precheck results**.

   - If prechecks pass, a green check mark icon is displayed.
   - If prechecks fail, a red exclamation mark icon is displayed.

4. Click the **Precheck results** link.

   The Patch Precheck Results dialog is displayed.

   - If the prechecks succeed, the following message is displayed in the dialog: `Precheck succeeded for patch` *`patch_name_and version`* `on` *`date_and_time`*`. No conflicts found.`

     Conflicts arise when a non-approved patch was applied using `opatch`, for example. Only approved patches available from the Oracle Java Cloud Service patching page should be applied.

   - If the prechecks fail, the following message is displayed in the dialog: `Precheck failed for patch`*`patch_name_and_version`* `on` *`data_and_time`*.

     Error messages are displayed in the dialog as well. If there are precheck failures, check the error messages to learn about the problems that you need to address.

The **Precheck results** link remains after prechecks complete.

Addressing Patching Precheck Issues

Running prechecks prior to patching lets you know of conditions that would prevent successful patching.

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.

Disk Space Shortage

If patching would fail because of insufficient disk space, you see the following error message:

```
Not enough space on vm. Dir size: 457 Free space: 350
```

If there is not enough disk space, try freeing up disk space by deleting unwanted backups, for example. For more information, see Deleting a Backup.

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.us.oracle.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. See Stopping and Starting an Oracle Java Cloud Service Instance and Individual VMs.

Applying a Patch to an Oracle Java Cloud Service Instance on OPCM

You can use the Patching page to apply a patch to the Oracle WebLogic servers, JDK, or load balancer of an Oracle Java Cloud Service instance. You can also apply a patch to the internal scripts that perform tooling operations such a backup and scaling.

You can not apply a patch to a service instance if you previously disabled backups on the service instance. See Disabling or Enabling Backups for an Oracle Java Cloud Service Instance on OPCM.

To apply a patch:

1. Navigate to the Patching page.
2. In the Available Patches section, click the  menu icon to the right of the patch you want to apply and select **Patch**.
3. In the Patch Service dialog, type notes pertaining to the patch.

   You must enter a note. If you do not, an error message will be displayed and you will not be able to patch.
4. (*Only when Oracle Coherence is enabled*) In the Patch Service dialog, select the **Confirm** checkbox if you want to override the default behavior of always reaching a `StatusHA` state of `NODE-SAFE` before stopping a node to apply the patch.

   - When you select **Confirm**, this means you agree to the possibility of data loss by not requiring `StatusHA` to reach `NODE-SAFE` before stopping a server to apply the patch.
   - When the **Confirm** checkbox is not selected, note that patching will fail if `NODE-SAFE` cannot be reached.
   - When there's only one Managed Server on the Coherence data tier, the **Confirm** checkbox is selected and disabled by default.
5. In the Patch Service dialog, click **Patch**.

The patching operation begins, and the patches are applied in a rolling fashion, having minimal impact on the functioning of the service.

- A message stating that a patch job has been submitted for the service is displayed in the upper left corner of the console.
- The Administration tile displays a message about the version of patch being applied.

- The menu is inactive, as are the menus for other available patches.
- A full backup of the service created during the patching operation is displayed on the Backup page. The Administration tile shows the time of the backup that occurred before the patching operation started.

When the patching operation completes, the information about the patch appears in the Patch History section of the Patching page. A **Roll Back** button is displayed.

Rolling Back a Patching Operation on OPCM

You can use the Patching page to view the results of patching operations and to roll back patches.

You can roll back a patch if you find that a patch is incompatible with applications deployed on your service, or for any other possible reason.

The rollback operation shuts down the Administration Server and Managed Servers while the patch is rolled back, so the service is temporarily unavailable.

Patches for the load balancer (Oracle Traffic Director) cannot be rolled back.

The patching history is displayed on the 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.

Each patch, whether failed or successful, has a backup, which also appears on the Backup page. Rolling back a patch restores the binaries to the specific version recorded at the time of the backup, and leaves the domain and user data intact. You can restore the domain configuration after rollback by using the Backup page.

Video

To roll back a patch:

1. Navigate to the Patching page.
2. In the Patch History section, click the **Roll Back** button to the right of the patch you want to roll back.
3. On the Roll Back Service dialog, click **Roll Back**.

   A message is displayed, stating that a roll back job has been submitted.

   The label on the **Roll Back** button changes to **Rolling Back...** and the button is grayed-out.

   When the operation is complete, the patch entry in the Patch History section is grayed out, preventing you from trying to roll back the patch again.

   The service returns to its previous patch state.

Exploring 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

You can perform these tasks from the Patching page:

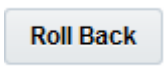
- View a list of available patches. See Viewing Details of Approved Patches.
- Perform patching prechecks and address precheck issues. See Performing Patching Prechecks and Addressing Patching Issues.
- Apply a patch. See Applying a Patch to an Oracle Java Cloud Service Instance.
- Check the patch history.
- Roll back a service to its previous patch level. See Rolling Back a Patching Operation.

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	Follow this link to return to the Oracle Java Cloud Service Console.

Element	Description
 (adjacent to the service instance name)	**Menu** icon provides the following options: • **Open WebLogic Server Console**—Open the WebLogic 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 the load balancer has been configured for the service instance. See Configuring a Load Balancer for an Oracle Java Cloud Service Instance. • **Start**—Start the VMs for the Administration Server, Managed Servers, load balancer, and Managed Servers on the Coherence data tier (if provisioned). • **Stop**—Stop the VMs for the Administration Server, Managed Servers, load balancer, and Managed Servers on the Coherence data tier (if provisioned). • **Scale Out**—Scales the Managed Server node. • **Add Load Balancer**—Adds a load balancer to the service instance. If you did not specify a load balancer when you created the service instance, you can add it now. This can also be a second load balancer. • **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. • **SSH Access**—Add public SSH keys to the VMs that make up this service instance. See Managing SSH Access for an Oracle Java Cloud Service Instance. • **View Activity**— Opens the Service Activities page. For a selected service instance: – Stop Service – Start Service – Stop Server – Start Server – Restart Server – Scale Out – Scale In – Scale Up – Scale Down
WebLogic Server Version	Version of Oracle WebLogic Server configured for the Oracle Java Cloud Service instance: 12.2.1.2, 12.1.3.0, or 11.1.1.7.
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.
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.
	Initiates an operation to roll back the service to its patch level prior to applying the patch.

8

Securing an Oracle Java Cloud Service Instance on Oracle Public Cloud Machine

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

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. If this default security configuration 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 Management instead of the embedded LDAP.

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 SSL 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:

- Understanding Users in Oracle Java Cloud Service
- Understanding Authentication Options
- Understanding the Default Access Ports

To learn more about the Java EE and WebLogic Server security architecture see:

- *Understanding Security for Oracle WebLogic Server* (12.2.1)
- *Understanding Security for Oracle WebLogic Server* (12.1.3)
- *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)
- Securing Oracle Coherence in Oracle WebLogic Server (12.1.3)

Configuring Network Security on OPCM

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're able to 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

Understanding the Default Access Ports

To use the resources available within your Oracle Java Cloud Service instances, access them through the default ports.

Refer to the diagram in About the Deployment Topology of Virtual Machines for an illustration of the default port allocation in an Oracle Java Cloud Service deployment topology.

Ports Available from Outside the Oracle Cloud Network

Resource	Protocol	Default Port
Oracle WebLogic Server Administration Console	HTTPS	7002*
Oracle Fusion Middleware Control	HTTPS	7002*
Integration with WebLogic Scripting Tool (WLST), Integrated Development Environment (IDE) or similar software	T3S	7002*

Resource	Protocol	Default Port
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
All VMs	SSH	22

** Note: 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 Oracle Java Cloud Service instance contains only one managed server, the server ports are 443 for HTTPS and disabled by default for HTTP.
- If the service instance was created by using the **REST API**, 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 Oracle Java Cloud 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 Oracle Java Cloud Service instance contains only one managed server, the managed server ports are set to 80 and 443 respectively. You can reconfigure these ports.

Ports Available from Within the Oracle Cloud Network

Resource	Protocol	Default Port
Oracle WebLogic Server Administration Console	HTTP	7001

Resource	Protocol	Default Port
Oracle Fusion Middleware Control	HTTP	7001
Managed Server	HTTP HTTPS	8001 8002
Database	SQL Net	1521

Deployment Port

The WebLogic Server administration 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 VM. Refer to Creating an SSH Tunnel for a secure method of making this port available to programs that are not running on the administration node.

Enabling HTTP Access to an Oracle Java Cloud Service Instance

If you create an Oracle Java Cloud Service instance by using the web console rather than the REST API, 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.

These instructions assume a load balancer has been enabled in your service instance. If there is no load balancer, you must instead create a 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)
- Configuring Network Resources (12.1.3)
- 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 VM automatically redirects incoming traffic on port 80 to port 8080.

Tasks:

- Enabling the HTTP Port on the Load Balancer

Enabling the HTTP Port on the Load Balancer

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.

1. Navigate to the Services page of 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. Click **OK**.
9. 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**.

The next task is to create an access rule for the port on the load balancer.

Managing SSH Access for an Oracle Java Cloud Service Instance

Use the SSH Access page to view and manage SSH keys for Oracle Java Cloud Service instances in your identity domain.

For information on creating SSH keys see Generating a Secure Shell (SSH) Public/Private Key Pair.

If the SSH private key that you use to access a service instance becomes lost or corrupted, you can add a new public key to the service instance. You may also need to add a new public key to a service instance in order to comply with your organization's security policies or regulations.

1. From the Oracle Java Cloud Service console click the **SSH Access** tab.
2. Locate your service instance and click **Add New Key**. See Managing SSH Access for Service Instances in an Identity Domain in *Using the Platform Services Console*.

For more information about using SSH with Oracle Java Cloud Service, see Accessing a VM Through a Secure Shell (SSH).

For more information about the SSH Access page, see Exploring the SSH Access Page in *Using the Platform Services Console*.

Configuring SSL for an Oracle Java Cloud 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 the client browser and the load balancer in your Oracle Java Cloud Service instance in order to ensure that applications are accessed securely.

By default SSL is already enabled within the software components of an Oracle Java Cloud Service instance, including Oracle WebLogic Server and the load balancer. They 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 CA for the certificate is unknown and not trusted.

You can update the load balancer to use a different certificate. Before you begin, ensure that you have enabled the load balancer in your Oracle Java Cloud Service instance and registered your custom domain name, as described in Defining a Custom Domain Name for an Oracle Java Cloud Service Instance.

Tasks:

- Creating a Self-Signed SSL Certificate in the Load Balancer
- Importing a CA-Issued SSL Certificate to the Load Balancer
- Associating the SSL Certificate With the Load Balancer

Creating a Self-Signed SSL Certificate in the Load Balancer

For development Oracle Java Cloud Serviceenvironments, you can use either a CA-issued or a self-signed certificate. You can create a self-signed certificate using the Load Balancer Console.

To obtain and use a CA-issued certificate instead, see Importing a CA-Issued SSL Certificate to the Load Balancer.

1. Navigate to the Services page of 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**.

Importing a CA-Issued SSL Certificate to the Load Balancer

For production Oracle Java Cloud Service environments, it is recommended that you use a CA-issued SSL certificate. A CA-issued SSL certificate reduces the chances of experiencing a man-in-the-middle attack.

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 need to provide the following information:

- Your custom domain name.
- Public information associated with the domain confirming you as the owner.
- Email address associated with the custom domain for verification.

Create a Certificate Signing Request (CSR) by using the Load Balancer Console and submit the CSR to the CA vendor. After receiving the CA-issued certificate, import it into the Load Balancer configuration.

