Hewlett Packard
Enterprise

HPE Codar

Software Version: 1.60

Online Help

Document Release Date: January 2016
Software Release Date: January 2016

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HPE shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Copyright Notice

Trademark Notices

About this PDF Version of Online Help

This document is a PDF version of the online help. This PDF file is provided so you can easily print multiple topics from the help information or read the online help in PDF format. Because this content was originally created to be viewed as online help in a web browser, some topics may not be formatted properly. Some interactive topics may not be present in this PDF version. Those topics can be successfully printed from within the online help.

Contents

Welcome to Codar

HPE Codar orchestrates the deployment of infrastructure compute resources and complex multi-tier application architectures. Codar integrates and leverages the strengths of several Hewlett Packard Enterprise data center management and automation products, adding resource modeling, service offering design, service design, and a customer portal to create a comprehensive service automation solution.

Based on your role, specific areas of Codar may not be available to you. See "Roles" on page 1.

Codar functions

Use the initial dashboard view to navigate to the area where you can complete your tasks. You can always click Codar in the title bar to return to the dashboard. Depending on your role, the following areas and functionality are available to you:

- "Organizations" on page 10
- "Providers" on page 18
- "Topology designs" on page 31
- "Release Automation" on page 58

Managing licenses

You can use the Managment Console to view, add, or remove software licenses if you are logged in as an administrator (other users can view licensing information, but cannot add or remove software licenses). Hewlett Packard EnterpriseCodar licensing is based on the number of operating system instances (OSI) used in current, active subscriptions. You can add more licenses at any time to increase your OSI capacity.

When users log in to the Managment Console, a licensing status banner will appear in the following situations:

> **Note:** Expiration information is displayed only for the trial license.

- The trial license is the only valid license. The banner displays the number of days remaining on the trial license.
- When you upgrade your product, a banner displays the number of days remaining on the trial license.
- The trial license has expired, there is no valid license, and you have exceeded the allowed number of operating system instances. The banner informs you that you have exceeded the allowed number of licenses.
- The trial license has expired, and there is no valid license. The banner displays the number of operating system instances allowed for an unlicensed version of the software. Any existing subscriptions will continue to work normally, except that flex up operations on existing subscriptions will not be supported.
- One or more valid licenses are installed, and you have exceeded the allowed number of operating system instances. The banner displays the number of active operating system instances by which you are exceeding the allowable license count.

The banner information updates when you refresh your view, log into the Managment Console, or navigate to a new section of the Managment Console by clicking a tile in the dashboard.

You must be logged in as an administrator to access the software licensing area and add and remove licensing.

> **Note:** The **Manage Software License** dialog box cannot be viewed in areas of the product that use Flash Player. It is recommended that you open the dialog box from the Dashboard.

License types

License management for Codar is the same as for CSA. See *Managing the Software License* in the CSA section of the online help for more information.

The following license types are available:

- CSA permanent license only.
- Codar permanent license only.
- Upgrade to Codar for a CSA installation.
- Upgrade to CSA for a Codar installation.

If you install CSA, then you must add a CSA license first; if you install Codar, then you must install a Codar license first. After you apply a base license, you can add an upgrade license, if desired. If you have licenses for both, you can apply a CSA & Codar license.

OSI capacity

The number of operating systems you can use in active applications or subscriptions is known as the OSI capacity.

If you have separate CSA and Codar licenses, then the OSI capacity is the lowest of the two. If you add a CSA & Codar license, its OSI capacity is added to the lowest of the two. Here’s an example: I have an CSA license with 100 OSI and a Codar license with 50 OSI, so my OSI capacity is 50. I add a CSA & Codar license that has a 25 OSI, so my OSI capacity is increased to 75.

Tasks

To perform the following tasks, go to the Managment Console masthead, click the arrow next to the user name, and select **Licensing**.

- **View the following information about your software license:**
  - Cumulative count of purchased operating system instances.
  - Current number of operating system instances being used in Codar active subscriptions.
  - Current license key(s).
- **Add a license:**
  - When you add a new license (one that is not a trial license), existing subscriptions will not be affected. If you exceed your OSI limit, a banner displays the number of active operating system instances by which you are exceeding the allowable license count.
- **Delete a license:**
  - When you delete a license (one that is not a trial license), a banner displays the number of operating system instances allowed. Any existing subscriptions will continue to work normally, except that flex up operations on existing subscriptions will not be supported.
  - You cannot delete the trial license.

