Oracle® Cloud

Using Oracle Application Container Cloud Service

E64179-32
Sep 2019

Oracle Cloud Using Oracle Application Container Cloud Service,

E64179-32

Primary Authors: Rebecca Parks, Marilyn Beck, Rob Gray

Contributing Authors: Michael W. Williams

Contents

Preface

Using Oracle Application Container Cloud Service tells you how to deploy applications and describes the web user interface.

Topics:

Audience

This document is primarily for administrators who are responsible for provisioning an Oracle Cloud service, monitoring service usage and performance, and deploying applications.

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

See `Oracle Public Cloud`.

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.

Convention	Meaning
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 Application Container Cloud Service

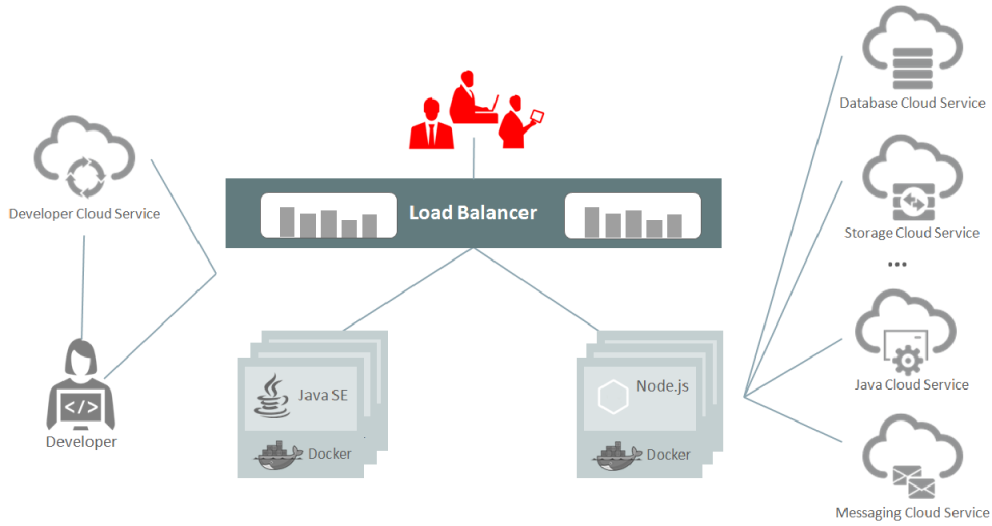
Get started with general information about using Oracle Application Container Cloud Service.

Topics:

About Oracle Application Container Cloud Service

Oracle Application Container Cloud Service lets you deploy Java SE, Node.js, PHP, Python, Ruby, Go, and .NET Core applications to the Oracle Cloud. You can also deploy Java EE web applications.

Subscribing to Oracle Application Container Cloud Service makes all types of applications available when you deploy your application. Your application runs in a Docker container.



With Oracle Application Container Cloud Service, you can use these key features:

- A preconfigured environment for Java SE, Java EE, Node.js, PHP, Python, Ruby, Go, and .NET Core applications.

- Java SE advanced features such as Java Flight Recorder, Java Mission Control, advanced memory management, and ongoing and timely security updates.
- An Open platform that supports all Java frameworks and containers such as Spring, Play, Tomcat, and Jersey.
- Support for Java Virtual Machine (JVM) based languages such as JRuby. You can run any language that uses the JVM on this service.
- Enterprise-grade support from Oracle.
- Web-based user interface and REST API.

In addition, you can integrate with other Oracle Cloud services, develop your application on your local system, or use Oracle Developer Cloud Service.

If you want to store and retrieve data, then you can use Oracle Cloud Infrastructure Object Storage Classic — a subscription is included. You can also subscribe to Oracle Database Cloud Service or Oracle MySQL Cloud Service.

To deploy an application and configure a database in a single step, you can use Oracle Cloud Stack.

For communication, you can subscribe to Oracle Messaging Cloud Service. Your application communicates with that service through its REST API, so no special security configuration is necessary.

Before You Begin with Oracle Application Container Cloud Service

Before using Oracle Application Container Cloud Service, ensure that your Oracle Cloud Infrastructure Object Storage Classic account is active. You must also have the necessary user roles. It's also useful to be familiar with other related Oracle Cloud services and the technologies that Oracle Application Container Cloud Service uses, such as Java SE, Java Flight Recorder, and Java Mission Control.

For details about service activation, see Requesting and Managing Free Oracle Cloud Promotions or Buying an Oracle Cloud Subscription in *Getting Started with Oracle Cloud*. For details about required user roles, see About Oracle Application Container Cloud Service Roles and Users.

When you purchase a subscription for Oracle Application Container Cloud Service, you automatically also get a subscription for Oracle Cloud Infrastructure Object Storage Classic and Oracle Developer Cloud Service. You can also purchase other subscriptions, such as Oracle Database Cloud Service and Oracle Messaging Cloud Service, if needed for your application.

Oracle Application Container Cloud Service uses the following technologies:

- Java SE

  If you're new to Java SE, learn about it at Java Platform, Standard Edition (Java SE).

- Java Flight Recorder

  Java Flight Recorder (JFR) generates detailed recordings of the Java Virtual Machine (JVM) and the application it is running. The recorded data includes an execution profile, garbage collection statistics, optimization decisions, object

allocation, heap statistics, and latency events for locks and I/O. See the *Java Flight Recorder Runtime Guide* in the Java Components Documentation.

- Java Mission Control

  The Java Mission Control (JMC) is a set of tools that run on the JDK and interact with a JVM to deliver advanced, unobtrusive Java monitoring and management. See the *Java Mission Control User's Guide* in the Java Components Documentation.

Access Oracle Application Container Cloud Service

Access Oracle Application Container Cloud Service through a web console, REST API, or a command-line interface (CLI).

Depending on how you sign up for Oracle Cloud, you are directed to either Oracle Cloud Infrastructure Console or Oracle Cloud Infrastructure Classic Console.