1. Navigate to the Services page of 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-----`.

   For example:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIC9jCCAd4CAQAwYDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAkNBMQwwCgYDVQQH
EwNTQ0ExDzANBgNVBAoTBk9yYWNsZTEPMA0GA1UECxMGT3JhY2xlMRQwEgYDVQQD
I+XY7ByYRma1XlM1cYoMUiKSnRHdllUZMRwYHu4AZvrEMIhKjB6YiC0F
-----END NEW CERTIFICATE REQUEST-----
```

The CSR includes the public key and other information that the CA vendor needs to verify the identity of the load balancer server.

8. Submit the CSR to your CA vendor to request a new CA-issued SSL certificate.

   For more information about submitting the CSR, refer to your CA vendor documentation.

   Your CA vendor uses the CSR information to validate the domain and provides you with a valid SSL certificate, typically via email.

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)

Associating the SSL Certificate With the Load Balancer

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. Navigate to the Services page of 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.

After modifying a listener's certificate you must also restart the load balancer node(s) in your service instance for the change to take effect. See Stopping, Starting, and Restarting Managed Server and Load Balancer VMs.

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)

Authenticating Users on OPCM

Oracle Java Cloud Service is comprised of multiple components, each with its own identity stores, authentication options and administrative tools.

Topics

Understanding Users in Oracle Java Cloud Service

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.

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 Administrator 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. To learn more about WebLogic security see the following:

- *Understanding Security for Oracle WebLogic Server* (12.2.1)
- *Understanding Security for Oracle WebLogic Server* (12.1.3)
- *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 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. For large production applications, Oracle recommends that you use a proper identity management system such as Oracle Identity Management 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.

Note: If your database is running Oracle Database 12*c*, 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)
- *Administering JDBC Data Sources for Oracle WebLogic Server* (12.1.3)
- *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 load balancer running Oracle Traffic Director. The load balancer distributes application traffic to the servers in the WebLogic Server domain. Traffic Director 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. See Configuring a Load Balancer for an Oracle Java Cloud Service Instance on OPCM.

VM OS Users

Each Oracle Java Cloud Service instance is associated with a Secure Shell (SSH) public key. Using the matching private key, you can SSH to the underlying virtual machines

(VMs) running WebLogic Server and the load balancer. SSH to a VM 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 Accessing a VM Through a Secure Shell (SSH).

Understanding Authentication Options

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.

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 while application users are found in a different LDAP.

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 an 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. Refer to one of these topics: • Managing the Embedded LDAP Server in *Administering Security for Oracle WebLogic Server (12.2.1)* • Managing the Embedded LDAP Server in *Administering Security for Oracle WebLogic Server (12.1.3)* • Managing the Embedded LDAP Server in *Securing Oracle WebLogic Server (11.1.1.7)*

Authentication Option	Description
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. Refer to one of these topics: • Configuring LDAP Authentication Providers in *Administering Security for Oracle WebLogic Server (12.2.1)* • Configuring LDAP Authentication Providers in *Administering Security for Oracle WebLogic Server (12.1.3)* • 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. Refer to one of these topics: • Configuring RDBMS Authentication Providers in *Administering Security for Oracle WebLogic Server (12.2.1)* • Configuring RDBMS Authentication Providers in *Administering Security for Oracle WebLogic Server (12.1.3)* • Configuring RDBMS Authentication Providers in *Securing Oracle WebLogic Server (11.1.1.7)*
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. Refer to these topics: • 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)* • Configuring Identity Assertion Providers and Configuring Single Sign-On with Web Browsers and HTTP Clients Using SAML in *Administering Security for Oracle WebLogic Server (12.1.3)* • 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)*

Managing Passwords for Oracle Java Cloud Service on OPCM

You may need to update the various credentials used to run a service instance, in order to meet 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, see Understanding Users in Oracle Java Cloud Service.

User	Updating the Password	Updating Dependencies
WebLogic Server Administrator	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) • Administration Console Online Help (12.1.3) • Administration Console Online Help (11.1.1.7)	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 VM 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: • Administration Console Online Help (12.2.1) • Administration Console Online Help (12.1.3) • Administration Console Online Help (11.1.1.7) For information on using SSH to access Oracle Java Cloud Service VMs, see Accessing a VM Through a Secure Shell (SSH).

User	Updating the Password	Updating Dependencies
Load Balancer Administrator	If you add a 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 load balancer at a later time, you have the option to provide different credentials. In either case use the Load Balancer Console to change this user's password. For service instances running Oracle Traffic Director 12*c*, 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 11*g*, see Securing Access to the Administration Server in *Oracle Traffic Director Administrator's Guide*.	None

User	Updating the Password	Updating Dependencies
Database User	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 an Oracle Database Cloud Service database deployment. You specify the database name and credentials for these data sources when you create the service instance. If you modify the password for one of the database users, the data sources in the WebLogic domain may fail to connect to the database. Use one of the standard WebLogic administrative interfaces to modify the connection properties of the existing data sources. See *Configuring JDBC Data Sources* in one of the following publications: • Administering JDBC Data Sources for Oracle WebLogic Server (12.2.1) • Administering JDBC Data Sources for Oracle WebLogic Server (12.1.3) • 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 Understanding Data Sources.	When you create a service instance, you select one database deployment to host the **Oracle Required Schema** and you provide appropriate database credentials. If you modify the password of this database user, you must perform an additional task. Use the WebLogic Scripting Tool (WLST) to execute the `modifyBootStrapCredential` command and then restart the Administration Server. `modifyBootStrapCredential(jpsConfigFile='/u01/data/domains/`*`DOMAIN_NAME`*`/config/fmwconfig/jps-config.xml',username='`*`SCHEMA_PREFIX`*`_OPSS', password='`*`NEW_PASSWORD`*`')` See Problems with Failure of a Running Service When the Schema User Password Expires

User	Updating the Password	Updating Dependencies
Application User	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) • Administration Console Online Help (12.1.3) • 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 or an LDAP server. 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 Understanding Authentication Options.	None

9

Administering the Load Balancer for an Oracle Java Cloud Service Instance on Oracle Public Cloud Machine

This section explains how to administer the load balancer for an Oracle Java Cloud Service instance.

Topics:

Overview of Load Balancer Administration Tasks for an Oracle Java Cloud Service Instance

You perform load balancer administration tasks for an Oracle Java Cloud Service instance as required.

Task	Description	More Information
Disable or enable the load balancer	Disable the load balancer to suspend an Oracle Java Cloud Service instance during maintenance, or enable the load balancer to resume a service instance after maintenance.	Disabling or Enabling the Load Balancer for an Oracle Java Cloud Service Instance

Task	Description	More Information
Add a load balancer	Add a load balancer to an Oracle Java Cloud Service instance that was created without a load balancer. Add up to two load balancer nodes.	Adding a Load Balancer to an Oracle Java Cloud Service Instance
Add a second load balancer	Add a second load balancer node to an Oracle Java Cloud Service instance that already has one load balancer node.	Adding a Second Load Balancer to an Oracle Java Cloud Service Instance on OPCM
Remove a load balancer	Delete an existing load balancer node from an Oracle Java Cloud Service instance.	Removing a Load Balancer from an Oracle Java Cloud Service Instance on OPCM
Configure the load balancer	Use the Oracle Traffic Director administration console to configure the Oracle Traffic Director software that is used as the load balancer for an Oracle Java Cloud Service instance.	Configuring a Load Balancer for an Oracle Java Cloud Service Instance
Configure a custom domain name	Configure a custom "vanity" domain name (such as `example.com`) in the load balancer.	Defining a Custom Domain Name for an Oracle Java Cloud Service Instance
Configure a custom application URL	Configure a custom "vanity" URL (such as `/apps/mystore`) in the load balancer.	Defining a Custom URL for an Application Deployed to an Oracle Java Cloud Service Instance
Configure SSL	Configure SSL between the client browser and the load balancer. Ensure applications are accessed securely using a custom SSL certificate.	Configuring SSL for an Oracle Java Cloud Service Instance

Disabling or Enabling the Load Balancer for an Oracle Java Cloud Service Instance on OPCM

You can disable the load balancer for an Oracle Java Cloud Service instance to suspend the service instance during maintenance, or enable the load balancer to resume a service instance after maintenance.

When enabled, the load balancer forwards the requests it receives from clients to the Oracle WebLogic Server managed servers in your service instance. When disabled, the load balancer stops forwarding such requests, but instead 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 service instance is being backed up.

To enable or disable the load balancer for a service instance:

1. From the Services page of the Oracle Java Cloud Service Console, click an existing service instance.

   The Overview page is displayed.

2. Click **Administration**.

3. Click **Load Balancer**.

4. Click the button that is available to change the status of the load balancer.

   - To disable a load balancer that is enabled, click **Disable Traffic**.
   - To enable a load balancer that is disabled, click **Enable Traffic**.

Adding a Load Balancer to an Oracle Java Cloud Service Instance on OPCM

You can add a load balancer to an Oracle Java Cloud Service instance after its creation.

For any Oracle Java Cloud Service instance that contains more than one Managed Server node, a load balancer provides these benefits:

- Manages the routing of requests across all Managed Servers.
- Enables you to configure the routing policy.
- Enables you to suspend a service instance temporarily to perform routine maintenance, as described in Suspending an Oracle Java Cloud Service Instance.

A service instance can include zero, one or two load balancer nodes (VMs). Each node is assigned a separate public IP address. A configuration with two active load balancer nodes provides these advantages:

- High availability if a load balancer becomes unavailable.
- Multiple load balancers for greater application throughput.

> **Note:** Oracle Java Cloud Service does not automatically fail over application requests to the other load balancer node if one becomes unavailable.

Oracle Traffic Director (OTD) is used as the load balancer for a service instance.

To add a load balancer to a service instance:

1. From the Services page of the Oracle Java Cloud Service Console, click an existing service instance.

   The Overview page is displayed.

2. Click  and select **Add Load Balancer**.

   The Add Load Balancer dialog box opens.

3. 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 is useful for smoothing distribution when Managed Servers get bogged down. 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 number of Oracle Compute Units (OCPUs) and amount of RAM memory that you want to allocate to the VM for the load balancer. The larger the compute shape, the greater the processing power. For more information about the defined compute shapes, contact your Oracle Cloud Administrator. Valid compute shapes 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 Additional shapes are available for more memory-intensive applications: • OC1M: 1 OCPU and 15 GB memory • OC2M: 2 OCPUs 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 You can also change the compute shape after creating the Oracle Java Cloud Service instance. See Scaling an Oracle Java Cloud Service Node on OPCM

Option	Description
Add Another Active OTD Node	Choose whether or not to add a second load balancer node to this service instance.
User Name	Enter your choice of user name for the Oracle Traffic Director administrator. 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 (?) **Note:** You can change the user name and password through the Load Balancer Console after you have added the load balancer.
Password	Define the password for the Oracle Traffic Director administrator. The password must meet these requirements: • 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.
Confirm Password	Re-enter the password for the Oracle Traffic Director administrator.

4. Click **Add Load Balancer**.

The Overview page is updated to show that the load balancer is being added.

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.

After the load balancer is added, information about the load balancer is also available on the Load Balancer tab on the Administration page.

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"*.

Adding a Second Load Balancer to an Oracle Java Cloud Service Instance on OPCM

If your Oracle Java Cloud Service instance has an existing load balancer node you can add a second active load balancer node to the same instance.

An Oracle Java Cloud Service instance can include zero, one or two load balancer nodes (VMs). Each node is assigned a separate public IP address. A configuration with two active load balancer nodes provides these advantages:

- High availability if a load balancer becomes unavailable.
- Multiple load balancers for greater application throughput.

However, in this configuration the client is responsible for utilizing both load balancer nodes, and for failing over to another node should one become unavailable. Oracle Java Cloud Service does not automatically fail over application requests to the other load balancer node.

Oracle Traffic Director (OTD) is used as the load balancer for an Oracle Java Cloud Service instance. The Failover Group feature of OTD is not supported in this configuration.

To add a second OTD node to a service instance:

1. From the Services page of the Oracle Java Cloud Service Console, click an existing service instance.

   The Overview page is displayed.

2. Expand the Load Balancer section of this page.

3. Click  next to the first load balancer node, and select **Add Active OTD Node**.

   The Add Active OTD Node dialog box is displayed.

4. Click **Add Active OTD Node**.

The Overview page is updated to show that the load balancer is being added.

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.

After the load balancer is added, information about the load balancer is also available on the Load Balancer tab on the Administration page.

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"*.

Removing a Load Balancer from an Oracle Java Cloud Service Instance on OPCM

You can remove an existing load balancer node from anOracle Java Cloud Service instance.

> **Note:** An Oracle Java Cloud Service instance can include zero, one or two load balancer nodes. You cannot remove the first load balancer node after adding it to a service instance, but you can remove the second load balancer node.

To remove a load balancer:

1. From the Services page of the Oracle Java Cloud Service Console, click an existing service instance.

   The Overview page is displayed.

2. Expand the Load Balancer section of this page.

3. Click  next to the load balancer node you want to remove, and select **Delete**.

   The Delete OTD Node dialog box is displayed.

4. Click **Delete OTD Node** to confirm that you want to remove the load balancer VM.

The Overview page is updated to show that the load balancer is being removed.

While the load balancer is being removed, the service instance is in maintenance status and you cannot start any other management operation on the service instance.

Configuring a Load Balancer for an Oracle Java Cloud Service Instance on OPCM

Oracle Traffic Director is used as the load balancer for an Oracle Java Cloud Service instance.

Oracle Java Cloud Service does not provide any interfaces for configuring the load balancer for an Oracle Java Cloud Service instance. Instead, you use the Oracle Traffic Director administration console to configure the load balancer.

1. Navigate to the Oracle Java Cloud Service Console.

2. Click  for the desired service instance and select **Load Balancer Console**.

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:

   - Features of Oracle Traffic Director in *Administering Oracle Traffic Director*
   - Overview of Administration Tasks in *Administering Oracle Traffic Director*

   For service instances running Oracle Traffic Director 11*g* refer to:

- Features of Oracle Traffic Director in *Oracle Traffic Director Administrator's Guide*
- Overview of Administration Tasks in *Oracle Traffic Director Administrator's Guide*

For information about the topology of Oracle Traffic Director instances in an Oracle Java Cloud Service instance, see About the Deployment Topology of Virtual Machines.

About Load Balancer Virtual Machines

If a load balancer is enabled for an Oracle Java Cloud Service instance, the Oracle Traffic Director (OTD) administration server is hosted on one virtual machine (VM). A second OTD node can optionally be hosted on another VM.