Organizations

Use the **Organizations** area to manage organizations. From this section, in the upper left corner, you can view the total number of organizations created, including the provider organization.

Using the organizations interface

Item	Description
↻	Reload the data in this view.
⊗	Delete a group DN from a role.
?	Display a tooltip for the associated field by placing the cursor over this icon.

Informational icons

Icon	Description
((•))	When this icon is adjacent to an organization, it denotes the provider organization. There can be only one provider organization and it is automatically configured. You may modify the provider organization, as needed. However, you cannot delete it.
*	Indicates the field is required, and you must enter information in order to successfully complete the organization's configuration.

View an organization

1. In the left navigation frame, select the organization.
2. In the organization's navigation frame, select **Summary** to view a summary of the organization. Select any of the other sections to view more detailed information.

Configure an organization

1. In the left navigation frame, select the organization.
2. In the organization's navigation frame, select a section in which you can configure information about the organization.

See the following topics for specific configuration tasks:

- "Organization summary" below
- "Organization general information" on the next page
- "Configure LDAP for an organization" on page 13
- "Organization access control" on page 16

Organization summary

View a summary of the selected organization's configuration. To configure or update this information, in the organization's navigation frame, select the appropriate section such as LDAP, Access Control, or Email Notifications.

Viewable summary information

Section	Displayed summary information
LDAP	• Hostname - The host name used to connect to the LDAP server. • Port - The port used to connect to the LDAP server.

Viewable summary information, continued

Section	Displayed summary information
Access Control	• List of roles - Roles in the organization to which group DNs can be assigned.
Email Notifications	• Sender Email Address - Email address that appears as the sender of email notifications. • Port - The port used to connect to the mail server when sending email notifications. Email notification is not supported by Codar.

Organization general information

General information appears at the top of the organization's page.

To configure general information for an organization

1. In the organization's navigation frame, select **General Information**.
2. Provide or update the following information:

Item	Description
Organization Identifier	A unique name that Codar assigns to the organization. This cannot be changed.
Organization Display Name	A unique name that identifies the organization.
Description	A description of the organization.
Organization Logo	An image that represents the logo of the organization. The logo appears in the top left of an organization's page. Click **Upload Image** to add your own image. Supported file extensions include `.jpg`, `.jpeg`, `.gif`, and `.png`. The recommended image size is 256 by 256 pixels, and the image will be scaled to the appropriate size. The images are stored in the `%CSA_HOME%\jboss-as\standalone\deployments\csa.war\images\library` folder of the Codar server.

3. Click **Save**.

Configure LDAP for an organization

LDAP (Lightweight Directory Access Protocol) is configured in the Managment Console.

To completely configure access to Codar, you must configure LDAP to

- Authenticate a user's credentials
- Allow an organization to authenticate a user's access to information
- Allow an organization to authorize a user's access to information

From this page you can:

- Configure LDAP for authentication to log in to Codar.
- Configure LDAP for access control in Codar.

When you configure LDAP for the provider organization, you are configuring the set of users who can log in and be authenticated to perform actions in the Managment Console.

To configure authorization to access information in Codar for organizations, see "Organization access control" on page 16.

To configure LDAP

1. In the organization's navigation frame, select **LDAP**.
2. Provide or update the following information:

   LDAP Server Information

   Configure the LDAP server and a user with access to the server.

Item	Description
Hostname	The fully-qualified LDAP server domain name (server.domain.com) or IP address. Example: `ldap.xyz.com`
Port	The port used to connect to the LDAP server (by default, 389). Example: `389`
Connection Security	If the LDAP server is configured to require ldaps (LDAP over SSL), select the **SSL** checkbox.

Item	Description
Base DN	Base distinguished name. The Base DN is the top level of the LDAP directory that is used as the basis of a search. Example: `o=xyz.com`
User ID (Full DN)	The fully distinguished name of any user with authentication rights to the LDAP server. If the LDAP server does not requre a User ID or password for authentication, this value can be omitted. Example: `uid=admin@xyz.com,ou=People,o=xyz.com`
Password	Password of the User ID. If the LDAP server does not requre a User ID or password for authentication, this value can be omitted.