Topics:

Access Oracle Application Container Cloud Service from Oracle Cloud Infrastructure Console

Oracle Application Container Cloud Service has limited access to the accounts entitled to use it.

To access the Oracle Application Container Cloud Service console from Oracle Cloud Infrastructure Console:

1. Sign in to Oracle Cloud.

   If you receive a welcome email, use to sign in to Oracle Cloud. The email provides you with your cloud account details and sign in credentials.

2. From Oracle Cloud Infrastructure Console, click the navigation menu  in the top left corner, expand **Platform Services**, and then click **Application Container.**
3. When you access Oracle Application Container Cloud Service the first time, you will see the welcome page.
4. Click **Applications** or **Go to Console** to display the applications page.

The Oracle Application Container Cloud Service console is displayed.

Access Oracle Application Container Cloud Service from Oracle Cloud Infrastructure Classic Console

To access the Oracle Application Container Cloud Service console from Oracle Cloud Infrastructure Classic Console:

1. Sign in to Oracle Cloud.

If you receive a welcome email, use to sign in to Oracle Cloud. The email provides you with your cloud account details and sign in credentials.

2. From Oracle Cloud Infrastructure Classic Console, click the navigation menu in the top left corner, and then click **Application Container.**
3. When you access Oracle Application Container Cloud Service the first time, you will see the welcome page.
4. Click **Applications** or **Go to Console** to display the applications page.

The Oracle Application Container Cloud Service console is displayed.

About Oracle Application Container Cloud Service Roles and Users

You must be assigned certain roles to access, administer, and use Oracle Application Container Cloud Service.

If you don't have all the roles that you need, then ask an administrator to assign these roles to you. If you're assigned the `TenantAdminGroup` role, then you can perform all tasks including assigning roles to other users. See Add Users and Assign Roles in *Getting Started with Oracle Cloud*.

The following are all the roles that are required to deploy applications to Oracle Application Container Cloud Service.

Role	Description
`APAAS APAAS Administrators`	Required for deploying and managing applications.
`Compute Compute Operations`	Required for using Oracle Cloud Infrastructure Compute Classic, on which Oracle Application Container Cloud Service depends. You need Compute resources to run your application from the Oracle Cloud platform.
`Storage Storage Read Only Group`	Required for reading the contents of Oracle Cloud Infrastructure Object Storage Classic, on which Oracle Application Container Cloud Service depends for log files.
`Storage Storage Read Write Group`	Required for reading and writing the contents of Oracle Cloud Infrastructure Object Storage Classic, on which Oracle Application Container Cloud Service depends for log files. In addition, to deploy an application using the REST API or the command-line interface, you must copy the application into Oracle Cloud Infrastructure Object Storage Classic first.

2

Administering Applications

Learn more about the Oracle Application Container Cloud Service user interface and application deployment.

Topics

Typical Workflow for Administering Applications

Complete these tasks to deploy and manage your applications in Oracle Application Container Cloud Service.

Task	Description	More Information
Sign up for a free credit promotion or purchase a subscription	Provide your information, and sign up for a free trial or purchase a subscription to Oracle Cloud.	See Purchasing a Subscription to an Oracle Cloud Service in *Getting Started with Oracle Cloud*.
Add and manage users and roles	Optionally create additional accounts for your cloud users.	Adding Users and Assigning Roles in *Getting Started with Oracle Cloud*
Access the service console	Access the service console.	Access Oracle Application Container Cloud Service
Prepare the application for Cloud deployment	Learn how to create a new application, or modify an existing application, for this service. An existing program must be modified to read certain environment variables when it's deployed. In addition, it needs a launch point and one or two metadata files.	See Design Considerations and The Packaging Process in *Developing for Oracle Application Container Cloud Service*.

Task	Description	More Information
Deploy your application to the service	Use the user interface, the REST API, or deploy from Oracle Developer Cloud Service.	To deploy using the REST API, see Create an Application in *REST API for Managing Applications*. To deploy from Oracle Developer Cloud Service, see Deploying an Application to Oracle Application Container Cloud in *Using Oracle Developer Cloud Service*. To deploy an application and configure a database in a single step, you can use Oracle Cloud Stack. See Deploy an Application and Configure a Database with Stack Manager.
Connect your application to other Oracle Cloud services.	Use a service binding to connect fromOracle Application Container Cloud Service to Oracle Database Cloud Service, MySQL Cloud Service, Oracle Java Cloud Service, Oracle Event Hub Cloud Service, Oracle Data Hub Cloud Service, or an application cache.	See Managing Service Bindings.
Test your application remotely		
Monitor your application by using Java Flight Recorder (Only Java applications)		See Java Mission Control and Java Flight Recorder in *Developing for Oracle Application Container Cloud Service*.
Manage your application	You can manage your applications using the web user interface or the REST API. As your application is running, you can: • Change the number of instances. • Alter the amount of memory allocated to each instance. • Review application logs. • Upload a new version of the application. • Upload new metadata files.	See *REST API for Managing Applications*.

Explore the Applications Page

On the Applications page, you can create an application and you can see all of the created applications.

The following table describes all the elements that you'll see on the Applications page.

Element	Description
Applications	See all the applications that are running in the identity domain. To refresh the list, click the icon. The date and time of the last refresh is displayed.
Search	To filter the list, enter an application name. Enclose each tag in single quotation marks. You can search using tag expressions.
Create Application	To deploy an application, click **Create Application.**
Java SE, Java EE, Node, PHP, Python, Ruby, Go, or .NET	See the version of the runtime environment used by the deployed application.
application name	Click to open the Application Console and view the details of the application.
Version	See the version of the deployed application.
Runtime	See the runtime version.
Tag	See the tags assigned to the application instance. The first tag is displayed. To see all tags assigned to the service instance, hover over the tag name.
Last Deployed On	See the date-time stamp of the most recent deployment.
Created On	See the date-time stamp of when this application was created.
Memory	View the memory allotted to each instance.
Instances	View the number of instances of the application.
URL	To connect to your deployed application, click the URL. A **Type: worker** label in place of a URL indicates a worker application. See Preparing a Worker Application for Deployment in *Developing for Oracle Application Container Cloud Service*.
	Click to perform these actions: • **Open Application**: Starts the application using the URL provided in the application list. • **View Service Metrics**: Opens the Monitoring graph, which displays memory usage or memory data. See View Service Metrics for an Application. • **Start**: Starts any instances that aren't running. • **Stop**: Stops all instances of the application. • **Restart**: Restarts all instances of the application. • **Delete**: Stops all instances of the application, and deletes the application and its associated files.
Application creation and deletion history	Expand to view all applications that have been created or deleted during the time period that you select. To see system messages about the action that occurred, click **Details**.

