

# Oracle® Cloud

## Using Oracle Cloud Stack



Release 19.3.3

E77672-24

September 2019

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Cloud Using Oracle Cloud Stack, Release 19.3.3

E77672-24

Copyright © 2016, 2019, Oracle and/or its affiliates. All rights reserved.

Primary Author: TJ Palazzolo

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

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

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

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

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

# Contents

## Preface

---

Audience	vii
Documentation Accessibility	vii
Related Resources	vii
Conventions	vii

## 1 Getting Started with Oracle Cloud Stack

---

About Oracle Cloud Stack	1-1
What is a Cloud Stack	1-1
What is a Template	1-2
About Oracle Stack Templates	1-2
About the Interfaces to Oracle Cloud Stack	1-4
Before You Begin with Oracle Cloud Stack	1-4
Accessing Oracle Cloud Stack	1-5
Accessing Cloud Stack from Infrastructure Console	1-5
Accessing Cloud Stack from the Infrastructure Classic Console	1-6
Accessing Cloud Stack from Cloud at Customer	1-7
Accessing the CLI	1-7
About Oracle Cloud Stack Roles and Users	1-8

## 2 Creating Templates with Oracle Cloud Stack

---

About Template Components	2-1
What are Parameters	2-2
What are Resources	2-3
What are Dependencies	2-4
What are Attributes	2-4
What are Conditions	2-5
What are Maps	2-5
What are Custom Actions	2-6
What are Functions	2-6
Creating a Template	2-6

Creating a Template with the Template Builder	2-7
Creating Template Parameters	2-8
Creating Parameter Groups	2-9
Creating Resources	2-10
Creating Dependencies	2-11
Adding a Resource Dependency	2-11
Creating Conditions	2-12
Defining Global Conditions	2-13
Adding a Condition to a Resource	2-14
Adding a Condition to a Resource Parameter	2-14
Adding a Condition to a Template Parameter	2-15
Creating Custom Actions	2-15
Creating a Software Component	2-16
Adding a Software Component to a Resource	2-17
Adding a Condition to a Software Component	2-17
Custom Action Best Practices	2-18
Creating Maps	2-18
Defining a Map	2-19
Using a Map in a Resource Parameter	2-19
Creating Attributes	2-20
Editing Template Files	2-21
About Template Files	2-21
Using Template Parameters	2-22
Using Resources	2-28
Using Attributes	2-30
Using Conditions	2-31
Using Custom Actions	2-34
Using Validations	2-40
Using Maps	2-44
Template Best Practices	2-45
List of Resource Types	2-46
List of Functions	2-48
List of Implicit Template Parameters	2-53

### 3 Managing Templates with Oracle Cloud Stack

---

About the Template Life Cycle	3-1
Configuring Template Storage	3-3
Viewing a Template Resource Topology	3-4
Copying a Template	3-5
Cloning an Oracle Template	3-5

Copying an Unpublished Template	3-5
Copying a Custom Published Template	3-6
Editing a Template	3-6
Editing with the Console	3-7
Editing Offline	3-7
Validating a Template	3-8
Validating a Template with the Console	3-8
Validating a Template with the CLI	3-8
Publishing a Template	3-9
Importing a Template	3-9
Importing a Published Template with the Console	3-10
Importing an Unpublished Template with the Console	3-10
Importing a Template with the CLI	3-10
Exporting a Template	3-11
Exporting a Published Template with the Console	3-12
Exporting an Unpublished Template with the Console	3-12
Exporting a Template with the CLI	3-12
Deleting a Template	3-13
Deleting a Template with the Console	3-13
Deleting a Template with the CLI	3-13

## 4 Creating and Managing Stacks with Oracle Cloud Stack

---

Typical Workflow for Managing Stacks	4-1
Creating a Cloud Stack	4-2
Creating a Cloud Stack with the Console	4-2
Creating a Cloud Stack with the CLI	4-4
Viewing the Details of a Cloud Stack	4-5
Viewing Cloud Stacks with the Console	4-5
Viewing Cloud Stacks with the CLI	4-7
Starting and Stopping a Cloud Stack	4-8
Starting and Stopping a Cloud Stack with the Console	4-8
Starting and Stopping a Cloud Stack with the CLI	4-8
Using Tags with Cloud Stacks	4-9
Creating, Assigning, and Unassigning Tags	4-9
Find Tags and Instances Using Search Expressions	4-10
Starting and Stopping a Resource in a Cloud Stack	4-13
Starting and Stopping a Resource with the Console	4-13
Managing SSH Access for a Resource in a Cloud Stack	4-13
Managing Access Rules for a Resource in a Cloud Stack	4-14
Deleting a Cloud Stack	4-15

Deleting a Cloud Stack with the Console	4-15
Deleting a Cloud Stack with the CLI	4-16

## 5 Frequently Asked Questions for Oracle Cloud Stack

## 6 Troubleshooting for Oracle Cloud Stack

Problems with Creating a Stack	6-1
Problems with Deleting a Stack	6-3
Problems with Importing a Template	6-3
Get Help for a Specific Template	6-5

## 7 Exploring the Oracle Cloud Stack Console

Welcome Page	7-1
Stacks Page	7-2
Stack Details Page	7-4
Published Templates Page	7-6
Unpublished Templates Page	7-7
Template Builder	7-8
Activity Page	7-11

# Preface

## Topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Resources](#)
- [Conventions](#)

## Audience

*Using Oracle Cloud Stack Manager* is intended for experienced users of Oracle Cloud who want to automate the provisioning of their cloud environments by using stacks.

## 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 Public Cloud](#)  
<http://cloud.oracle.com>
- *PaaS Service Manager Command Line Interface Reference*

## Conventions

The following text conventions are used in this document:

<b>Convention</b>	<b>Meaning</b>
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	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 Cloud Stack

To get started with Oracle Cloud Stack, you should become familiar with its concepts, architecture, interfaces, and prerequisites.

### Topics

- [About Oracle Cloud Stack](#)
- [About Oracle Stack Templates](#)
- [About the Interfaces to Oracle Cloud Stack](#)
- [Before You Begin with Oracle Cloud Stack](#)
- [Accessing Oracle Cloud Stack](#)
- [About Oracle Cloud Stack Roles and Users](#)

## About Oracle Cloud Stack

Oracle Cloud Stack is a feature of Oracle Cloud that empowers users to automate the provisioning of multiple cloud services as a single unit called a stack.

In order to build and deploy their applications, businesses often require sophisticated environments that consist of multiple, integrated cloud services. Consider a development environment whose needs include a Java application server along with two relational databases. Provisioning these three service instances (1 Java and 2 databases) for every member of your development team is time consuming and error prone, regardless of whether you're using service consoles or REST APIs to provision the services.

When you use Oracle Cloud Stack to create an environment, it provisions the specified Oracle Cloud resources by using each service's public REST APIs. It orchestrates the execution of these APIs in a standardized, repeatable way, ensuring that you create consistent environments in your cloud accounts or even in different cloud data centers. You can also define the dependencies between your resources, so that Oracle Cloud Stack creates and destroys the resources in your cloud stacks in a logical sequence.

Oracle Cloud Stack is supported on Oracle Cloud and on Oracle Cloud at Customer.

## What is a Cloud Stack

A *cloud stack* is a collection of related cloud services that you provision together with Oracle Cloud Stack .

It represents a complete environment whose functionality extends beyond a single service in Oracle Cloud. The services or *resources* that make up a cloud stack are created, deleted, started, and stopped together as a unit, but these services can also be individually accessed, configured and managed by using their service-specific interfaces. A cloud stack is created from a template.

See [List of Resource Types](#) for the types of cloud resources that can be included as components of a cloud stack.

## What is a Template

A *template* in Oracle Cloud Stack defines the cloud services that are part of a stack as well as how they are provisioned in Oracle Cloud. They act like blueprints for the creation of cloud environments.

You can rapidly provision similar environments, in the same cloud data center or in different ones, by creating multiple stacks from a single template. After a stack is created from a template, its lifecycle is completely independent from the template's lifecycle. In other words, modifications you make to the template will not affect existing cloud stacks.

Templates are text files that use the standard [YAML](#) syntax. You can either use the template authoring tools in the Oracle Cloud Stack console, or import template files into Oracle Cloud Stack that you authored with some other tool.

A template is comprised of several elements:

- *Resources* define the cloud services to create and the dependencies between them.
- *Template parameters* allow users of the template to customize the template for a specific cloud stack.
- *Attributes* enable you to use the runtime characteristics of one resource as parameters for the creation of another resource.
- *Conditions* change a template's behavior based on different criteria like the values of template parameters.

For more information, see [About Template Components](#).

## About Oracle Stack Templates

In order to help you get started creating cloud stacks, all Oracle Cloud accounts include templates that are ready-to-use with Oracle Cloud Stack.

These certified, Oracle-defined templates address popular use cases and deployment patterns. You do not have to import these templates into Oracle Cloud Stack to begin using them. You can also copy an Oracle template, and then customize its name and definition to meet your requirements. See [Cloning an Oracle Template](#).

The available Oracle templates include:

Name	Description	More Information
Oracle-AnalyticsCloud-Autonomous-Template	Provision Oracle Analytics Cloud on Oracle Cloud Infrastructure, and using an existing cloud storage bucket.	Creating Services in <i>Administering Oracle Analytics Cloud</i>
Oracle-IntegrationCloud-CM-Template	Provision an Oracle Integration instance, an Oracle Database Cloud Service deployment, and a cloud storage container for backups.	Creating an Oracle Integration Cloud Instance in <i>Administering Oracle Integration</i>

Name	Description	More Information
Oracle-IoT-Asset-Monitoring	Provision an Oracle IoT Asset Monitoring Cloud Service instance, along with cloud storage containers for backups and content.	Creating an Oracle IoT Asset Monitoring Cloud Service Instance in <i>Administering Oracle Internet of Things Cloud Service</i>
Oracle-IoT-Asset-Monitoring-CX	Provision an Oracle IoT Asset Monitoring CX Cloud Service instance, along with cloud storage containers for backups and content.	Creating an Oracle IoT Asset Monitoring CX Cloud Service Instance in <i>Administering Oracle Internet of Things Cloud Service</i>
Oracle-IoT-Connected-Worker	Provision an Oracle IoT Connected Worker Cloud Service instance, along with cloud storage containers for backups and content.	Creating an Oracle IoT Connected Worker Cloud Service Instance in <i>Administering Oracle Internet of Things Cloud Service</i>
Oracle-IoT-Enterprise	Provision a standard Oracle Internet of Things Cloud Service instance, along with cloud storage containers for backups and content.	Creating an Oracle Internet of Things Cloud Service Instance in <i>Administering Oracle Internet of Things Cloud Service</i>
Oracle-IoT-Enterprise-Analytics	Provision an Oracle Internet of Things Cloud Service instance with Analytics, and cloud storage containers for backups and content.	Creating an Oracle Internet of Things Cloud Service Instance with Analytics in <i>Administering Oracle Internet of Things Cloud Service</i>
Oracle-IoT-Fleet-Monitoring	Provision an Oracle IoT Fleet Monitoring Cloud Service instance, along with cloud storage containers for backups and content.	Creating an Oracle IoT Fleet Monitoring Cloud Service Instance in <i>Administering Oracle Internet of Things Cloud Service</i>
Oracle-IoT-Production-Monitoring	Provision an Oracle IoT Production Monitoring Cloud Service instance, along with cloud storage containers for backups and content.	Creating an Oracle IoT Production Monitoring Cloud Service Instance in <i>Administering Oracle Internet of Things Cloud Service</i>
Oracle-JCS-DBCS-Template	Provision an Oracle Java Cloud Service instance, an Oracle Database Cloud Service deployment, and a cloud storage container for backups.	Creating an Oracle Java Cloud Service Instance with Stack Manager in <i>Administering Oracle Java Cloud Service</i>
Oracle-LMP	Provision one or more PHP containers in Oracle Application Container Cloud Service, along with an Oracle MySQL Cloud Service instance. Optionally deploy an existing PHP application to the container.	Deploying an Application and Configuring a Database with Stack Manager in <i>Using Oracle Application Container Cloud Service</i>
Oracle-MFTCS-DBCS-Template	Provision an Oracle Managed File Transfer Cloud Service instance, an Oracle Database Cloud Service deployment, and a cloud storage container for backups.	<b>This template is not supported in a production environment. Use the template only in non-production environments or for testing.</b>
Oracle-Mobile-Bots-Standard-Rel	Provision an Oracle Mobile Cloud, Enterprise service instance that includes only the Bots component.	Create an Oracle Mobile Cloud, Enterprise Bots-Only Stack in <i>Administering Oracle Mobile Cloud Enterprise in a Customer-Managed Environment</i>

Name	Description	More Information
Oracle-Mobile-Cloud-Enterprise-Template	Provision an Oracle Mobile Cloud, Enterprise service instance, including the Mobile Core, Custom Code, Analytics, and Bots components.	Create an Oracle Mobile Cloud, Enterprise Stack in <i>Administering Oracle Mobile Cloud Enterprise in a Customer-Managed Environment</i>
Oracle-OEHCS-BDCSCE-StackTemplate	Provision an Big Data Cloud instance, including an Oracle Event Hub Cloud Service - Platform instance, two Oracle Event Hub Cloud Service instances (input and output), and a cloud storage container for backups.	Creating a Cluster with Stack Manager in <i>Using Oracle Big Data Cloud</i>
Oracle-SOACS-DBCS-Template	Provision an Oracle SOA Cloud Service instance, an Oracle Database Cloud Service deployment, and a cloud storage container for backups.	<b>This template is not supported in a production environment. Use the template only in non-production environments or for testing.</b>
Oracle-VisualBuilder-CM-Template	Provision an Oracle Visual Builder instance, an Oracle Database Cloud Service deployment, and a cloud storage container for backups.	Creating an Instance with the Oracle Cloud Stack Template in <i>Administering Oracle Visual Builder Classic (User-Managed)</i>

## About the Interfaces to Oracle Cloud Stack

All Oracle Cloud Stack operations are performed through a web console, the standard Oracle Cloud command line interface or a REST API.

Type of Access	Description	More Information
Web Console	Create and manage cloud stacks and templates from a web browser. The console also includes tools to edit template files.	<a href="#">Accessing Oracle Cloud Stack</a>
Command Line Interface (CLI)	Create and manage cloud stacks and templates from a command line or script.	<ul style="list-style-type: none"> <li><a href="#">Accessing Oracle Cloud Stack</a></li> <li><a href="#">About the PaaS Service Manager Command Line Interface</a></li> <li><a href="#">psm stack Commands</a></li> </ul>
REST API	Use REST endpoints to manage cloud stacks and templates from a terminal, script or custom program.	<a href="#">REST API for Oracle Cloud Stack</a>

## Before You Begin with Oracle Cloud Stack

Before using Oracle Cloud Stack you may have to perform certain prerequisite tasks.

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

- You must have a valid Oracle Cloud account with subscriptions to the cloud resources that you plan to provision as part of your cloud stacks. However, you do not have to explicitly request or purchase a subscription to Oracle Cloud Stack itself in order to use it.
- In order to edit templates with the Oracle Cloud Stack console, you must first create an object storage container in your cloud account, if one doesn't already exist. See *Creating Containers in Using Oracle Cloud Infrastructure Object Storage Classic*.

## Accessing Oracle Cloud Stack

Use one of the available interfaces to access Oracle Cloud Stack.

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

 [Tutorials](#) are also available to help you get started with Oracle Cloud Stack.

### Topics

- [Accessing Cloud Stack from Infrastructure Console](#)
- [Accessing Cloud Stack from the Infrastructure Classic Console](#)
- [Accessing Cloud Stack from Cloud at Customer](#)
- [Accessing the CLI](#)

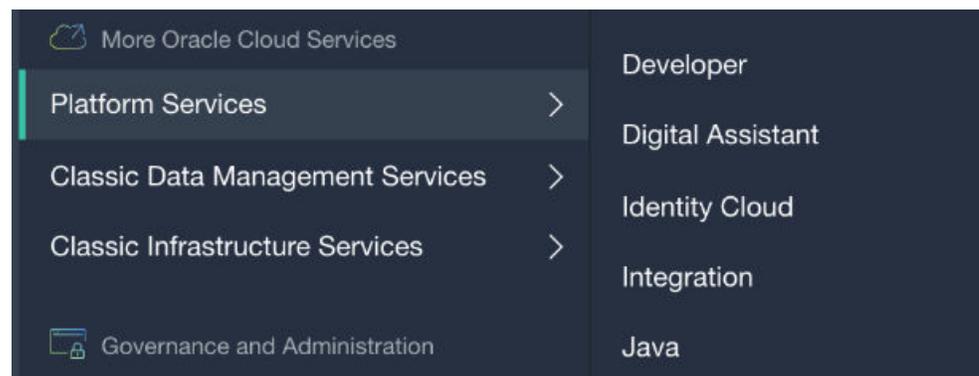
## Accessing Cloud Stack from Infrastructure Console

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

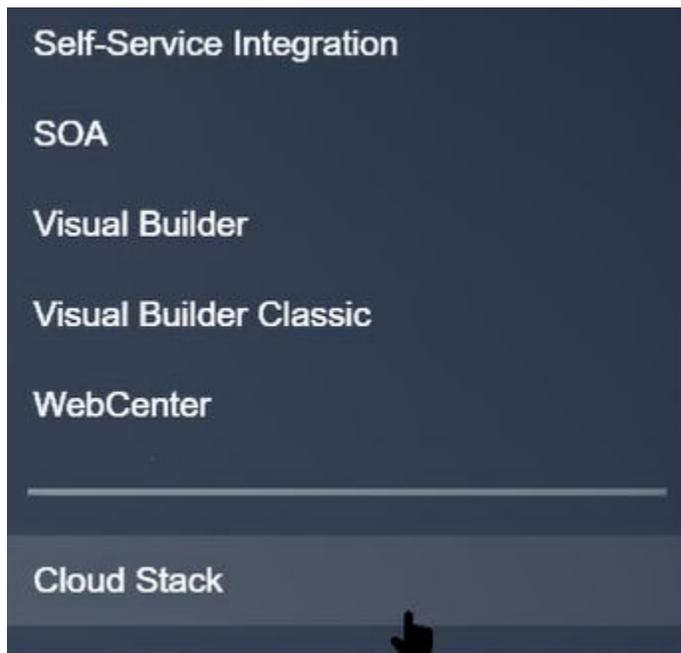
1. Sign in to Oracle Cloud.

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

2. From the Infrastructure Console, click the navigation menu  in the top left corner, expand **Platform Services**, and then select any available service.



3. From the service console, click the navigation menu  in the top left corner, and then select **Cloud Stack**.



The Oracle Cloud Stack console is displayed.

When you access the console for the first time, you'll see the [Welcome Page](#). Otherwise you'll see the [Stacks Page](#).

Click **Help**  in the top right corner of a page for links to documentation, tutorials and videos related to the features on that page. Refer to these topics to learn how to create templates and cloud stacks:

- [Creating a Template](#)
- [About the Template Life Cycle](#)
- [Typical Workflow for Managing Stacks](#)

## Accessing Cloud Stack from the Infrastructure Classic Console

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

1. Sign in to Oracle Cloud.  
If you received a welcome email, use it to identify the URL, your user name, and your temporary password. After signing in, you will be prompted to change your password.
2. From the Infrastructure Classic Console, click the navigation menu  in the top left corner, and then click the name of any available service.



3. From the service console, click the navigation menu  in the top left corner, and then select **Cloud Stack**.

The Oracle Cloud Stack console is displayed.

When you access the console for the first time, you'll see the [Welcome Page](#).  
Otherwise you'll see the [Stacks Page](#).

## Accessing Cloud Stack from Cloud at Customer

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

1. Sign in to Oracle Cloud at Customer.
2. From the My Services Dashboard, click the navigation menu  in the top left corner, and then click the name of any available service.



3. From the service console, click the navigation menu  in the top left corner, and then select **Cloud Stack**.

The Oracle Cloud Stack console is displayed.

When you access the console for the first time, you'll see the [Welcome Page](#).  
Otherwise you'll see the [Stacks Page](#).

## Accessing the CLI

After installing and configuring the Oracle Cloud command line interface (CLI), you can use it to send commands to Oracle Cloud Stack

1. If you have not yet installed the CLI, perform the tasks described in *PaaS Service Manager (PSM) CLI Reference*:
  - a. Performing Prerequisite Tasks for the Command Line Interface
  - b. Downloading the Command Line Interface
  - c. Installing the Command Line Interface
  - d. Configuring the Command Line Interface
2. Execute this CLI command:

```
psm help
```

3. Use the output to verify that the Oracle Cloud Stack commands (`stack`) are available to this account. Similarly, note the other cloud services that are available to this account.

For example:

```
psm help (or psm h)
```

```
o dbcs
  Oracle Database Cloud Service
```

- o `jcs`  
Oracle Java Cloud Service
- o `accs`  
Oracle Application Container Cloud Service
- o **stack**  
Cloud Stack Manager

 **Note:**

If the `stack` commands are not present, update your CLI installation to the latest version. See *Updating the Command Line Interface in PaaS Service Manager (PSM) CLI Reference*.

4. Refer to these topics to learn how to use the CLI to manage templates and create cloud stacks:
  - [About the Template Life Cycle](#)
  - [Typical Workflow for Managing Stacks](#)

## About Oracle Cloud Stack Roles and Users

Unlike other Oracle Cloud services, there are no user roles specific to using Oracle Cloud Stack.

Any cloud user in your identity domain can view and manage stack templates. However, in order to create a cloud stack from a template the user must have the necessary roles to create the specific resources in the stack. For example, in order to create an Oracle Java Cloud Service instance as a resource in a stack, you must have the Java Administrator role.

Only the Identity Domain Administrator is allowed to create user accounts and assign roles. See *Managing Users, User Accounts, and Roles in Managing and Monitoring Oracle Cloud*.

# 2

## Creating Templates with Oracle Cloud Stack

Templates in Oracle Cloud Stack provide a blueprint for the creation of cloud stacks, and orchestrate the creation of cloud services, or resources. Creating a template involves working with parameters, attributes, conditions, and functions.

Templates are text files that use the standard [YAML](#) syntax. You can either use the template authoring tools found in the Oracle Cloud Stack console, or edit template files offline with some other tool. After creating a template, you can publish and test it. See [About the Template Life Cycle](#).