The following table lists the file paths found on the load balancer VM for service instances running Oracle Traffic Director 12*c*:

Name	Path	Description
`JAVA_HOME`	`/u01/jdk`	Java installation
`ORACLE_HOME`	`/u01/app/oracle/middleware/otd`	Oracle Traffic Director installation
`DOMAIN_HOME`	`/u01/data/otd-instance/otd_domain`	Oracle WebLogic Server domain that is used to manage and monitor Oracle Traffic Director

The following table lists the file paths found on the load balancer VM for service instances running Oracle Traffic Director 11*g*:

Name	Path	Description
`JAVA_HOME`	`/u01/jdk`	Java installation
`ORACLE_HOME`	`/u01/app/oracle/middleware/otd`	Oracle Traffic Director installation
`INSTANCE_HOME`	`/u01/data/otd-instance`	Scripts and configuration data for this Oracle Traffic Director instance

Exploring the Load Balancer Page

You can use the Load Balancer page to view and change the status of the load balancer for an Oracle Java Cloud Service instance.

What You Can Do from the Load Balancer Page

You can perform these tasks from the Load Balancer page:

- Disabling or Enabling the Load Balancer for an Oracle Java Cloud Service Instance

What You See on the Load Balancer Page

The following table describes the key information shown on the Load Balancer page.

Element	Description
Oracle Java Cloud Service link	Follow this link to return to the Oracle Java Cloud Service Console.
 (adjacent to the service instance name)	**Menu** icon provides the following options: • **Open WebLogic Server Console**—Open the WebLogic 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 the load balancer has been configured for the service instance. See Configuring a Load Balancer for an Oracle Java Cloud Service Instance. • **Start**—Start the VMs for the Administration Server, Managed Servers, load balancer, and Managed Servers on the Coherence data tier (if provisioned). • **Stop**—Stop the VMs for the Administration Server, Managed Servers, load balancer, and Managed Servers on the Coherence data tier (if provisioned). • **Scale Out**—Scales the Managed Server node. • **Add Load Balancer**—Adds a load balancer to the service instance. If you did not specify a load balancer when you created the service instance, you can add it now. This can also be a second load balancer. • **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. • **SSH Access**—Add public SSH keys to the VMs that make up this service instance. See Managing SSH Access for an Oracle Java Cloud Service Instance. • **View Activity**— Opens the Service Activities page. For a selected service instance: – Stop Service – Start Service – Stop Server – Start Server – Restart Server – Scale Out – Scale In – Scale Up – Scale Down
	Click to refresh the page. The date and time the page was last refreshed is displayed adjacent to this button.
Disable Traffic	Click **Disable Traffic** to disable the load balancer. You see this button only if the load balancer is enabled.
Enable Traffic	Click **Enable Traffic** to enable the load balancer. You see this button only if the load balancer is disabled.
Version	The version of Oracle Traffic Director that the Oracle Java Cloud Service instance is running.

Element	Description
JDK Version	The version of the JDK that the Oracle Traffic Director is running.
Content URL	Resource URL for accessing the deployed applications using HTTP. If the sample application is deployed during provisioning of the service instance, the URL for the sample application is shown.
Load Balancer Status	One of the following statuses of the load balancer: • Submitting • Submitted • Preparing • Starting • Enabling • Enabled • Disabling • Disabled • Error • Deleting • Deleted • Not configured
Origin Servers	The host name and port number of each Oracle WebLogic Server managed server to which Oracle Traffic Director forwards the requests that it receives from clients, and from which it receives responses to client requests.
OTD Node Count	The number of Oracle Traffic Director (OTD) load balancer nodes in this Oracle Java Cloud Service instance.

10

Managing Compute Virtual Machines in Oracle Java Cloud Service on Oracle Public Cloud Machine

Oracle Compute virtual machines (VMs) are automatically configured for you when you create an Oracle Java Cloud Service instance.

Topics:

About the Oracle Compute Cloud Service Environment in Oracle Java Cloud Service

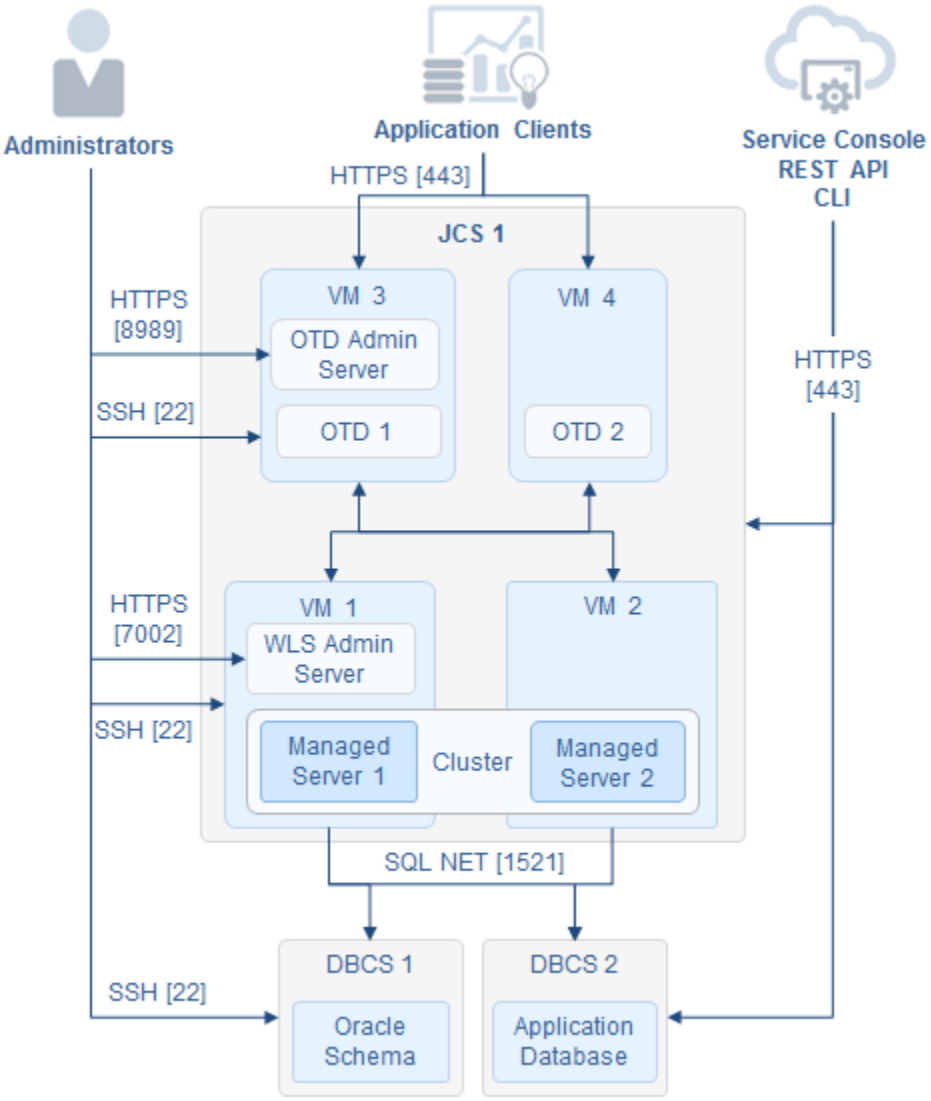
Oracle Java Cloud Service is built on top of infrastructure and functionality that are provided by Oracle Compute Cloud Service. The work involved in setting up and managing the compute virtual machines and storage resources for your Java EE applications is done for you.

Topics:

About the Deployment Topology of Virtual Machines

Using Oracle Java Cloud Service simplifies the work you've to do in order to provision and configure an Oracle WebLogic Server domain and cluster, Oracle Coherence as the caching and data tier, and Oracle Traffic Director as the software load balancer.

The following illustration shows an example of the Oracle Java Cloud Service virtual machine (VM) deployment topology that is set up and configured for you when you provision an Oracle Java Cloud Service instance with two Managed Servers and also enable two load balancers:



Note:

For information about the network protocols and default ports that can be used from within Oracle Cloud and from outside Oracle Cloud, see Understanding the Default Access Ports. Note that the HTTP port is disabled if you created the Oracle Java Cloud Service instance by using the service instance creation wizard available through the Oracle Java Cloud Service Console.

As shown in the illustration, an Oracle Java Cloud Service instance is a single Oracle WebLogic Server domain that consists of one WebLogic Administration Server and one WebLogic Server cluster of Managed Servers for hosting applications. The example in the topology illustration shows a cluster of two Managed Servers.

When Oracle Coherence is enabled for a service instance, there is a second WebLogic Server cluster of Managed Servers for storing Coherence data. Both WebLogic Server clusters are associated with the single Coherence cluster in the domain. For information about the VM deployment topology when Oracle Coherence is enabled

for a service instance in Oracle Java Cloud Service, see About Oracle Coherence and the Compute Environment in Oracle Java Cloud Service.

About the Virtual Machines

The Oracle Compute virtual machines in an Oracle Java Cloud Service environment are based on the Oracle Linux 6 (OEL6) 60-GB disk image. These virtual machines are highly available and the underlying infrastructure contains built-in capabilities to migrate an unhealthy virtual machine (VM) to a separate hardware cluster.

Each Oracle Java Cloud Service instance that you create can contain one or more Oracle Compute VM instances. The first VM always contains the WebLogic Administration Server and the first Managed Server. Each remaining Managed Server is hosted on its own VM. When the service instance is scaled out, each additional Managed Server is also on its own VM.

If a load balancer is enabled for a service instance, the Oracle Traffic Director (OTD) administration server is hosted on one VM. A second OTD node can optionally be hosted on another VM.

When using the wizard to provision an Oracle Java Cloud Service instance, you can create up to four Managed Servers in the cluster. You can also create up to two load balancers VMs. The following table summarizes the number of Managed Servers you can have in the WebLogic Server cluster, and the corresponding VMs:

VM Instance	1–Node Cluster	2–Node Cluster	4–Node Cluster
1st VM	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 VM		Contains Managed Server 2	Contains Managed Server 2
3rd VM			Contains Managed Server 3
4th VM			Contains Managed Server 4
5th VM	If present, the VM contains the OTD administration server	If present, the VM contains the OTD administration server	If present, the VM contains the OTD administration server
6th VM	If present, the VM contains a second OTD node	If present, the VM contains a second OTD node	If present, the VM contains a second OTD node

Note: By default a load balancer is not enabled for a service instance that has a single-node cluster in the WebLogic Server domain, so the Oracle Traffic Director VM 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 VM would be present.

When Oracle Coherence is enabled for a service instance, a VM on the Coherence data tier can have one or more storage-enabled Managed Servers. The number of Coherence VMs and the number of Managed Servers configured per VM are based on the capacity unit definition that was declared for the service instance during initial provisioning.

The following table summarizes the number of VMs on the application tier and Coherence data tier, and the corresponding Managed Servers contained in the VMs 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) that's created when you use a default capacity unit defined with three VMs and one Managed Server running on each VM:

VM Instance	Contains	WLS Cluster
1st VM	WebLogic Administration Server, Managed Server 1(storage-disabled)	Application Tier
2nd VM	Managed Server 2 (storage-disabled)	Application Tier
3rd VM	Managed Server 3_DG (storage-enabled)	Coherence Data Tier
4th VM	Managed Server 4_DG (storage-enabled)	Coherence Data Tier
5th VM	Managed Server 5_DG (storage-enabled)	Coherence Data Tier

Appropriate security rules are configured on the Oracle Java Cloud Service VMs to enable communication among the different VMs hosting the WebLogic managed servers, and also with the Oracle Traffic Director VM and the Oracle Database.

You have access to all VM instances, including the VM on which the WebLogic Administration Server is running. You can use a Secure Shell (SSH) client to log into a VM, as described in Accessing a VM Through a Secure Shell (SSH).

For information about all Oracle Compute VM objects that are created or configured in an Oracle Java Cloud Service instance, see About Compute Virtual Machine Objects in Oracle Java Cloud Service.

About the Disk Volumes

You have access to all Oracle Compute Cloud Service virtual machine instances created for Oracle Java Cloud Service, including the virtual machine on which the WebLogic Administration Server is running.

The following table lists the disk volumes that are attached to Oracle Java Cloud Service virtual machines and the mount points:

Disk Volume	Purpose	Mount Point
Boot/OS volume	The boot volume as provided by the machine image. Contains the OS binaries.	Local disk, no mount point
Backup volume	Contains a copy of backups up to seven days old.	`/u01/data/backup`

Disk Volume	Purpose	Mount Point
DOMAIN_HOME	Contains data for the domain corresponding to the Oracle Java Cloud Service instance.	/u01/data/domains
APPLICATION_HOME	Contains deployed applications and application configuration files.	/u01/data/domains
MW_HOME	Contains Oracle WebLogic Server binaries and OTD binaries.	/u01/app/oracle/middleware
JCS_RESERVED	Contains files required by Oracle Java Cloud Service, that is, any binaries and related metadata that are required by the Oracle Java Cloud Service management layer.	/u01/app/oracle/tools
JDK_HOME	Contains JDK binaries.	/u01/jdk

Note:

- All volumes under /u01, except DOMAIN_HOME and APPLICATION_HOME, should be treated as read-only volumes.
- The Backup volume is writable by the oracle user; the opc user has read-only access.
- The Boot/OS volume of a service instance is persistent. Content added to a persistent Boot/OS volume is retained if the service instance is restarted.

About Compute Virtual Machine Objects in Oracle Java Cloud Service

When you provision a service instance in Oracle Java Cloud Service, all the Oracle Compute virtual machines (VMs) required to support your service instance are provisioned and configured for you.

The Oracle Compute virtual machine (VM) objects that are created or configured in a service instance include:

- Block storage volumes: Oracle Compute Cloud Service provides persistent virtual disks for storing applications and data.
- Secure Shell (SSH) public keys: You provide an SSH public key when creating a service instance. The key is attached to VMs to enable secure access to your service instance VMs from remote hosts.
- IP reservations: Oracle Compute Cloud Service assigns public IP addresses to the WebLogic Server administration server VM and the Oracle Traffic Director administration server VM (if a load balancer is enabled on the service instance).

11

Using Oracle Coherence in Oracle Java Cloud Service on Oracle Public Cloud Machine

Enable Oracle Coherence in Oracle Java Cloud Service to use in-memory data grid and caching for Coherence applications that are deployed to Oracle WebLogic Server environments in the cloud.

Topics:

Workflow for Using Oracle Coherence in Oracle Java Cloud Service

You can enable Oracle Coherence for an Oracle Java Cloud Service instance to use in-memory data grid and caching for your Coherence applications deployed to the cloud.

> **Note:** For clarification purposes in this document and others, service instances provisioned in Oracle Java Cloud Service to use Oracle Coherence are referred to as Oracle Java Cloud Service—Coherence instances.

To start using Oracle Coherence in Oracle Java Cloud Service, use the following tasks as a guide:

Task	Description	More Information
Understand how capacity units are used in provisioning Oracle Java Cloud Service—Coherence instances	You must declare to use a default capacity unit or define a custom capacity unit when you create a service instance.	About Capacity Units for Oracle Java Cloud Service—Coherence Instances
Create an Oracle Java Cloud Service—Coherence instance	Be sure you have completed the prerequisite steps in Before You Begin with Oracle Java Cloud Service. When using the Oracle Java Cloud Service Console to create a service instance, there are certain options you must select and specific Coherence data tier details you must specify, so that an environment for using Oracle Coherence within Oracle Java Cloud Service is correctly provisioned and configured for you: • Select the service level Oracle Java Cloud Service or Oracle Java Cloud Service—Virtual Image. Note that if you enable Oracle Coherence for an Oracle Java Cloud Service—Virtual Image instance, you won't be able to use advanced tooling in the Oracle Java Cloud Service Console to scale, back up and restore, or patch your service instance. • Select the Oracle WebLogic Server 12*c* (12.1.3) or Oracle WebLogic Server 12*c* (12.2.1) software release • Select Enterprise Edition with Coherence • Select a default capacity unit or select Advanced to define your own custom capacity unit by specifying details such as the compute shape to use and the number of nodes to configure. Note that you can't change the capacity unit after the service instance has been provisioned.	About Oracle Java Cloud Service—Coherence Instances Creating an Oracle Java Cloud Service—Coherence Instance

Task	Description	More Information
Deploy Coherence applications to a service instance	Use the WebLogic Server Administration Console, the Fusion Middleware Control, WebLogic Scripting Tool commands, or an IDE to deploy and undeploy applications.	About Deploying Applications to an Oracle Java Cloud Service—Coherence Instance Deploying and Undeploying Applications for an Oracle Java Cloud Service Instance
Scale the Coherence data tier of a service instance	You can increase or decrease the total amount of cache capacity you require in preparation for an increased or reduced load on a service instance. A scaling operation will either add or remove storage-enabled nodes based on the capacity unit definition declared for the service instance during initial provisioning.	About Scaling an Oracle Java Cloud Service—Coherence Instance Scaling the Coherence Data Tier of an Oracle Java Cloud Service Instance
Remove a specific capacity unit	Perform a scale in of the Coherence data tier by deleting specific storage-enabled nodes as identified by a specific capacity unit.	Scaling the Coherence Data Tier by Using the Oracle Java Cloud Service Console
Back up a 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).	About Backing Up and Restoring an Oracle Java Cloud Service—Coherence Instance Backing Up and Restoring an Oracle Java Cloud Service Instance
Patch a service instance	Apply a patch or roll back a patch.	About Patching an Oracle Java Cloud Service—Coherence Instance Patching an Oracle Java Cloud Service—Coherence Instance Rolling Back a Patching Operation
Add a Coherence data tier	Add a data tier WebLogic Server cluster of storage-enabled servers to an existing service instance that doesn't already have Oracle Coherence enabled (REST API only).	Add a Coherence Data Tier Cluster in *REST API for Oracle Java Cloud Service*
Remove a Coherence data tier	Delete the entire data tier WebLogic Server cluster from a service instance (REST API only).	Delete a Coherence Data Tier Cluster in *REST API for Oracle Java Cloud Service*
Manage access to a service instance	Suspend a service instance (if load balancer is enabled). Stop or start a service instance. Note that when you stop or start a service instance, all the virtual machines 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.	Suspending an Oracle Java Cloud Service Instance About Stopping and Starting an Oracle Java Cloud Service Instance and Individual VMs Stopping and Starting an Oracle Java Cloud Service Instance

Task	Description	More Information
Remove a service instance	Delete an entire service instance.	About Deleting an Oracle Java Cloud Service—Coherence Instance Deleting an Oracle Java Cloud Service Instance
Use the REST API to create and manage Oracle Java Cloud Service—Coherence instances	The Oracle Java Cloud Service REST API provides an alternative to using the web-based Oracle Java Cloud Service Console to create and manage Oracle Java Cloud Service—Coherence instances. For example, you can use the REST API to enable Oracle Coherence for an Oracle Java Cloud Service instance that's already been provisioned and configured to run Oracle WebLogic Server 12*c* (12.1.3) or Oracle WebLogic Server 12*c* (12.2.1), and Enterprise Edition with Coherence.	About Using the REST API to Create and Manage Oracle Java Cloud Service—Coherence Instances

About Oracle Coherence in Oracle Java Cloud Service

Oracle Coherence is a fault-tolerant, in-memory data management solution for clustered applications and application servers, enabling Java EE applications to predictably scale by providing fast, reliable, and scalable access to frequently used data. You can enable Oracle Coherence for Oracle Java Cloud Service instances to use in-memory data grid and caching out-of-the-box for Coherence applications deployed to Oracle WebLogic Server environments in the cloud.