Create an Application

From the Applications page, you can create and deploy your application.

1. Open the Oracle Application Container Cloud Service Applications page.

   See Access Oracle Application Container Cloud Service.

2. Click **Create Application**.

3. Click an application type: **Java SE**, **Java EE**, **Node**, **PHP**, **Python**, **Ruby**, **Go**, or **.NET.**

4. In the Create Application dialog box, complete the fields described in the following table. All fields are optional except for **Name** and **Application**.

Element	Description
Region	**(Available only for cloud accounts with Oracle Identity Cloud Service)** Select a region to which to deploy the application. The default is **No Preference**. The others Oracle Cloud services that you intend to bind with your application must be in the same region that you select in this field.
Name	Enter the application name. The application name must start with a letter and must contain letters and numbers only. Spaces and other special characters aren't allowed.
Application	Select one of these options for identifying the application archive location: • **Upload Archive**— Click **Browse** to upload your application archive. • **Deploy Sample**— Select the sample application for your application type.
Instances	Specify the number of container instances that you need for your uploaded application. This number can be changed after deployment. The default is 2. The maximum is 64.
Memory	Specify the memory (in GB) to be allotted to each container instance. This number can be changed after deployment. The default is 2 GB. The maximum is 20 GB.
Authentication	Select the type of authentication for a Java SE 7 or 8, Node.js, or PHP application: • None — Provides no authentication. • Basic — Prompts for a username and password set up in Oracle Identity Cloud Service. • OAuth — Creates a corresponding application in Oracle Identity Cloud Service to control who can access your application, and redirects to Oracle Identity Cloud Service for authentication. See Use Oracle Identity Cloud Service with Oracle Application Container Cloud Service.
Application Cache	For an application that uses a cache, select a cache from the list. This option appears only if at least one cache exists. To create a cache and use it in your application, see Typical Workflow for Creating and Using Caches in *Using Caches in Oracle Application Container Cloud Service*.

5. Click **More Options** if you want to upload configuration files or specify a metering frequency, runtime version, or notification email. All these fields are optional.

Element	Description
Metering Frequency	Select the application's billing frequency for a metered subscription. Select **Hourly** to be billed for the number of hours you use the service. Select **Monthly** to be billed monthly regardless of hourly usage. After creating the application, you can't change the subscription type. If your Oracle Application Container Cloud Service subscription is non-metered, then the value of this field is ignored.
Manifest	Click **Browse** to upload a `manifest.json` file. This file is required if your application requires a launch command and your archive doesn't include this file. See Creating Metadata Files in *Developing for Oracle Application Container Cloud Service*.
Deployment Configuration	Click **Browse** to upload a `deployment.json` file. See Creating Metadata Files in *Developing for Oracle Application Container Cloud Service*.
Notification Email	Enter an optional email address to which to send notification of whether application creation succeeded or failed.
Version	What you specify indicates the type and runtime version of the uploaded application: • Java SE — 7, 8, 9, 10 • Java EE — 7 • Node.js — 0.10, 0.12, 4, 6, 8 • PHP — 5.6, 7.0, 7.1 • Python — 2.7.13, 3.6.0, 3.6.1 • Ruby — 2.3.4, 2.4.1 • Go — 1.7.6, 1.8.3 • .NET — 1.1.2-runtime, 2.0.0-runtime You can specify the version here or in the `manifest.json` file. If you do both, then the version specified here takes precedence.
Tags	(Optional) Select existing tags or add tags to associate with the application. To select existing tags, select one or more check boxes from the list of tags that are displayed on the pull-down menu. To create tags, enter a tag that can be a key or a key:value pair and press Enter. If you do not assign tags during provisioning, you can create and manage tags after the service instance is created. See Explore the Application Overview Page.
Notes	(Optional) Enter the release notes or the application description.

6. Click **Create**.
7. Click **OK**.

   A Creating Application message is displayed until the application is fully deployed.

Configure a Vanity URL

A vanity URL is a unique, customized web address that's branded for marketing purposes and helps users remember and find a specific page of your website. To

customize the user sign-in experience, customers may use their own vanity URL prefix instead of the Oracle Application Container Cloud Service application URL.

To configure your vanity URL, you must have a custom domain name, access to an Oracle Cloud Infrastructure Load Balancing Classic account, and a Secure Socket Layer (SSL) certificate. See Obtaining a Digital Certificate in *Using Oracle Cloud Infrastructure Load Balancing Classic*.

1. Add the digital certificate to Oracle Cloud Infrastructure Load Balancing Classic. See Importing a Load Balancer Digital Certificate in *Using Oracle Cloud Infrastructure Load Balancing Classic*.
2. Update the Oracle Cloud Infrastructure Load Balancing Classic listener of your application. Include the digital certificate in and add the vanity URL of the certificate as one of the virtual hosts. See Creating Listeners for a Load Balancer in *Using Oracle Cloud Infrastructure Load Balancing Classic*.
3. Create a `CNAME` record in your domain name system, entering your vanity URL as the name and the Oracle Application Container Cloud Service application URL as the value.
4. To test your vanity URL, enter it in a browser. Oracle Application Container Cloud Service application is displayed.