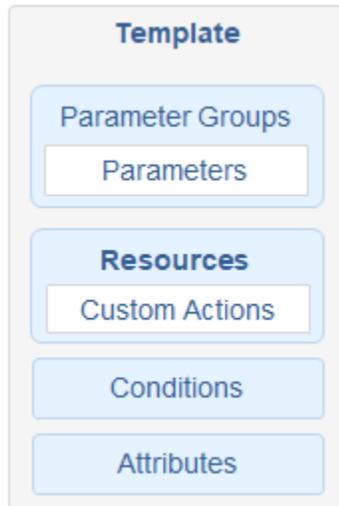
An alternative to creating a new template is to copy an existing one. See [Copying a Template](#).

### Topics

- [About Template Components](#)
- [Creating a Template](#)
- [Editing Template Files](#)
- [Template Best Practices](#)
- [List of Resource Types](#)
- [List of Functions](#)
- [List of Implicit Template Parameters](#)

## About Template Components

A template in Oracle Cloud Stack is a YAML document that's comprised of several components, including parameters, resources, attributes, and conditions.



### Topics

- [What are Parameters](#)
- [What are Resources](#)
- [What are Dependencies](#)
- [What are Attributes](#)
- [What are Conditions](#)
- [What are Maps](#)
- [What are Custom Actions](#)
- [What are Functions](#)

## What are Parameters

Template parameters define the inputs that users of this template can provide to Oracle Cloud Stack when they create a cloud stack from the template.

Parameters make your templates more customizable and reusable. For example, a template's parameters might include inputs like:

- `cloudUserName`
- `databasePassword`
- `serverComputeShape`

The definition for a template parameter includes its name, data type (text, number, boolean, SSH key, and so on), and whether the parameter is mandatory or optional. If a parameter is optional, you can also define a default value if one is not provided by the user. Template parameters also support various constraints that are validated at runtime, such as the allowed values, minimum value, and maximum value.

The values of these input parameters are referenced in other parts of the template by using the `GetParam` function.

You can optionally group related template parameters together. Each parameter group is given a label, which is used as a heading in the Details page of the Oracle Cloud

Stack creation wizard. The image below depicts a parameter group named *Oracle Database Cloud Service Details*. This group consists of four template parameters:

### Oracle Database Cloud Service Details

DBCS Version	<input type="text" value="12.1.0.2"/>	?
Compute Shape	<input type="text" value="oc3"/>	?
* DB Name (SID)	<input type="text" value="ORCL"/>	?
Usable Database Storage (GB)	<input type="text" value="25"/>	?

If you don't assign a parameter to a group, it will be displayed under a generic *Configuration* heading in the creation wizard. Standard parameters that are displayed for all templates, including **Name** and **Description**, are always displayed under the *Stack Details* heading.

#### Note:

Parameter groups have no effect on the CLI or REST API.

## What are Resources

Oracle Cloud Stack provisions one or more Oracle Cloud resources when a cloud stack is created from the template.

For a list of the resource types supported by Oracle Cloud Stack see [List of Resource Types](#).

Each resource in a template has a name and parameters, which provide the specific names and values that Oracle Cloud Stack sends to the REST endpoint for creating this type of cloud resource. For example, the REST endpoint for creating an instance of Oracle Java Cloud Service supports parameters like:

- `serviceName`
- `cloudStorageContainer`
- `backupDestination`

As a template author, you assign values to these resource parameters. The value can be typed directly into the template, or it can be determined dynamically through the use of functions. For instance, you can use the `GetParam` function to retrieve a user-supplied template parameter.

Many REST endpoints in Oracle Cloud require a request payload in JSON format, which organizes parameters into a hierarchical structure. Consequently, resource parameters in Oracle Cloud Stack are also defined as a hierarchy, with parent

parameters and child parameters. For example, Oracle Java Cloud Service supports a JSON structure like the following:

```
serviceName: myjcs
components: {
  WLS: {
    adminUserName: myadmin
    managedServerCount: 3
  }
}
```

When a stack is created, Oracle Cloud Stack uses your resource definitions in the template to automatically construct these JSON payloads.

## What are Dependencies

A template contains one or more resources. Dependencies control the order in which Oracle Cloud Stack provisions these resources when a new cloud stack is created from a template.

As a template author, you can define *explicit* dependencies between resources. For example, a template can explicitly state that the creation of `ResourceC` depends on the creation of `ResourceA` and `ResourceB`. In other words, when Oracle Cloud Stack executes this template it must successfully create `ResourceA` and `ResourceB` before creating `ResourceC`, regardless of the order in which these resources are defined in the template.

Oracle Cloud Stack will also automatically identify and honor any *implicit* dependencies between resources in a template, even if the author does not explicitly define them. This occurs when a template states that the value of a resource's parameter is the runtime attribute of another resource. For example, consider a template in which the `databaseURL` parameter of `ResourceB` is configured to use the `url` attribute of `ResourceA`. Because Oracle Cloud Stack cannot retrieve this `url` attribute until after it successfully creates `ResourceA`, there is an implicit dependency between these two resources. `ResourceA` must be created prior to creating `ResourceB`.

To optimize performance, Oracle Cloud Stack creates multiple resources in parallel if no dependencies exist between them.

## What are Attributes

The values of some resource parameters may not be known in advance because they are generated as part of provisioning a resource in the cloud. Therefore, these values must be retrieved at runtime by Oracle Cloud Stack after a resource has been successfully created.

Consider a scenario in which a stack template includes an application resource and a database resource. The application resource requires a parameter `dbURL`, which is the URL of a running database resource. The database resource's `connectionString` attribute has this information. Therefore, when configuring the application resource, you can set its `dbURL` to the value of the database's `connectionString` attribute.

Similar to creating a resource, Oracle Cloud Stack uses a REST endpoint to collect resource attributes. For example, the [REST endpoint](#) for viewing an instance of Oracle Database Cloud Service responds with attributes like:

- `service_name`
- `num_nodes`
- `listenerPort`

Use the `GetAtt` function in your template to retrieve the value of a resource's attribute. The response payload of these REST endpoints are often in JSON format, which organizes the attributes as a hierarchy like the following example:

```
{
  "serviceId":1104,
  "serviceName":"MyDatabase",
  "cloudStorageContainer":
  {
    "value":"MyStorageURL",
  }
}
```

In order to retrieve attributes within this JSON hierarchy, templates use a dot notation. For example: `cloudStorageContainer.value`

When you monitor running stacks from the CLI or REST API, Oracle Cloud Stack returns a set of standard runtime attributes for each stack, such as who created the stack. But each template can also define custom stack attributes. For example, you can define a stack attribute named `database` that maps to a database's `connectionString` attribute.

## What are Conditions

Stack templates include features that allow you to customize how a stack is created under different conditions.

Working with conditions in Oracle Cloud Stack is similar to working with standard programming constructs like `If/Then/Else`. You can define conditional expressions in a template and then refer to these conditions within other template elements:

- Display a template parameter to the user only if the specified conditions are true.
- Provision a resource only if the specified conditions are true.

Consider the following scenario:

- If the template parameter `production` is true, then set a resource's `clusterSize` to 3.
- If the template parameter `production` is false, then do not configure a resource's `clusterSize`.

You can also use conditional functions like `If`, `Equals`, `Not`, and `GreaterThan` to further control the configuration of template parameters and resources.

## What are Maps

Maps are data structures that allow you to define static values within a template. You refer to these values within other template components such as resources.

You organize the data within a map as follows:

- A map contains one or more submaps.
- A submap contains one or more key-value pairs.

For example, you could use a map to define:

- The compute shapes to use in development, test, and production stacks
- The user names for the resources in a stack

```
mymap:  
  shapes: { "dev": "oc3", "test": "oc4", "prod": "oc5" }  
  users: { "db": "mydbuser", "java": "myjavauser" }
```

## What are Custom Actions

Use Oracle Cloud Stack to execute custom operating system (OS) commands and scripts after the creation of a resource in your template.

For example, you can use custom actions to:

- Install an OS package
- Modify the default OS configuration
- Create a database schema
- Deploy an application
- Download files from cloud storage

Custom actions are only supported for Oracle Cloud services that provide access to their administration node (virtual machine) through a Secure Shell (SSH) interface. Custom actions are not supported on Oracle-managed or Autonomous cloud services.

Custom actions are defined within a special type of resource called a Software Component. These Software Components and their actions are assigned to other cloud resources in the same template.

## What are Functions

Oracle Cloud Stack provides template authors with many functions to help validate, evaluate, and manipulate values.

These functions are executed at runtime by Oracle Cloud Stack when a stack is being created. Some basic functions include:

- `GetParam` — Retrieve the value of a template parameter.
- `GetAtt` — Retrieve the value of the runtime attribute from a specific resource.
- `If` — Evaluate a condition in this template and return different values if the condition evaluates to true or false.

See [List of Functions](#).

## Creating a Template

Use the Template Builder in Oracle Cloud Stack to define template parameters, resources, and conditions.



[Tutorials](#) are also available to help you create your first template.

### Template Tasks

- **Task 1:** [Creating a Template with the Template Builder](#)
- **Task 2:** [Creating Template Parameters](#)
- **Task 3:** [Creating Parameter Groups](#)
- **Task 4:** [Creating Resources](#)
- **Task 5:** [Creating Dependencies](#)
- **Task 6:** [Creating Conditions](#)
- **Task 7:** [Creating Custom Actions](#)
- **Task 8:** [Creating Maps](#)
- **Task 9:** [Creating Attributes](#)

## Creating a Template with the Template Builder

Create and edit templates from the Oracle Cloud Stack web console.

A template in Oracle Cloud Stack is in one of two states:

- **Published** — Users in this cloud account can create stacks from this template.
- **Unpublished** — This template is still in-progress, and is not available for stack creation.

An alternative to creating a new template is to copy an existing one. See [Copying a Template](#).

Before creating your first template, configure the storage location for unpublished files. See [Configuring Template Storage](#).

1. Access the Oracle Cloud Stack console, and then click **Templates**.
2. Click **Unpublished**.
3. Click **Add Template**, and then select **Create new Template**.  
The Template Builder is displayed.
4. In the text field at the top of the page, enter a name for this template.
5. Optional: Click , and then enter a **templateVersion** and **templateDescription**.
6. Click **Save the current template** .

Oracle Cloud Stack automatically saves your template as you make changes, but as a precaution Oracle recommends that you routinely click **Save the current template**.

After creating a template, see [Creating Template Parameters](#).

To learn more about the user interface, see [Template Builder](#).

## Creating Template Parameters

Use parameters to define user inputs for your Oracle Cloud Stack template.

For an introduction, see [What are Parameters](#).

Some parameters are predefined and available to all templates. You do not have to explicitly define them. See [List of Implicit Template Parameters](#).



### Tip:

To define constraints on the stack's name, create a template parameter named `serviceName` of type `String`.

1. Access the Oracle Cloud Stack console, and then edit your template.
2. To the right of **Parameters**, click **Add parameter** .
3. Enter a name for the parameter.
4. Select a parameter **type**:

Parameter Type	Description
<b>String</b>	Users can enter any text value.
<b>Number</b>	Users can enter only a numeric value.
<b>Boolean</b>	Users can select either true or false (check box).
<b>Password</b>	Users can enter any text value, but this value is not displayed in the Oracle Cloud Stack console.
<b>Ssh</b>	Users can supply a Secure Shell (SSH) public key. Oracle Cloud Stack console users can supply the key value as text or as a file, or let Oracle Cloud generate a value dynamically. If the key is generated, users must download a file containing the public key and its corresponding private key.
<b>ComputeShape</b>	Users can select from a list of standard Oracle Cloud compute shapes ( <code>oc3</code> , <code>oc2m</code> , and so on). A shape specifies the number of Oracle Compute Units (OCPU) and the amount of memory allocated to a cloud resource.
<b>Storage</b>	Users can specify a storage container in Oracle Cloud Infrastructure Object Storage Classic.
<b>Region</b>	Users can select from a list of Oracle Cloud regions, such as a geographical locations. The available regions vary depending on the user's account settings. The default value is "No Preference", in which case Oracle Cloud automatically selects a region.
<b>File</b>	Users can select a file to upload from their local machine.
<b>Service_Type</b>	Users can select from a list of existing instances of cloud resource <code>Type</code> . See <a href="#">List of Resource Types</a> .

5. Edit the attributes and constraints of the parameter:  
The available attributes and constraints vary for each parameter type.

Field	Description
<b>default</b>	The value of this parameter if none is provided
<b>description</b>	A description of the parameter. This description is displayed if the user clicks the help icon.
<b>label</b>	The parameter's display name.
<b>mandatory</b>	(true/false) Whether or not this parameter is mandatory or optional
<b>readOnly</b>	(true/false) Whether or not the default value of this parameter can be modified
<b>sensitive</b>	(true/false) Whether or not this parameter's value should be displayed to users
<b>allowedPattern</b>	The parameter's value must match this regular expression, such as "[a-zA-Z]*"
<b>allowedValues</b>	A list of value options. Use one of these formats: <ul style="list-style-type: none"> <li>• A comma-separated list of values</li> <li>• A list of key:value pairs. The values are displayed to the user and the <code>GetParam</code> function returns the selected key.</li> </ul>
<b>constraintDescription</b>	A custom message that is displayed to the user if the parameter's constraints are not met
<b>maxLength</b>	The maximum number of characters
<b>minLength</b>	The minimum number of characters
<b>maxValue</b>	The minimum numeric value
<b>minValue</b>	The maximum numeric value

6. Click **Apply Changes**.
7. Repeat from step 2 for each additional template parameter that you want to create.
8. Click **Save the current template** .

If you delete an existing parameter, Oracle Cloud Stack does not delete any references to the parameter in the rest of the template (`Fn::GetParam: paramName`). Your template will not have valid syntax until you remove these parameter references.

After creating parameters in a template, see:

- [Creating Parameter Groups](#) (optional)
- [Creating Resources](#)

## Creating Parameter Groups

Use parameter groups in a template to control how parameters are presented to users in the Oracle Cloud Stack console.

Each group is given a label, which is used as a heading in the Details page of the Oracle Cloud Stack creation wizard. For more information, see [What are Parameters](#).

1. Access the Oracle Cloud Stack console, and then edit your template.
2. To the right of **Parameter Groups**, click **Add parameter group** .
3. Enter a name for the parameter group.
4. To the right of each parameter that you want to add to this group, click **Add parameter** .

- To remove a parameter from the group, click **Remove parameter** .
5. Click **Apply Changes**.
  6. Repeat from step 2 for each additional parameter group that you want to create.
  7. Click **Save the current template** .

After creating parameters and groups, see [Creating Resources](#).

## Creating Resources

Configure the cloud resources that Oracle Cloud Stack provisions when your template is used to create a stack.

For an introduction, see [What are Resources](#).

The Template Builder in Oracle Cloud Stack provides a canvas to help you visualize and configure the resources in your template. It also provides an interface to help construct expressions and to invoke functions.

1. Access the Oracle Cloud Stack console, and then edit your template.
2. To the left of **Resources**, click .  
The console displays a list of resource types that are available to your cloud account.
3. Identify the resource type that you want to add to the template. If necessary, use the **Search** field above **Resources**.
4. Perform one of the following actions:
  - Drag and drop the resource onto the canvas on the right side of the console.
  - To the right of the resource, click **Add resource to canvas** .
5. Either click the resource on the canvas, or right-click it and select **Edit**.
6. Using the text field at the top of the dialog, update the identifier for the resource.
7. Edit the parameter values for this resource. Click the text area to the right of the parameter name, and then choose from one of the following options:
  - Enter a static value, such as `myusername` or `10`.
  - Use the `GetParam` function to assign the value of a template parameter. For example, `'Fn::GetParam': dbPassword`
  - Use the `GetAtt` function to assign the value of another resource's attribute. Enter the resource name and the attribute name as a YAML sequence. For example, `'Fn::GetAtt': [dbresource, connectionString]`

Refer to the REST API documentation for this type of cloud resource in order to identify its attributes. See [List of Resource Types](#).

**Tip:**

Place your cursor over a parameter's name to view its description.

All required parameters for this resource type are displayed in **bold** and have the text *(required)* beside their names. All other parameters are optional. To hide all optional parameters, select **Show Only Required**.

You can search for specific parameters by name.

To configure a parameter whose value is an array, to the right of the parameter name, click **Add a parameter to this array** .

8. Click **Apply Changes**.
9. Repeat from step 2 for each additional resource that you want to create in this template.
10. Click **Save the current template** .

If you are ready to use your template, see [Publishing a Template](#).

See these topics for more template features:

- [Creating Dependencies](#)
- [Creating Conditions](#)
- [Creating Attributes](#)
- [List of Functions](#)

## Creating Dependencies

Use dependencies in Oracle Cloud Stack to modify the order in which cloud resources are created within a stack.

For an introduction, see [What are Dependencies](#).

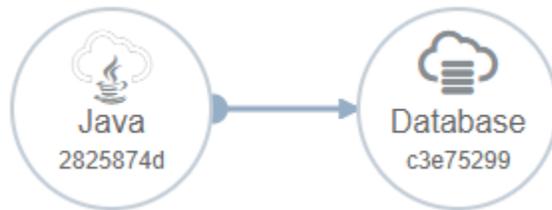
### Topics

- [Adding a Resource Dependency](#)

## Adding a Resource Dependency

Control the order in which Oracle Cloud Stack provisions the resources in your template.

Use the canvas in Template Builder to visually define the relationships between your resources. In the example below, the Java resource depends on the Database resource.



Oracle Cloud Stack will also automatically identify and honor any implicit dependencies between resources in a template, even if you don't explicitly define them. This occurs when you set the value of a resource's parameter to the runtime attribute of another resource in the template.

1. Access the Oracle Cloud Stack console, and then edit your template.
2. On the canvas, place your cursor over the resource on which you want to create a dependency.

A circle appears on the edge of the resource.

3. Click and drag an arrow to another resource on the canvas.
4. Repeat from step 2 if the resource has additional dependencies.

5. Click **Save the current template** .

To delete a dependency from the canvas, place your mouse over the arrow, and then

click .

As an alternative to using the canvas, you can also add and delete dependencies when editing a resource's configuration.

## Creating Conditions

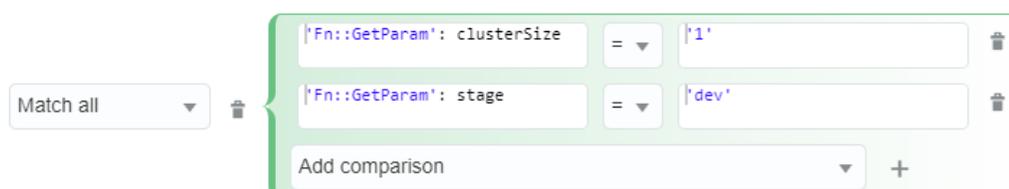
Create and use conditions within your Oracle Cloud Stack template in order to change how the template behaves for different scenarios.

For an introduction, see [What are Conditions](#).

A conditional expression includes at least one **comparison**, which compares two arguments by using a comparison operator (equals, greater than, and so on). Each argument in a comparison can be a literal value or you can use functions like `GetParam`. A comparison evaluates to true or false.

A **filter** in a conditional expression combines the results of *two or more comparisons*. A filter can evaluate if any, all, or none of the comparisons are true.

The following image depicts a filter that contains two comparisons:



For more complex conditional expressions, you can also add filters to a filter.

### Topics

- [Defining Global Conditions](#)
- [Adding a Condition to a Resource](#)
- [Adding a Condition to a Resource Parameter](#)
- [Adding a Condition to a Template Parameter](#)

## Defining Global Conditions

If you need to use the same conditional expression at multiple locations in your Oracle Cloud Stack template, create a condition at the template level. You can then refer to these global conditions throughout the template.

1. Access the Oracle Cloud Stack console, and then edit your template.
2. To the right of **Conditions**, click **Add condition** .
3. Enter a name for the condition.
4. If your condition requires a *single comparison*, perform these steps:
  - a. Select **Add comparison**, and then click **Add condition expression** .
  - b. Enter two expressions and select a comparison operator.
5. If your condition requires *two or more comparisons*, perform these steps:
  - a. Select **Add filter**, and then click **Add condition expression** .
  - b. In the second input field, select **Add comparison**, and then click **Add condition expression** .
  - c. For the first comparison, enter two expressions and select a comparison operator.
  - d. Below the first comparison, verify that **Add comparison** is selected, and then click **Add condition expression** .
  - e. For the second comparison, enter two expressions and select a comparison operator.
  - f. Repeat this process for each comparison you want to add to this condition.
  - g. Select a logical operator to evaluate the results of the comparisons: **Match all**, **Match any**, or **Match none**.

#### **Tip:**

If you prefer to enter a conditional expression manually, select the **Add value** option. You can use any combination of functions in the expression, but the result must return either true or false. See [List of Functions](#).

6. Click **Apply Changes**.
7. Repeat from step 2 for each additional condition that you want to create.

8. Click **Save the current template** .

When editing an existing condition:

- Click **Add condition expression**  to add filters or comparisons.
- Click **Delete condition expression**  to delete filters or comparisons.

## Adding a Condition to a Resource

Assign a global condition to a resource in your Oracle Cloud Stack template.

If this condition evaluates to true, the resource will be created as part of the stack. If it evaluates to false the resource (and any resources on which it depends) will not be created.

1. Access the Oracle Cloud Stack console, and then edit your template.
2. On the canvas, right click the resource to which you want to add a condition.
3. Select **Assign ConditionName**.

Resources on the canvas that have been assigned a condition are identified with the  icon.

4. Click **Save the current template** .

As an alternative to using the canvas, you can also create, assign, and remove conditions when editing a resource's configuration.

## Adding a Condition to a Resource Parameter

Use global conditions to dynamically change a resource's configuration in an Oracle Cloud Stack template.

For example, you can set a resource's `computeShape` to the value `oc3` if a condition is true, or to the value `oc4` if the condition is false.

1. Access the Oracle Cloud Stack console, and then edit your template.
2. On the canvas, right-click the resource you want to modify, and then select **Edit**.
3. Click the value of the parameter to edit it.
4. Use the `If` function. Enter the name of the global condition along with two string values as a YAML sequence.