LDAP Attributes

Enter the names of the attributes whose values are used for email notifications and authentication.

Item	Description
User Email	The name of the attribute of a user object that designates the email address of the user. The email address is used for notifications. If a value for this attribute does not exist for a user, the user does not receive email notifications. Default: `mail`
Group Membership	The name of the attribute(s) of a group object that identifies a user as belonging to the group. If multiple attributes convey group membership, the attribute names should be separated by a comma. Default: `member,uniqueMember`
Manager Identifier	The name of the attribute of a user object that identifies the manager of the user. Default: `manager`
Manager Identifier Value	The name of the attribute of a user object that describes the value of the Manager Identifier's attribute. For example, if the value of the Manager Identifier attribute is a distinguished name (such as `cn=John Smith, ou=People, o=xyz.com`) then the value of this field could be `dn` (distinguished name). Or, if the Manager Identifier is an email address (such as `admin@xyz.com`) then the value of this field could be `email`. Default: `dn`
User Avatar	LDAP attribute whose value is the URL to a user avatar image that will display for the user. If no avatar is specified, a default avatar will be used.

User Information

Codar uses a user search-based log-in method to authenticate access to information.

Item	Description
User Name Attribute	The name of the attribute of a user object that contains the user name that will be used to log in. The value for this field can be determined by looking at one or more user objects in the LDAP directory to determine which attribute consistently contains a unique user name. Often, you will want a **User Name Attribute** whose value in a user object is an email address. Examples: `userPrincipalName` or `sAMAccountName` or `uid`
User Search Base	The location in the LDAP directory where users' records are located. This location should be specified relative to the Base DN. If users are not located in a common directory under the Base DN, leave this field blank. Examples: `cn=Users` or `ou=People`
User Search Filter	Specifies the general form of the LDAP query used to identify users during login. It must include the pattern `{0}`, which represents the user name entered by the user when logging in. The filter is generally of the form `<attribute>= 0}`, with `<attribute>` typically corresponding to the value entered for **User Name Attribute**. Examples: `userPrincipalName={0}` or `sAMAccountName={0}` or `uid={0}`
Search Option (Search Subtree)	When a user logs in, the LDAP directory is queried to find the user’s account. The **Search Subtree** setting controls the depth of the search under **User Search Base**. If you want to search for a matching user in the **User Search Base** and all subtrees under the **User Search Base**, leave **Search Subtree** selected. If you want to restrict the search for a matching user to only the **User Search Base**, excluding any subtrees, unselect **Search Subtree**.

3. Click **Save**.

Example LDIF content record

The following is a sample LDIF (LDAP Data Interchange Format) content record that shows the `uniqueMember` group membership attribute being used to define users `cn=User1,ou=providers,ou=users,ou=system` and `cn=Manager1,ou=managers,ou=users,ou=system` as members of the group `cn=ResourceSupplyManagers,ou=providergrp,ou=groups,ou=system`.

```
dn: cn=ResourceSupplyManagers,ou=providergrp,ou=groups,ou=system
objectclass: groupOfUniqueNames
objectclass: top
cn: ResourceSupplyManagers
uniqueMember: cn=User1,ou=providers,ou=users,ou=system
uniqueMember: cn=Manager1,ou=managers,ou=users,ou=system
```

To assign this group or DN to the Resource Supply Manager Role, go to the Access Control section of the Organizations area and add the `cn=ResourceSupplyManagers,ou=providergrp,ou=groups,ou=system` DN to the Resource Supply Manager role.

Look up a user

The **Look Up User** button allows for the displaying of common LDAP attribute values for a specified user. Also, this button validates the User Login Information (User Name Attribute, User Search Base, and User Search Filter).

Provide the user name attribute value of a user to display that user's DN, common name, display name, email address, and manager.

The lookup also validates the User Name Attribute, User Search Base, and User Search Filter (if attribute information is displayed, these fields are correctly defined).