Use the Application Console

The Application Console provides access to all application configuration parameters, including the static application manifest values and the service dependencies. You can start, stop, and restart applications, and view the activity log in the Application Console.

1. Go to the Applications page.

   The page appears after you sign in or when you click on the **Applications** link at the top of any page. See Access Oracle Application Container Cloud Service.
2. In the **Applications** list, click the desired application name.

At the top of the Application Console are the application name, the application Menu  , and the application URL.

A **Type: worker** label in place of a URL indicates a worker application. See Preparing a Worker Application for Deployment in *Developing for Oracle Application Container Cloud Service*.

The application Menu  displays these options:

- **Open Application** — Starts the application using the URL.
- **View Service Metrics** — Opens the Monitoring graph, which displays memory usage or memory data. See View Service Metrics for an Application.
- **Start** — Starts all instances of the application.
- **Stop** — Stops all instances of the application.
- **Restart** — Restarts all instances of the application.
- **View Activity** — Displays the activity log.

If you expand the **Service Details** icon next to the menu icon, the following additional information is displayed:

- Location of the application in Oracle Cloud Infrastructure Object Storage Classic
- User who created the application
- Notes about the application entered when the application was created, if any
- Date-time stamp of application creation
- Metering frequency of the application, either Hourly or Monthly
- Identity domain name of the application

The following table describes the key pages available in the Application Console. By default, the application Overview page opens when you access the Application Console.

Element	Description
Overview	Overview of the application. View and change application instances and memory information. See Explore the Application Overview Page.
Deployments	Details of the deployed application. View and change service bindings, environment variables, the application launch command, and the history of the application. See Explore the Application Deployments Page.
Administration	Application administration. Apply service and runtime updates, view and download application logs, and view and download Java Flight Recorder recordings for Java applications. See Explore the Application Administration Page.

Stop, Start, and Restart an Application

You can stop, start, and restart an application from the Service Console, the Application Console, or the REST API. This section describes using the Service and Application Consoles.

Stopping an application stops the running instances and frees up the memory. The metering of the resources also stops, which is useful for hourly subscriptions. While an application is stopped, you can't perform any of the following actions:

- Increase or decrease allocation instances
- Manage memory allocation of instances
- Redeploy the application
- Add, edit, or remove service bindings
- Add, edit, or remove environment variables
- Generate logs and recordings

Starting an application starts all stopped instances, and the load balancer starts routing traffic. Metering of the resources also starts. An application starts by default when it's created.

Restarting an application stops it and starts it immediately. You have two options:

- Rolling restart — Instances stop and start one at a time, so the application has no downtime.

- Concurrent restart — Instances stop and start at the same time. The application is down briefly, but the overall restart time is faster.

You have these options for all actions that require restarting an application:

- Restarting the application
- Scaling memory up or down
- Updating the application runtime
- Updating Oracle Application Container Cloud Service
- Updating the application, which includes:
  - Uploading a new archive
  - Changing the launch command
  - Changing service bindings
  - Changing environment variables

Setting rolling mode in the `manifest.json` file causes a rolling restart upon redeployment. See Creating Metadata Files in *Developing for Oracle Application Container Cloud Service*.

To stop, start, and restart an application from the Service Console:

1. Open the **Services Console**.
2. In the **Applications** table, click next to the application that you want to stop, start, or restart.
3. Select the desired menu option.
4. In the confirmation dialog box, click **OK**.
5. If you selected the **Restart** menu option, then select **Rolling Restart** or **Concurrent Restart** on the Restart Application dialog, then click **Restart** .

To stop, start, and restart an application from the Application Console:

1. Open the **Application Console**.
2. In the **Applications** table, click the application name.
3. In the Application Console header, click next to the application path.
4. Select the desired menu option.
5. In the confirmation dialog box, click **OK**.
6. If you selected the **Restart** menu option, then select **Rolling Restart** or **Concurrent Restart** on the Restart Application dialog box, and then click **Restart**.

Based on your action, an event is generated in the **Activity** section of the Overview page.

Delete an Application

You can delete an application using the Service Console.

1. Open the **Service Console**.
2. In the **Applications** table, click next to the application that you want to delete.

3. Click **Delete**.

   Caution: You can't undo the deletion of an application.

Any running instances of the deployed application are stopped. The deployed application and its configuration data are deleted.

Explore the Application Overview Page

The Overview page of an application displays detailed information about the deployed application.

The following table describes key information that you'll see on the application Overview page.

Element	Description
Summary	Manages application instances and the memory allotted to each application instance. In **Instances**, specify the number of application instances. By default, two application instances are allotted when the application is created. In **Memory**, specify the memory (in GB) allotted to each application instance. By default, 2 GB of memory is allotted. In **Average Memory Usage**, see the average for all application instances of the percentage of the total allocated memory that's used.
More Information	Displays the following information about the application: • Current version of the deployed application • Date-time stamp of the last deployment • Runtime environment and version used to execute the application. For Python, Ruby, .NET, and Go applications, you can use this value to find the application in DockerHub and download it from there. • Type of the application, web if public and worker if private. See Preparing a Worker Application for Deployment in *Developing for Oracle Application Container Cloud Service*. • Region to which the application was deployed • Security IP List, which you can add to a security rule for a virtual machine (VM) in Oracle Cloud Infrastructure Compute Classic. This allows your application and the VM to communicate. • Application cache used, if any • A **Manage Access** link if you selected Basic or OAuth authentication during application creation. Click this link to go to Oracle Identity Cloud Service. See Use Oracle Identity Cloud Service with Oracle Application Container Cloud Service. • Tags assigned to the application. You can assign or remove tags without redeploying the application. See Create, Assign, and Unassign Tags.
Resources	Lists all instances allotted for the application, including their allocated memory and memory usage. A public application has instances named **web.***N*. A worker application has instances named **worker.***N*. See Preparing a Worker Application for Deployment in *Developing for Oracle Application Container Cloud Service*.