Format: `'Fn::If': [conditionName,trueValue,falseValue]`

For example: `'Fn::If': [mycondition,oc3,oc4]`

If the specified condition evaluates to true, the function returns the first value. Otherwise, the function returns the second value.

5. Click **Apply Changes**.
6. Click **Save the current template** .

 **Tip:**

A YAML sequence can also be expressed with the following syntax:

```
'Fn::If':  
  - conditionName  
  - trueValue  
  - falseValue
```

## Adding a Condition to a Template Parameter

Assign a global condition to a parameter or parameter group in your Oracle Cloud Stack template.

If the condition evaluates to false, the parameter or group is hidden in the Oracle Cloud Stack console when a user creates a stack from this template. If it evaluates to true, the parameter or group is displayed to the user. These conditions are dynamically reevaluated each time the user modifies the value of a template parameter.

1. Access the Oracle Cloud Stack console, and then edit your template.
2. To the left of **Parameters** or **Parameter Groups**, click .
3. To the right of the parameter or group that you want to modify, click **Edit** .
4. From the **Conditions** field, select an existing condition.
5. Click **Apply Changes**.
6. Click **Save the current template** .

## Creating Custom Actions



This topic does not apply to Oracle Cloud at Customer.

Use Software Components in your Oracle Cloud Stack template to execute custom operating system (OS) commands or scripts after a stack resource is provisioned.

For example, you can use custom actions to automatically install an OS package, create a database schema, or deploy an application.

For an introduction, see [What are Custom Actions](#).

### Topics

- [Creating a Software Component](#)
- [Adding a Software Component to a Resource](#)
- [Adding a Condition to a Software Component](#)
- [Custom Action Best Practices](#)

## Creating a Software Component

Define custom actions in your Oracle Cloud Stack template within a special type of resource called a software component.

1. Access the Oracle Cloud Stack console, and then edit your template.
2. To the right of **Software components**, click **Add software component** .
3. Enter a name for the software component.
4. Enter a **name** for the first configuration in this software component.
5. Set the **actions** to **CREATE**. All custom actions run after the target cloud resource is created.
6. For **config**, enter one or more Bash OS commands as a Base64 encoded string. Use the following YAML structure:

```
Fn::Base64:
  Fn::Join:
    - "\n"
    -
      - bash_command
      - bash_command
```

For example:

```
'Fn::Base64':
  'Fn::Join':
    - "\n"
    -
      - touch ~/stack.out
      - echo 'Updating hosts' >> ~/stack.out
      - echo 'myhost 192.168.1.10' >> /etc/hosts
```

### Tip:

Be sure that all members of the same sequence have the same indentation. The following example is invalid:

```
'Fn::Base64':
  'Fn::Join':
    - "\n"
    -
      - This is
      - not valid
```

7. For **runAsUser**, select an OS user. Oracle Cloud Stack will run the commands in this configuration as the selected user.

8. Enter a **timeout** in minutes. If this configuration does not complete after the specified time period, it will fail.
9. Set **ContinueOnFailure** to **true** if stack creation should continue if this configuration fails or times out (a warning is added to the activity log).
10. For each additional configuration that you want to create in the software component, click **Add another software component configuration** , and then repeat from step 4.
11. If the OS commands in any of your configurations utilize environment variables, then define an input for each environment variable:
  - a. Click **Add another software component input** .
  - b. For **name**, enter the name of the environment variable.  
For example, if a configuration includes `${dbPassword}`, then enter `dbPassword`.
12. Click **Apply changes**.
13. Repeat from step 2 for each additional software component that you want to create.
14. Click **Save the current template** .

## Adding a Software Component to a Resource

Configure a template resource so that Oracle Cloud Stack executes custom actions after provisioning the resource.

1. Access the Oracle Cloud Stack console, and then edit your template.
2. From the canvas area, right-click a resource and select **Edit**.
3. To the right of **Software component**, click **Add software component** .
4. Select the name of an existing software component.
5. If the selected software component defines one or more inputs, enter values for these input fields.  
You can enter static text, or use template functions like `GetParam`.
6. Click **Apply changes**.
7. Click **Save the current template** .

## Adding a Condition to a Software Component

Assign a global condition to a software component in your Oracle Cloud Stack template.

If the condition evaluates to false during the creation of a stack, the custom actions in this software component will not be executed. See [Defining Global Conditions](#).

1. Access the Oracle Cloud Stack console, and then edit your template.
2. To the left of **Software components**, click .

3. To the right of the software component that you want to modify, click **Edit** .
4. From the **Conditions** field, select an existing condition.
5. Click **Apply Changes**.
6. Click **Save the current template** .

## Custom Action Best Practices

Consider these best practices when creating Software Components in an Oracle Cloud Stack template.

- The use of template functions, such as `GetParam`, is not supported within the OS commands of a custom action.
- Program a custom action so that it can run multiple times without failing or behaving unexpectedly (idempotent). A custom action might be run more than once in the following scenarios:
  - Oracle Cloud Stack retries the execution of a custom action after a communication failure with the target resource.
  - If you create a cloud stack with the `RETAIN` option, and then resume the stack after a resource provisioning failure, Oracle Cloud Stack also (re)executes any custom actions that are associated with the failed resource.
- In a custom action, return 0 to indicate success and return 1 to indicate a failure.
- By default, all existing resources are terminated if there is a failure during the creation of a stack, which includes the deletion of any log files generated by a custom action. To facilitate troubleshooting, record log messages to an external location like cloud storage.
- You can use `wget`, `curl`, or similar OS commands to download scripts and other files to the administration node in the target cloud resource. For example, you could download and extract an archive file found in an Oracle Cloud Infrastructure Object Storage container, and then execute one or more scripts found in the archive.

## Creating Maps



This topic does not apply to Oracle Cloud at Customer.

Use maps in your Oracle Cloud Stack template to define static values, and then refer to these values in other template components such as resources.

A map contains one or more submaps. A submap contains one or more key-value pairs.

For an introduction, see [What are Maps](#).

### Topics

- [Defining a Map](#)
- [Using a Map in a Resource Parameter](#)

## Defining a Map

Define a map in your Oracle Cloud Stack template.

1. Access the Oracle Cloud Stack console, and then edit your template.
2. To the right of **Mappings**, click **Add map** .
3. Enter a name for the map.
4. Enter a name for the first submap.
5. To the right of the first submap, click **Add submap key/value pair** .
6. Enter a **Key** and a **Value**.
7. Repeat from step 5 for each additional key/value pair that you want to create in the submap.
8. Repeat from step 4 for each additional submap that you want to create in this map.  
If you need to create more than 4 submaps in this map, click **Add more submap rows** .
9. Click **Apply Changes**.
10. Click **Save the current template** .

## Using a Map in a Resource Parameter

Use values defined in a map to configure resources in an Oracle Cloud Stack template.

1. Access the Oracle Cloud Stack console, and then edit your template.
2. On the canvas, right-click the resource you want to modify, and then select **Edit**.
3. Click the value of the parameter to edit it.
4. Use the `FindInMap` function. Enter the map name, submap name, and key name as a YAML sequence.

Format:

```
'Fn::FindInMap':  
  - mapName  
  - submapName  
  - keyName
```

For example:

```
'Fn::FindInMap':  
  - shapes  
  - dev  
  - DB
```

Any of the function parameters can be determined programmatically. For example:

```
'Fn::FindInMap':  
  - shapes  
  - 'Fn::GetParam': appState  
  - DB
```

5. Click **Apply Changes**.
6. Click **Save the current template** .

 **Tip:**

A YAML sequence can also be expressed with the following syntax:

```
'Fn::FindInMap': [mapName, submapName, keyName]
```

## Creating Attributes

Identify the key runtime attributes that Oracle Cloud Stack should monitor on stacks that are created from your template.

For an introduction, see [What are Attributes](#).

1. Access the Oracle Cloud Stack console, and then edit your template.
2. To the right of **Attributes**, click **Add attribute** .
3. Enter a name for the attribute.
4. Optional: Enter a **label** and a **description** for the attribute.
5. Edit the **value**, and use the `GetAtt` function to retrieve the value of a resource's attribute. Enter the resource name and the attribute name as a YAML sequence.

Format: `'Fn::GetAtt': [resourceName, attribute]`

For example: `'Fn::GetAtt': [dbresource, connectionString]`

Refer to the REST API documentation for this type of cloud resource in order to identity its available attributes. See [List of Resource Types](#).

The `attribute` argument supports a dot notation in order to access nested attributes within a JSON response payload. For example, `components.mysql.attributes.CONNECT_STRING.value`.

6. Click **Apply Changes**.
7. Repeat from step 2 for each additional attribute that you want to create.
8. Click **Save the current template** .

## Editing Template Files

Advanced users in Oracle Cloud Stack may prefer to directly edit a template's source file. Use YAML to define and edit parameters and resources, and to use functions for tasks like retrieving parameter values or evaluating conditions.

You can edit a template file using one of these methods:

- Use the Oracle Cloud Stack console to edit the template, and then click **Open YAML editor**  .
- Export the template file from Oracle Cloud Stack, and then edit the file offline. See [Exporting a Template](#).

### Topics

- [About Template Files](#)
- [Using Template Parameters](#)
- [Using Resources](#)
- [Using Attributes](#)
- [Using Conditions](#)
- [Using Custom Actions](#)
- [Using Validations](#)
- [Using Maps](#)

## About Template Files

A template in Oracle Cloud Stack is a YAML file that's comprised of several components, including parameters, resources, attributes, and conditions.

Files that contain [YAML](#) documents can use any file extension but `.yaml` is a typical convention. The first line in the file identifies it as a YAML document by using 3 dashes:

```
---
```

In the template's definition provide its name, version and description:

```
---
```

```
template:
  templateName: MyTemplateName
  templateVersion: 1.0
  templateDescription: My Oracle cloud stack template
```

The values of `templateName` and `templateVersion` are used to uniquely identify a template in Oracle Cloud Stack. The name of the template source file is not used to identify a template. Two templates can have identical names if they have different versions.

Template documents have the following general structure:

```
Template
  Parameters (Optional)
  Parameter Groups (Optional)
  Attributes (Optional)
  Conditions (Optional)
  Resources
```

YAML supports two different methods of expressing sequences (collections of items). You can write a YAML sequence by using multiple lines and dashes, as shown in the following example:

```
depends_on:
  - HRStorage
  - HRDatabase
```

Alternatively, you can write a YAML sequence as an array by using square brackets. For example:

```
depends_on: [HRStorage,HRDatabase]
```

## Using Template Parameters

Edit your Oracle Cloud Stack template file and configure user inputs by using parameters.

For an introduction, see [What are Parameters](#).

Some parameters are predefined and available to all templates. You do not have to explicitly define them. See [List of Implicit Template Parameters](#).

### Topics

- [Basic Syntax](#)
- [Stack Name Parameter](#)
- [Parameter Groups](#)
- [Hidden Parameters](#)
- [Password Parameters](#)
- [SSH Parameters](#)
- [Compute Shape Parameters](#)
- [Tag Parameters](#)
- [Storage Parameters](#)
- [Region Parameters](#)
- [Service Parameters](#)

## Basic Syntax

Define template parameters within the `parameters` node of your template file. For each parameter, specify a `type` and other optional attributes and constraints. For example:

```
parameters:
  dbAdminUser:
    label: Database Username
    description: The admin user for the new database
    type: String
    mandatory: true
    sensitive: true
    maxLength: 15
  nodeCount:
    label: Database Nodes
    description: The number of nodes in the database cluster
    type: Number
    default: 1
    minValue: 1
    maxValue: 3
```

To learn more about configuring parameters, see [Creating Template Parameters](#).

Specify the `allowedValues` attribute in one of these formats:

- Enter a sequence of values. For example: `[oc1,oc2]`
- Enter a single JSON object with key/value pairs. The values are displayed to the user and the `GetParam` function returns the selected key. For example: `{"10.3.6": "WebLogic Server 11g", "12.1.3": "WebLogic Server 12c"}`
- Enter a sequence of JSON objects, which each object being a key/value pair. The values are displayed to the user and the `GetParam` function returns the selected key. For example: `[{"10.3.6": "WebLogic Server 11g"}, {"12.1.3": "WebLogic Server 12c"}]`

## Stack Name Parameter

You do not need to explicitly define a parameter for the stack's name. Retrieve the stack name in your template with the implicit parameter `serviceName`:

```
dbSchema: { "Fn::Join": ["-", [{"Fn::GetParam": serviceName, DB}]} }
```

However, if you want to define constraints on the stack's name, such as the number of characters, create a `String` parameter with the same name in your template:

```
parameters:
  serviceName:
    label: Name
    type: String
    minLength: 10
    allowedPattern: "^[A-Z]+[A-Z0-9]"
```

## Parameter Groups

You can optionally group related template parameters together. Each parameter group is given a label, which is used as a heading in the Details page of the Oracle Cloud Stack creation wizard.

Define parameter groups within the `parameterGroups` node of your template file. This node contains a sequence. Each item in the sequence contains a group's label and the group's list of parameters. For example:

```
parameterGroups:
  - label: Database Configuration
    parameters: [dbVersion,dbComputeShape,dbSid]
  - label: Storage Configuration
    parameters: [storageUser,storagePassword]
```



### Tip:

The `parameters` sequence can also be expressed with the following syntax:

```
parameters:
  - storageUser
  - storagePassword
```

## Hidden Parameters



This topic does not apply to Oracle Cloud at Customer.

If you set the `visible` attribute to `false` for a parameter, it will not be displayed as an input field in the Oracle Cloud Stack console:

```
parameters:
  dbSchema:
    ...
    visible: false
```

CLI and REST API clients can specify values for hidden parameters when creating a stack.

## Password Parameters

This parameter type enables users to securely enter a password in the Oracle Cloud Stack console. The password value is not displayed in the browser. The user must also enter the same password value twice in order for it to be accepted.

```
parameters:
  dbPassword:
    label: Database Password
    type: Password
```

The name of the confirmation input field is the value of `label` attribute prefixed with the text “Confirm”. In the example above, the confirmation text field would have the label “Confirm Database Password”.

Unlike other parameter types, the password type is case sensitive by default.

Use the `GetParam` function to retrieve the password value:

```
adminPassword: { "Fn::GetParam": dbPassword }
```

### SSH Parameters

This parameter type enables users to enter a Secure Shell (SSH) public key. Oracle Cloud Stack console users can supply the key value as text or as a file, or let Oracle Cloud generate a value dynamically. If the key is generated, users can download it as a file.

```
parameters:
  sshkey:
    label: SSH Public Key
    type: Ssh
```

Use the `GetParam` function to retrieve the user-supplied or generated key value:

```
vmPublicKey: { "Fn::GetParam": sshkey }
```

### Compute Shape Parameters

This parameter type enables users to select a compute shape. Oracle Cloud defines standard shapes (`oc3`, `oc2m`, and so on) that specify the number of Oracle Compute Units (OCPU) and the amount of memory allocated to a cloud resource.

```
parameters:
  databaseShape:
    label: Database Compute Shape
    type: ComputeShape
    default: oc3
```

You can use the `allowedValues` constraint with the `ComputeShape` parameter type to limit the Oracle Cloud compute shapes that the user may choose. The Oracle Cloud Stack console will automatically display the number of OCPUs and the amount of RAM for each shape option in your list. If you do not provide an `allowedValues` constraint, the user can select any standard compute shape.

```
parameters:
  databaseShape:
    label: Database Compute Shape
    type: ComputeShape
    allowedValues: [ oc3, oc4, oc5, oc6, oc1m, oc2m, oc3m, oc4m ]
```

Use the `GetParam` function to retrieve the selected compute shape:

```
weblogicShape: { "Fn::GetParam": databaseShape }
```

### Tag Parameters



This topic does not apply to Oracle Cloud at Customer.

The parameter type enables users to select one or more existing tags in this cloud account, or to create new tags.

```
parameters:
  dbTags:
    label: Database Tags
    type: Tag
```

Each tag can be a simple key, or a `key:value` pair. Use the `GetParam` function to retrieve the tags as a sequence of `key:value` objects. If a tag does not have a specific value, it will be an empty string.

```
tags: { "Fn::GetParam": dbTags }
```

### Storage Parameters

This parameter type enables users to specify a storage container in Oracle Cloud Infrastructure Object Storage Classic. See [About Oracle Storage Cloud Service in Using Oracle Storage Cloud Service](#).

```
parameters:
  backupStorage:
    label: Cloud Storage
    type: Storage
```

Each parameter of this type results in three separate input fields in the Oracle Cloud Stack console:

- The URL of the storage container
- The name of an Oracle Cloud user that has access to this container
- The password for this Oracle Cloud user

Use the `GetParam` function to retrieve these three input values. The storage parameter's value is a JSON object. Use a dot notation in order to access the values within this object:

- `dbBackupContainer`: { "Fn::GetParam": backupStorage.**cloudStorageContainer** }
- `dbBackupUser`: { "Fn::GetParam": backupStorage.**cloudStorageUser** }
- `dbBackupPassword`: { "Fn::GetParam": backupStorage.**cloudStoragePassword** }

By default, the labels and descriptions for these three input fields are generated from the template parameter's `label` attribute. In the previous example, the label for the first text field would be "Cloud Storage – Backup Storage Container". Alternatively, you can specify your own label and description for each input field:

```
parameters:
  backupStorage:
    label: {"cloudStorageContainer": "Backup Storage Container",
"cloudStorageUser": "Storage User", "cloudStoragePassword": "Storage
Password"}
    description: {"cloudStorageContainer": "Oracle Storage Cloud Service
container URL", "cloudStorageUser": "User with access to Oracle Storage
Cloud Service", "cloudStoragePassword": "Password for the storage user"}
    type: Storage
```

When you use the CLI to create a stack from a template that has storage type parameters, enter the parameter value as a JSON object with these three attributes. For example:

```
-p backupStorage:
{"cloudStorageContainer": "value", "cloudStorageUser": "value", "cloudStoragePa
ssword": "value" }
```

## Region Parameters

The Oracle Cloud Stack console automatically displays a list of available Oracle Cloud regions, such as geographical locations, which will vary depending on the user's account settings. The default value is "No Preference", in which case Oracle Cloud will automatically select a region. Some Oracle Cloud features like IP networks and IP reservations may not be supported if a resource is not assigned to a specific region.

The `Region` parameter type displays a single input field in the console:

```
parameters:
  cloudRegion:
    label: Cloud Region
    type: Region
```

Use the `GetParam` function to retrieve the user-selected region:

```
dbRegion: { "Fn::GetParam": cloudRegion }
```

## Service Parameters

The `Service_Type` parameter type enables users to select from a list of existing Oracle Cloud service instances. `Type` is the name of a resource type in Oracle Cloud Stack (see [List of Resource Types](#)). For example, if you define a parameter of type `Service_dbaas`, the user can select from a list of existing Oracle Database Cloud Service deployments in this cloud account:

```
parameters:
  dbaaSService:
```

```
label: Database
type: Service_dbaas
```

Use the `GetParam` function to retrieve the name of the selected service instance:

```
dbService: { "Fn::GetParam" : dbaaSService }
```

## Using Resources

Edit your Oracle Cloud Stack template and configure resource parameters and dependencies.

For an introduction, see [What are Resources](#).

### Topics

- [Basic Syntax](#)
- [Get Template Parameters](#)
- [Get Resource Attributes](#)
- [Dependencies](#)

### Basic Syntax

Define resources within the `resources` node of your template file. For each node, specify its `type`. For example:

```
resources:
  HRJavaApp:
    type: jaas
```

See [List of Resource Types](#).

Add a `parameters` node within your resource node. Refer to the REST API documentation for this type of cloud resource to identify the available parameters. See [List of Resource Types](#). For example, this [endpoint](#) is used to create a service instance in Oracle Java Cloud Service.

```
resources:
  HRJavaApp:
    type: jaas
    parameters:
      adminUserName: myuser
      clusterSize: 3
```

If the request payload for this resource's REST endpoint supports a hierarchy of parameters, model this same hierarchy in your template file. For example:

```
parameters:
  adminUserName: myuser
  virtualMachine:
```

```
shape: oc3
volumeSize: 10G
```

Similarly, if the request payload supports arrays, model these arrays in your template file as sequences. For example:

```
parameters:
  adminUserName: myuser
  virtualMachines:
    -
      shape: oc3
      volumeSize: 10G
    -
      shape: oc3
      volumeSize: 20G
```

### Get Template Parameters

Use the `GetParam` function to assign the value of a template parameter to a resource's parameter. For example:

```
adminUserName: { "Fn::GetParam": userName }
```

For more information, see [Using Template Parameters](#).

### Get Resource Attributes

Use the `GetAtt` function to assign the value of another resource's runtime attribute to this resource's parameter. Enter the resource name and the attribute name as a sequence. For example:

```
adminUserName: { "Fn::GetAtt": [HRDatabase, userName] }
```

Refer to the REST API documentation for this type of cloud resource in order to identify its attributes. See [List of Resource Types](#). For example, this [endpoint](#) is used to view a service instance in Oracle Java Cloud Service.

For more information, see [Using Attributes](#).

### Dependencies

Optionally add a `depends_on` attribute to your resource node. Place it after the `parameters` node. Use this attribute to identify one or more resources in this template on which this resource depends. Enter the value as a sequence:

```
resources:
  HRJavaApp:
    type: jaas
    parameters:
      resource parameters here
    depends_on:
      - HRStorage
      - HRDatabase
```

**Tip:**

YAML sequences can also be expressed as arrays, such as `[HRStorage,HRDatabase]`.

## Using Attributes

Edit your Oracle Cloud Stack template, and use the runtime attributes of resources to configure other resources within the same template.

For an introduction, see [What are Attributes](#).

### Topics

- [The GetAtt Function](#)
- [Nested Attributes](#)
- [Stack Attributes](#)

### The GetAtt Function

While provisioning a stack, Oracle Cloud Stack uses REST endpoints to collect the specified resource attributes. For example, the response payload for the endpoint [View a Service Instance](#) describes the available attributes for an Oracle Database Cloud Service resource.

Refer to the REST documentation for a specific resource type in order to identify its list of attributes. See [List of Resource Types](#).