To look up a user:

1. In the organization's navigation frame, select **LDAP**.
2. Provide all the required LDAP service access information.
3. Click **Save**.
4. Click **Look Up User**.
5. Provide the user name attribute value of a user to look up.
6. Click **Search**.

Organization access control

Roles control what a user can accessCodar. For more information about available roles, see "Roles in " on page 1.

Adding a DN to the roles authorizes members of the LDAP directory organizational units access to Codar.

Access control allows you to add or remove directory service groups or organization units (ou) to a Codar role by associating the ou's distinguished name (DN) to the desired role. Authenticated LDAP users, who are members of a group or organization unit that is assigned to a predefined role, can perform specific tasks and access specific parts of Codar .

Only members of a group or organization unit are assigned to the role. To ensure secure role assignment, access control inheritance stops at the assigned organizational unit. This does not follow the traditional directory service pattern where inheritance flows down the organizational unit's hierarchy. Instead, assignments to roles must be assigned to individual organizational units (ou).

A group or organization unit DN can be assigned to more than one role.

LDAP must be configured in order to authenticate users so that they can log in. See "Configure LDAP for an organization" on page 13 for more information.

To add a DN to a role

1. Locate the role to which you want to add a DN.
2. Below the role, click **Add DN**.
3. Provide the following information, and click **Save**:

   To select an existing named DN:

Item	Description
Select from existing named DNs	Select an existing named DN (that identifies a group or organization unit DN) to add to the role. If there are no existing named DNs, this item is not selectable.

   To add a new named DN:

Item	Description
Enter a name for the group or organization unit DN	Enter a name to identify the DN.
Enter a group or organization unit DN	Enter the group or organization unit DN to add to the role. This DN must be relative to the Base DN you configured in the LDAP section of this organization. If the base DN is empty, supply the full DN of the group.

To update a name or DN in a role

1. Locate the role whose DN you want to update.
2. Below the role, locate the DN you want to update.
3. Move your cursor over the DN and click the **Edit** button.
4. In the **Update DN** dialog, update the DN name and/or the DN.
5. Click **Update**.

To remove a named DN from a role

Note: The named DN (group) is not deleted; instead, it is disassociated from the role. You will still see the group when you click **Add DN** and then click **Select from existing named DNs**.

1. Locate the role from which you want to remove a named DN.
2. Below the role, locate the group you want to remove.

3. Click the **Remove DN** icon.

4. Click **Yes**.

Providers

Providers are management platforms that offer centralized control over the infrastructure and resources used in a cloud computing environment. For example, a provider such as infrastructure orchestration can deploy virtual machines, while a provider such as HPE SiteScope can monitor applications.

A provider corresponds to the specific instance of an application that Codar can integrate with to help instantiate service designs. For example, to enable service designs that target infrastructure orchestration, you must first create a provider (with a provider type of HPE Matrix Operating Environment (Matrix OE)).

Provider types

A provider type allows you to classify providers for improved filtering and identification. CodarCodar includes some predefined, out-of-the-box provider types. Each instance of a provider can have a single provider type, and each instance of a resource offering can also have a single provider type. In addition, resource offerings can be associated only with providers that share the same provider type.

Tasks

You can perform the following tasks in this area:

- **View providers by type.** In the drop-down box, select to view providers **By Type**. Provider types are listed in the left pane. The list of providers contained by the type displays in the right pane. Disabled providers are indicated by the disabled icon and will not be selected when provisioning new services.

- **Manage provider types.** In the drop-down box, select **By Type**. Select the manage icon. For more information, see "Manage provider types" on the next page.

- **Create a Provider.** In the drop-down box, select **By Type**. In the left pane, select **All Providers**, or select the type for which you want to create a provider. In the right pane, select the **Providers** tab. Click **Create** to add a provider with the selected provider type. See "Provider overview" on page 20 for more information about configuring a provider.

- **See more information about a provider.** Click a provider to see more information about the provider.

Best practices

- Locked items cannot be deleted.

- You can select whether to see the content in card view or table view.

Components (per provider type)

The components tab for a selected resource provider type shows all the components available for that provider type. When components are imported into CodarCodar, they are associated with a single provider type, and all provider instances of that type support the component.