Scale an Application Out or In

Scaling an application out or in means that you're to increasing or decreasing the number of instances.

1. Under **Instances,** enter the number of desired instances, or use the up and down arrows to set the number of instances.

   After you change the number of instances, you'll see the **Apply** and **Cancel** buttons.

2. After you've set the number of instances to your desired value, click **Apply**.

   A confirmation dialog box appears. Its message indicates that new instances are started and old instances are still running.

3. Click **Apply**. Your application scales or in to the number of instances requested.

Scale an Application Up or Down

Scaling an application up or down means that you're to increasing or decreasing the size of each instance.

1. Under **Instances,** enter the desired size of each instance in GB, or use the up and down arrows to set the size of instances.

   After you change the size, you'll see **Apply** and **Cancel** buttons.

2. After you've set the memory size of each instance to your desired value, click **Apply**.

   A confirmation dialog box appears. Its message indicates that all instances are restarted.

3. Select **Rolling Restart** or **Concurrent Restart**.

4. Click **Apply**. Your application scales all instances up or down to the desired memory size.

Considerations when Scaling Applications

Your instances start or restart depending on the scaling operation.

The direction that you scale determines if your instances restart or just start:

- If you're scaling out, then new instances start.
- If you're scaling in, then all instances are restarted.
- If you're scaling up, then all instances are restarted.
- If you're scaling down, then all instances are restarted.

Explore the Application Deployments Page

The Deployments page of an application enables you to redeploy the application, configure environment variables and service bindings, and view the deployment history.

The following table describes the key information that you'll see on the application Deployments page.

Element	Description
Deployments	Displays the following information about the deployed application: • Current version of the deployed application • Date-time stamp of the last deployment • Size of the archive file • Name of the archive file • Application source, the account that deployed the application • Commit ID, a string assigned in the `manifest.json` file when the application was packaged • Build number, a string assigned in the `manifest.json` file • Release notes, notes about the application when it was deployed See Redeploy an Application.
Runtime	Displays the command that launches the application and the runtime version.
Topology	Displays the number of application instances and the amount of memory per instance in gigabytes.
Service Bindings	Lists other Oracle Cloud services to which your application connects. See Manage Service Bindings.
Environment Variables	Lists all environment variables available to the application. Click **Hide System Variables** to remove the system variables from the list. See Configure Environment Variables.
Deployment History	Lists the deployment history of the application. To download the application `manifest.json` file or the `deployment.json` file, click  .

Redeploy an Application

If you've modified your application, then you can redeploy it from the Deployments page of the Application Console. This procedure assumes that the application has been deployed previously.

1. Open the Service Console.
2. In the Applications table, click the application name.
3. In the left pane, click **Deployments**.
4. Click **Update**.
5. Click **Browse** to upload a new archive, `manifest.json`, or `deployment.json` file.
6. Select the file and click **Open**.
7. Edit the **Launch Command**, **Version**, **Instances**, or **Memory (GB)** values if necessary.
8. Add service bindings or environment variables if necessary.

   See Manage Service Bindings or Configure Environment Variables.
9. Click **Apply Edits** to save the changes and restart the application.

10. If prompted, select **Rolling Restart** or **Concurrent Restart**, then click **Restart**.

Manage Service Bindings

A service binding provides connectivity to other subscribed Oracle Cloud services.

Currently, you can use a service binding to connect from Oracle Application Container Cloud Service to Oracle Database Cloud Service, Oracle MySQL Cloud Service, Oracle Java Cloud Service, Oracle Event Hub Cloud Service, Oracle Data Hub Cloud Service, or an application cache.

To add a service binding:

1. Open the **Service Console**.
2. In the **Applications** table, click the application name.
3. In the **Application Console**, click the **Deployments** page.
4. In the Service Bindings section of the Deployments page, click **Add**.
5. In the Add Service Binding dialog box, specify these values:
   - **Service Type** – Type of the service: Oracle Java Cloud Service, Oracle Database Cloud Service, Oracle MySQL Cloud Service, Oracle Event Hub Cloud Service, Oracle Data Hub Cloud Service, or caching service. For Oracle Event Hub Cloud Service, specify OEHCS to publish or subscribe to a topic or OHEPCS to access a cluster at the platform level.
   - **Service Name** – Name of the service, the name of an Oracle Java Cloud Service instance, Oracle Database Cloud Service database, Oracle MySQL Cloud Service database, Oracle Event Hub Cloud Service topic or cluster, Oracle Data Hub Cloud Service instance, or cache service.
   - **Username** – User name used to access the service. See the note after the steps for details.
   - **Password** – Password for the user name. See the note after the steps for details.
6. Click **Save**.
7. To apply the changes, click **Apply Edits** at the top of the page. If prompted, select **Rolling Restart** or **Concurrent Restart**, then click **Apply**.

> **Note:**
>
> The **Username** and **Password** aren't automatically used to authenticate against the target service. The values are placed in the `SERVICE_USER_NAME` and `SERVICE_USER_PASSWORD` environment variables, which your application can access. If the target service requires authentication, then your application must handle it. If the target service doesn't require authentication, then you can't omit the user name and password for Oracle Java Cloud Service, Oracle Database Cloud Service, or Oracle MySQL Cloud Service, but you can specify any values. For Oracle Event Hub Cloud Service, the **Username** and **Password** fields are optional. For a cache service, the **Username** and **Password** fields are grayed out and aren't used.

The service binding is added to the list and its environment variables are automatically added to the Environment Variables list. For Oracle Java Cloud Service, these variables have names beginning with the prefix `JAAS_`. For Oracle Database Cloud Service, the prefix is `DBAAS_`. For Oracle MySQL Cloud Service, the prefix is `MYSQLCS_`. For Oracle Event Hub Cloud Service, the prefix is `OEHCS_` or `OEHPCS_`. For Oracle Data Hub Cloud Service, the prefix is `DHCS_`. For a cache service, there is one variable, named `CACHING_INTERNAL_CACHE_URL`. See Configure Environment Variables.