Use the `GetAtt` function in your template to retrieve an attribute from a resource:

```
Fn::GetAtt: [resourceName, attributeName]
```

For example:

```
parameters:
  dbURL: { "Fn::GetAtt": [mydb, connect_descriptor] }
  clusterSize: { "Fn::GetAtt": [mydb, num_nodes] }
```

### Nested Attributes

The JSON response payload from a REST endpoint may include a hierarchy of attributes. For example:

```
{
  "keyComponentInstance": "mysql",
  "adminHostName": "mytestinstance-mysql-1",
  "components": {
    "mysql": {
      "attributes": {
        "CONNECT_STRING": {
          "displayName": "Connect Descriptor",
          "type": "STRING",
```

```
        "value": "142.114.43.1:3306\mytestinstance"  
      }  
    }  
  }  
}
```

The attribute name input for the `GetAtt` function supports a dot notation in order to access nested attributes within a JSON payload. For example, to access the highlighted attribute in the previous JSON response:

```
parameters:  
  dbURL: { "Fn::GetAtt": [mydb,  
    components.mysql.attributes.CONNECT_STRING.value] }
```

### Stack Attributes

When you monitor running stacks from the CLI or REST API, Oracle Cloud Stack returns a set of standard runtime attributes for each stack, such as who created the stack. But a template can also define custom stack attributes.

Define attributes within the `attributes` node of your template file. Give each attribute a name and an optional description. For example:

```
attributes:  
  OraConnectString:  
    description: Connection string for Oracle database
```

Assign a value to each attribute. Use the `GetAtt` function to retrieve the value of a resource's attribute. For example:

```
attributes:  
  OraConnectString:  
    value: { "Fn::GetAtt": [mydb,  
    components.mysql.attributes.CONNECT_STRING.value] }  
    description: Connection string for Oracle database
```

## Using Conditions

Edit your Oracle Cloud Stack template and customize how a stack is created under different conditions.

For an introduction, see [What are Conditions](#).

### Topics

- [Using Conditional Functions](#)
- [Defining Global Conditions](#)
- [Using Global Conditions with Functions](#)
- [Assigning a Global Condition to a Resource](#)
- [Using Conditions in Template Parameters](#)

## Using Conditional Functions

Oracle Cloud Stack supports these conditional functions: `Equals`, `GreaterThan`, `LessThan`, `Contains`, `Empty`, `Match`, `Not`, `And`, `Or`.

These functions are executed at runtime by Oracle Cloud Stack when a stack is being created. Each function evaluates two input strings and returns either the string `true` or the string `false` at the function's location in the template. The inputs to a conditional function can be literal strings or functions that return strings, such as `GetParam`.

For example, consider a resource definition within a template that contains three parameters:

```
myresource:
  type: xyz
  parameters:
    dbEnabled: { "Fn::Equals": [{"Fn::GetParam": "dbVersion"}, "12.2"] }
    cacheEnabled: { "Fn::GreaterThan": [{"Fn::GetParam": "cacheSize"}, "0"] }
    backupEnabled: { "Fn::Not": [{"Fn::Equals":
[{"Fn::GetParam": "container"}, "" ]}] }
```

These conditional functions result in the following logic:

- The `dbEnabled` parameter is set to `"true"` only if the value of the `dbVersion` template parameter is set to the value `"12.2"`.
- The `cacheEnabled` parameter is set to `"true"` only if the value of the `cacheSize` template parameter is greater than zero.
- The `backupEnabled` parameter is set to `"true"` only if the value of the `container` template parameter is **not** an empty string.

For more information, see [List of Functions](#).

## Defining Global Conditions

You often need to use the same conditional expression (`cacheSize` is greater than zero) at multiple locations in your template. Alternatively, you can define these conditions in a single location within the template and then refer to them throughout the template.

Define global conditions within the `conditions` node of your template file. Give each condition a name and an expression whose value evaluates to `true` or `false`. For example:

```
conditions:
  includeMyApp: { "Fn::Equals": [{"Fn::GetParam": "dbVersion"}, "12.2"] }
```

This example template defines a condition named `includeMyApp` whose value is `true` only if the value of the `dbVersion` template parameter is equal to `"12.2"`.

## Using Global Conditions with Functions

Use the `If` function to evaluate a global condition at some location in your template. Provide the name of the global condition along with two string values. If the specified

condition evaluates to `true` the `If` function returns the first value. Otherwise the function returns the second value.

In the following example, the `shape` parameter for a resource is assigned the value `"oc3"` if the `includeMyApp` condition is true, or the value `"oc4"` if the condition is false:

```
myresource:
  type: xyz
  parameters:
    shape: { "Fn::If": [ "includeMyApp", "oc3", "oc4" ] }
```

The `If` function can also be used to remove the current object, attribute or sequence item after evaluating its condition. Use the variable `OPC::NoValue` for either the true or false arguments. In this example, the `dbService` parameter is dynamically removed from this resource's definition if the `useDB` condition is false.

```
myresource:
  type: xyz
  parameters:
    clusterSize: 2
    dbService: { "Fn::If": [ "useDB", "MyOracleDB-1", "OPC::NoValue" ] }
```

### Assigning a Global Condition to a Resource

Each resource in a template can optionally be assigned the name of a global condition. If this condition evaluates to `true`, the resource will be created as part of the stack. If it evaluates to `false`, the resource (and any resources on which it depends) will not be created.

Place the `condition` attribute within a resource's definition, after the `parameters`. For example:

```
resources:
  myapp:
    type: xyz
    parameters:
      param1: value1
      param2: value2
    condition: includeMyApp
```

### Using Conditions in Template Parameters

You can also use conditional functions and global conditions within the definition of a template parameter or a parameter group. These conditions control whether or not the parameter or group is displayed in the Oracle Cloud Stack console when creating a stack. If the condition evaluates to `false`, the parameter or group is hidden in the user interface. If it evaluates to `true`, the parameter or group is shown. These conditions are dynamically reevaluated each time the user modifies the value of a template parameter.

For example, you can use a boolean (check box) parameter to determine whether or not another parameter is shown to the user:

```
parameters:
  useBackup:
    label: Use Backup
    type: Boolean
    mandatory: false
    default: false
  backupPath:
    label: Backup Path
    type: String
    condition: {"Fn::Equals":["Fn::GetParam":"useBackup"],"true"}}
```

Alternatively, use the same parameter and a global condition to control when an entire parameter group is shown to the user:

```
parameters:
  useBackup:
    label: Use Backup
    type: Boolean
    mandatory: false
    default: false
  ...
parameterGroups:
  - label: Backup Configuration
    parameters: [ backupPath, backupTime ]
    condition: backupEnabled
conditions:
  backupEnabled: {"Fn::Equals":["Fn::GetParam":"useBackup"],"true"}}
```

## Using Custom Actions



This topic does not apply to Oracle Cloud at Customer.

Edit your Oracle Cloud Stack template file and use Software Components to execute custom operating system (OS) commands or scripts after a stack resource is provisioned.

For example, you can use custom actions to:

- Install an OS package
- Modify the default OS configuration
- Create a database schema
- Deploy an application
- Download files from cloud storage

Custom actions are only supported for cloud resources that provide a Secure Shell (SSH) interface to their administration node (virtual machine). The specified OS commands are run on the administration node after the resource is created. If a

resource consists of multiple administration nodes, the commands are run only from the first node.

 **Note:**

Custom actions are not supported on Oracle-managed or Autonomous cloud services.

 **Tutorials** are also available to help you create custom actions.

## Topics

- [Basic Syntax](#)
- [Assigning a Custom Action to a Resource](#)
- [Inputs and Environment Variables](#)
- [Downloading Files](#)
- [Activity Log](#)
- [Best Practices](#)

## Basic Syntax

Add a resource of type `SoftwareComponent` to the template. Within the `parameters` node of this resource, add a sequence named `configs`.

```
resources:
  ...
  resource_name:
    type: SoftwareComponent
    parameters:
      configs:
        - configuration
        - configuration
```

Each object in the `configs` sequence describes a custom action. Specify one or more Bash OS commands to run on the target cloud resources.

```
parameters:
  configs:
    - actions: [CREATE]
      name: configuration_name
      config:
        Fn::Base64:
          bash_commands
      runAsUser: OS_user
      timeout: minutes
      continueOnFailure: true_or_false
```

The available attributes for a custom action are:

- `actions` – The only supported value is `[CREATE]`. All custom actions run after the target cloud resource is created.
- `name` – An optional name for this custom action.
- `config` – One or more Bash commands as a Base64 encoded string. Separate multiple commands with new line (`\n`) characters.
- `runAsUser` – The OS user name against which to run the commands. The default user is `opc`.
- `timeout` – The number of minutes within which the custom action is expected to complete. If the action does not complete within this time, stack creation fails. The default value is 5 minutes.
- `continueOnFailure` – If `false`, the creation of the stack fails if this custom action fails. If `true`, a warning is added to the activity log, but the creation of the stack continues.

You can also use the `Join` function as a convenient method of defining an action with multiple commands:

```
name: configuration_name
config:
  Fn::Base64:
    Fn::Join:
      - "\n"
      -
      - bash_command
      - bash_command
```

Below is an example Software Component resource that defines a custom action:

```
resources:
  updateDatabaseOS:
    type: SoftwareComponent
    parameters:
      configs:
        - actions: [CREATE]
          name: updateHosts
          config:
            Fn::Base64:
              Fn::Join:
                - "\n"
                -
                - touch ~/stack.out
                - echo 'Updating hosts' >> ~/stack.out
                - echo 'myhost 192.168.1.10' >> /etc/hosts
          continueOnFailure: true
```

 **Tip:**

When using the `Join` function, be sure that all members of the same sequence have the same indentation. The following example is invalid:

```
Fn::Join:
- "\n"
-
- This is
- not valid
```

### Assigning a Custom Action to a Resource

After defining Software Components in your template, you refer to them within the definitions of other cloud resources in the same template. Add a `configs` sequence within the resource, and place it after the `parameters` node. The `configs` sequence lists the names of the Software Component resources that Oracle Cloud Stack will execute after this resource is successfully created.

Use the `GetResource` function to retrieve each Software Component, and also give each a logical configuration name.

```
resources:
  ...
  resource_name:
    type: resource_type
    parameters:
      ...
    configs:
      - name: configuration_name
        config: { "Fn::GetResource": software_component_name }
      - name: configuration_name
        config: { "Fn::GetResource": software_component_name }
```

For example:

```
HRDatabase:
  type: dbaas
  parameters:
    ...
  configs:
    - name: HRDatabaseUpdateOS
      config: { "Fn::GetResource": updateDatabaseOS }
```

### Inputs and Environment Variables

The use of template functions, such as `GetParam`, is not supported within the OS commands of a custom action. However, you can parameterize the actions in a template by using environment variables in your OS commands. Add a sequence

named `inputs` within the `parameters` node of your Software Component resource, and then declare the environment variables that are used in this custom action.

```
resources:
  ...
  resource_name:
    type: SoftwareComponent
    parameters:
      configs:
        ...
      inputs:
        - name: env_variable
        - name: env_variable
```

Refer to these environment variables in the OS commands of the Software Component. For example:

```
resources:
  updateDatabaseOS:
    type: SoftwareComponent
    parameters:
      configs:
        - actions: [CREATE]
          name: updateHosts
          config:
            Fn::Base64:
              Fn::Join:
                - "\n"
                -
                  - echo 'Updating database OS...'
                  - echo "myhost ${storage_address}" >> /etc/hosts
      inputs:
        - name: storage_address
```

When you assign the Software Component to another cloud resource using the `configs` sequence, you must specify the values for any input parameters by adding an object named `params` to the sequence.

```
resources:
  ...
  HRDatabase:
    type: dbaas
    parameters:
      ...
    configs:
      - name: configuration_name
        config: { "Fn::GetResource": software_component_name }
        params:
          input_name: value
          input_name: value
      - name: configuration_name
        config: { "Fn::GetResource": software_component_name }
        params:
```

```
input_name: value
input_name: value
```

You can use template functions when specifying an input's value, such as `GetParam` and `GetAtt`. For example:

```
configs:
- name: HRDatabaseUpdateOS
  config: { "Fn::GetResource": updateDatabaseOS }
  params:
    log_file: stack.out
    storage_address: { "Fn::GetParam": storageIPAddress }
```

## Downloading Files

In your custom actions, you can use `wget`, `curl`, or similar OS commands to download scripts and other files to the administration node in the target cloud resource. For example, you could download and extract an archive file found in an Oracle Cloud Infrastructure Object Storage container, and then execute one or more scripts found in the archive.

## Activity Log

You can verify the execution of your custom actions by viewing the activity log for a cloud stack. The log messages specify the name of the target resource, the name of the action that was executed, and the node on which it was executed.

```
[stack_name] : Attempting to execute SoftwareComponent script -
[identity_domain/resource_name] : [action_name] on host [node_ip_address]
...
[stack_name] : Execution completed successfully. SoftwareComponent -
[identity_domain/resource_name] : [action_name] on host [node_ip_address]
```

For example:

```
[MyStack] : Execution completed successfully. SoftwareComponent -
[MyIdentityDomain/HRDatabase] : [updateHosts] on host [192.168.10.10]
```

See [Viewing the Details of a Cloud Stack](#).

## Best Practices

- Program a custom action so that it can run multiple times without failing or behaving unexpectedly (idempotent). A custom action might be run more than once in the following scenarios:
  - Oracle Cloud Stack retries the execution of a custom action after a communication failure with the target resource.
  - If you create a cloud stack with the `RETAIN` option, and then resume the stack after a resource provisioning failure, Oracle Cloud Stack also (re)executes any custom actions that are associated with the failed resource.
- In a custom action, return 0 to indicate success and return 1 to indicate a failure.

- By default, all existing resources are terminated if there is a failure during the creation of a stack, which includes the deletion of any log files generated by a custom action. To facilitate troubleshooting, record log messages to an external location like cloud storage.

## Using Validations

Edit your Oracle Cloud Stack template file and configure advanced validation criteria.

When a stack is created from a template, any validations defined in the template are executed prior to the provisioning of the resources in the template. If the validations succeed, Oracle Cloud Stack attempts to create the resources. Any failed validations are reported to the user and the stack is not created.

### Topics

- [Basic Syntax](#)
- [Database Connectivity](#)
- [Storage Connectivity](#)
- [Generic Constraint](#)

### Basic Syntax

Define template validations within the `validations` node of your template file. For each validation, specify a `validationType` and add a `parameters` node.

```
validations:
  myFirstValidation:
    validationType: DB_CONNECT
    parameters:
      ...
  mySecondValidation:
    validationType: STORAGE_CONNECT
    parameters:
      ...
```

The available parameters depend on the type of validation. You can assign static values to these validation parameters, or assign the value of a template parameter by using the `GetParam` function.

You can optionally assign the name of an existing condition. The validation will not execute unless the specified condition evaluates to true:

```
validations:
  myValidation:
    condition: connectToDB
    validationType: DB_CONNECT
    parameters:
      ...
```

## Database Connectivity

Check the connectivity to an existing Oracle Database Cloud Service deployment prior to creating a stack. Provide the connection details for the database deployment in the validation definition. For example:

```
dbConnectivityValidation:
  validationType: DB_CONNECT
  parameters:
    serviceName: MyDBInstance
    userName: SYS
    password: password
    requiredSize: 10GB
    constraintDescription: Failed to connect to the MyDBInstance database
```

Parameter	Description
serviceName (Optional)	Enter the name of an existing Oracle Database Cloud Service deployment. You must specify either <code>serviceName</code> or <code>connectString</code> .
connectString (Optional)	Enter a Java Database Connectivity (JDBC) URL for an existing Oracle Database Cloud Service deployment. For example: <code>jdbc:oracle:thin:@203.0.113.100:1521:orcl.</code> You must specify either <code>serviceName</code> or <code>connectString</code> .
userName	The database user with which to make a connection
password	The database user's password
requiredSize (Optional)	The validation fails if the database's available capacity is less than this size.
pdbName (Optional)	The validation fails if the specified pluggable database (PDB) does not exist in the database.
schemaName (Optional)	The validation fails if the specified schema already exists in the database.
constraintDescription (Optional)	The error message to display to the user if this validation fails

## Storage Connectivity



This topic does not apply to Oracle Cloud at Customer.

Check the connectivity to Oracle Cloud Infrastructure Container Service Classic prior to creating a stack. Provide the storage connection details in the validation definition.

You can connect to cloud storage with a username and password, as shown in the example below:

```
storageConnectivityValidation:
  validationType: STORAGE_CONNECT
```

```

parameters:
  cloudStorageContainer: https://foo.storage.oraclecloud.com/v1/
MyService-bar/MyContainer
  cloudStorageUser: john@example.com
  cloudStoragePassword: your_password
  checkIfContainerExists: false
  constraintDescription: Failed to connect to cloud storage as john

```

If you do not provide storage credentials, Oracle Cloud Stack will attempt to connect to cloud storage by using the OAuth protocol. You can optionally specify the name of an Oracle Cloud service that has been previously configured for OAuth authentication to cloud storage. For example:

```

storageConnectivityValidation:
  validationType: STORAGE_CONNECT
  parameters:
    cloudStorageContainer: https://foo.storage.oraclecloud.com/v1/
MyService-bar/MyContainer
  serviceType: jaas
  checkIfContainerExists: false
  constraintDescription: Failed to connect to cloud storage using jaas
OAuth configuration

```

Alternatively, you can specify an Oracle Cloud entitlement ID that has been previously configured for authentication to cloud storage by using the OAuth protocol. For example:

```

storageConnectivityValidation:
  validationType: STORAGE_CONNECT
  parameters:
    cloudStorageContainer: https://foo.storage.oraclecloud.com/v1/
MyService-bar/MyContainer
  serviceEntitlementId: 123456
  checkIfContainerExists: false
  constraintDescription: Failed to connect to cloud storage using
entitlement 123456

```

Parameter	Description
cloudStorageContainer	The URL for a cloud storage container If checkIfContainerExists is true, then this container must already exist for this validation to succeed.
cloudStorageUser (Optional)	The Oracle Cloud user with which to make a connection If you do not provide a value for this parameter, Oracle Cloud Stack attempts to connect to cloud storage by using the OAuth protocol.
cloudStoragePassword (Optional)	The password for the specified cloud user If you do not provide a value for this parameter, Oracle Cloud Stack will attempt to connect to cloud storage by using the OAuth protocol.

Parameter	Description
<code>checkIfContainerExists</code> (Optional)	If set to true, then the validation will succeed only if the specified storage container exists. If set to false, then the validation simply checks for connectivity to the storage cloud service.
<code>serviceType</code> (Optional)	The name of an Oracle Cloud service that has been previously configured for authentication to cloud storage by using the OAuth protocol. Oracle Cloud Stack uses this service's OAuth configuration to validate connectivity to cloud storage. See <a href="#">List of Resource Types</a> . If not specified, Oracle Cloud Stack automatically selects a service that is configured for OAuth.
<code>serviceEntitlementId</code> (Optional)	An Oracle Cloud entitlement ID that has been previously configured for authentication to cloud storage by using the OAuth protocol. Oracle Cloud Stack uses this account's OAuth configuration to validate connectivity to cloud storage. If not specified, Oracle Cloud Stack automatically selects an account that is configured for OAuth. If both <code>serviceType</code> and <code>serviceEntitlementId</code> are specified, <code>serviceEntitlementId</code> takes precedence.
<code>constraintDescription</code> (Optional)	The error message to display to the user if this validation fails

### Generic Constraint



This topic does not apply to Oracle Cloud at Customer.

Evaluate a conditional expression that uses standard template functions. Validation fails if the expression evaluates to `false`. For example:

```
passwordValidation:
  validationType: CONSTRAINT
  parameters:
    constraint: {"Fn::Equals": [{"Fn::GetParam": "password"}, "oracle"]}
    constraintDescription: The password oracle is not permitted. Enter a
different password.
```

To learn more about conditional expressions, see [Using Conditions](#) and [List of Functions](#).

Parameter	Description
<code>constraint</code>	A YAML expression that evaluates to either <code>true</code> or <code>false</code> . Validation fails if the expression evaluates to <code>false</code> .
<code>constraintDescription</code> (Optional)	The error message to display to the user if this validation fails

## Using Maps



This topic does not apply to Oracle Cloud at Customer.

Edit your Oracle Cloud Stack template file to define and use data structures.

Maps allow you to define static values within a template and refer to these values within other template components such as resources. You organize the data within a map as follows:

- A **map** contains one or more **submaps**.
- A **submap** contains one or more **key-value pairs**.

### Topics

- [Defining Maps](#)
- [Using Map Values](#)

### Defining Maps

Define maps within the `mappings` node of your template file. Within this node, add one or more map nodes.

```
mappings:
  map1:
    ...
  map2:
    ...
```

Within each map node, add one or more submap nodes. Each submap contains a JSON object with one or more key-value pairs.

```
mappings:
  map1:
    submap1: {"key1":"value1","key2":"value2", ...}
    submap2: ...
    ...
```

For example:

```
mappings:
  shapeMap:
    database: {"dev":"oc3", "prod":"oc4"}
```

### Using Map Values

Retrieve a value within a map by using the `FindInMap` function, which takes three arguments:

- The name of the map
- The name of the submap

- The key whose value you want to use

For example, suppose you want to retrieve the value for the `prod` key in the following map:

```
mappings:
  shapeMap:
    database: {"dev": "oc3", "prod": "oc4"}
```

Running `FindInMap` with the following arguments results in the value `oc4`:

```
"Fn::FindInMap": ["shapeMap", "database", "prod"]
```

Any of these three arguments can be determined dynamically through the use of other functions. For example, if the key name (`dev` or `prod`) is supplied by the user as a template parameter, use the `GetParam` function:

```
"Fn::FindInMap": ["shapeMap", "database", {"Fn::GetParam": stage}]
```

## Template Best Practices

Follow these best practices to ensure that your Oracle Cloud Stack templates are flexible and maintainable, and work as expected.