Topics:

- About Oracle Coherence in Oracle Java Cloud Service and Supported Software Releases
- About Using Oracle Coherence in Oracle Java Cloud Service
- About Oracle Coherence and the Compute Environment in Oracle Java Cloud Service

Note: For clarification purposes in this document and others, service instances provisioned in Oracle Java Cloud Service to use Oracle Coherence are referred to as Oracle Java Cloud Service—Coherence instances.

About Oracle Coherence in Oracle Java Cloud Service and Supported Software Releases

Use Oracle Java Cloud Service to create a service instance and enable Oracle Coherence for the service instance. Depending on the selected software release of the service instance, Oracle Java Cloud Service can also provision and manage the Oracle Coherence cache capacity running on the service instance.

You can enable Oracle Coherence on any service instance by selecting the **Enterprise Edition with Coherence** option. If the service instance is running Oracle WebLogic

Server 12*c* (12.1.3 or 12.2.1) you can also use Oracle Java Cloud Service to automatically provision, scale and manage your Oracle Coherence cache capacity. If the service instance is running Oracle WebLogic Server 11*g* you must manually configure and start Oracle Coherence after the service instance is created, by accessing the VMs that comprise your service instance.

When you use Oracle Java Cloud Service to set your initial cache capacity in a 12*c* service instance, the following infrastructure is added to the service instance:

- A Coherence cluster, the system-level `CoherenceClusterSystemResource`, is configured in the WebLogic Server (WLS) domain.
- A data tier WLS cluster of Managed Servers is provisioned and configured for storing data related to Coherence. The Coherence data tier cluster is created in the same WLS domain as the application tier WLS cluster that's used for running applications.
- Both the storage-enabled Coherence data tier cluster and the storage-disabled application tier cluster are associated with the Coherence cluster. Thus, all Managed Servers in an Oracle Java Cloud Service—Coherence instance are configured as Coherence cluster members and are referred to as Managed Coherence Servers. The Coherence data tier can be scaled independently of the application tier.

About Using Oracle Coherence in Oracle Java Cloud Service

By enabling Oracle Coherence for Oracle Java Cloud Service when you provision a cloud environment, you can quickly create and configure an Oracle WebLogic Server domain, set up the Coherence cluster, and set up the Coherence data tier and application tier WebLogic Server clusters without worrying about setting up any infrastructure or platform details yourself. You can also use advanced tooling for all major service lifecycle operations such as scaling, backup and restoration, and patching.

As long as you are a Tenant User in Oracle Java Cloud Service, you can perform all operations related to Oracle Java Cloud Service—Coherence instances. All tasks can be performed using the Oracle Java Cloud Service Console.

See Accessing Oracle Java Cloud Service for details about how to sign in to use the Oracle Java Cloud Service Console.

To start using Oracle Coherence in Oracle Java Cloud Service, refer to the tasks in Workflow for Using Oracle Coherence in Oracle Java Cloud Service as a guide.

To use the REST API to create and manage Oracle Java Cloud Service—Coherence instances, see *REST API for Oracle Java Cloud Service*.

About Oracle Coherence and the Compute Environment in Oracle Java Cloud Service

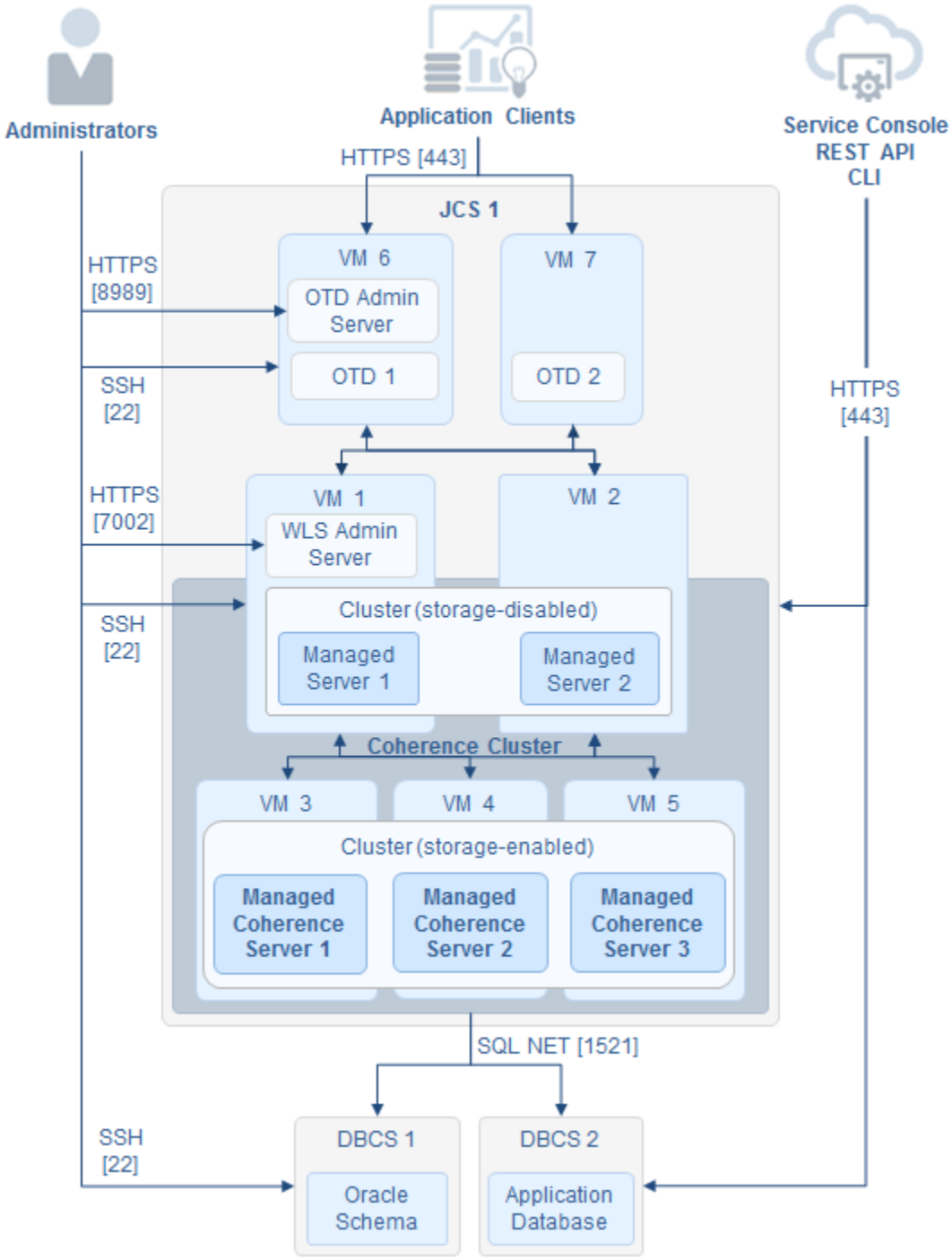
When Oracle Coherence is enabled for a service instance in Oracle Java Cloud Service, the Oracle Compute VM deployment topology is similar to the topology for a service instance that doesn't have Coherence enabled, with the addition of VMs and Managed Servers for a Coherence data tier.

When enabled for a service instance, Oracle Coherence is built on top of infrastructure and functionality created for Oracle Java Cloud Service:

- The Oracle Coherence Enterprise Edition software is installed and configured within the Oracle Java Cloud Service environment.

- All the underlying infrastructure work involved in setting up and managing the compute VMs, network settings, and storage resources to run applications is done for you.
- The Oracle WebLogic Server (WLS) domain, Coherence cluster, and WLS clusters are set up for you. All the Managed Servers you need are also created, configured, and started automatically for you when provisioning of the service instance is complete.

The following illustration shows the Oracle Java Cloud Service VM deployment topology that is set up and configured for you when you provision an Oracle Java Cloud Service—Coherence instance. The example uses a cluster of two Managed Servers for the application tier (storage-disabled), and the default SMALL capacity unit for the Coherence data tier (resulting in a storage-enabled cluster of three VMs and three Managed Servers):



Note: For information about the network protocols and default ports that can be used from within Oracle Cloud and from outside Oracle Cloud, see Understanding the Default Access Ports. Note that the HTTP port is disabled if you created the Oracle Java Cloud Service instance by using the service instance creation wizard available through the Oracle Java Cloud Service Console.

As shown in the illustration, an Oracle Java Cloud Service—Coherence instance is a single WLS domain that consists of the following resources:

- One WebLogic Administration Server
- One Coherence cluster (the system-level CoherenceClusterSystemResource). The default name is `DataGridConfig`.
- One application tier WLS cluster of storage-disabled Managed Servers for running application cache clients. EAR files, containing application WAR and Coherence GAR files, are deployed on the application tier. The example in the illustration shows two Managed Servers in the application tier cluster.
- One data tier WLS cluster of storage-enabled Managed Servers for running cache servers. This is the Coherence data tier. By default, the cluster name of the Coherence data tier is generated from the first eight characters of the service instance name using the following format: *`first8charsOfServiceInstanceName`*`_DGCluster`. Coherence GAR files are deployed on the Coherence data tier. The example in the illustration shows three Managed Servers in the Coherence data tier cluster.
- Both the WLS cluster for hosting applications and the WLS cluster for storing Coherence data are associated with the Coherence cluster. Thus, all Managed Servers are members of the Coherence cluster.

For information about the VMs and the Managed Servers contained in each VM, see About the Virtual Machines.

About Oracle Java Cloud Service—Coherence Instances

You can enable Oracle Coherence when you create a service instance in Oracle Java Cloud Service using the Oracle Java Cloud Service Console or the REST API. For clarification purposes in this document and others, service instances provisioned to use Oracle Coherence in Oracle Java Cloud Service are referred to as Oracle Java Cloud Service—Coherence instances.

When you provision a service instance using the Create Java Cloud Service wizard, there are certain options you must select and specific Coherence details you must specify, so that an environment for using Oracle Coherence in Oracle Java Cloud Service is correctly provisioned and configured for you.

> **Note:** You can enable Oracle Coherence on any service instance but only those running Oracle WebLogic Server 12*c* (12.1.3 or 12.2.1) provide tooling to provision and manage your cache capacity. If the service instance is running Oracle WebLogic Server 11*g* you must manually configure and start Oracle Coherence by directly accessing the VMs that comprise the service instance.

To enable Oracle Coherence for an Oracle Java Cloud Service instance that's already been provisioned and configured, you have to use the REST API. Note that you can add Oracle Coherence only to a service instance that's configured to run Oracle WebLogic Server 12*c* (12.1.3) or Oracle WebLogic Server 12*c* (12.2.1), and Enterprise Edition with Coherence. For information about using the REST API to enable Coherence for a service instance, see Add a Coherence Data Tier Cluster in *REST API for Oracle Java Cloud Service*.

Before creating an Oracle Java Cloud Service—Coherence instance, ensure you're familiar with capacity units. See About Capacity Units for Oracle Java Cloud Service—Coherence Instances for information about using default and custom capacity units.

Service Level and Software Choices

The following table describes the service level and software choices, and the selections you must make when using the wizard to create an Oracle Java Cloud Service—Coherence instance:

Choice	Select This	Description
Service Level	Oracle Java Cloud Service or Oracle Java Cloud Service—Virtual Image	If you select Oracle Java Cloud Service—Virtual Image, you won't be able to use advanced tooling in the Oracle Java Cloud Service Console for major service lifecycle operations such as scaling, backup and restoration, and patching. Use Oracle Java Cloud Service—Virtual Image if you're planning to create an environment for development purposes.
Software Release	Oracle WebLogic Server 12*c* (12.1.3) or Oracle WebLogic Server 12*c* (12.2.1)	To use all of the capabilities provided for Oracle Coherence in Oracle Java Cloud Service, you must run Oracle WebLogic Server 12*c* on the service instance.
Software Edition	Enterprise Edition with Coherence	To use Oracle Coherence in Oracle Java Cloud Service, you must run Enterprise Edition with Coherence.

Coherence Data Tier Details

When configuring details for the Coherence data tier, you can use a default capacity unit or define your own custom capacity unit. The following table describes the Coherence data tier choices and the selections you can make when using the wizard to create an Oracle Java Cloud Service—Coherence instance:

Choice	Select	Description
Capacity Unit	Select: • A default capacity unit definition that provides the best combination of processing power and cache capacity for your Coherence application and workflow requirements. • **Advanced** to define your own configuration for a custom capacity unit (instead of using a default capacity unit), to get the processing power and cache capacity your business needs.	A default capacity unit comprises one virtual machine (VM) or three virtual machines (VMs) with a predefined shape, running a number of JVMs or Managed Coherence Servers to provide a predefined amount of primary cache storage for Coherence. A shape defines the processing power of the virtual machines to provision, and the amount of memory that would be available for JVM heap. Coherence high availability is achieved when you select a default capacity unit that has three VMs. See About Default Capacity Units for Oracle Java Cloud Service—Coherence Instances for information about the different configurations provided by default capacity unit definitions. When defining a custom capacity unit: You must specify a number for VMs and JVMs, and a compute shape for the VMs. Note that after a service instance has been provisioned, you can't change the default capacity unit you used or the configuration of a custom capacity unit you defined.
Cache Size	A display-only field. The field is displayed when you select a default capacity unit or **Advanced** from the Capacity Unit dropdown list.	Whether you're using a default capacity unit or defining a custom capacity unit, each capacity unit provides an amount of cache, which is the primary cache storage to allocate for Coherence in the domain. The primary cache storage to allocate is based on the general rule of splitting the JVM heap size into thirds, using 1/3rd for primary cache storage, 1/3rd for backup storage, and 1/3rd for scratch space. See About Capacity Units for Oracle Java Cloud Service—Coherence Instances for more information about default and custom capacity units.

Choice	Select	Description
Shape	Select a VM compute shape that provides the best combination of processing power and RAM.	Advanced configuration only: A shape represents the number of Oracle Compute Units (OCPUs) and amount of RAM (system memory) available for a VM instance. All VMs in a capacity unit have the same shape. For information about the defined compute shapes, contact your Oracle Cloud administrator.
Nodes	Select the number of VMs to create per capacity unit.	Advanced configuration only: The number of VMs to create when a capacity unit is provisioned. Coherence high availability is achieved when you specify three VMs per custom unit. See About Custom Capacity Units and Advanced Configuration for Oracle Java Cloud Service—Coherence Instances for more information.
Managed Servers Per Node	Select the number of JVMs to configure per VM.	Advanced configuration only: The number of Managed Coherence Servers to configure per VM. Note that the maximum heap size that will be configured is 16 GB. See About Custom Capacity Units and Advanced Configuration for Oracle Java Cloud Service—Coherence Instances for more information.

About Capacity Units for Oracle Java Cloud Service—Coherence Instances

Oracle Java Cloud Service uses capacity units to provision Oracle WebLogic Server and Oracle Coherence environments that provide a preconfigured amount of Coherence cache capacity. A capacity unit is a group of virtual machines (VMs) running a number of JVMs (Managed Coherence Servers), and is meant to scale linearly in the context of your application. Capacity is defined as a fixed amount of primary cache storage to allocate for Coherence, based on the general rule of splitting the JVM heap size into thirds, using 1/3rd for primary cache storage, 1/3rd for backup storage, and 1/3rd for scratch space.

Topics:

When provisioning an Oracle Java Cloud Service—Coherence instance, a capacity unit must be declared for the service instance. A capacity unit is a well-defined unit of VMs and JVMs, preconfigured by properties that specify:

- The processing power of the VMs to provision on the Coherence data tier of the WebLogic Server domain
- The amount of primary cache storage to allocate for Coherence

When provisioning a service instance using the Create Java Cloud Service wizard, you can choose from a list of default capacity unit definitions (for example, Basic, Small, Medium, or Large) or define a custom capacity unit that meets your resource requirements. Each default capacity unit definition provides a different amount of primary cache capacity for Coherence, based on the number of VMs and JVMs configured in the unit. Each default capacity unit (except the Basic configuration) is defined as a group of three VMs to provide a high availability (HA) unit, three being the smallest unit of VMs that will provide Coherence HA. When using a custom capacity unit, Coherence HA is achieved when you specify three VMs per custom unit. After a service instance has been provisioned, you can't change the default capacity unit you used or the configuration of a custom capacity unit you defined.

Note:

- Only one capacity unit is provisioned upon initial creation of an Oracle Java Cloud Service—Coherence instance. The default capacity unit you choose or the custom unit you define during provisioning determines how Oracle Java Cloud Service scales the service instance when you initiate a scaling operation on the Coherence data tier.
- Based on the capacity unit definition declared for the service instance, a scaling operation always increases or decreases the total cache capacity for Coherence by a preconfigured, fixed amount, adding or removing a group of VMs at a time.

About Properties of a Capacity Unit Definition

Oracle Java Cloud Service uses a capacity unit to create the Coherence data tier when you enable Oracle Coherence for a service instance during provisioning. A capacity unit definition has properties that determine the amount of Coherence cache capacity to provide, based on how the VMs, JVMs, and JVM heap are configured.

When creating an Oracle Java Cloud Service—Coherence instance, you don't have to worry about defining any properties if you use a default capacity unit.

If you want to define your own configuration for a custom capacity unit (by using Advanced configuration during provisioning), you will have to specify a collection of properties for the custom unit.

The properties of a default or custom capacity unit definition are:

- VM count: Each capacity unit has a predefined number of VMs.
- VM shape: Each VM has the same shape. A shape, which is the number of Oracle Compute Units (OCPUs) and amount of RAM, determines the processing power of the VM and the amount of memory available for JVM heap. For example, the shape `oc3` has 1 OCPU and 7.5 GB of memory.

- JVM count: Each VM is configured to run the same number of JVMs (Managed Coherence Servers).
- JVM heap: The same heap size is configured with each JVM. The memory available for heap is 75% of the remaining shape memory after reserving 1.5 GB for the operating system. The heap size per JVM is the available memory divided by the number of JVMs per capacity unit. The general rule is to use 1/3rd of heap for primary cache storage.

The cache size or cache capacity of a capacity unit is the amount of primary cache storage to configure for Coherence, which is based on:

- The predefined number of JVMs (Managed Coherence Servers) on each VM
- The predefined heap size on each JVM (Managed Coherence Server)

Note: Only one capacity unit of VMs and JVMs is provisioned and configured upon initial creation of an Oracle Java Cloud Service—Coherence instance. To increase the cache capacity provided by a service instance, you scale out the Coherence data tier by adding one capacity unit at a time.

About Default Capacity Units for Oracle Java Cloud Service—Coherence Instances

When you provision an Oracle Java Cloud Service instance using the Create Java Cloud Service wizard, you can use a default capacity unit to enable Oracle Coherence for the service instance. Oracle Java Cloud Service provides a set of default capacity units that are optimized for different use cases.

Each default capacity unit definition provides a different combination of primary cache storage and VM processing power that fits most application and workflow requirements.

A default capacity unit has:

- A name (for example, Basic, Small, Medium, or Large)
- A shape (for example, low or high processing power, and standard memory or memory-intensive)
- A cache capacity or cache size, which is a fixed amount of primary cache storage to allocate for Coherence, based on the unit's predefined properties for number of VMs, number of JVMs per VM, and heap size for each JVM.