> **Note:**
>
> The environment variables generated from an Oracle MySQL Cloud Service service binding are all uppercase, but are displayed in the console with mixed case. Ensure your code references the uppercase names.

To edit a service binding, click the **Edit** icon, edit the values in the Service Binding – Edit dialog box, and click **Save**. To delete a service binding, click the **Delete** icon.

As an alternative, you can define service bindings in the `deployment.json` file. See Creating Metadata Files in *Developing for Oracle Application Container Cloud Service*.

See Service Bindings in *Developing for Oracle Application Container Cloud Service*.

Configure Environment Variables

Environment variables provide details of the runtime environment for the application and are accessible to the running application.

Some predefined environment variables are available when you create an application. Creating service bindings for your application creates related environment variables. You can create additional environment variables, and manage all environment variables, from the Deployments page of the Application Console.

The following predefined environment variables are available when you create an application:

- `$HOSTNAME` – The system generated host name for this application.
- `$PORT` – The port that most application types should listen to for traffic from the load balancer.
- `$APP_HOME` – The directory in which the deployed application archive file is extracted.
- `$ORA_APP_NAME` – The name of the application, user-specified at creation time.
- `$ORA_APP_PUBLIC_URL` – The public URL of the application.
- `$ORA_INSTANCE_NAME` – The name of the specific application instance, such as `web.1` or `web.2`.
- `$ORA_PORT` – The same value as the `$PORT` environment variable. Your application can use either `$PORT` or `$ORA_PORT` to read the port.

If you've added a service binding to a subscribed Oracle Cloud service, more predefined environment variables are available, enabling the deployed application to interact with the Oracle Cloud service. See Manage Service Bindings.

> **Note:**
>
> The environment variables generated from an Oracle MySQL Cloud Service service binding are all uppercase, but are displayed in the console with mixed case. Ensure that your code references the uppercase names.

You can also define your own environment variables as key-value pairs in the Deployments page of the Application Console or in the `deployment.json` file. For details about the `deployment.json` file, see Creating Metadata Files in *Developing for Oracle Application Container Cloud Service*.

The best practice for connecting to a service with which you can't use a service binding or a REST API is to create environment variables for the connection information. For example, to connect to a virtual machine (VM) in Oracle Cloud Infrastructure Compute Classic, create environment variables for the IP address and port of the VM.

An environment variable can reference other environment variables. For example, suppose you have a service binding and environment variables for a connection to an Oracle Database Cloud Service database but don't want to use the default pluggable database (PDB). Suppose the service name for your PDB is `MYPDB.identity-domain.oraclecloud.com`. You can create your own connection descriptor in an environment variable named `MYPDB_CONNECT_DESCRIPTOR` that references the host name and port of the database but includes your PDB:

```
$DBAAS_LISTENER_HOST_NAME:$DBAAS_LISTENER_PORT/MYPDB.identity-
domain.oraclecloud.com
```

To add an environment variable in the Deployments page of the Application Console:

1. Open the **Service Console**.
2. In the **Applications** table, click the application name.
3. In the **Application Console**, click the **Deployments** page.
4. In the Environment Variables section of the **Deployments** page, click **Add**.
5. In the Add Environment Variable dialog box, in the **Name** field, enter the variable name , in the **Value** field, enter a value. To mark the environment variable as secured, uncheck the **Show** check box. The value of the secure environment variables is hidden on the user interface.
6. Click **Save**.

To edit a variable, click the **Edit** icon, edit the values in the Edit Environment Variable dialog, and click **Save**.
To delete a variable, click the **Delete** icon.

To apply the changes, click **Apply Edits** at the top of the page. Select **Rolling Restart** or **Concurrent Restart**, then click **Restart**.

Explore the Application Administration Page

The Administration page of an application lets you to apply service and runtime updates, view and download application logs, and view and download Java Flight Recorder recordings for Java applications.

The following table describes key information that you'll see on the application Administration page.

Element	Description
Timestamp and Counts	The Administration tab in the left pane displays the following information: • Count of runtime updates available • Timestamp of most recent logs or recordings • Count of logs for all application instances • Count of recordings for all application instances
Updates	To apply updates to Oracle Application Container Cloud Service or to the Java SE, Java EE, Node.js, PHP, Python, Ruby, Go, or .NET runtime environments, click the **Updates** tab. To apply an update, click the **Update** button. To activate a runtime update immediately, you must restart the application. On the page, click **Restart**. Select **Rolling Restart** or **Concurrent Restart**, then click **Restart**. To view the update history, expand the **Update and Rollback History**. Applications running on older runtime versions continue to run unchanged. New applications must use the latest supported runtime versions.
Logs	To download application logs, click the **Logs** tab. A new log is generated every three minutes. To retrieve additional application logs, click **Logs**, select **Get More**, and select an application instance. Time range options for logs already retrieved are Last hour (the default), Last 24 hours, Last 7 days, Last month, and Custom Period. Expand the logs list and click a log file name to download it. To view the log history, expand **Log Capture History**.
Recordings	To download Java Flight Recorder recordings of JVM instances, click the **Recordings** tab. To generate a recording of profile and event information for the running application, click the **Get Recording** button, and select an application instance or **All** (the default) for all instances. Time range options for recordings already retrieved are Last hour (the default), Last 24 hours, Last 7 days, Last month, and Custom Period. To download the recording file, expand the recordings list and click the file name. You can analyze the recording with Java Mission Control. To view the recordings history, expand **Recordings History**.

View Service Metrics for an Application

The Monitoring graph displays either percent memory usage or absolute memory for an Oracle Application Container Cloud Service application.

1. Access the Monitoring graph in one of the following ways:

- On the Applications page, from the **Menu**, select **View Service Metrics**.
- On the Application Console, the **Menu**, select **View Service Metrics**.
- On the Application Overview page, click **Average Memory Usage**.

The Monitoring graph is displayed.