Best Practice	More Information
Describe your template	Use <code>templateDescription</code> to explain your template's purpose and to describe the cloud resources that it creates. Also explain any limitations and/or conditional behavior in your template.
Create input parameters	If your template uses static values for a resource's configuration, users who want to change this configuration must copy and edit your template. Instead, use parameters to make your template more flexible and reusable.  Provide a sensible default value for each template parameter. This minimizes the amount of time that a typical user spends creating stacks from your template.
Describe your parameters	Use the <code>description</code> attribute to explain the purpose of each template parameter. Describe the default behavior if the parameter is optional, and any dependencies between this parameter and other parameters. Users can view this help text when they use the web console to create a stack from your template.
Configure parameter constraints	Stacks can require significant time to provision. Avoid costly user errors by configuring appropriate constraints on your template parameters, such as a minimum value, a maximum number of characters, or a regular expression. See <a href="#">Creating Template Parameters</a> .
Use parameters for passwords and keys	Oracle does not recommend the use of default values for passwords, SSH keys, and other secret information. Instead, use template parameters so that users can provide their own inputs for these values. Do not specify a default value for these parameters.
Configure template validations	Stacks can require significant time to provision. Avoid costly user errors by configuring validations in your template. For example, a validation can check the connectivity to an existing database prior to creating a stack. See <a href="#">Using Validations</a> .

Best Practice	More Information
Validate your template syntax	Before publishing and testing your template, check its contents for common syntax errors. See <a href="#">Validating a Template</a> .
Follow best practices when developing custom actions for a resource	See <a href="#">Best Practices</a> .

## List of Resource Types

Oracle Cloud Stack templates can provision the following types of resources in Oracle Cloud.

User-defined stack templates can create any combination of these resources. Popular combinations are also available as standard templates. See [About Oracle Stack Templates](#).

If you're editing a stack template offline, refer to the specific resource's REST API documentation for a complete list of available parameters and attributes.

Resource Type	Description	REST Documentation
apaas	An application in Oracle Application Container Cloud Service	<a href="#">Create an Application</a> <a href="#">View Application Details</a>
APICatalog	A service instance in Oracle API Catalog Cloud Service	
APICS	A service instance in API Platform Cloud Service	<a href="#">Create a Service Instance</a> <a href="#">View a Service Instance</a>
AUTOANALYTICS	A service instance in Oracle Analytics Cloud	
BDCSCE	A service instance in Oracle Big Data Cloud	<a href="#">Create a Service Instance</a> <a href="#">View a Service Instance</a>
BigDataAppliance	A service instance in Oracle Big Data Cloud Service	
BOTS	A Bot component in Oracle Mobile Cloud, Enterprise	
caching	An application cache in Oracle Cache Cloud Service	<a href="#">Create a Cache</a> <a href="#">View Cache Details</a>
CEC	A suite in Oracle Content and Experience	
CECS	A service instance in Oracle Content and Experience	
CONTAINER	A service instance in Oracle Cloud Infrastructure Container Service Classic	
dbaas	A service instance in Oracle Database Cloud Service	<a href="#">Create a Service Instance</a> <a href="#">View a Service Instance</a>
DevServiceApp	A project in Oracle Developer Cloud Service	

Resource Type	Description	REST Documentation
DHCS	A service instance in Oracle Data Hub Cloud Service	<a href="#">Create a Service Instance</a> <a href="#">View a Service Instance</a>
dics	A service instance in Oracle Data Integration Platform Cloud	
ggcs	A service instance in Oracle GoldenGate Cloud Service	
IDCS	A service instance in Oracle Identity Cloud Service	
IntegrationCloud	A service instance in Oracle Integration	
INTEGRATIONCAUTO	A service instance in Oracle Autonomous Integration Cloud	
IOT	A component of Oracle Internet of Things Cloud Service <b>Note:</b> Use only the standard Oracle templates to provision Oracle Internet of Things Cloud Service resources.	
jaas	A service instance in Oracle Java Cloud Service	<a href="#">Create a Service Instance</a> <a href="#">View a Service Instance</a>
Mobile	A component of Oracle Mobile Cloud Service	
MySQLCS	A service instance in Oracle MySQL Cloud Service	<a href="#">Create a Service Instance</a> <a href="#">View a Service Instance</a>
OEHCS	A service instance in Oracle Event Hub Cloud Service	<a href="#">Create a Service Instance</a> <a href="#">View a Service Instance</a>
OEHPCS	A dedicated instance in Oracle Event Hub Cloud Service - Platform	<a href="#">Create a Service Instance</a> <a href="#">View a Service Instance</a>
OMCE	A service instance in Oracle Mobile Cloud, Enterprise <b>Note:</b> Use only the standard Oracle templates to provision Oracle Mobile Cloud, Enterprise resources.	
OMCEExternal	A service instance in Oracle Management Cloud Service	
OSS.Container	A storage container in Oracle Cloud Infrastructure Object Storage Classic	Not applicable
SoftwareComponent	A custom action that executes one or more operating system commands on a target resource after it has been successfully provisioned. See <a href="#">Using Custom Actions</a> .	Not applicable
SOA	A service instance in Oracle SOA Cloud Service	<a href="#">Provision a New Service Instance</a> <a href="#">View a Service Instance</a>
VisualBuilder	A service instance in Oracle Visual Builder	

Resource Type	Description	REST Documentation
VISUALBUILDER AUTO	An autonomous service instance in Oracle Visual Builder	

## List of Functions

Oracle Cloud Stack offers several intrinsic functions that you can use when creating a template. These functions are executed at runtime by Oracle Cloud Stack when a stack is being created.

Function	Description and Syntax	Examples
And	Return true if the sequence of expressions all evaluate to true; otherwise return false.  <code>Fn::And: [exp1, exp2, expN]</code>	<code>"Fn::And": [{"Fn::Equals": [{"Fn::GetParam": stage}, "dev"]}, {"Fn::Equals": [{"Fn::GetParam": container}, ""]}]</code>
Base64	Convert the given string to base64 encoding.  <code>Fn::Base64: string</code>	<code>publicKeyText: { "Fn::Base64": { "Fn::GetParam": "key" } }</code>
Contains	Return true if the first string is found in the second string, or if the specified value is found in a sequence of values; otherwise return false.  <code>Fn::Contains: [valueToSearchFor, stringOrSequence]</code>	<code>"Fn::Contains": ["us", { "Fn::GetParam": region}]</code>
Empty	Return true if the provided string is empty; otherwise return false.  <code>Fn::Empty: string</code>	<code>"Fn::Empty": { "Fn::GetParam": region}</code>
Equals	Compare a sequence containing two expressions and return true if the expressions evaluate to the same value; otherwise return false.  <code>Fn::Equals: [exp1, exp2]</code>	<code>"Fn::Equals": [{"Fn::GetParam": appType}, "Java"]</code>

Function	Description and Syntax	Examples
FindInMap	Retrieve the value for the specified key from a map that's defined in this template.  Fn::FindInMap: [mapName, submapName, keyName]	usableStorage: "Fn::FindInMap": ["dbMap", "size", {"Fn::GetParam": projectName}]
GetAtt	Retrieve the value of the runtime attribute from a specific resource. The attribute name supports a dot notation to access nested attributes within a JSON payload.  Fn::GetAtt: [resourceName, attributeName]	dbServiceName: { "Fn::GetAtt": [HRDatabase, serviceName] }  shape: { "Fn::GetAtt": [HRWeb, options.clusters.shape] }
GetParam	Retrieve the value of a template parameter.  Fn::GetParam: parameterName	network: { "Fn::GetParam": networkName }  description: { "Fn::GetParam": serviceName }  container: { "Fn::GetParam": backupStorage.cloudStorageCo ntainer }
GetRandom	Generate a string of random characters.  Fn::GetRandom: [size, characterSet]  Use <i>characterSet</i> to limit the characters that are allowed in the generated string: <ul style="list-style-type: none"> <li>ALPHANUMERIC - The characters A-Z, a-z, 0-9</li> <li>ALPHABET - The characters A-Z, a-z</li> <li>NUMERIC - The characters 0-9</li> <li>A custom sequence of characters. For example: [ \$, !, #, &amp; ]</li> </ul>	password: { "Fn::GetRandom": [10, "ALPHANUMERIC" ] }  dbServiceName: { "Fn::Join": [ "", [ "DB", { "Fn::GetRandom" : [8, "ALPHABET" ] } ] ] }

Function	Description and Syntax	Examples
GetResource	Retrieve a resource's definition.  Fn::GetResource: <i>resourceName</i>	configs: - name: DB1-dbScripts config: { "Fn::GetResource": dbScripts }  serviceName: { "Fn::GetTimestamp": "DB'yyMMdd" }
GetTimestamp	Return the current time in the specified format.  Fn::GetTimestamp: <i>format</i>  See <a href="#">SimpleDateFormat</a> for the available date and time options in the format string.	For example: DB180521  description: { "Fn::GetTimestamp": "Created on' EEE, MMM dd" }  For example: Created on Wed, May 21
GreaterThan	Compare a sequence containing two expressions and return true if the first expression evaluates to a value that is greater than the value of the second expression; otherwise return false.  Fn::GreaterThan: [ <i>exp1</i> , <i>exp2</i> ]	"Fn::GreaterThan": [{"Fn::GetParam": clusterSize}, 1]
If	Evaluate a global condition defined in this template. If the specified condition evaluates to true, return the second item in the sequence; otherwise return the third item in the sequence.  Fn::If: [ <i>conditionName</i> , <i>trueValue</i> , <i>falseValue</i> ]	shape: { "Fn::If": ["prodEnv", "oc4", "oc3"] }

Function	Description and Syntax	Examples
Join	<p>Combine a sequence of values into a single string, and using the specified delimiter character between the values.</p> <p>The function takes a sequence (delimiter, values) as its argument, and the values themselves are a sequence. To simply join the values without a delimiter specify an empty string for the first argument.</p> <pre>Fn::Join: [delimiterChar, [value1, value2, valueN]]</pre>	<pre>serviceName: { "Fn::Join": ["_", [HRApp, "Fn::GetParam": userName]] }</pre> <p>If the <code>userName</code> parameter's value is <code>bill5</code>, the <code>serviceName</code> parameter would be <code>HRApp_bill5</code>.</p>
LessThan	<p>Compare a sequence containing two expressions and return <code>true</code> if the first expression evaluates to a value that is less than the value of the second expression; otherwise return <code>false</code>.</p> <pre>Fn::LessThan: [exp1, exp2]</pre>	<pre>"Fn::LessThan": [{"Fn::GetParam": clusterSize}, 2]</pre>
Match	<p>Return <code>true</code> if a string matches a regular expression; otherwise return <code>false</code>.</p> <pre>Fn::Match: [regularExpression, string]</pre>	<p>This expression returns <code>true</code> if the <code>region</code> parameter begins with the text <code>us</code>:</p> <pre>"Fn::Match": ["^us", {"Fn::GetParam": region}]</pre> <p>This expression returns <code>true</code> if the <code>schemaName</code> parameter contains only letters and the numbers 0-3:</p> <pre>"Fn::Match": ["^(?=.*?[A-Z]) (?=.*?[a-z])(?=.*?[0-3])[a- zA-Z]([a-zA-Z0-3_#])+\$", {"Fn::GetParam": schemaName}]</pre>
Not	<p>Negate the boolean value of an expression. If the expression evaluates to <code>true</code>, return <code>false</code>.</p> <pre>Fn::Not: booleanExpression</pre>	<pre>"Fn::Not": {"Fn::Equals": [{"Fn::GetParam": "userName"}, ""]}</pre>

Function	Description and Syntax	Examples
Or	Return true if at least one expression in the sequence evaluates to true; otherwise return false.  <code>Fn::Or: [exp1, exp2, expN]</code>	<code>"Fn::Or": [{"Fn::Equals": [{"Fn::GetParam": "shape"}, "oc3"]}, {"Fn::Equals": [{"Fn::GetParam": "shape"}, "oc4"]}]</code>
Replace	Dynamically find and replace one or more values in a string ( <i>body</i> ). The replacements (substitutions) object can contain one or more name/value attributes and their values can be static or dynamic.  <code>Fn::Replace: {"body" : string, "substitutions" : {find:replace, find:replace, ...}}</code>	<code>appURL: { "Fn::Replace": {"body" : "host:port/myapp", "substitutions" : {"host" : "Fn::GetParam": "publicIP", "port" : "7002"}} }</code>  If the <code>publicIP</code> parameter's value is 192.0.2.10, the <code>appURL</code> parameter would be 192.0.2.10:7002/myapp.
Select	Select an object from a list of indexed objects, by passing in the index of the desired object.  <code>Fn::Select: [indexToFetch, objectList]</code>	<code>serverHost: { "Fn::Select": [0, "Fn::Split": [":", "host123:9001"]] }</code>  The <code>serverHost</code> parameter is assigned the value <code>host123</code> .
Size	Return the number of characters in the specified string, or the number of elements in the specified sequence.  <code>Fn::Size: stringOrSequence</code>	<code>"Fn::Size": {"Fn::GetParam": dbName}</code>  If the <code>dbName</code> parameter is the string "mydb", the value returned is 4. If the parameter is a sequence ["db1", "db2"], the value returned is 2.

Function	Description and Syntax	Examples
Split	<p>Split a single string into a sequence of strings by using the specified delimiter character.</p> <p>The function takes a sequence (delimiter, string) as its argument, and returns a sequence.</p> <p>You can also use the <code>Select</code> function on the result of <code>Split</code> to retrieve a specific item in the sequence.</p> <pre>Fn::Split: [delimiterChar, string]</pre>	<pre>databases: { "Fn::Split": ["_", "mydb1_mydb2"] }</pre> <p>The <code>databases</code> parameter is assigned the sequence <code>[mydb1, mydb2]</code>.</p>

## List of Implicit Template Parameters

Oracle Cloud Stack template authors can use several predefined template parameters when creating stack resources. These parameters do not have to be explicitly defined within a template, nor do users of a template have to specify them in the list of parameters.

Parameter Name	Description	Example
<code>serviceName</code>	The name of the stack being created. If you want to define constraints on the stack's name, such as the number of characters, create a parameter with this name in your template.	<code>Fn::GetParam: serviceName</code>
<code>identityDomain</code>	The Oracle Cloud identity domain in which this stack is being created	<code>Fn::GetParam: identityDomain</code>
<code>userName</code>	The name of the Oracle Cloud user who is creating this stack	<code>Fn::GetParam: userName</code>

# 3

## Managing Templates with Oracle Cloud Stack

Templates in Oracle Cloud Stack provide a blueprint for the creation of cloud stacks, and orchestrate the creation of cloud resources. You can create new templates, copy or import existing ones, and publish them.

### Topics

- [About the Template Life Cycle](#)
- [Configuring Template Storage](#)
- [Viewing a Template Resource Topology](#)
- [Copying a Template](#)
- [Editing a Template](#)
- [Validating a Template](#)
- [Publishing a Template](#)
- [Importing a Template](#)
- [Exporting a Template](#)
- [Deleting a Template](#)

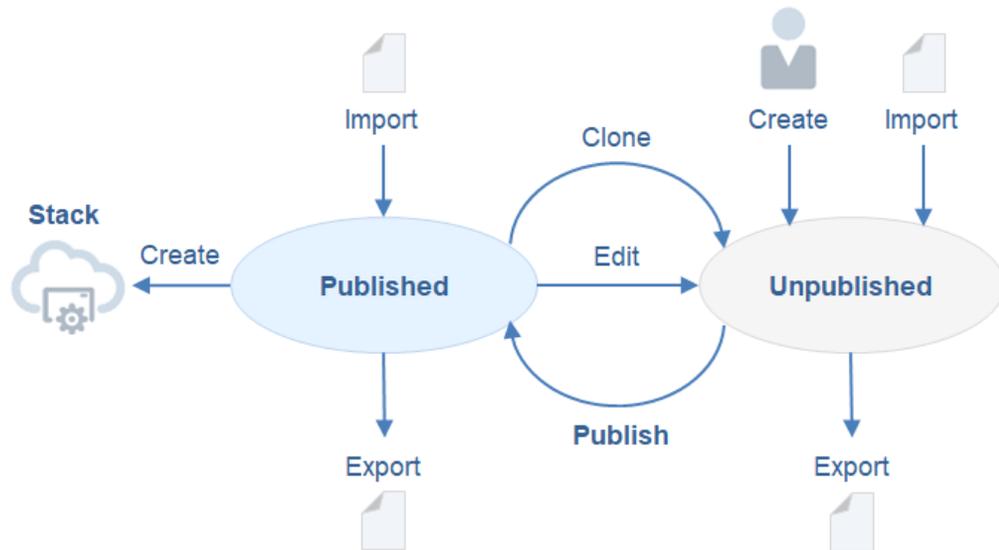
### About the Template Life Cycle

A template in Oracle Cloud Stack is a YAML document that's managed according to a standard life cycle.

A template is in one of two states:

- **Published** — Users in this cloud account can create stacks from this template.
- **Unpublished** — This template is still in-progress, and is not available for stack creation.

These states and their supported operations are illustrated in the following diagram:



You can edit templates by using the console, or edit them offline on your local computer.

### Editing with the Console: Typical Workflow

Task	Description	More Information
Create a template	Choose one of these options: <ul style="list-style-type: none"> <li>Use the console to create a new template.</li> <li>Use the console to copy an existing a template.</li> </ul> For a list of standard Oracle templates that are available to all accounts, see <a href="#">About Oracle Stack Templates</a> .	<ul style="list-style-type: none"> <li><a href="#">Creating a Template</a></li> <li><a href="#">Copying a Template</a></li> </ul>
Edit a template	Use the console to add parameters, resources, and other elements to your template.	<a href="#">Editing a Template</a>
Validate a template (optional)	Validate the syntax of your template before publishing it.	<a href="#">Validating a Template</a>
Publish a template	Publish your template so that it can be used to create cloud stacks.	<a href="#">Publishing a Template</a>
Delete a template	Delete your template when it's no longer required.	<a href="#">Deleting a Template</a>

## Editing Offline: Typical Workflow

Task	Description	More Information
Export a template	Export an existing template to your local computer. For a list of standard Oracle templates that are available to all accounts, see <a href="#">About Oracle Stack Templates</a> .	<a href="#">Exporting a Template</a>
Edit a template	Use your own text editor to add parameters, resources, and other elements to your template.	<a href="#">Editing a Template</a>
Validate a template (optional)	Validate the syntax of your template before importing it.	<a href="#">Validating a Template</a>
Import a template	Import the updated template back into Oracle Cloud Stack.	<a href="#">Importing a Template</a>
Delete a template	Delete your template when it's no longer required.	<a href="#">Deleting a Template</a>

## Configuring Template Storage

Specify the location in which Oracle Cloud Stack stores your template files.

The Template Builder in the Oracle Cloud Stack console saves your unpublished templates to a storage container in Oracle Cloud Infrastructure Object Storage Classic.

Oracle Cloud Stack can create a storage container for you, or you can use an existing one. To create a storage container without using Oracle Cloud Stack, see [Creating Containers in Using Oracle Cloud Infrastructure Object Storage Classic](#).

1. Sign in to the Oracle Cloud Stack console and navigate to the **Templates** page.
2. Click **Unpublished**.
3. Click **Configure cloud storage** .

### Note:

If this is the first time anyone in this cloud account has accessed this page, you do not need to click **Configure cloud storage**. The Cloud Storage Credentials dialog is displayed automatically.

4. For **Cloud Storage Container**, enter the URL for an Oracle Cloud Infrastructure Object Storage container.

Format: `rest_endpoint_url/containerName`

The default container name is `stackTemplates`.

See [Finding the REST Endpoint URL for Your Service Instance in Using Oracle Cloud Infrastructure Object Storage Classic](#).

5. Optional: If you select the option **Create a new container**, Oracle Cloud Stack will create the specified container if it doesn't already exist.
6. For **Username** and **Password**, enter the credentials of an Oracle Cloud user who has write access to the specified storage container.

If you selected **Create a new container**, then this user must have the Storage Administrator role.

7. Click **Apply Changes**.

If you change the storage location at a later time, Oracle Cloud Stack will **not** automatically move your template files from the previous container. To move or copy files between containers, see Typical Workflow for Managing Objects in *Using Oracle Cloud Infrastructure Object Storage Classic*.

## Viewing a Template Resource Topology

Use the Oracle Cloud Stack console to explore the topology of a published stack template as an interactive diagram.

A template contains one or more cloud resources. Dependencies control the order in which Oracle Cloud Stack provisions these resources when a new cloud stack is created from a template. For more information, see [What are Dependencies](#).

Oracle Cloud Stack helps you to visualize these resource dependencies by inspecting a template's definition and creating a diagram that consists of icons and arrows.

1. Access the Oracle Cloud Stack console, and then click **Templates**.
2. Click the name or icon of a published template.

### Tip:

To view an unpublished (draft) template, click **Unpublished**, and then click the name of the template.

To view the template that was used create an existing stack, from the **Stacks** page, click the **Template** link for the stack.

The **Topology** page of the Stack Template Details dialog is displayed. This page has an interactive diagram that depicts the resources created by this template. The diagram also uses arrows to show the dependencies between resources and therefore the order in which they will be created by Oracle Cloud Stack.

3. Position your mouse over an icon in the template diagram.

A dialog box displays the name of this cloud resource as defined in the template, as well as its resource type. Each type of resource is represented by a different icon.

In addition, the diagram is updated so that only those arrows that are linked to this resource (both incoming and outgoing) are displayed.

4. Position your mouse over an arrow that links two of the resources in this template.

A dialog box describes the relationship between these two resources.

The direction of the arrow indicates the direction of the dependency. For example, if an arrow begins at `ResourceA` and ends at `ResourceB`, `ResourceA` depends on `ResourceB`.

5. Optional: Click the **Template** tab.

This page displays the definition of this template as a YAML document.

6. Optional: Click **Export** to save this template as a file to your local machine.
7. When you are finished viewing this template, click **Done** to return to the **Templates** page.

## Copying a Template

Copy an existing template in Oracle Cloud Stack as an alternative to creating a new template.

Before copying a template for the first time, configure the storage location for unpublished template files. See [Configuring Template Storage](#).

### Topics

- [Cloning an Oracle Template](#)
- [Copying an Unpublished Template](#)
- [Copying a Custom Published Template](#)

## Cloning an Oracle Template

Create a copy of an Oracle-defined Oracle Cloud Stack template.

To learn more about these templates, see [About Oracle Stack Templates](#).

1. Access the Oracle Cloud Stack console, and then click **Templates**.
2. To the right of the Oracle template that you want to copy, click **Clone this template** .

The new template is added to the list of unpublished (draft) templates and it is displayed in the Template Builder. By default, the new template has the same name as the source template, but the text “Oracle” is removed.