With the exception of the Basic definition, a default capacity unit is a group of three VMs, which is the smallest unit that will provide Coherence high availability. The Basic definition is configured with one VM only.

Use the following capacity unit properties to help you decide which default capacity unit definition to use:

Default Capacity Unit Name	Number of VMs per Unit	Shape	OCPUs	Memory (RAM in GB)	Number of JVMs per VM	Heap Size per JVM (MB)	Total Primary Cache (MB) per Unit
Basic (non-HA)	1	oc3	1	7.5	1	4608	1536
Small	3	oc3	1	7.5	1	4608	4608
Medium	3	oc4	2	15	2	5120	10240
Large	3	oc5	4	30	4	5632	22528

Note the following when using a default capacity unit:

- The amount of memory for JVM heap is based on a shape's memory (RAM). The total memory that's used for heap configuration is 75% of the remaining shape memory after reserving 1.5 GB for the operating system. The heap size per JVM is the available memory divided by the number of JVMs per capacity unit.
- The total primary cache storage to allocate for Coherence is based on the general rule of splitting the JVM heap size into thirds, using 1/3rd for primary cache storage, 1/3rd for backup storage, and 1/3rd for scratch space.
- Only one capacity unit of VMs and JVMs is provisioned and configured upon initial creation of an Oracle Java Cloud Service—Coherence instance. To increase the cache capacity provided by a service instance, you scale out the Coherence data tier by adding one capacity unit at a time.
- The smaller the cache size of a capacity unit (total primary cache capacity), the smaller the cache increment (or decrement) by which the cluster will grow (or shrink) when you initiate a scaling operation on the cluster.
- You can scale out the cluster to the maximum cache capacity supported by the data center site.

About Custom Capacity Units and Advanced Configuration for Oracle Java Cloud Service—Coherence Instances

When you provision an Oracle Java Cloud Service—Coherence instance using the Create Java Cloud Service wizard, instead of using a default capacity unit, you can choose the Advanced option and then define your own properties for a custom capacity unit.

When defining a custom capacity unit in the Create Java Cloud Service wizard, use these guidelines when you specify the following properties:

Property	Description	Guideline
Shape	The compute shape to use for VMs on the Coherence data tier. All VMs have the same shape, which is the number of Oracle Compute Units (OCPUs) and amount of RAM (system memory) available for a VM instance.	For information about the defined compute shapes, contact your Oracle Cloud administrator.
Nodes	The number of VMs to configure per capacity unit.	Depending on the given Compute Shape, use a number from 1 to 3. To achieve Coherence high availability, select 3. This is because three is the smallest number of VMs that will provide Coherence high availability.
Managed Servers per Node	The number of JVMs or Managed Coherence Servers to start on each VM for the selected Compute Shape. All VMs have the same number of JVMs.	Depending on the given shape, use a number from 1 to 8. Using more servers can improve concurrency and memory management but can require more processors. The maximum number of servers you can configure per VM is based on using the minimum of 1 GB for individual JVM heap size. For example, if the memory available for heap is 4.5 GB, then the maximum number of servers you can select is 4.

Consider the following when defining a custom capacity unit:

- The memory available for JVM heap configuration is based on a shape's memory (RAM). The total memory that's available is 75% of the remaining shape memory after reserving 1.5 GB for the operating system. The heap size to configure for each JVM is the available memory divided by the number of JVMs per unit.

  For example, if a capacity unit uses a shape that has 7.5 GB of memory, the memory available for heap configuration is 75% of 6 GB or 4.5 GB. If the capacity unit is predefined with 1 VM and is configured to run 2 JVMs per VM, then the heap size per JVM is half of 4.5 GB or 2.25 GB.

  When you provision a service instance using the service creation wizard available through the Oracle Java Cloud Service Console, the maximum individual heap size that will be configured is 16 GB. While you can use the WebLogic Server Administration Console to increase the heap size after the service instance is created, the change is not reflected in the Oracle Java Cloud Service Console or capacity unit definition for your service instance. When you initiate a scale out operation, the original heap size will be configured for new JVMs.

- As with default capacity units, the total amount of primary cache storage to allocate for Coherence in the domain is based on the general rule of splitting the JVM heap size into thirds, using 1/3rd for primary cache storage, 1/3rd for backup storage, and 1/3rd for scratch space.

Following the example described earlier: Provisioning one capacity unit would provide (4.5/3) or 1.5 GB of primary cache to allocate for Coherence.

However, if your custom capacity unit is configured with a single VM, there might not be space to store a backup copy of the Coherence data, so the actual data available might be more than 1/3 of the JVM heap size.

- The larger the cache size of a capacity unit (total primary cache capacity), the larger the cache increment (or decrement) by which the cluster will grow (or shrink) when you initiate a scaling operation on the cluster.

Note: Only one capacity unit of VMs and JVMs is provisioned and configured upon initial creation of an Oracle Java Cloud Service—Coherence instance. To increase the cache capacity provided by a service instance, you scale out the Coherence data tier by adding one capacity unit at a time.

About Service Lifecycle Operations and Oracle Java Cloud Service—Coherence Instances

Your role determines the type of tasks you can perform on Oracle Java Cloud Service—Coherence instances. All service tasks and lifecycle operations can be performed by using the Oracle Java Cloud Service Console.

Topics:

To use the REST API to perform service tasks and lifecycle operations, see About Using the REST API to Create and Manage Oracle Java Cloud Service—Coherence Instances.

About Creating an Oracle Java Cloud Service—Coherence Instance

You can enable Oracle Coherence for an Oracle Java Cloud Service instance to use in-memory data grid and caching for your Coherence application deployed to the cloud. Use the Oracle Java Cloud Service Console or the REST API to create an Oracle Java Cloud Service—Coherence instance.

Before you provision an Oracle Java Cloud Service—Coherence instance, ensure you're familiar with capacity units. See About Capacity Units for Oracle Java Cloud Service—Coherence Instances.

To create a service instance using the Create Java Cloud Service wizard (which is launched from the Oracle Java Cloud Service Console), see Creating an Oracle Java Cloud Service Instance.

When you provision an Oracle Java Cloud Service—Coherence instance using the wizard, there are certain options you must select so that an environment for using Oracle Coherence in Oracle Java Cloud Service is correctly provisioned and configured. See About Oracle Java Cloud Service—Coherence Instances.

For information about using the REST API to create a service instance, see Create a Service Instance in *REST API for Oracle Java Cloud Service*.

To enable Oracle Coherence for an existing Oracle Java Cloud Service instance, use the REST API, as described in Adding a Coherence Data Tier Cluster to a Service Instance in *REST API for Oracle Java Cloud Service*. See also Considerations When Adding a Coherence Data Tier to an Oracle Java Cloud Service Instance.

About Deploying Applications to an Oracle Java Cloud Service—Coherence Instance

You can deploy and undeploy Oracle Coherence applications to an Oracle Java Cloud Service—Coherence instance by using the WebLogic Server Administration Console, Fusion Middleware Control, WLST commands, or an IDE. You cannot deploy and undeploy applications directly through the Oracle Java Cloud Service Console.

Deploying applications to your Oracle WebLogic Server environment in the cloud is the same as deploying to an on-premises environment. The GAR is deployed to the Coherence data tier cluster in the domain; the EAR (WAR and GAR) is deployed to the application tier cluster.

For more information about deploying applications, see Deploying and Undeploying Applications for an Oracle Java Cloud Service Instance.

About Scaling an Oracle Java Cloud Service—Coherence Instance

In the Oracle Java Cloud Service Console, you initiate a scaling operation on the Coherence data tier by choosing to add or remove a capacity unit. The capacity unit definition declared for an Oracle Java Cloud Service—Coherence instance during initial provisioning predetermines how many storage-enabled nodes to add or remove when you scale the Coherence data tier.

By adding or removing a capacity unit, the total primary cache storage to allocate for Coherence is increased or decreased by the amount of cache capacity as preconfigured by one capacity unit.

Before scaling the Coherence data tier, understanding certain aspects of the scaling operation will help ensure a successful experience. For details, see Considerations When Scaling the Coherence Data Tier of an Oracle Java Cloud Service Instance.

To scale the Coherence data tier using the Oracle Java Cloud Service Console, see Scaling the Coherence Data Tier by Using the Oracle Java Cloud Service Console.

To scale the Coherence data tier using the REST API, see Scale a Coherence Data Tier in *REST API for Oracle Java Cloud Service*.

The application tier is scaled independently of the Coherence data tier. To scale the application tier of an Oracle Java Cloud Service—Coherence instance, see Scaling an Oracle Java Cloud Service Instance.

For service instances provisioned with the Oracle Java Cloud Service—Virtual Image service level, you must use the WebLogic Server Administration Console to add or

remove Managed Servers. Scaling by using the Oracle Java Cloud Service Console or the REST API is not supported by Oracle Java Cloud Service—Virtual Image instances.

About Backing Up and Restoring an Oracle Java Cloud Service—Coherence Instance

Weekly full backups and daily incremental backups are already scheduled for you when your Oracle Java Cloud Service—Coherence instance is provisioned. Backups contain only the volumes that are needed for a proper restoration of a service instance.

Note that data stored in the Coherence data tier is not backed up or restored as part of the Oracle Java Cloud Service backup and restoration functionality.

When a service instance contains Managed Server hosts on the Coherence data tier that are not in a backup, you should scale in the service instance by removing the capacity unit that manages those specific nodes and the corresponding Managed Server hosts before trying to restore the service instance from the backup. For information about removing a specific capacity unit, see Scaling the Coherence Data Tier by Using the Oracle Java Cloud Service Console.

In terms of initiating an operation related to backup or restoration (for example, configuring when automated backups are performed, initiating a backup on-demand, restoring a service instance from a backup, or deleting a backup), there are no differences between Oracle Java Cloud Service—Coherence instances and service instances without Coherence enabled. For complete information about backup or restoration, such as backup contents and storage, and what happens during a backup or restoration operation, see Backing Up and Restoring an Oracle Java Cloud Service Instance.

You can use the Oracle Java Cloud Service Console or the REST API to perform backup and restoration operations. See Typical Workflow for Backing Up and Restoring an Oracle Java Cloud Service Instance for instructions on how to perform all operations using the Oracle Java Cloud Service Console. For information about using the REST API to back up or restore a service instance, see Backups and Restorations in *REST API for Oracle Java Cloud Service*.

Using the Oracle Java Cloud Service Console or the REST API to back up or restore a service instance is not supported by Oracle Java Cloud Service—Coherence instances that are provisioned with the Oracle Java Cloud Service—Virtual Image service level.

About Patching an Oracle Java Cloud Service—Coherence Instance

When released, patches for Oracle WebLogic Server, Oracle Traffic Director (load balancer), Oracle Fusion Middleware, and JDK are made available on the Oracle Java Cloud Service Console. When there is a PSU patch for Oracle Coherence, the patch is included as part of the Fusion Middleware patch. You can use the Oracle Java Cloud Service Console or the REST API to patch your Oracle Java Cloud Service—Coherence instance on your own schedule.

In terms of initiating an operation related to viewing details of a patch and rolling back a patch, there are no differences between an Oracle Java Cloud Service instance without Coherence and a service instance that has Coherence enabled. When applying a patch to an Oracle Java Cloud Service—Coherence instance, note that by default the patching operation checks that the `StatusHA` state for a Coherence member reaches `NODE-SAFE` before shutting down the node to apply the patch.

For information about how to perform all patching and rollback operations using the Oracle Java Cloud Service Console, see Typical Workflow for Patching an Oracle Java Cloud Service Instance. For information about using the REST API to apply or roll back a patch, see Patches in *REST API for Oracle Java Cloud Service*.

Applying or rolling back a patch by using the Oracle Java Cloud Service Console or the REST API is not supported in Oracle Java Cloud Service—Coherence instances that are provisioned at the Virtual Image service level.

What Happens During Patching

All general information about patching are found in About Patching an Oracle Java Cloud Service Instance.

Before patching Oracle Java Cloud Service—Coherence instances, understanding certain aspects of the patching operation will help ensure a successful patching experience.

- All Managed Coherence Servers on a VM are shut down before the patch is applied to the VM.
- If the topology consists of a few VMs with multiple Managed Coherence Servers on each VM, then the load on the remaining cluster nodes may increase significantly.
- A Managed Coherence Server is stopped gracefully prior to being shut down.
- By default, the patching process performs a rolling restart of Managed Coherence Servers. Maintaining data is achieved by checking the `StatusHA` state of the services within the cluster and ensuring that they are always in state `NODE-SAFE` before shutting down a node to apply the patch.
- If there is no high availability requirement for your application and you can accept the possibility of data loss, you may choose to override the default behavior by selecting the Confirm checkbox on the Patch Service dialog (if patching through the Oracle Java Cloud Service Console). When selected, the patching process will shut down a node to apply the patch even if `NODE-SAFE` cannot be reached. When the Confirm checkbox is not selected, note that patching will fail if `NODE-SAFE` cannot be reached.

About Stopping and Starting an Oracle Java Cloud Service—Coherence Instance

Stopping an Oracle Java Cloud Service—Coherence instance stops the virtual machines on which the Administration Server, the application tier Managed Servers, the Coherence data tier Managed Servers, and the Load Balancer (if present) are running.

Note the following about stopping and starting:

- When an Oracle Java Cloud Service—Coherence instance is stopped, all data in the Coherence cache is lost.
- Starting an Oracle Java Cloud Service—Coherence instance starts all the virtual machines on which the Administration Server, the application tier Managed Servers, the Coherence data tier Managed Servers, and the Load Balancer (if present) are running.
- Unlike application tier virtual machines, you cannot stop or start Coherence data tier virtual machines individually.

For more information about stopping and starting a service instance, see About Stopping and Starting an Oracle Java Cloud Service Instance and Individual VMs.

About Deleting an Oracle Java Cloud Service—Coherence Instance

You use the Oracle Java Cloud Service Console or the REST API to delete Oracle Java Cloud Service—Coherence instances that you no longer need.

To delete a service instance using the Oracle Java Cloud Service Console, see Deleting an Oracle Java Cloud Service Instance. For information deleting a service instance using the REST API, see Delete a Service Instance in *REST API for Oracle Java Cloud Service*.

When you delete an Oracle Java Cloud Service—Coherence instance, the entire environment is removed. If you want to remove only the resources and entities related to Oracle Coherence in Oracle Java Cloud Service, see Deleting the Coherence Data Tier of an Oracle Java Cloud Service Instance.

About Using the REST API to Create and Manage Oracle Java Cloud Service—Coherence Instances

The Oracle Java Cloud Service REST API provides an alternative to using the web-based Oracle Java Cloud Service Console to create and manage Oracle Java Cloud Service instances that have Oracle Coherence enabled.

For information about how to access the Oracle Java Cloud Service REST resources, and the authentication information to provide, see Quick Start in *REST API for Oracle Java Cloud Service*.

The following table shows some of the supported REST API you can use when creating and managing Oracle Java Cloud Service—Coherence instances:

Task	REST Resource	More Information
List and view all default capacity unit configurations	`GET /datagrid/{identityDomainId}/config/scalingUnits/defaultScalingUnits`	View Default Capacity Units (Reference)
Create a service instance (using a default or custom capacity unit)	`POST /instances/{identityDomainId}`	Creating a Service Instance (Use Case) Create a Service Instance (Reference)
Add a Coherence data tier cluster to a service instance that's already provisioned	`POST /instances/{identityDomainId}/{serviceId}/clusters`	Adding a Coherence Data Tier Cluster to a Service Instance (Use Case) Add a Coherence Data Tier Cluster (Reference)
Scale the Coherence data tier of a service instance	`PUT /instances/{identityDomainId}/{serviceId}/clusters/{clusterName}`	Scaling the Coherence Data Tier of a Service Instance (Use Case) Scale a Coherence Data Tier (Reference)

Task	REST Resource	More Information
Delete the Coherence data tier cluster of a service instance	`DELETE /instances/{identityDomainId}/{serviceId}/clusters/{clusterName}`	Deleting the Coherence Data Tier Cluster of a Service Instance (Use Case) Delete a Coherence Data Tier Cluster (Reference)