2. On the Monitoring graph, select one of the following graph types:
   - **Memory Usage** (default) — The percentage of the total allocated memory that's used.
   - **Memory**— The actual amount of consumed memory.

   The graph displays either memory usage or memory data.
3. In the **Instance** selection field, select one of the following:
   - **web.1** or **worker.1** instance (default)
   - Another instance
   - **All**

   A graph of data for your selected instance or for all instances is displayed. The graph provides a snapshot of up to 24 hours of data in 5-minute intervals. If you selected All, then each application instance has its own data points.
4. To return to the Application Overview page, click the name of the application.

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

View Activity for Service Instances

Use the Activity page to view activities for service instances. In Oracle Application Container Cloud Service, a service instance is an application or a cache. You can restrict the list of activities displayed by using search filters.

To view activities for your Oracle Application Container Cloud Service instances:

1. Navigate to the Oracle Application Container Cloud Service Console.
2. Click the **Activity** tab.
3. By default, the Activity page shows the list of all activities started within the past 24 hours. To specify a start time range other than the default of the previous 24 hours, enter the desired time in the **Start Time Range** field.
4. To locate a specific activity, complete these fields in the **Search Activity Log** area:
   - **Start Time Range**: Select the time range to search for operations started within a specified time range. The range defaults to the previous 24 hours.
   - **Status**: Select the operation status. The default value is `All`.
   - **Service Name**: Enter a full or partial service name to filter the results by a specific service name.

- **Service Type**: Select the service type to filter the results by a specific service type. The default value is the current cloud service.
- **Operation**: Select the operation to filter the results by a specific operation type. The default value is `All`.

Then click **Search**.

5. If necessary, you can limit the maximum number of search results by specifying that number in **Results per page**.
6. When finished, click the **Services** tab.

Retrieve the Application Logs

You may want to check your application logs to monitor the application or troubleshoot a problem. Information that your application sends to stdout or stderr is captured in the logs. The logs are stored on Oracle Cloud Infrastructure Object Storage Classic.

1. Go to the **Applications** page.

   The page appears after you sign in or when you click on the Applications link at the top of any page. See How to Access Oracle Application Container Cloud Service.
2. In the applications list, click the desired application name.
3. Click the **Administration** tab.
4. Click **Logs** and then click **Get Logs**  .
5. In the **Upload Logs** window, select the instance and click **OK.**
6. Click **Refresh**  , then click the name of the zip file and save the file in your local system.

Automatically Scaling an Application

The Autoscaling Rules page lets you create rules for when to scale out, in, up, or down based on memory usage.

You can create a pair of rules for scaling out and in or a pair of rules for scaling up and down, but not both. To create an autoscaling rule:

1. On the Application Console, select **Auto Scaling** from the  Menu.

   The Autoscaling Rules page appears.
2. On the Autoscaling Rules page, click **Create Rule**.

   The New Rule dialog appears.
3. In the New Rule dialog, edit the following settings:
   - **Perform** — Select **Scale Up** to increase memory per application instance, **Scale Out** to create additional application instances, **Scale Down** to decrease memory per application instance, **Scale In** to reduce number of application instances.

- **GB Memory** — Specify the maximum or minimum amount of memory per application instance that can exist as a result of a Scale Up or Scale Down rule.
- **Cluster Size** — Specify the maximum or minimum number of application instances that can exist as a result of a Scale Out or Scale In rule.
- **Aggregation** — Select **Average**, **Maximum**, or **Minimum**.
- **Metric** — Select **Memory Utilization** for percent memory usage or **Memory** for absolute memory usage in gigabytes.
- **Threshold** — Specify the value for the aggregated metric that triggers the rule.
- **Consecutive Periods** — Specify the number of consecutive periods for which the threshold condition must be met.
- **Minutes** — Specify the length of each period in minutes.
- **Instances** — Select All for all instances or select a specific instance.
- **Cool Down** — Specify the length of time in minutes during which the rule can't be triggered again.

For example: Perform Scale Out to Maximum Cluster Size of 5 whenever Average of Memory Utilization is >= 20% for at least 2 consecutive periods of 5 minutes on All Instances and wait for 30 minutes of cool down period.

4. Click **Create**.

   The new rule is listed on the Autoscaling Rules page. Events that activate the rule appear in the **Alert History** list.

5. To return to the Application Console, select **Open Application** from the Menu.

After you create a rule, you can select **Edit**, **Delete**, or **View Metrics** from the Menu for the rule. **View Metrics** displays a list of metrics on which the rule is based.

Manage Tags

A tag is an arbitrary key or a key-value pair that you can create and assign to your Oracle Application Container Cloud Service applications. You can use tags to organize and categorize your applications, and to search for them.

Topics

Create, Assign, and Unassign Tags

To assign tags to an application or to unassign tags:

1. Navigate to the Overview page for the application for which you want to assign or unassign tags.
2. If at least one tag is assigned to the application, the Overview page shows the first tag assigned to the application. If there aren't any tags assigned to the application you see the **Click to Assign** label.

a. Hover over the tag or the **Click to Assign** label, until a **More** link is displayed.

b. Click the **More** link.

3. In the Manage Tags dialog box, assign or unassign tags, as required:

   - In the **Assign** section, from the **Tags** field, select the tags that you want to assign to the instance.
   - If the tags that you want to assign don't exist, select **Create and Assign** in the **Tags** field, and then enter the required tags in the **Enter New Tags** field.
   - To unassign a tag, in the **Unassign** section, look for the tag that you want to unassign, and click the **X** button next to the tag.
   - To exit without changing any tag assignments for the instance, click **Cancel**.

4. After assigning and unassigning tags, click **OK** for the tag assignments to take effect.

Find Tags and Applications Using Search Expressions

A tag is an arbitrary key or a key-value pair that you can create and assign to your Oracle Application Container Cloud Service applications. You can use tags to organize and categorize your applications, and to search for them. Over time, you might create dozens of tags, and you might assign one or more tags to several of your applications. To search for specific tags and to find instances that are assigned specific tags, you can use filtering expressions.