3. In the text field at the top of the page, update the default name for this template.
4. Optional: Click , and then update the **templateVersion** and **templateDescription**.
5. Click **Save the current template** .

## Copying an Unpublished Template

Copy an unpublished Oracle Cloud Stack template.

1. Access the Oracle Cloud Stack console, and then click **Templates**.
2. Click **Unpublished**.
3. Click the template that you want to copy.

4. Click **Copy the current template**  .  
The new template is displayed in the Template Builder and by default has the name `SourceTemplateName-copy`.
5. In the text field at the top of the page, update the default name for this template.
6. Optional: Click , and then update the **templateVersion** and **templateDescription**.
7. Click **Save the current template**  .

## Copying a Custom Published Template

Copy a previously-published, user-defined template by editing it in the Oracle Cloud Stack console.

1. Access the Oracle Cloud Stack console, and then click **Templates**.
2. To the right of the template that you want to copy, click **Edit this template**  .  
The new template is added to the list of unpublished (draft) templates and it is displayed in the Template Builder. By default the new template has the same name and version as the source template.
3. To the left of the template's name, click  .
4. Choose from one of these options:
  - If the new template will replace the source template, modify the **templateVersion**. Do not modify the template's name.
  - If the new template will be used in addition to the source template, modify the template's name, and optionally modify the **templateVersion**.In order to publish a template, it must have a unique name or version.
5. Optional: Update the **templateDescription**.
6. Click **Save the current template**  .

## Editing a Template

Modify an existing stack template by either using the Oracle Cloud Stack console or by editing the template file offline.

 [Tutorials](#) are also available to help you customize an existing template.

### Topics

- [Editing with the Console](#)
- [Editing Offline](#)

## Editing with the Console

The Oracle Cloud Stack console provides a web-based Template Builder and YAML editor.

To edit a published template with the console, you must first create a copy of it. See [Copying a Template](#).

1. Access the Oracle Cloud Stack console, and then click **Templates**.
2. Click **Unpublished**.
3. Click the template that you want to edit.

The Template Builder is displayed.

4. Edit the template.

For more information, refer to the topics in [Creating Templates with Oracle Cloud Stack](#).

To directly edit the template's source, click **Open YAML editor** .

### Tip:

If an existing resource is not visible on the canvas area, click **Snap to grid** , and then click **Sort resources** .

5. Click **Save the current template** .

After making your modifications, you must publish the template before cloud users can create stacks from it. See [Publishing a Template](#).

## Editing Offline

Export an Oracle Cloud Stack template if you want to edit it on your local computer in your choice of text editor.

Export the template file, modify its contents, and then import the updated file. See:

- [Exporting a Template](#)
- [Editing Template Files](#)
- [Importing a Template](#)

The values of the `templateName` and `templateVersion` attributes in the template file are used to uniquely identify a template in Oracle Cloud Stack. The actual name of the template source file is not used by Oracle Cloud Stack to identify a template.

If you are editing a published template, modify the `templateVersion` attribute. You cannot publish a template that has the same name and version as an existing published template. For example:

```
---
template:
  templateName: MyTemplateName
```

```
templateVersion: 2.0
templateDescription: My updated Oracle cloud stack template
. . .
```

## Validating a Template

As you develop an Oracle Cloud Stack template, you can use the validation feature to help identify template errors before you publish or import the template.

Oracle Cloud Stack validates your template's syntax. It does not verify that:

- Your resource definitions adhere to the REST APIs of the underlying cloud services.
- Your use of the `GetAtt` function adheres to the REST APIs of the underlying cloud services.

## Validating a Template with the Console

Use the web console to validate an unpublished template in Oracle Cloud Stack.

To validate a published template with the console, you must first create a copy of it. See [Copying a Template](#).

1. Access the Oracle Cloud Stack console, and then click **Templates**.
2. Click **Unpublished**.
3. Click the template that you want to validate.  
The Template Builder is displayed.
4. Click **Validate current template** .

## Validating a Template with the CLI

When you import a template into Oracle Cloud Stack the template's syntax is automatically validated. You can also optionally validate your template file and its template parameters prior to importing it into the cloud by using the CLI.

Execute the `stack validate-template` command.

Provide the fully-qualified path of the template YAML file on your local file system. You can also test the template against specific input parameters, if desired. Template parameters are separated by spaces:

```
psm stack validate-template -f filePath -p param1:value1 param2:value2 ...
paramN:valueN
```

For example:

```
psm stack validate-template -f /home/oracle/cloud/dev_stack_template.yaml -
p shape:OC3 clusterSize:2
```

For more information about the CLI commands for Oracle Cloud Stack, see [psm stack Commands](#).

## Publishing a Template

Publish an Oracle Cloud Stack template so that you and other cloud users can create stacks from the template.

Oracle recommends that you validate your template prior to publishing it. See [Validating a Template](#).

You cannot publish a template that has the same name and version as an existing published template. If a published template already exists with the same name and version, you must modify the name and/or version of your template prior to publishing it.

If you publish a template that has the same name as an existing published template, but has a newer version, your template will replace the existing published template. New stacks will be created from the latest version of the template. Any existing stacks that were created from previous versions of this template are not affected.

### Tip:

If you want to create a stack using an older version of a template, you must edit and then publish the template again prior to creating the stack. Alternatively, you can publish these versions as separate templates with different names.

1. Access the Oracle Cloud Stack console, and then click **Templates**.
2. Click **Unpublished**.
3. To the right of the template that you want to publish, click **Publish this template**  
.

You can also publish a template while editing it within the Template Builder. Be sure to save your changes before publishing.

4. Click **Published**.
5. Verify the presence on your template.

You can also use the **Search** input field to locate your template.

To create a stack from your published template, see [Creating a Cloud Stack](#).

## Importing a Template

Import a custom template file into Oracle Cloud Stack.

The value of the `templateName` attribute in the file is used as the name of the template in Oracle Cloud Stack. The actual name of the template source file is not used by Oracle Cloud Stack to identify a template. If you import a template with the same name as an existing template but with a newer `templateVersion`, you replace the existing one. Any existing stacks that were created from previous versions of this template are not affected. See [Editing a Template](#).

## Topics

- [Importing a Published Template with the Console](#)
- [Importing an Unpublished Template with the Console](#)
- [Importing a Template with the CLI](#)

## Importing a Published Template with the Console

Use the web console to import a template file into Oracle Cloud Stack. The template is automatically published and is available to all users in this account for the creation of cloud stacks.

1. Access the Oracle Cloud Stack console, and then click **Templates**.
2. Click **Import**.
3. Select one of the following options:
  - **Template File** — Click **Choose File** and select the YAML template file on your local machine.
  - **Template URL** — Enter the URL to the YAML template file.
4. Click **Import**.

The format and contents of the template file are automatically validated. If any validation errors are found, you must edit the template file, correct the problems and then import again.

5. Verify the presence of your template on the **Templates** page.

You can also use the **Search** input field to locate your template.

## Importing an Unpublished Template with the Console

Use the web console to import a unpublished (draft) template file into Oracle Cloud Stack. The template cannot be used to create stacks until it is published.

Before importing an unpublished template for the first time, see [Configuring Template Storage](#).

1. Access the Oracle Cloud Stack console, and then click **Templates**.
2. Click **Unpublished**.
3. Click **Add Template**, and then select **Import existing Template**.
4. Navigate to and select the YAML template file on your local machine.
5. Verify the presence of your template on the Unpublished Templates page.

You can also use the **Search** input field to locate your template.

After making any modifications to the imported template with the console, you can publish it. See [Publishing a Template](#).

## Importing a Template with the CLI

Use the CLI to import a template file into Oracle Cloud Stack.

1. Execute the `stack import-template` command.

Provide the fully-qualified path of the template YAML file on your local file system:

```
psm stack import-template -f filePath
```

For example:

```
psm stack import-template -f /home/oracle/cloud/  
myappdev_stack_template.yaml
```

```
"status": "Template [myappdev:1.0.0] saved successfully"
```

The format and contents of the template file are automatically validated. Any validation errors are provided in the output of this command.

2. Execute the `stack describe-template` command to verify the presence of your new template:

```
psm stack describe-template -n name
```

For example:

```
psm stack describe-template -n myappdev
```

#### Note:

You can also view a list of all templates in this account by using the `stack list-templates` command. The output of both of these commands include a list of stack names that were created from this template.

You can also validate a template file without importing it. See [Validating a Template](#).

For more information about the CLI commands for Oracle Cloud Stack, see [psm stack Commands](#).

## Exporting a Template

Export an existing stack template as a YAML document from Oracle Cloud Stack. After exporting the template you can update it or use it to create a new template.

### Topics

- [Exporting a Published Template with the Console](#)
- [Exporting an Unpublished Template with the Console](#)
- [Exporting a Template with the CLI](#)

## Exporting a Published Template with the Console

Use the web console to export a published template file from Oracle Cloud Stack.

1. Access the Oracle Cloud Stack console, and then click **Templates**.
2. To the right of the template that you want to export, click **Download this template** .
3. Follow the instructions in your web browser to save the YAML template file on your local machine.

## Exporting an Unpublished Template with the Console

Use the web console to export an unpublished (draft) template file from Oracle Cloud Stack.

1. Access the Oracle Cloud Stack console, and then click **Templates**.
2. Click **Unpublished**.
3. Click the template that you want to export.
4. Click **Export the current template** .
5. Enter a file name for the template, and then click **OK**.
6. Follow the instructions in your web browser to save the YAML file on your local machine.

## Exporting a Template with the CLI

Use the CLI to export a template file from Oracle Cloud Stack.

1. Identify the name of the template to export. Execute the `stack list-templates` command.
2. Execute the `stack export-template` command. Provide the name of the template and redirect the output to a file:

```
psm stack export-template -n templateName > fileName
```

For example:

```
psm stack export-template -n MyTemplateName > MyTemplateName.yaml
```

The command results in a YAML document that contains the template definition:

```
---
template:
  templateName: MyTemplateName
  templateVersion: 1.0
  templateDescription: My Oracle cloud stack template
  . . .
```

After exporting the template you can modify its definition using any text editor. If you want to create a new template from the exported document modify the document's `templateName` property. After you're finished changing the template, import the document back into Oracle Cloud Stack. See [Importing a Template](#).

For more information about the CLI commands for Oracle Cloud Stack, see [psm stack Commands](#).

## Deleting a Template

Use Oracle Cloud Stack to delete a stack template.

### Note:

- User-created templates can be deleted. Oracle templates cannot be deleted.
- Deleting a template does not affect any existing stacks that were created from the template.

## Deleting a Template with the Console

Use the web console to delete a template file from Oracle Cloud Stack.

1. Access the Oracle Cloud Stack console, and then click **Templates**.
2. Optional: To delete an unpublished (draft) template, click **Unpublished**.
3. To the right of the template that you want to delete, click **Delete this template** .
4. When prompted for confirmation, click **OK**.

## Deleting a Template with the CLI

Use the CLI to delete an existing template from Oracle Cloud Stack.

1. Identify the name of the template to delete. Execute the `stack list-templates` command.
2. Execute the `stack delete-template` command. Supply the name of the template:

```
psm stack delete-template -n name
```

For example:

```
psm stack delete-template -n MyTemplate
```

For more information about the CLI commands for Oracle Cloud Stack, see [psm stack Commands](#).

# 4

## Creating and Managing Stacks with Oracle Cloud Stack

Use Oracle Cloud Stack to create one or more cloud stacks from an existing template.

You can use one of the standard Oracle templates, or use a custom template that you previously created or imported into Oracle Cloud Stack.

- For more information on using a specific Oracle template, see [About Oracle Stack Templates](#).
- For more information about custom templates, see [Creating a Template](#).

### Topics

- [Typical Workflow for Managing Stacks](#)
- [Creating a Cloud Stack](#)
- [Viewing the Details of a Cloud Stack](#)
- [Starting and Stopping a Cloud Stack](#)
- [Using Tags with Cloud Stacks](#)
- [Starting and Stopping a Resource in a Cloud Stack](#)
- [Managing SSH Access for a Resource in a Cloud Stack](#)
- [Managing Access Rules for a Resource in a Cloud Stack](#)
- [Deleting a Cloud Stack](#)

## Typical Workflow for Managing Stacks

To manage the lifecycle of a cloud stack with Oracle Cloud Stack, refer to the typical task workflow.

Task	Description	More Information
Create a stack	Create a cloud stack from an existing template.	<a href="#">Creating a Cloud Stack</a> .
View stack details	Periodically check the status and configuration of your cloud stacks.	<a href="#">Viewing the Details of a Cloud Stack</a>
Start and stop a stack	Start and stop all of the resources within a cloud stack.	<a href="#">Starting and Stopping a Cloud Stack</a>
Manage stack resources	Configure and manage the individual cloud resources that make up a cloud stack.	<ul style="list-style-type: none"><li>• <a href="#">Starting and Stopping a Resource in a Cloud Stack</a></li><li>• <a href="#">Managing SSH Access for a Resource in a Cloud Stack</a></li><li>• <a href="#">Managing Access Rules for a Resource in a Cloud Stack</a></li></ul>

Task	Description	More Information
Delete a stack	Delete an existing cloud stack.	<a href="#">Deleting a Cloud Stack</a>

## Creating a Cloud Stack

Create a cloud stack in Oracle Cloud Stack to provision a group of related cloud services. After selecting a template and initiating the creation of a stack, monitor its progress and status.

 [Video](#)

 [Tutorials](#) are also available to help you create your first cloud stack.

You can use one of the standard Oracle stack templates, or use a custom template that you previously created or imported into Oracle Cloud Stack.

- For more information on using a specific Oracle template, see [About Oracle Stack Templates](#).
- For more information about custom templates, see [Creating a Template](#).

You cannot create a stack from a custom template that is in the unpublished state. See [Publishing a Template](#).

In order to create a cloud stack you must have a valid subscription to all of the Oracle Cloud services of which the stack is comprised. For example, in order to use a stack template that includes an Oracle Java Cloud Service instance, your Oracle Cloud account must have a subscription to Oracle Java Cloud Service and it also must have the appropriate entitlements to use the desired features of Oracle Java Cloud Service. If these requirements are not met, you will receive a `Forbidden` error message.

A stack template defines zero or more named template parameters. Values for these parameters can be provided when a stack is created. A template parameter may be mandatory (does not have a default value) or optional. A parameter's value can also be limited to certain values or have to meet certain requirements.

If the creation of a stack fails, Oracle Cloud Stack can take one of these actions:

- **Rollback** (default): Any resources that were created as part of this stack will be deleted.
- **Retain**: For troubleshooting purposes, any resources that were created as part of this stack are not deleted. See [Troubleshooting for Oracle Cloud Stack](#).

## Creating a Cloud Stack with the Console

Use the web console to create a cloud stack in Oracle Cloud Stack.

1. Sign in to the Oracle Cloud Stack console and navigate to the **Stacks** page.
2. Click **Create Stack**.

Alternatively, go to the **Templates** page and click the  icon to the right of the template you want to use.

The Create New Oracle Cloud Stack — Details page is displayed.

3. Enter a **Name** and optional **Description** for the stack.  
Hyphens are not permitted in a stack's name.
  4. Select a **Template** from which to create the stack.
  5. Optional: For **Notification Email**, enter an email address that should receive a message when the stack's creation is complete.
  6. Optional: For **Tags**, select one or more existing tags to assign to this stack, and/or click **Create a tag**  to create and assign new tags to this stack.  
Tags can be simple values or `key:value` pairs.
  7. Optional: Select the **On Failure Retain Resources** check box to retain (not delete) any successfully created resources, in the event the stack creation fails.
  8. Enter values for the remaining template parameters. The available parameters are specific to the template you selected.
    - Mandatory parameters are indicated with an asterisk.
    - Read-only parameters cannot be modified.
    - Click **Help**  next to a parameter to learn more information about it.
  9. For any secure shell (SSH) key parameters, do the following:
    - a. Click the **Edit** button.
    - b. Choose from one of these options:
      - Select **Key File Name**. Click **Choose File** and select a public key file on your local machine.
      - Select **Key Value** and paste the value of the public key into the text area.
      - Select **Create a New Key** if you want Oracle Cloud to generate a public/private key pair for you.
    - c. Click **Enter**.
    - d. If you selected the **Create a New Key** option, click **Download** and save the generated keys to your local machine. Then click **Done**.
-  **Note:**

Be sure to download the generated keys to a permanent location. Without the private key file, you will not be able to access and administer the nodes that are created as part of this cloud stack.
10. Click **Next**.
  11. Review the values you provided for the template parameters and click **Confirm** to create your stack.

 **Tip:**

Click **Download Payload**  to save your parameter values as a JSON file. You can use this file to create a stack with the CLI or REST API.

Click **Back** if you want to change the parameter values. Click **Cancel** if you want to exit the creation wizard without creating a stack.

12. Your new stack appears on the Stacks page. Click its status to monitor the progress of your stack's creation.

You can also monitor this operation by:

- Expanding the **Stack Create and Delete History** section of the Stacks page.
- Navigating to the [Activity Page](#).

## Creating a Cloud Stack with the CLI

Use the CLI to create a cloud stack in Oracle Cloud Stack.

To discover the available parameters for a template, you can download it as a file and view its contents. See [Exporting a Template](#).

You can specify parameter values for a stack using either of these methods:

- Enter a single string, and separate each parameter with a space.

```
param1:value1 param2:value2 ... paramN:valueN
```

- Provide the location of a JSON file.

```
{
  "param1": "value1",
  "param2": "value2",
  ...,
  "paramN": "valueN"
}
```

To create a stack with the CLI:

1. Execute the `stack create` command.

Provide a name for the new stack and the name of the existing template. You can also optionally provide a description, template parameters and a failure action (ROLLBACK or RETAIN).

```
psm stack create -n name -d description -t template -f RETAIN|ROLLBACK
[-p parameters | -pf parameterFile]
```

For example:

```
psm stack create -n dev1 -t dev_template -f RETAIN -p shape:OC3
clusterSize:2
psm stack create -n dev2 -t dev_template -pf params_dev2.json
```

Hyphens are not permitted in a stack's name.

2. The output of the `stack create` command includes a job ID number. Use the `stack operation-status` command to periodically check the state of the create operation:

```
psm stack operation-status -j jobID
```

3. If you created the stack with the `-f RETAIN` or `--on-failure RETAIN` options and stack creation failed, you can attempt to correct the cause of the failure and then execute the `stack resume` command:

```
psm stack resume -n name
```

Oracle Cloud Stack will resume the creation of the stack and attempt to recreate the failed resource.

4. Execute the `stack describe` command to view details about the stack and the services of which it is comprised:

```
psm stack describe -n name
```

 **Note:**

You can also view a list of all cloud stacks in this account by using the `stack list` command.

For more information about the CLI commands for Oracle Cloud Stack, see [psm stack Commands](#).

## Viewing the Details of a Cloud Stack

View the resources and other attributes of your cloud stacks in Oracle Cloud Stack.

## Viewing Cloud Stacks with the Console

Use the web console to monitor your cloud stacks in Oracle Cloud Stack.

1. Sign in to the Oracle Cloud Stack console and navigate to the **Stacks** page.

The Stacks page lists the cloud stacks in this identity domain. A stack's icon is used to indicate its status and health. This page also shows the number of Oracle CPUs (OCPU), memory and block storage allocated to the resources in each stack.

 **Tip:**

To view the template definition that was used to create a stack, click its **Template** link.

2. Locate the stack you want to view.

If tags are assigned to stacks, you can also search for stacks using tag expressions and wild cards. See [Find Tags and Instances Using Search Expressions](#).

3. Click the name of the stack.

The Stack Details page is displayed.

4. Click **Stack Details** , located at the top of the page, for additional information about the stack.

5. Locate the *Resources* area of the Stack Details page.

This section lists the cloud resources of which this stack is made up. Different icons are used to distinguish different types of cloud resources, as well as their status.

For each resource, this page shows its Name and Type, along with the number of OCPUs, memory and block storage allocated to the resource (if applicable).

6. Click the **Name** link for a specific resource to go to the console for this cloud service.

Use a resource's service console for more detailed monitoring and troubleshooting.

 **Note:**

You may not be able to click on some types of cloud resources, such as Oracle Cloud Infrastructure Object Storage Classic containers.

7. At the top of the page, click the link with the name of your stack in order to return back to the Stack Details page in the Oracle Cloud Stack console.

8. Click the **Stacks** link at the top of the page to go back to the Stacks page.

9. Click the  icon to expand the *Stack Create and Delete History* area

This area lists the operations that Oracle Cloud Stack has executed on all stacks, or is in the process of executing.

10. Click the  icon for a specific operation to view its log.

 **Tip:**

You can also use the **Activity** page to monitor stack operations. To view only those operations performed by Oracle Cloud Stack, set the **Service Type** to Oracle Cloud Stack.

For more information about the console user interface see these topics:

- [Stacks Page](#)
- [Stack Details Page](#)

## Viewing Cloud Stacks with the CLI

Use the CLI to monitor your cloud stacks in Oracle Cloud Stack.

1. Execute the `stack list` command.

Use the output to identify the names of your existing cloud stacks. For example:

```
psm stack list

{
  "identityDomain": "MyIdentityDomain",
  "stacks": [
    {
      "serviceName": "MyStack1",
      "description": "",
      "serviceURI": "http://example.org:7200//paas/api/v1.1/
instancemgmt/MyIdentityDomain/services/stack/instances/MyStack1",
      "state": "READY",
      "stateDetail": "Ready",
      "identityDomain": "MyIdentityDomain",
      "createdBy": "joe@example.com",
      "creationJobId": "80161",
      "creationTime": "2016-07-26T15:48:18.366+0000",
      "lastModifiedTime": "2016-07-26T15:48:18.365+0000",
      "template": {
        "templateName": "MyTemplate",
        "templateVersion": "1.0.0",
        "templateURI": "http://example.org:7200//paas/api/v1.1/
instancemgmt/MyIdentityDomain/templates/cst/instances/MyTemplate/export?
version=1.0.0"
      }
    },
    {
      "serviceName": "MyStack2",
      "description": "",
      "serviceURI": "http://example.org:7200//paas/api/v1.1/
instancemgmt/MyIdentityDomain/services/stack/instances/MyStack1",
      "state": "READY",
      "stateDetail": "Ready",
      "identityDomain": "MyIdentityDomain",
      "createdBy": "joe@example.com",
      "creationJobId": "80161",
      "creationTime": "2016-07-27T15:48:18.366+0000",
      "lastModifiedTime": "2016-07-27T15:48:18.365+0000",
      "template": {
        "templateName": "MyTemplate",
        "templateVersion": "1.0.0",
        "templateURI": "http://example.org:7200//paas/api/v1.1/
instancemgmt/MyIdentityDomain/templates/cst/instances/MyTemplate/export?"
```

```
version=1.0.0"  
    }  
  }  
]  
}
```

2. To retrieve additional details about a specific cloud stack, use the CLI to execute the `stack describe` command.

```
psm stack describe -n stackName
```

For example:

```
psm stack describe -n MyStack1
```

For more information about the CLI commands for Oracle Cloud Stack, see [psm stack Commands](#).