View details of a service instance	`GET /instances/{identityDomainId}/{serviceId}`	View a Service Instance (Reference)
View all managed servers for a service instance	`GET /instances/{identityDomainId}/{serviceId}/servers`	View All Managed Servers (Reference)
View details of a managed server	`GET /instances/{identityDomainId}/{serviceId}/servers/{serverName}`	View a Managed Server (Reference)
Delete a service instance	`PUT /instances/{identityDomainId}/{serviceId}`	Deleting a Service Instance (Use Case) Delete a Service Instance (Reference)

For a list of all REST endpoints, see *REST API for Oracle Java Cloud Service*.

Creating an Oracle Java Cloud Service—Coherence Instance on OPCM

Oracle Java Cloud Service—Coherence instances are service instances created in Oracle Java Cloud Service that have Oracle Coherence enabled. You can use the Oracle Java Cloud Service Console or the REST API to create and manage service instances.

Creating an Oracle Java Cloud Service—Coherence instance is the same as creating an Oracle Java Cloud Service instance without enabling Coherence, with the exception of specifying details for the Coherence data tier:

- To create a service instance using the Create Java Cloud Service wizard (which is launched from the Oracle Java Cloud Service Console), see Creating an Oracle Java Cloud Service Instance.
- To create a service instance using the REST API, see Create a Service Instance in *REST API for Oracle Java Cloud Service*.

When you provision an Oracle Java Cloud Service—Coherence instance using the wizard, there are certain options you must select so that an environment for using Oracle Coherence in Oracle Java Cloud Service is correctly provisioned and configured. See About Oracle Java Cloud Service—Coherence Instances for information about those options and the Coherence data tier details you've to select or specify.

Adding a Coherence Data Tier to an Oracle Java Cloud Service Instance on OPCM

Only the REST API can be used to add a Coherence data tier to an Oracle Java Cloud Service instance that's already been provisioned. You cannot use the Oracle Java

Cloud Service Console to enable Oracle Coherence on a service instance that's already created.

Before adding a Coherence data tier, see Considerations When Adding a Coherence Data Tier to an Oracle Java Cloud Service Instance.

To use the REST API to add a Coherence data tier, see Add a Coherence Data Tier Cluster in *REST API for Oracle Java Cloud Service*.

Considerations When Adding a Coherence Data Tier to an Oracle Java Cloud Service Instance

Adding a Coherence data tier to an Oracle Java Cloud Service instance that's already provisioned means you're enabling Oracle Coherence for the service instance by creating a storage-enabled, data tier WebLogic Server cluster of Managed Servers. If a service instance already has Coherence enabled, you cannot add a second data tier cluster to the service instance.

Note: Adding a Coherence data tier to an existing service instance can be performed by using the REST API only.

A Coherence data tier can be added only to a service instance that's been provisioned to run Oracle WebLogic Server 12*c* (12.1.3) or Oracle WebLogic Server 12*c* (12.2.1), and Enterprise Edition with Coherence. You cannot add a Coherence data tier to an Oracle Java Cloud Service instance that's running Oracle WebLogic Server 11*g*.

To provision a Coherence data tier of Managed Servers, you've to use a default capacity unit or define a custom capacity unit for the service instance. Be sure you're familiar with capacity units before you proceed with adding a Coherence data tier. See About Capacity Units for Oracle Java Cloud Service—Coherence Instances for information about using default and custom capacity units.

During provisioning of a Coherence data tier:

- Based on the capacity unit definition you declare for the service instance, a number of Managed Servers are created in the data tier WebLogic Server cluster.
- Both the data tier and application tier WebLogic Servers clusters are associated with the Coherence cluster.
- The Coherence data tier members are storage-enabled by default.
- The application tier members are storage-disabled by default.
- A scale out operation is added to the Activity page.

Scaling the Coherence Data Tier of an Oracle Java Cloud Service Instance on OPCM

The Coherence data tier of an Oracle Java Cloud Service—Coherence instance is scaled independently of the application tier. You can use the Oracle Java Cloud Service Console or the REST API to scale the Coherence data tier.

To use the REST API to scale the Coherence data tier, see Scale a Coherence Data Tier in *REST API for Oracle Java Cloud Service*.

To scale the application tier of a service instance, see Scaling Out an Oracle Java Cloud Service Cluster, Scaling In an Oracle Java Cloud Service Cluster, and Scaling an Oracle Java Cloud Service Node.

Note: This section does not apply to Oracle Java Cloud Service—Coherence instances provisioned with the Oracle Java Cloud Service—Virtual Image service level. Scaling by using the Oracle Java Cloud Service Console or the REST API is not supported by Oracle Java Cloud Service—Virtual Image instances.

Topics:

Considerations When Scaling the Coherence Data Tier of an Oracle Java Cloud Service Instance

Scaling the Coherence data tier of an Oracle Java Cloud Service—Coherence instance means you're adding or removing a predefined number of virtual machines (VMs) and Managed Coherence Servers (or JVMs), thereby increasing or decreasing the total cache capacity to allocate for Coherence in the domain.

When your Oracle Java Cloud Service—Coherence instance was created, you either used a default capacity unit or defined a custom capacity unit for the service instance. The number of VMs and JVMs to add or remove during a scaling operation is predetermined by the capacity unit definition you declared for the service instance during initial provisioning. You cannot change the definition of your capacity unit when you scale the Coherence data tier.

At any time during a scaling operation, you can check the job status by opening the Activity page. See Viewing Scaling Requests.

About Using the Oracle Java Cloud Service Console

When you use the Oracle Java Cloud Service Console to initiate a scaling operation, you can add or remove only one capacity unit at a time. Whether you're scaling out or scaling in, the scaling process always adds or removes the appropriate number of VMs and server nodes as defined by one capacity unit, increasing or decreasing the total primary cache storage to allocate for Coherence by the amount of cache capacity as preconfigured by one capacity unit.

Note the following when scaling in the Coherence data tier:

- You can choose to remove a specific capacity unit or let the scaling process remove any one capacity unit.
- When there's only one capacity unit on the Coherence data tier, the remove option is disabled.
- During a scale in operation, data is maintained by checking that the Coherence cluster has a `StatusHA` value of `NODE-SAFE` before removing a node. If the Coherence data tier has less than three nodes in total, the check isn't performed.

  Select the Force Remove checkbox to remove the capacity unit even if there's any failure during the scaling process.

About Using the REST API

When you use the REST API to initiate a scaling operation, you specify the number of capacity units you want for the entire Coherence data tier. Based on the capacity unit count requested, the current capacity unit count, and the capacity unit definition declared for the service instance when the Coherence data tier was created, the scaling process then performs either a scale out operation to add more server nodes or a scale in operation to remove nodes.

For example, if the Coherence data tier currently has a capacity unit count of 2 and you specify a unit count of 3, the scaling process initiates a scale out operation to provision 1 more capacity unit, adding and configuring the appropriate number of Managed Coherence Servers as defined in one capacity unit. Note that you can scale out by one capacity unit at a time only.

For more information, see Scale a Coherence Data Tier in *REST API for Oracle Java Cloud Service*.

Scaling the Coherence Data Tier by Using the Oracle Java Cloud Service Console

To increase or decrease the Coherence cache capacity for an Oracle Java Cloud Service —Coherence instance, you can use the Oracle Java Cloud Service Console to scale out or scale in the Coherence data tier by adding or removing one capacity unit at a time.

Based on the capacity unit configuration that was declared for the service instance when it was provisioned, the following occurs when you scale the Coherence data tier:

- The appropriate number of nodes as was provisioned by one capacity unit are added to or removed from your service instance
- The total primary cache storage to allocate for Coherence is increased or decreased by the amount of cache capacity as was predefined by one capacity unit

---

Note:

- You cannot use the scaling operation to delete the entire Coherence data tier by removing one capacity unit at a time until all nodes are deleted.
- When scaling in, you can remove a specific capacity unit or let the scaling process remove any one capacity unit.
- You cannot scale the Coherence data tier when the Oracle Java Cloud Service instance is in a stopped state.

To scale the Coherence data tier by using the Oracle Java Cloud Service Console:

1. From the Services page of the Oracle Java Cloud Service Console, click an existing service instance.

   The Overview page is displayed.

2. Navigate to the Topology page.
3. To scale out by adding one capacity unit, do the following:

   a. In the Coherence Data Tier section, click  (next to OCPUs) and choose **Add Capacity Unit** to add nodes and increase capacity.

A confirmation dialog is displayed, asking you to confirm that you want to add a capacity unit.

b. Click **Add Capacity Unit** to confirm.

A message saying that the request has been submitted is displayed.

c. Proceed to step 5.

4. To scale in by letting the operation remove *any* one capacity unit, do the following:

a. In the Coherence Data Tier section, click  (next to OCPUs) and choose **Remove Capacity Unit** to remove nodes and decrease capacity.

Note that the Remove Capacity Unit option is disabled when there's only one capacity unit on the data tier.

A confirmation dialog is displayed, asking you to confirm that you want to delete any one capacity unit.

b. Select the Force Remove checkbox to allow the operation to remove the capacity unit even if there's any failure during the process.

c. Click **Remove Capacity Unit**.

A message saying that the request has been submitted is displayed.

d. Proceed to step 5.

5. To scale in by removing a *specific* capacity unit, do the following:

a. Click next to the Coherence Data Tier icon.

The section expands to show the individual capacity units configured for your service.

b. Review the list of capacity units and their Managed Servers to confirm the capacity unit you want to remove. Then on the row of the capacity unit that you want to remove, click  and choose **Remove**.

Note that if there's only one capacity unit, the  icon is disabled or not available.

A confirmation dialog is displayed, asking you to confirm that you want to remove that particular capacity unit and its Managed Servers.

c. Select the Force Remove checkbox to allow the operation to remove the capacity unit even if there's any failure during the process.

d. Click **Remove Capacity Unit**.

A message saying that the request has been submitted is displayed.

e. Proceed to step 5.

6. After a few moments, click  to update the console (you might need to do this more than once).

After a few moments of processing, the number of nodes and cache capacity will be updated in the Coherence Data Tier section. If a scale out operation is performed, the new node information is displayed when you expand the section to view the capacity unit details. See Viewing Scaling Requests for information about how to check the status of an ongoing scaling operation as well as previous scaling operations.

Patching an Oracle Java Cloud Service—Coherence Instance on OPCM

You use the Oracle Java Cloud Service Console or the REST API to apply a patch to an Oracle Java Cloud Service—Coherence instance.

Note: This section does not apply to Oracle Java Cloud Service—Coherence instances provisioned with the Oracle Java Cloud Service—Virtual Image service level. Patching is not supported by Oracle Java Cloud Service—Virtual Image instances.

Patching an Oracle Java Cloud Service—Coherence instance is very similar to patching a service instance that doesn't have Oracle Coherence enabled. For general information about patching, see About Patching an Oracle Java Cloud Service Instance.

The only difference is by default, the patching process on an Oracle Java Cloud Service —Coherence instance checks that the `StatusHA` state for a Coherence member is `NODE- SAFE` before shutting down the node to apply the patch. This means no data will be lost if a node fails during patching. If there is no high availability requirement for your application and you can accept losing data, you may choose to override the default behavior.

For more information about patching a service instance when Oracle Coherence is enabled, see About Patching an Oracle Java Cloud Service—Coherence Instance.

To apply a patch, see Applying a Patch to an Oracle Java Cloud Service Instance.

To use the REST API to apply a patch, see Patches in *REST API for Oracle Java Cloud Service*.

Deleting the Coherence Data Tier of an Oracle Java Cloud Service Instance on OPCM

You cannot use the Oracle Java Cloud Service Console to remove the Coherence data tier. You can perform the task by using the REST API only.

Only specific resources and entities related to Oracle Coherence in Oracle Java Cloud Service are deleted when you remove the Coherence data tier cluster of storage-enabled Managed Servers from a service instance. A scale in operation is added to the Activity page.

To use the REST API to remove the Coherence data tier, see Delete a Coherence Data Tier Cluster in *REST API for Oracle Java Cloud Service*.

12

Troubleshooting Oracle Java Cloud Service on Oracle Public Cloud Machine

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

Before you begin troubleshooting, perform the following tasks.

Ensure that you are using best practices. For information, see Keeping Your Instances Manageable by Oracle Java Cloud Service.

Finding 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:

- Finding Status Messages for Oracle Java Cloud Service Instances

Using 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. Open the WebLogic Administration Console.
3. Click the menu icon and select **WebLogic Server Console**.

   A new browser opens and you are redirected to the WebLogic Server Administration Console's login page.

   If the console is protected by a self-signed certificate, you will be warned that this certificate is not trusted.
4. Accept the certificate.
5. When the console log-in page appears, enter the log-in credentials you entered for the WebLogic Administrator when you created the service instance.

   The WebLogic Administration Console is displayed.
6. In the Domains area, expand **Diagnostics**.
7. Click on the diagnostics that interests you.

   For information on the diagnostic choices, click on **Diagnostics**.

Using 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 Cloud Service Console.
2. Open the WebLogic Administration Console.
   a. Click the Menu icon and select Open WebLogic Server 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.

   b. Accept the certificate.

   c. When the WebLogic Server Console appears, enter the username and password your provided when you created the service instance.
3. In the Domains area, expand **Diagnostics**.
4. Click **Log Files**.

5. The Log Files table is displayed.
6. Click the button to the left of the log file you want to view.
7. Click **View**.
8. 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.

Finding 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 Viewing Activity for Service Instances.

To find status messages for a failed attempt to create a service instance:

1. Navigate to the Oracle Java Cloud Service Console.
2. Click on the arrow next to **Instance create or 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.

Finding 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 Compute Cloud Service. See Viewing the Boot Log of an Instance in *Using Oracle Compute Cloud Service (for PaaS)*.

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 is due to the fact that the user password for the infrastructure repository schemas is set to expire in 180 days after an Oracle Java Cloud Service instance is created. You will get 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.

You must be able to log onto the VM that hosts the WebLogic Server Administration Server as `opc`. Doing so allows you to `sudo` as `oracle`. All actions on files on the VM that hosts the Administration Server or VMs that host Managed Servers should be executed as `oracle`.

Summary of steps in this procedure:

1. Update each infrastructure repository schema's password on the database deployment.
2. If WebLogic Servers are running and the WebLogic Server Console is accessible, change the password for all the corresponding datasources from the Weblogic Administration Console.
3. If WebLogic Servers are not running and WebLogic Server Console is inaccessible, manual change the passwords in WebLogic Server configuration.
4. Run offline WLST command `modifyBootStrapCredential`.
5. Start the Administration Server through Node Manager, and then start the Managed Servers.

To remedy this problem:

1. Update each infrastructure repository schema's password on the database deployment.

   If the schema prefix is already known, jump to Step b.

   a. Login to the Java Cloud Service Instance VM that hosts the Administration Server, and get the value of the schema prefix.

      ```
      ssh -i <private_key> opc@<IP_address_of_admin_server_VM>
      sudo su oracle
      curl http://192.1.1.192/latest/user-data/chef/initial_attributes/wlss/
      schema_prefix
      ```

      The Schema Prefix value returned would be similar to the following:

      ```
      SP255951777
      ```

   b. Login to the Oracle Database Cloud Service database deployment VM.

      ```
      ssh -i <ssh_key> opc@<DB_vm_ip_address>
      sudo su oracle
      ```

   c. Connect to the Oracle Database Cloud Service database deployment.

      ```
      sqlplus / as sysdba
      ```

Use the username provided when provisioning the database deployment.

If your database deployment version is 12*c*, the following step is also required:

```
alter session set container=PDB1
```

Use the PDB name provided during Oracle Java Cloud Service.

d. Change the password for the infrastructure repository schema users to new password.

Note:

The following schema users must have their passwords changed:

For Fusion Middleware 11*g*	For Fusion Middleware 12.1.3	For Fusion Middleware 12.2.1.x
<schema_prefix>_IAU	<schema_prefix>_IAU	<schema_prefix>_IAU
<schema_prefix>_IAUOES	<schema_prefix>_IAU_APPEND	<schema_prefix>_IAU_APPEND
<schema_prefix>_IAUOES_APPEND	<schema_prefix>_IAU_VIEWER	<schema_prefix>_IAU_VIEWER
<schema_prefix>_IAUOES_VIEWER	<schema_prefix>_MDS	<schema_prefix>_MDS
<schema_prefix>_IAU_APPEND	<schema_prefix>_OPSS	<schema_prefix>_OPSS
<schema_prefix>_IAU_VIEWER	<schema_prefix>_STB	<schema_prefix>_STB
<schema_prefix>_MDS		<schema_prefix>_UMS
<schema_prefix>_OPSS		

Change the password for each of the schema users pertaining to the WebLogic Server version on the database deployment. For example:

```
ALTER USER <schema_prefix>_IUA identified by <new_password>;
```

e. Unlock all the user accounts on the database to cover for the case that they are locked due to repeated login failures after password expiry.