For example, on the Application page of the web console, you can search for the applications that are assigned a tag with the key `env` and any value starting with `dev` (example: `env:dev1, env:dev2`), by entering the search expression `'env':'dev%'` in the **Search** field.

3

Using Oracle Application Container Cloud Service with Other Oracle Cloud Services

Oracle Application Container Cloud Service typically integrates with other Oracle Cloud services using service bindings. The services described in these topics integrate in other ways, and you must configure the integrations differently.

Topics

Use Oracle Identity Cloud Service with Oracle Application Container Cloud Service

This feature is only available if you have Universal Cloud Credits and subscriptions to both Oracle Identity Cloud Service and Oracle Cloud Infrastructure Load Balancing Classic.

Topics

What is Oracle Identity Cloud Service?

Oracle Identity Cloud Service provides Oracle Cloud administrators with a central security platform to manage the relationships that your users have with your applications, including with other Oracle Cloud services like Oracle Application Container Cloud Service. With Oracle Identity Cloud Service you can create custom password policies and email notifications, onboard new users, assign users and groups to applications, and run security reports. See these topics in *Administering Oracle Identity Cloud Service*:

- About Oracle Identity Cloud Service Concepts
- How to Access Oracle Identity Cloud Service

Managing Service Administrators

When your Oracle Cloud account includes Oracle Identity Cloud Service, use it to create users and groups and to give them access to Oracle Application Container Cloud Service. Assign users the `APAAS APAASAdministrators` role in order to grant them rights to create and manage applications. See these topics in *Administering Oracle Identity Cloud Service*:

- Creating User Accounts

- Creating Groups
- Adding or Removing a User Account from an Administrator Role
- Assigning Users to Oracle Applications
- Assigning Groups to Oracle Applications

Managing Application Roles in Oracle Identity Cloud Service

After you deploy your application with Basic or OAuth selected, you can go to the Application Overview page and click the **Manage Access** link. This takes you to Oracle Identity Cloud Service, where you can manage users and roles with access to the application.

For each Oracle Application Container Cloud Service application with OAuth that it manages, Oracle Identity Cloud Service provides the following predefined roles:

- `App Administrators`
- `App Operators`
- `App Monitors`

The application deployer is automatically assigned the `App Administrators` role.

You can use these roles to implement custom authorization in the application code, which would understand and enforce the role-based grants. Typically, users with the `App Administrators` role have full access to the application, users with the `App Operators` role have limited access to the application, and users with the `App Monitors` role have read-only access to the application.

For example, suppose the Oracle Application Container Cloud Service application performs inventory management. The application code might permit users with the `App Administrators` role to add and delete items, while users with the `App Operators` role can only change amounts for existing items and users with the `App Monitors` role can only view items and amounts.

Deploy an Application and Configure a Database with Stack Manager

Use Oracle Cloud Stack to deploy an application to Oracle Application Container Cloud Service and configure an Oracle MySQL Cloud Service database in a single operation.

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

Stacks are created from templates. Oracle Cloud Stack includes many certified Oracle stack templates for Oracle Application Container Cloud Service. Most have names beginning with **Oracle-ACCS**.

One of the most basic stack templates for Oracle Application Container Cloud Service is Oracle-LMP, which implements the popular LAMP pattern (Linux, Apache, MySQL,

PHP), although by default Apache isn't configured. This template creates a stack that's composed of these resources:

- A service instance in Oracle MySQL Cloud Service
- An optional PHP application instance in Oracle Application Container Cloud Service that's connected to the database

The optional application can be modified to have a different runtime, such as Java or Node.js, or multiple instances.

> **Note:**
>
> Prior to creating a cloud stack from this template that includes the application, you must upload the application to Oracle Cloud Infrastructure Object Storage Classic. The template doesn't automatically create a storage container or upload the application file.

Get Started

Create a stack using the Oracle-LMP template. Refer to these topics in *Using Oracle Cloud Stack*:

- Accessing Oracle Cloud Stack
- Creating a Cloud Stack

A video and a tutorial are also available.

Video

Tutorial

Template Parameters

In the Oracle-LMP template, the values of these input parameters can be customized for each stack creation:

- MySQL database VM compute shape (CPU, memory, storage)
- MySQL database system user name and password
- Secure Shell (SSH) public key for MySQL database VM administration
- Location of your PHP application in Oracle Cloud Infrastructure Object Storage Classic
- Number of instances of your PHP application
- Memory (GB) allocated to each instance of your PHP application

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

Customize the Template

Export and update the Oracle-LMP template to customize your stack's behavior. Modify the template's name and contents, such as adding a template parameter or changing the parameters used to create the application instance. See:

- Exporting a Template in *Using Oracle Cloud Stack*
- Creating a Template
- Oracle Application Container Cloud Service *REST API for Managing Applications*
- REST API for Oracle MySQL Cloud Service

See the following for some examples of customizing this stack template.

Set Environment Variables

Your PHP application may rely on environment variables so that you can customize its behavior or tune its performance without modifying code. Use Oracle Application Container Cloud Service to set one or more environment variables (name/value pairs).

```
resources:
  phpContainer:
    type: apaas
    parameters:
      ...
      deployment:
        memory: { "Fn::Join": ["", ["Fn::GetParam": instanceMemory, G]] }
        instances: { "Fn::Join": ["", ["Fn::GetParam": numberOfInstances]]}
        environment:
          NUM_CONNECTIONS: 5
          SCHEMA_NAME: Order
```

Change the Runtime

If your application is written in a language other than PHP, then you can change the runtime environment. For example, to change the runtime to Java, make these changes to the template:

- Change all occurrences of `phpContainer` to `javaContainer`.
- Within the resource definition of `javaContainer`, change the runtime parameter to `Java`.

```
resources:
   javaContainer:
     type: apaas
     parameters:
       ...
       runtime: Java
```