## Starting and Stopping a Cloud Stack

Use Oracle Cloud Stack to start and stop all of the cloud resources within an existing stack. The dependencies between the resources within a stack determine the order in which they are started and stopped.

## Starting and Stopping a Cloud Stack with the Console

Use the web console to start or stop all of the resources in a stack.

1. Sign in to the Oracle Cloud Stack console and navigate to the **Stacks** page.  
You can also start and stop a stack from the [Stack Details Page](#) for a specific stack.
2. Click the  menu to the right of the stack you want to start or stop.
3. Select **Start** or **Stop**.
4. If any of the resources in your stack support start or stop parameters, you will be prompted to provide values for them.
5. When prompted for confirmation, click **OK**.

The stack's icon changes to indicate that it is in either the Starting or Stopping state. You can monitor the operation's progress from the [Activity Page](#).

## Starting and Stopping a Cloud Stack with the CLI

Use the CLI to start or stop all of the resources in a stack.

1. Execute the `stack start` or `stack stop` command. Supply the name of the cloud stack:

```
psm stack start -n name
psm stack stop -n name
```

2. The output of the command includes a job ID number. Use the `stack operation-status` command to periodically check the state of the start or stop operation:

```
psm stack operation-status -j jobID
```

For more information about the CLI commands for Oracle Cloud Stack, see `psm stack Commands`.

## Using Tags with Cloud Stacks



This topic does not apply to Oracle Cloud at Customer.

A tag is an arbitrary key or a key-value pair that you can assign to your cloud stacks in Oracle Cloud Stack. You can use tags to organize and categorize your stacks, and to search for them.

### Topics

- [Creating, Assigning, and Unassigning Tags](#)
- [Find Tags and Instances Using Search Expressions](#)

## Creating, Assigning, and Unassigning Tags

You can create and assign tags to Oracle Cloud Stack instances while creating the instances or later. When you no longer need certain tags for an instance, you can unassign them.

To assign tags to an instance or to unassign tags:

1. Navigate to the Overview page for the instance for which you want to assign or unassign tags.
2. Click **Manage this instance**  in the instance name bar at the top.
3. Select **Manage Tags** or **Add Tags**.

If any tags are already assigned, then the menu shows **Manage Tags**; otherwise, it shows **Add Tags**.

4. In the Manage Tags dialog box, create and assign the required tags, or unassign tags:
  - In the **Assign** section, in the **Tags** field, select the tags that you want to assign to the instance.
  - If the tags that you want to assign don't exist, then select **Create and Assign** in the **Tags** field, and click just above the field. Enter the required new tags in the **Enter New Tags** field.

- To unassign a tag, in the **Unassign** section, look for the tag that you want to unassign, and click the **X** button next to the tag.

 **Note:**

You might see one or more tags with the key starting with `ora_`. Such tags are auto-assigned and used internally. You can't assign or unassign them.

- To exit without changing any tag assignments for the instance, click **Cancel**.
- After assigning and unassigning tags, click **OK** for the tag assignments to take effect.

## Find Tags and Instances Using Search Expressions

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

### Search for Instances with Tags

From the Instances page of the web console, select **Tags**, and then enter a *search expression* in the **Search** field.

For example, you can search for the instances that are assigned a tag with the key `env` and any value starting with `dev` (example: `env:dev1`, `env:dev2`), by entering the search expression `'env':'dev%'`.

### Instances



Similarly, when you use the REST API to find tags or to find instances that are assigned specific tags, you can filter the results by appending the optional `tagFilter=expression` query parameter to the REST endpoint URL.

- To find specific tags: `GET paas/api/v1.1/tags/{identity_domain}/tags?tagFilter={expression}`
- To get a list of instances that are assigned specific tags: `GET paas/api/v1.1/instancemgmt/{identity_domain}/instances?tagFilter={expression}`

### Syntax and Rules for Building Tag-Search Expressions

- When using cURL to send tag-search API requests, enclose the URL in double quotation marks.

**Example:**

```
curl -s -u username:password -H "X-ID-TENANT-NAME:acme"
"restEndpointURL/paas/api/v1.1/instancemgmt/acme/instances?
tagFilter='env'"
```

This request returns all the tags that have the key `env`.

- Enclose each key and each value in single quotation marks. And use a colon (:) to indicate a key:value pair.

**Examples:**

```
'env'
'env': 'dev'
```

- You can include keys or key:value pairs in a tag-filtering expression.

Sample Expression	Description	Sample Search Result
'env'	Finds the tags with the key <code>env</code> , or the instances that are assigned the tags with that key.	The following tags, or the instances that are assigned any of these tags:  env:dev env:qa
'env': 'dev'	Finds the tag with the key <code>env</code> and the value <code>dev</code> , or the instances that are assigned that tag.	The following tag, or the instances that are assigned this tag  env:dev

- You can build a tag-search expression by using actual keys and key values, or by using the following wildcard characters.

% (percent sign): Matches any number of characters.

\_ (underscore): Matches one character.

Sample Expression	Description	Sample Search Result
'env': 'dev%'	Finds the tags with the key <code>env</code> and a value starting with <code>dev</code> , or the instances that are assigned such tags.  <b>Note:</b> When you use <code>curl</code> or any command-line tool to send tag-search REST API requests, encode the percent sign as <code>%25</code> .	The following tags, or the instances that are assigned any of these tags:  env:dev env:dev1

Sample Expression	Description	Sample Search Result
'env': 'dev_'	Finds the tags with the key <code>env</code> and the value <code>devX</code> where <code>X</code> can be any one character, or finds the instances that are assigned such tags.	The following tags, or the instances that are assigned any of these tags:  <code>env:dev1</code> <code>env:dev2</code>

- To use a single quotation mark (`'`), the percent sign (`%`), or the underscore (`_`) as a literal character in a search expression, escape the character by prefixing a backslash (`\`).

Sample Expression	Description	Sample Search Result
'env': 'dev \_%'	Finds the tags with the key <code>env</code> and a value starting with <code>dev_</code> , or the instances that are assigned such tags.	The following tags, or the instances that are assigned any of these tags:  <code>env:dev_1</code> <code>env:dev_admin</code>

- You can use the Boolean operators `AND`, `OR`, and `NOT` in your search expressions:

Sample Expression	Description	Sample Search Result
'env' OR 'owner'	Finds the tags with the key <code>env</code> or the key <code>owner</code> , or the instances that are assigned either of those keys.	The following tags, or the instances that are assigned <i>any of these tags</i> :  <code>env:dev</code> <code>owner:admin</code>
'env' AND 'owner'	Finds the instances that are assigned the tags <code>env</code> <i>and</i> <code>owner</code> .  <b>Note:</b> This expression won't return any results when used to search for tags, because a tag can have only one key.	The instances that are assigned <i>all of the following tags</i> :  <code>env:dev</code> <code>owner:admin</code>
NOT 'env'	Finds the tags that have a key other than <code>env</code> , or the instances that are assigned such tags.  <b>Note:</b> Untagged instances as well will satisfy this search expression.	The following tags, or the instances that are assigned <i>any of these tags</i> or no tags:  <code>owner:admin</code> <code>department</code>

Sample Expression	Description	Sample Search Result
<code>('env' OR 'owner') AND NOT 'department'</code>	Finds the tags that have the key <code>env</code> or the key <code>owner</code> but not the key <code>department</code> , or the instances that are assigned such tags.	The following tags, or the instances that are assigned <i>any of these tags</i> :  <code>env:dev</code> <code>owner:admin</code>

## Starting and Stopping a Resource in a Cloud Stack

For convenience, you can use Oracle Cloud Stack to start and stop individual cloud resources within an existing stack. You don't have to use a separate cloud interface to manage each type of resource.

### Starting and Stopping a Resource with the Console

Use the web console to start or stop a specific resource in a cloud stack.

1. Sign in to the Oracle Cloud Stack console and navigate to the **Stacks** page.
2. Click the stack's name.
3. Click the  menu to the right of the resource you want to start or stop.
4. Select **Start** or **Stop**.



**Note:**

Some resources may also support a **Restart** option. Resources of type Oracle Cloud Infrastructure Object Storage Classic cannot be started or stopped.

5. When prompted for confirmation, click **OK**.

## Managing SSH Access for a Resource in a Cloud Stack

Some Oracle Cloud resources can be accessed through the Secure Shell (SSH) protocol by using SSH keys. You typically provide a public key when you initially create these resources, but you can also use Oracle Cloud Stack to add more keys to an existing resource in your cloud stack.

Use the web console to add SSH keys to a resource in a cloud stack.

1. Sign in to the Oracle Cloud Stack console and navigate to the **Stacks** page.
2. Click the stack's name.

3. Click the  menu to the right of the resource you want to update.
4. Select **SSH Access**.  
The SSH Access page is displayed.

 **Note:**

Some cloud resources, such as Oracle Cloud Infrastructure Object Storage Classic containers, are not associated with SSH keys.

5. Click **Add New Key**.
6. Choose one of these options:
  - Select **Upload a new SSH Public Key value from file** and then click **Browse** to add a new SSH public key from an existing file.
  - Select **Key Value** and then paste the new public key value in the text area.
7. Click **Add New Key**.
8. To return to the Stack Details page, click the name of your stack in the bread crumb trail at the top of the page. Or click **Stacks** to return to the Stacks page.

## Managing Access Rules for a Resource in a Cloud Stack

Some Oracle Cloud resources use access rules to control access to the components that make up the resource, including access by other Oracle Cloud resources and by external clients. Oracle Cloud typically creates default access rules for a resource when it is provisioned, but you can also use Oracle Cloud Stack to create custom rules on existing resources.

Use the web console to add an access rule to a resource in a cloud stack.

1. Sign in to the Oracle Cloud Stack console and navigate to the **Stacks** page.
2. Click the stack's name.
3. Click the  menu to the right of the resource you want to update.
4. Select **Access Rules**.

 **Note:**

Some cloud resources, such as Oracle Cloud Infrastructure Object Storage Classic containers, are not associated with access rules.

5. Click **Create Rule**.
6. Enter a name for the access rule in the **Rule Name** field.  
Enter an optional description of the access rule in the **Description** field.

7. Select a **Source** for the rule. The available options will vary for each cloud resource, depending on its type and configuration.

For example, **PUBLIC-INTERNET** is any host on the internet.

Choose **<custom>** to provide a custom list of IP addresses from which traffic should be allowed. In the field that displays below, enter a comma-separated list of the subnets (in CIDR format, such as 192.0.2.1/24) or IPv4 addresses for which you want to permit access.

8. Select a **Destination** for the rule. The available options will vary for each cloud resource, depending on its type and configuration.
9. Enter the port or port range through which the source will access the destination in the **Destination Port(s)** field.
10. For **Protocol**, select **TCP** or **UDP**.
11. Click **Create**.
12. To return to the Stacks page, click **Stacks** in the bread crumb trail at the top of the page.

## Deleting a Cloud Stack

When an environment is no longer needed, delete the cloud stack with Oracle Cloud Stack in order to free up account resources.

Deleting a stack terminates all cloud resources that are components of the stack. Oracle Cloud Stack does not clean up any related resources that are not directly part of the stack. For example, if a stack resource is configured to use an existing object storage container for backups, these backups may not be deleted when the stack resource is deleted.

If a stack contains more than one resource, you must delete these resources with Oracle Cloud Stack. You cannot delete the resources individually from their respective service consoles or APIs.

## Deleting a Cloud Stack with the Console

Use the web console to delete a cloud stack from Oracle Cloud Stack.

1. Sign in to the Oracle Cloud Stack console and navigate to the **Stacks** page.
2. Click the  menu to the right of the stack you want to delete.
3. Select **Delete**.
4. Choose a value for **Force service deletion**:
  - **No (default)** – If an error occurs while a resource in this stack is being deleted, the rest of the stack will not be deleted.
  - **Yes** – The stack and all of its resources will be deleted, even if an error occurs during the delete operation.
5. If any of the cloud resources in your stack support additional deletion parameters, you will be prompted to provide values for these parameters.

6. Click **Delete**.

## Deleting a Cloud Stack with the CLI

Use the CLI to delete a cloud stack from Oracle Cloud Stack.

1. Execute the `stack delete` command. Supply the name of the cloud stack:

```
psm stack delete -n name
```

If any of the resources in your stack support deletion parameters, use the `-c` option to provide values for them.

By default, the stack and all of its resources will be deleted, even if an error occurs during the delete operation. If you specify the `-f false` option and an error occurs while a resource in this stack is being deleted, the rest of the stack will not be deleted.

2. The output of the `stack delete` command includes a job ID number. Use the `stack operation-status` command to periodically check the state of the delete operation:

```
psm stack operation-status -j jobID
```

For more information about the CLI commands for Oracle Cloud Stack, see [psm stack Commands](#).

# 5

## Frequently Asked Questions for Oracle Cloud Stack

Get answers to frequently asked questions for Oracle Cloud Stack.

### Topics:

- [What is Oracle Cloud Stack?](#)
- [What Oracle Cloud services can I create with Oracle Cloud Stack?](#)
- [What does Oracle Cloud Stack cost?](#)
- [How do I access Oracle Cloud Stack?](#)
- [What is a stack template?](#)
- [How do I create a custom stack template?](#)
- [Who uses Oracle Cloud Stack?](#)

### What is Oracle Cloud Stack?

Oracle Cloud Stack is a feature of Oracle Cloud that empowers users to automate the provisioning and management of multiple cloud services as a single unit called a stack.

See [About Oracle Cloud Stack](#).

### What Oracle Cloud services can I create with Oracle Cloud Stack?

Oracle Cloud Stack supports the creation of specific types of cloud resources, and Oracle is continually adding to this list.

See [About Oracle Stack Templates](#) and [List of Resource Types](#).

### What does Oracle Cloud Stack cost?

You do not have to explicitly obtain a subscription for Oracle Cloud Stack — it is free to use for all Oracle Cloud customers. However, in order to create a cloud stack you must have a valid subscription to all of the Oracle Cloud services of which the stack is comprised. For example, in order to use a stack template that includes an Oracle Java Cloud Service instance, your Oracle Cloud account must have a subscription to Oracle Java Cloud Service and it also must have the appropriate entitlements to use the desired features of Oracle Java Cloud Service.

### How do I access Oracle Cloud Stack?

The features of Oracle Cloud Stack are accessible from a web console, a command-line interface (CLI) or a REST API.

See [Accessing Oracle Cloud Stack](#).

**What is a stack template?**

Cloud stacks are created from templates. A template is a text file that defines what cloud resources to create, how to create them and any dependencies between them. Templates can also define input parameters that enable cloud users to further customize each stack. Oracle Cloud Stack provides you with standard templates for common scenarios, but you can also create your own templates.

See [About Oracle Stack Templates](#) and [About Template Components](#).

**How do I create a custom stack template?**

You can create a new template from scratch, or you can copy one of the standard Oracle templates and use it as a starting point.

The Oracle Cloud Stack console provides a graphical Template Builder tool to simplify the creation and maintenance of templates. By using the Template Builder and learning a few simple template functions, you can implement most cloud stack requirements. More experienced authors can use additional template features and functions in order to implement more advanced requirements like conditional logic.

If you prefer to use your own favorite editor for template development, you can also import and export template files to and from Oracle Cloud Stack.

See [Creating a Template](#) and [Editing Template Files](#).

**Who uses Oracle Cloud Stack?**

Oracle Cloud Stack gives IT Operations more control over how your development and testing environments are created and managed in Oracle Cloud, and also ensures that these tasks are easily repeatable. Your Operations team can create custom stack templates that meet their precise setup requirements. Working with templates requires some familiarity with YAML documents and the underlying REST APIs used to create cloud services.

Developer Operations (or DevOps) advocates automating the tasks of provisioning and managing environments for developers, testers and end users. Similarly, DevOps treats all of the resources used to provision these environments just like other code artifacts. Oracle Cloud Stack aligns with these IT objectives by providing users with stack templates, a command-line interface and a REST API.

# 6

## Troubleshooting for Oracle Cloud Stack

This section describes common problems that you might encounter when using Oracle Cloud Stack and explains how to solve them.

### Topics

- [Problems with Creating a Stack](#)
- [Problems with Deleting a Stack](#)
- [Problems with Importing a Template](#)
- [Get Help for a Specific Template](#)

### Problems with Creating a Stack

The creation of a cloud stack will fail if an error occurs during the creation of any of the resources that comprise the new stack. By default Oracle Cloud Stack rolls back (deletes) any resources that were created prior to the error.

Some Oracle-defined templates provide additional troubleshooting resources. See [Get Help for a Specific Template](#).

Follow this procedure in the Oracle Cloud Stack console to help identify the source of the stack creation error:

1. Create the same stack again, but select the **On Failure Retain Resources** checkbox on the Create New Oracle Cloud Stack — Stack Details page. When the creation of this stack fails, any resources that were created will not be automatically deleted.
2. From the main Stacks page of the console click **Activity**.
3. Verify that the **Service Type** is set to Oracle Cloud Stack. Search for and locate your failed Create Service operation.

4. Click  to the left of the operation to view its log.

5. From the log identify the resource in the stack that caused the failure and note its Service Type. For example, Oracle Java Cloud Service.

Alternatively, from the main Stacks page click the name of the failed stack. Then use the Stack Details page to identify the type of the failed resource.

6. From the same Activity page change the **Service Type** to the type of the failed resource. Search for and locate the failed Create Service operation for this specific cloud resource.

7. Click  to the left of the operation to view its log. Use this information to determine the exact cause of the resource creation failure.

8. Take an appropriate corrective action:

- If the resource creation problem was due to the value of a template parameter, you will need to delete the stack and create it again with a different parameter value. See [Deleting a Cloud Stack](#).
- If the resource creation problem was due to some external issue in the environment and you can correct it, Oracle Cloud Stack can attempt to create the failed resource again. Return to the Stacks page, click the  menu to the right of the stack and select **Resume**.

To troubleshoot a stack failure from the command line interface (CLI):

1. Turn on the debug logging level in the CLI:

```
psm log -l debug
```

2. Run the same `stack create` command again, but specify the `RETAIN` option:

```
psm stack create -f RETAIN . . .
```

The output includes the Oracle Cloud Stack job ID.

3. Monitor the progress of your stack by using the `stack operation-status` command. From the output of this command identify the resource in the stack that caused the failure.

For example:

```
"message":"Failed to create service MyApp-JCS. Please refer to the  
service activity logs for more details"
```

4. Use the `operation-status` or similar command for the specific cloud service that failed.

For example, if the failed resource is an Oracle Java Cloud Service instance refer to `psm jcs operation-status` in *PaaS Service Manager Command Line Interface Reference*.

Use this information to determine the exact cause of the resource creation failure.

5. Take an appropriate corrective action:
  - If the resource creation problem was due to the value of a template parameter, you will need to `delete` the stack and `create` it again with a different parameter value.
  - If the resource creation problem was due to some external issue in the environment and you can correct it, use the `stack resume` command. Oracle Cloud Stack will attempt to create the failed resource again:

```
psm stack resume -n YourStackName
```

## Problems with Deleting a Stack

Solve problems related to deleting a stack in Oracle Cloud Stack.

### Database is Stopped

Error message when attempting to delete a cloud stack:

```
Error while either uploading sql file or executing sql command on VM
```

Some cloud resources such as Oracle Java Cloud Service and Oracle SOA Cloud Service instances depend on a schema in a relational database. By default, before a service instance is deleted, the service attempts to delete its associated schema in the database. If the database is stopped or is otherwise not available, the deletion of the service instance with fail.

Choose from one of the following options:

- Start the database on which the resource depends, and then delete the stack.
- When you delete the stack, select the option **Force service deletion**.

## Problems with Importing a Template

Solve problems related to importing custom templates into Oracle Cloud Stack.

### Topics

- [Not a YAML Document](#)
- [Unsupported Template Parameter Type](#)
- [Invalid Indentation for a Sequence](#)
- [Resource Configuration is not Displayed in the Template Builder](#)

### Not a YAML Document

You can import a template by either uploading a YAML file from your local machine, or providing the location of a YAML file as a URL.

You receive this error message when uploading a template from a URL:

```
Template import failed! Mapping values are not allowed here.
```

Be sure that your URL resolves to a file of type YAML, and not HTML. For example, if your template is located in GitHub, enter its **Raw** URL.

### Unsupported Template Parameter Type

You receive this error message when importing a template:

```
Template cannot use reserved parameter type Type
```

Some parameter types are supported only in Oracle-defined templates. Oracle does not support these reserved parameter types in custom templates. Modify your template and use a different parameter type. See [Using Template Parameters](#).

### Invalid Indentation for a Sequence

You receive an error message similar to the following when you import a template:

```
Fn::Join - within template expects 2 members in json array. However, a
Fn::Join usage was found in template with 4 member(s)
```

A YAML sequence can be expressed in either of these equivalent formats:

```
[item1,item2,item3]
```

```
- item1
- item2
- item3
```

You can also nest sequences. For example:

```
- item1
- item2
-
  - item3a
  - item3b
```

Members of same sequence must have the same preceding whitespace. For example, when you use the `Join` function, you specify a sequence that contains two arguments. The second argument in this sequence is another sequence:

```
Fn::Join:
- "\n"
-
  - This is line 1
  - This is line 2
```

The following `Join` example is not valid YAML because the whitespace, or indentation, does not match the expected syntax of the function:

```
Fn::Join:
- "\n"
-
- This is
- not valid
```

### Resource Configuration is not Displayed in the Template Builder

The provisioning API for some Oracle Cloud services supports this older payload format for backwards compatibility:

```
parameters:  
  serviceParameters:  
    ...  
  componentParameters:  
    ...
```

Oracle Cloud Stack continues to support templates whose resources use this payload format. However, you cannot edit these resources using the Template Builder in the web console because the Template Builder supports only the most recent API payload:

```
parameters:  
  ...  
  components:  
    ...
```

If you want to modify a template that uses the deprecated provisioning API payload for a resource, choose from one of the following options:

- Do not use the Template Builder. Edit the template offline, and then import it into Oracle Cloud Stack. See [Editing Offline](#).
- From the Template Builder, edit the YAML source for the template, and then convert the resource(s) to the new payload format. See [Editing with the Console](#).