```
ALTER USER <schema_prefix>_IAU ACCOUNT UNLOCK;
```

Note: If the WebLogic Administration Server is running and the WebLogic Administration Console is accessible, follow Step 2, else jump to Step 3.

2. Update all the datasources from the WebLogic Administration Console to reflect the new password.

a. Log in to the WebLogic Administration Console and navigate to the Services > Datasources menu on the Domain Structure box.

b. Click **Lock & Edit**.

c. For each datasource, navigate to the Datasource Name > Configuration > Connection Pool tab and update the Password and Confirm Password field with the new password.

d. Click on **Save** button on this page, and then **Activate**.

e. Stop all the WebLogic Servers.

From the WebLogic Administration Console, navigation to the Environment > Servers > Control tab.

Select all the servers and click Shutdown > Force Shutdown Now.

Proceed to Step 4.

3. If the WebLogic Server is not running or the Administration Console is not accessible:

a. Encrypt the new schema password and Update Data Source Configuration files

```
ssh -i <private_keyopc@<ipaddress_of Admin_VM>
sudo su oracle; cd /u01/data/domain/<domain_name>
```

Ensure WebLogic Servers are not running. If running, to stop the processes:

```
kill -9 <pid>
```

then run:

```
. <domain_home/bin/setDomainEnv.sh
```

a. Run the WebLogic Encryption Utility and enter the password you set for the database schemas.

```
/u01/jdk/bin/java weblogic.security.Encrypt
 password: <Enter the new password for the schema user>
```

b. An encrypted password will be required such as:

```
AES}JHyrhOMB5hVRuDU/pV0qX86qz98ZV0xWXBSEAANA4Gs=
```

Note the encrypted password output for future reference.

c. Update the new password in the datasource xml files.

```
cd <domain_home>/<domain_name>/config/jdbc
```

Open the datasource xml files found in the `<domain_home/<domain_name>/config/jdbc` directory that need to be updated with the new encrypted password:

For Fusion Middleware 11*g*	For Fusion Middleware 12.1.3	For Fusion Middleware 12.2.1.x

`mds-owsm-jbdc.xml`
`opss-ds-jdbc.xml`
	`LocalSvcTblDataSource-jdbc.xml`
	`opss-auditview-jdbc.xml`
	`mds-owsm-jdbc.xml`
	`opss-datasource-jdbc.xml`
	`opss-audit-jdbc.xml`
		`LocalSvcTblDataSource-jdbc.xml`
		`opss-auditview-jdbc.xml`
		`mds-owsm-jdbc.xml`
		`opss-datasource-jdbc.xml`
		`opss-audit-jdbc.xml`

4. Run the `Modify BootStrapCredential` WLST command to update `jps-config.xml` with the new password for `<SCHEMA_PREFIX>_OPSS` user. See Step 2b.

a. Login to the Oracle Java Cloud Service VM hosting the Administration Server.

```
ssh -i <private_key> opc@<IP_of_admin_server>
sudo su oracle
```

b. Invoke WLST.

```
/u01/app/oracle/middleware/oracle_common/common/bin/wlst.sh
```

c. Run the `modifyBootStrapCredential` command to update the `jps-config` file with the new password for the schema user.

> **Note:** The full path to the JPS config file must be specified.

For example:

```
wls:/offline>modifyBootStrapCredential(jpsCconfig_File='<domain_home>/
<domain_name>/config/fmwconfig/jps-
config.xml',username='<schema_prefix>_OPSS',password='<new_password_set_for_thi
s_schema_user>')
```

5. Start the Administration Server through the Node Manager and then the Managed Server(s).

a. Log in to the Oracle Java Cloud Service VM hosting the WebLogic Administration Server.

b. Start WLST.

```
/u01/app/oracle/middleware/oracle_common/common/bin/wlst.sh
```

c. Connect to the Node Manager.

Before running the command, get the required values of some of the variables involved.

- Host name — On the command prompt, type `hostname`.
- Node Manager port number, domain name, domain home — Open the nodemanager.properties files to determine the respective values.

  For 11*g*:

  ```
  u01/app/oracle/middleware/wlserver_10.3/common/nodemanager/
  nodemanager.properties
  ```

  For 12*c*:

  ```
  /u01/data/domains/<domain_name>/nodemanager/nodemanager.properties
  ```

- Administration Server name —

  ```
  cd /u01/data/domains/<domain_name>/servers.
  ```

  Look for the server name ending in `adminserver`.

Run the `nmConnect` command.

```
nmConnect('<weblogic_username>','<weblogic_password>','<hostname>','<domain_nam
e>','<domain_home>/<domain_name>','ssl')
```

d. Start the Administration Server.

```
nmStart("<admin_server_name>")
```

e. If the Administration Server has status RUNNING, access the WebLogic Administration Console and start the Managed Servers.

   - Click on Servers under Environments in the Domain Structure section.
   - Under the Control tab, select the Managed Servers and click **Start**.

The password expiry problem has been fixed.

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 or delete history** section.
5. Click on **Details** to receive progress and error messages.

   You can locate additional error messages using the procedure in Finding 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
- Exceeding resource quota (for trial subscriptions)

The following solutions apply to problems creating service instances for Oracle Java Cloud Service.

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 Instances in *Using 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.

In my trial subscription, the Details message states “Failed to create Compute resources”

If you have an Oracle Java Cloud Service trial subscription, a common reason instance creation fails is exceeding the amount of compute resources allocated to your account. If you exceed your quota of compute resources, new services will not be created. To determine whether you have a quota breach, open the Dashboard to find a message such as `1 Ongoing Quota Breach`.

If you have exceeded your quota, delete unused or unwanted service instances. Once you delete the service instances, wait. The quota mechanism takes some to recalculate. After waiting, try again to create a new service instance.

There is no quota for paid subscriptions.

Problems with Deploying and Accessing Applications

Problems might occur when you attempt deploy or access an application.

The following solutions apply to problems with application deployment and access for Oracle Java Cloud Service.

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 are WebLogic Server 12*c* (12.2.1), which supports Java EE 7, or WebLogic Server 12*c* (12.1.3), which supports Java EE 6.

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 Accessing 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 Create New Oracle Java Cloud Service Instance 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 OTD, then create an access rule. If your service instance has a load balancer, see Enabling HTTP Access to an Oracle Java Cloud 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:

- Understanding the Default Access Ports in *Using Oracle Java Cloud Service*.
- 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.

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.

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. To protect your key store and trust store, use OPSS KeyStoreService or put them in the WebLogic domain.

For WebLogic Server patching, do not:

- Put CA certificates in the existing demo keystores
- Put custom key stores and trust stores in the `<MW_HOME>/wlserver/lib` directory
- 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.

The following solutions apply to problems with backup and restoration operations for Oracle Java Cloud Service.

Backups fail after the identity domain password changes

To prevent backup failure after you change your identity domain password, update the password in the following locations:

- Configure Backups dialog box, which you invoke from the Backup page in the Oracle Java Cloud Service user interface

  See Configuring 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 *Using Oracle Database Cloud Service*.

The Oracle Traffic Director is not backed up

Typically, this occurs when the traffic director is currently busy servicing other requests.

Verify that Oracle Traffic Director is running and in a healthy state, and try backup 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, 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.

To check for available space, log in to the VM and check the size of the backup mounted directory under `/u01/data/backup`. See Accessing VM Through Secure Shell (SSH).

If there is not enough space for the backup, do one of the following:

- Delete any unwanted backups. See Deleting a Backup.
- Archive one or more backups to your NFS remote backup destination. See Downloading or Archiving a Backup for a Service Instance in *REST API for Oracle Java Cloud Service*.

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. Sign in to the Oracle Enterprise Manager Cloud Control on your Oracle Public Cloud Machine with the URL and credentials that your administrator gave you.
   b. From the **Enterprise** menu, choose **Cloud** and then **Self Service Portal**.
   c. From the **Manage** list, select **Exalogic Public Cloud**.
   d. If you have access to more than one site, select the site where you want to work from the **Oracle Compute Site** list.
   e. In the Oracle Java Cloud Service section, click **Service Console**.

      The Oracle Java Cloud Service sign in page is displayed.
   f. Sign in with your Oracle Java Cloud Service credentials.
   g. The Oracle Java Cloud Service Console is displayed.
   h. 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.
   i. Click the Administration tile.

      The Oracle Java Cloud Service Instance page is refreshed with the Administration tile in focus.
   j. 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, the software automatically attempts a scale-in operation. Try restoring the service again.

If there is not enough space for the backup, do one of the following:

- Delete any unwanted backups. For more information, see Deleting a Backup.
- Archive one or more backups to your NFS remote backup destination. For more information, see Downloading or Archiving a Backup for a Service Instance 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 NFS backup destination.

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, ensure that your NFS backup destination is correct and that it has the correct permissions. See Configuring Automated 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.

   For more information, see Accessing the Administrative Consoles Used by Oracle Java Cloud Service.
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.

   For more information, see Using the WebLogic Server Administration Console to Shut Down Servers Without Stopping VMs and Using 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.

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 Scaling 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 Disk Volumes.

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.

My 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 Accessing a VM Through 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.
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 Accessing a VM Through 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 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 Using WLST Commands to Restart the Admininistration 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 12.1.3, 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 user interface.

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 12.1.3, 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 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 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
```

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 12.1.3 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*.