## Get Help for a Specific Template

Additional help resources are available for specific Oracle Cloud Stack templates.

Name	More Information
Oracle-Mobile-Bots-Standard-Rel	Troubleshoot Your OMCE Service in <i>Administering Oracle Mobile Cloud Enterprise in a Customer-Managed Environment</i>
Oracle-Mobile-Cloud-Enterprise-Template	

# 7

## Exploring the Oracle Cloud Stack Console

Get help on the various pages of the Oracle Cloud Stack console.

### Topics

- [Welcome Page](#)
- [Stacks Page](#)
- [Stack Details Page](#)
- [Published Templates Page](#)
- [Unpublished Templates Page](#)
- [Template Builder](#)
- [Activity Page](#)

## Welcome Page

The Welcome page is shown the first time you access the Oracle Cloud Stack console.

### What You Can Do From the Welcome Page

Use this page to access documentation, tutorials, videos and other getting started content for Oracle Cloud Stack.

### What You See on the Welcome Page

The following table describes the key information shown on the Welcome page.

Element	Description
<i>Header</i>	
	Access other Oracle Cloud services. Depending on how you signed up for Oracle Cloud, you can also access either the Oracle Cloud Infrastructure Console or the Oracle Cloud Infrastructure Classic Console.
	Access help for this page, including documentation, tutorials, videos and FAQs.
	Access help, accessibility options, console version information and sign-out.
<b>Welcome!</b>	Go to the <a href="#">Welcome Page</a> .
<b>Stacks</b>	Go to the <a href="#">Stacks Page</a> .
<b>Templates</b>	Go to the <a href="#">Published Templates Page</a> .
<b>Activity</b>	Go to the <a href="#">Activity Page</a> .
<i>Welcome</i>	

Element	Description
<b>Follow Tutorial</b>	Access the Oracle Cloud Stack <a href="#">Tutorials</a> .
<b>Go to Console</b>	Go to the <a href="#">Stacks Page</a> .

## Stacks Page

The Stacks page in the Oracle Cloud Stack console allows the user to create and manage cloud stacks, and to also view the recent stack creation history.

### What You Can Do From the Stacks Page

Use this page to perform the following tasks:

- [Creating a Cloud Stack](#)
- [Viewing the Details of a Cloud Stack](#)
- [Starting and Stopping a Cloud Stack](#)
- [Deleting a Cloud Stack](#)
- [Viewing a Template Resource Topology](#)

### What You See on the Stacks Page

The following table describes the key information shown on the Stacks page.

Element	Description
<i>Header</i>	
	Access other Oracle Cloud services. Depending on how you signed up for Oracle Cloud, you can also access either the Oracle Cloud Infrastructure Console or the Oracle Cloud Infrastructure Classic Console.
	Access help for this page, including documentation, tutorials, videos and FAQs.
	Access help, accessibility options, console version information and sign-out.
<b>Welcome!</b>	Go to the <a href="#">Welcome Page</a> .
<b>Stacks</b>	Go to the <a href="#">Stacks Page</a> .
<b>Templates</b>	Go to the <a href="#">Published Templates Page</a> .
<b>Activity</b>	Go to the <a href="#">Activity Page</a> .

Element	Description
Summary	Shows a summary of the cloud resources being used by Oracle Cloud Stack in this identity domain: <ul style="list-style-type: none"> <li>• <b>Stacks</b> — The total number of cloud stacks</li> <li>• <b>OCPUs</b> — The total number of Oracle CPUs allocated across all cloud stacks</li> <li>• <b>Memory</b> — The total amount of memory allocated across all cloud stacks</li> <li>• <b>Storage</b> — The total amount of block storage allocated across all cloud stacks</li> <li>• <b>Public IPs</b> — The total number of public IP addresses allocated across all cloud stacks</li> </ul>
<i>Stacks</i>	
<b>Search by stack or template</b>	Enter a full or partial stack name to filter the list of stacks to include only those that contain the string in their name. Enter a full or partial template name to filter the list of stacks to include only those that contain the string in the name of the template from which they were created.
<b>Create Stack</b>	Click to create a new stack. See <a href="#">Creating a Cloud Stack</a> .
	This icon indicates a running stack. Click the icon or the stack's name to view the <a href="#">Stack Details Page</a> .
	This icon indicates a stack that is partially running. Possible reasons include: <ul style="list-style-type: none"> <li>• Some of the resources in this stack were not created successfully.</li> <li>• The stack or some of its resources are in the process of being started or stopped.</li> </ul> Click the icon or the stack's name to view the <a href="#">Stack Details Page</a> .
	This icon indicates a stack that is being terminated. Click the icon or the stack's name to view the <a href="#">Stack Details Page</a> .
Template	The name of the template from which this stack was created. Click the template's name to view it. See <a href="#">Viewing a Template Resource Topology</a> .
Status	The current status of the stack. This field is not shown if the stack is in a healthy, running state.
Description, Created On	The stack's description (if one was provided) and the date on which it was created.
OCPUs, Memory, Storage	The number of Oracle CPUs, memory and block storage allocated to this stack.
	Click the menu for a specific stack to select from the following options: <ul style="list-style-type: none"> <li>• <b>Start</b>— See <a href="#">Starting and Stopping a Cloud Stack</a></li> <li>• <b>Stop</b>— See <a href="#">Starting and Stopping a Cloud Stack</a></li> <li>• <b>Delete</b> — See <a href="#">Deleting a Cloud Stack</a></li> <li>• <b>Resume</b> — Retry the creation of a failed stack, starting with the resource that caused the failure. This menu option is only available if you selected <b>On Failure Retain Resources</b> during the creation of the stack. See <a href="#">Creating a Cloud Stack</a>.</li> </ul>

*Stack Create and Delete History*

Element	Description
	Choose from the following options to filter the list of stack creation and deletion jobs that are displayed: <ul style="list-style-type: none"> <li>• <b>Last 24 hours</b></li> <li>• <b>Last 7 days</b></li> <li>• <b>Last month</b></li> <li>• <b>All failed</b></li> </ul>
<b>Show only failed attempts</b>	Show only stack creation and deletion jobs that failed. Click the job for more details. Alternatively, use the <a href="#">Activity Page</a> .

## Stack Details Page

The Stack Details page in the Oracle Cloud Stack console allows you to monitor a stack's resource usage, monitor the status of the resources in the stack, and view a summary of the operations that were performed on this stack. It also provides convenient access to the service console for each resource in the stack.

### What You Can Do From the Stack Details Page

Use this page to perform the following tasks:

- [Starting and Stopping a Cloud Stack](#)
- [Starting and Stopping a Resource in a Cloud Stack](#)
- [Managing SSH Access for a Resource in a Cloud Stack](#)
- [Managing Access Rules for a Resource in a Cloud Stack](#)

### What You See on the Stack Details Page

The following table describes the key information shown on the Stack Details page.

Element	Description
<i>Header</i>	
	Access other Oracle Cloud services. Depending on how you signed up for Oracle Cloud, you can also access either the Oracle Cloud Infrastructure Console or the Oracle Cloud Infrastructure Classic Console.
	Access help for this page, including documentation, tutorials, videos and FAQs.
	Access help, accessibility options, console version information and sign-out.
<b>Welcome!</b>	Go to the <a href="#">Welcome Page</a> .
<b>Stacks</b>	Go to the <a href="#">Stacks Page</a> .
<b>Templates</b>	Go to the <a href="#">Published Templates Page</a> .
<b>Activity</b>	Go to the <a href="#">Activity Page</a> .

Element	Description
	Click to view general information about this stack: <ul style="list-style-type: none"> <li>• <b>Identity Domain</b></li> <li>• <b>Created By</b></li> <li>• <b>Created On</b></li> </ul>
	Click the stack menu to select from the following options: <ul style="list-style-type: none"> <li>• <b>Start</b> — See <a href="#">Starting and Stopping a Cloud Stack</a></li> <li>• <b>Stop</b> — See <a href="#">Starting and Stopping a Cloud Stack</a></li> <li>• <b>View Activity</b> — Go to the <a href="#">Activity Page</a></li> </ul>
<b>Oracle Cloud Stack</b>	Click to view the <a href="#">Stacks Page</a> .
<b>Stacks</b>	Click to view the <a href="#">Stacks Page</a> .
	Start this stack. See <a href="#">Starting and Stopping a Cloud Stack</a> .
	Stop this stack. See <a href="#">Starting and Stopping a Cloud Stack</a> .
	View the template from which this stack was created. See <a href="#">Viewing a Template Resource Topology</a> .
Stack Overview	Shows a summary of the cloud resources being used by this stack: <ul style="list-style-type: none"> <li>• <b>Resources</b> — The number of resources that comprise this stack</li> <li>• <b>OCPUs</b> — The total number of Oracle CPUs allocated for the resources in this stack</li> <li>• <b>Memory</b> — The total amount of memory allocated for the resources in this stack</li> <li>• <b>Storage</b> — The total amount of block storage allocated for the resources in this stack</li> <li>• <b>Public IPs</b> — The total number of public IP addresses allocated for the resources in this stack</li> </ul>
<i>Resources</i>	
	An icon with no badges indicates a cloud resource in this stack that was created successfully and is running. The icon is specific to the type of cloud service (Oracle Database Cloud Service in this example).
	An icon with the hourglass badge indicates a cloud resource in this stack that has not yet been created. The icon is specific to the type of cloud service. This resource likely has a dependency on another resource in the stack, which either is still being created or failed to create successfully.
	An icon with the red badge indicates a cloud resource in this stack that is either being deleted or was not created successfully. The icon is specific to the type of cloud service (Oracle MySQL Cloud Service in this example). Click the resource's <b>Name</b> to go to this cloud service's console and to learn more about the cause of the error.
Name	Click the name of a resource to go to the console for this cloud service.
Type, Version, Created On	The type of cloud service that was used to create this resource (Oracle Database Cloud Service for example), the software version running on this resource, if applicable, and the date/time on which this resource was created.
OCPUs, Memory, Storage	The number of Oracle CPUs, memory and block storage allocated to this resource.

Element	Description
Container, Username, URL	If this resource is an Oracle Cloud Infrastructure Object Storage Classic container, the name of the storage container, the cloud user that Oracle Cloud Stack used to create and access the container, and the base URL of the container.
<b>One or more patches available</b> 	Click this link to go to the console for this cloud service. Use the service's Patching page to view the available patches.  Click the menu for a specific resource to select from the following options. Some options may not be available on all resource types: <ul style="list-style-type: none"> <li>• <b>Start</b> — See <a href="#">Starting and Stopping a Resource in a Cloud Stack</a></li> <li>• <b>Stop</b> — See <a href="#">Starting and Stopping a Resource in a Cloud Stack</a></li> <li>• <b>Restart</b> — See <a href="#">Starting and Stopping a Resource in a Cloud Stack</a></li> <li>• <b>SSH Access</b> — See <a href="#">Managing SSH Access for a Resource in a Cloud Stack</a></li> <li>• <b>Access Rules</b> — See <a href="#">Managing Access Rules for a Resource in a Cloud Stack</a></li> </ul>

## Published Templates Page

The Published Templates page in the Oracle Cloud Stack console shows the stack templates that are available to the current cloud user. Users can manage these templates and also create cloud stacks from a template.

### What You Can Do From the Published Templates Page

Use this page to perform the following tasks:

- [Viewing a Template Resource Topology](#)
- [Importing a Template](#)
- [Creating a Cloud Stack](#)
- [Copying a Template](#)
- [Exporting a Template](#)
- [Deleting a Template](#)

### What You See on the Published Templates Page

The following table describes the key information shown on the Published Templates page.

Element	Description
<i>Header</i> 	Access other Oracle Cloud services.  Depending on how you signed up for Oracle Cloud, you can also access either the Oracle Cloud Infrastructure Console or the Oracle Cloud Infrastructure Classic Console.
	Access help for this page, including documentation, tutorials, videos and FAQs.

Element	Description
	Access help, accessibility options, console version information and sign-out.
<b>Welcome!</b>	Go to the <a href="#">Welcome Page</a> .
<b>Stacks</b>	Go to the <a href="#">Stacks Page</a> .
<b>Templates</b>	Go to the <a href="#">Published Templates Page</a> .
<b>Activity</b>	Go to the <a href="#">Activity Page</a> .
<b>Unpublished</b>	Create a new template, or manage other unpublished (draft) templates. See <a href="#">Unpublished Templates Page</a> .
<i>Stack Templates</i>	
<b>Search by template name</b>	Enter a full or partial template name to filter the list of templates to include only those that contain the string in their name.
<b>Import</b>	Click to import a stack template. See <a href="#">Importing a Template</a> .
	This icon indicates a standard Oracle template. Click the icon or the template's name to view it. See <a href="#">Viewing a Template Resource Topology</a> and <a href="#">About Oracle Stack Templates</a> .
	This icon indicates a user-defined template. Click the icon or the template's name to view it. See <a href="#">Viewing a Template Resource Topology</a> .
Description, Created On, Created By	The template's description, the date on which it was published or imported, and the name of the user who published or imported it. For standard Oracle templates, the Created By field is "Oracle".
	Create a stack from this template. See <a href="#">Creating a Cloud Stack</a> .
	Copy this template in order to create a new one. See <a href="#">Copying a Template</a> .
	Export this template to your local machine. See <a href="#">Exporting a Template</a> .
	Delete this template. Note that Oracle templates cannot be deleted. See <a href="#">Deleting a Template</a> .

## Unpublished Templates Page

The Unpublished Templates page in the Oracle Cloud Stack console shows the draft stack templates that are not yet available to users for stack creation. Users can create, edit, and publish templates.

### What You Can Do From the Unpublished Templates Page

Use this page to perform the following tasks:

- [Configuring Template Storage](#)
- [Creating a Template](#)
- [Importing a Template](#)
- [Editing a Template](#)
- [Publishing a Template](#)

- [Deleting a Template](#)

### What You See on the Unpublished Templates Page

The following table describes the key information shown on the Unpublished Templates page.

Element	Description
<i>Header</i>	
	Access other Oracle Cloud services. Depending on how you signed up for Oracle Cloud, you can also access either the Oracle Cloud Infrastructure Console or the Oracle Cloud Infrastructure Classic Console.
	Access help for this page, including documentation, tutorials, videos and FAQs.
	Access help, accessibility options, console version information and sign-out.
<b>Welcome!</b>	Go to the <a href="#">Welcome Page</a> .
<b>Stacks</b>	Go to the <a href="#">Stacks Page</a> .
<b>Templates</b>	Go to the <a href="#">Published Templates Page</a> .
<b>Activity</b>	Go to the <a href="#">Activity Page</a> .
<b>Published</b>	Manage published templates and create stacks from them. See <a href="#">Published Templates Page</a> .
<i>Stack Templates</i>	
<b>Search by template name</b>	Enter a full or partial template name to filter the list of templates to include only those that contain the string in their name.
	Configure the cloud storage location for unpublished templates. See <a href="#">Configuring Template Storage</a> .
<b>Add Template</b>	Choose one of these options: <ul style="list-style-type: none"> <li>• <b>Import existing Template</b> — See <a href="#">Importing a Template</a>.</li> <li>• <b>Create new Template</b> — See <a href="#">Creating a Template</a>.</li> </ul>
	This icon indicates a user-defined template. Click the template's name to view or edit it. See <a href="#">Editing a Template</a> and <a href="#">Template Builder</a> .
<b>Description, Updated On, Updated By</b>	The template's description, the date on which it was last updated, and the name of the user who last updated it.
	Edit this template. See <a href="#">Editing a Template</a> and <a href="#">Template Builder</a> .
	Publish this template. See <a href="#">Publishing a Template</a> .
	Delete this template. See <a href="#">Deleting a Template</a> .

## Template Builder



This topic does not apply to Oracle Cloud at Customer.

The Template Builder page in the Oracle Cloud Stack console lets users edit a stack template by using graphical tools or by editing the YAML source. Users can also validate, publish, and copy templates.

### What You Can Do From the Template Builder Page

Use this page to perform the following tasks:

- [Creating a Template](#)
- [Copying a Template](#)
- [Publishing a Template](#)
- [Exporting a Template](#)
- [Validating a Template](#)
- [Editing Template Files](#)

### What You See on the Template Builder Page

The following table describes the key information shown on the Template Builder page.

Element	Description
<i>Header</i>	
	Access other Oracle Cloud services. Depending on how you signed up for Oracle Cloud, you can also access either the Oracle Cloud Infrastructure Console or the Oracle Cloud Infrastructure Classic Console.
	Access help for this page, including documentation, tutorials, videos and FAQs.
	Access help, accessibility options, console version information and sign-out.
<b>Welcome!</b>	Go to the <a href="#">Welcome Page</a> .
<b>Stacks</b>	Go to the <a href="#">Stacks Page</a> .
<b>Templates</b>	Go to the <a href="#">Published Templates Page</a> .
<b>Activity</b>	Go to the <a href="#">Activity Page</a> .
<i>Template Metadata</i>	
<b>Unpublished Templates</b>	Return to the list of unpublished templates. See <a href="#">Unpublished Templates Page</a> .
	Click to edit the template's version and description.
<i>Toolbar</i>	
	Create a new template. See <a href="#">Creating a Template</a> .
	Return to the list of unpublished templates. See <a href="#">Unpublished Templates Page</a> .
	Save the template.
	Create a new template by copying the current one. See <a href="#">Copying a Template</a> .

Element	Description
	Publish the template so that it can be used to create cloud stacks. See <a href="#">Publishing a Template</a> .
	Export the template file to your local machine. See <a href="#">Exporting a Template</a> .
	Delete the template.
	Undo the previous modification to the template.
	Redo the previous modification to the template.
	Display a grid of horizontal and vertical lines on the canvas area, behind the resources and connectors.
	Rearrange the resources on the canvas area so that all of them are positioned on the intersections of the horizontal and vertical lines of the grid.
	Display all connectors on the canvas area as straight lines (rays).
	Display all connectors on the canvas area using multiple lines (pipes).
	Rearrange the resources on the canvas area based on their dependencies.
	Zoom in and out on the canvas area.
	Validate the syntax of the entire template. See <a href="#">Validating a Template</a> .
	Open the template in the online YAML editor. See <a href="#">Editing Template Files</a> .
<i>Template Elements</i>	
	Enter a text string and click  in order to search the template for the string and then display a list of matching resource types and matching template elements (resources, parameters, attributes, conditions, and so on). Click one of the items in the search results to navigate to that element within the template.
	Click  to clear the search results.
	Click this icon to the left of each type of template element (resources, parameters, parameter groups, and so on) in order to display a list of elements of this type within the template. Click the icon again to hide the list of elements.
	Click this icon to the right of each type of template element (resources, parameters, parameter groups, and so on) in order to add a new element of this type to the template. Refer to the following topics: <ul style="list-style-type: none"> <li>• <a href="#">Creating Template Parameters</a></li> <li>• <a href="#">Creating Parameter Groups</a></li> <li>• <a href="#">Creating Resources</a></li> <li>• <a href="#">Creating Attributes</a></li> <li>• <a href="#">Creating Conditions</a></li> </ul>
	Click this icon to the right of each type of template element (resources, parameters, parameter groups, and so on) in order to sort the existing elements of this type alphabetically. Click the icon again to sort the existing elements reverse alphabetically.

Element	Description
	Click this icon to the right of each element in this template (resources, parameters, parameter groups, and so on) in order to edit the element's configuration.
	Click this icon to the right of each element in this template (resources, parameters, parameter groups, and so on) in order to delete the element from the template.

## Activity Page

Use the Activity page in the console to monitor activity across all of the platform cloud services in this identity domain, including operations performed by Oracle Cloud Stack. Search for operations by date, status, service type and service (stack) name.

### Tip:

To view only those operations performed by Oracle Cloud Stack, set the **Service Type** to Oracle Cloud Stack.

### What You See on the Activity Page

Element	Description
Start Time Range	Filters activity results to include only operations started within a specified time range. The range defaults to the previous 24 hours.
Status	Filters operations by status of the operation: <ul style="list-style-type: none"> <li>All</li> <li>Scheduled</li> <li>Running</li> <li>Succeeded</li> <li>Failed</li> </ul> <p>You can select any subset of status types. The default value is All.</p>
Service Name	Filters the activity results to include operations only for the specified service instance. You can enter a full or partial service instance name.
Service Type	Filters the activity results to include operations only for instances of the specified service type. The default value is the current cloud service.
Operation	Filters the activity results to include selected types of operations. You can select any subset of the given operations. The default value is All.
<b>Search</b>	Searches for activities by applying the filters specified by the Start Time Range, Status, Service Name, Service Type and Operation fields, and displays activity results in the table.
<b>Reset</b>	Clears the Start Time Range and Service Name fields, and returns the Status and Operation fields to their default values.
Results per page	Specifies the number of results you want to view per page. The default value is 10.
	Displays status messages for the given operation. Clicking on the resulting downward arrow hides the status messages.

---

<b>Element</b>	<b>Description</b>
Service Name	Shows the name of the service instance and its identity domain: <i>service_instance:identity_domain</i> You can sort the column in ascending or descending order.
Service Type	Shows the type of cloud service for this instance. You can sort the column in ascending or descending order.
Operation	Shows the type of operation performed on the service instance. You can sort the column in ascending or descending order.
Status	Shows the status of the operation performed on the service instance. You can sort the column in ascending or descending order.
Start Time	Shows the time the operation started. You can sort the column in ascending or descending order.
End Time	Shows the time the operation ended, if the operation is complete. You can sort the column in ascending or descending order.
Initiated By	Shows the user that initiated the operation. The user can be any user in the identity domain who initiated the operation or, for certain operations such as automated backup, System. You can sort the column in ascending or descending order.

---