> **Note:** Provider components are applicable only to topology designs and are not applicable to sequenced designs.

Tasks

- **View components associated with the selected provider type** - See the list of components in the right pane.
- **Launch the component management area** - Click a component. Or click the **Manage** button (or the **Manage Components** button if no components exist). You must have Service Designer role access to perform this task.

Manage provider types

You can perform the following tasks in this area:

- **Create a provider type.** Provide the information listed in the table below.
- **Edit a provider type.** See the table below for the items you can edit.
- **Delete a provider type.** A provider type cannot be deleted if any resource offerings or providers of that provider type exist. Out-of-the-box provider types also cannot be deleted.

Item	Description
Name	A name that is automatically generated by Codar, and which may be needed when importing components for topology designs.
Display Name	The display name you provide for the provider type.
Description	The description you provide for the provider type.
Image	An image that displays for the provider type. Click **Change Image**. Choose the image you want, and click **Select**. Click **Upload Image** to add your own image. Supported file extensions include `.jpg`, `.jpeg`, `.gif`, and `.png`. The recommended image size is 256 by 256 pixels, and the image will be scaled to the appropriate size. The images are stored in the `%CSA_HOME%\jboss-as\standalone\deployments\csa.war\images\library` folder of the Codar server.

Provider information

- "Provider overview" below
- "Provider properties" on the next page
- "Environments" on page 24
- "Components (per provider instance)" on page 24

Provider overview

You can perform the following tasks:

- **Create or Edit a provider.** In the right pane, select the **Providers** tab. Click **Create** in the main Providers area or click **Edit** in the **Overview** tab. See the table below for the items you can edit.
- **Delete a provider.** Click **Delete**. A provider referenced by topology designs or by topology or sequenced service instances cannot be deleted. All other providers can be deleted, and their associations (to environments and to resource offerings) are automatically removed on deletion. When deleting providers, make sure that any resource offerings referenced by the provider and used in a service design are still associated with at least one provider that can provision the resource offering.

Item	Description
Provider Type	The type selected for this provider. Note that the **Provider Type** cannot be changed after a provider is created.
Display Name	The name you provide for the provider.
Description	The description you provide for the provider.
User ID	The user ID for the specified Service Access Point.
Password	The password for the specified Service Access Point. Re-type the password in the **Confirm Password** field.

Item	Description
Service Access Point	Specify a URL for connecting to the provider. The following examples show how to connect to some common resource providers: - Matrix OE - `https://<MOE server IP>:51443/hpio/controller/soap/<v1/v2/v3>` - HPE Server Automation - `https://<SA server IP>:443` - SiteScope - `http://<SiteScope server IP>:8080` - HPE Universal CMDB - `http://<UCMB server IP>:8080` - VMware vCenter - `https://<vCenter server IP>:443` - Chef - `https://<Chef Server IP>:443` When creating or updating a provider, Codar attempts to contact the provider at the URL you specify. Codar uses a six (6) second timeout to attempt to validate the provider URL (for HTTP and HTTPS only). If the URL is not successfully contacted before the timeout expires, a validation failure message displays, and you have the option to ignore the validation failure or to correct the URL and try again. Note: User credentials are not validated at this time.
Image	Select an image to be displayed with the provider.
Enabled	This value determines whether the provider will be selected when provisioning a new service. The setting is either **Enabled** (when checked) or **Disabled** (when not checked). When **Disabled**, the provider will not be selected when provisioning new services. Disabling a provider will have no effect on existing services that are using that provider.

Provider properties

Custom properties on a provider can be used to capture additional configuration information about a particular provider. For example, you can use custom properties to model provider resources, such as datacenters, hypervisors, and datastores for a specific VMware vCenter provider.

When a topology design is provisioned, an Operations Orchestration flow can read and write provider property values.

Tasks

Custom properties are optional, and are needed only if the provisioning process requires them. For example, provider properties may be used by HPEOperations Orchestration flows during service provisioning.

- **Create provider properties.** Click **Create**. Provide the information listed in the following table.
- **View property description.** Click the information icon.

- **Edit provider properties.** Click the edit icon. See the following table for the properties you can edit.
- **Delete provider properties.** Click the delete icon.

Item	Description
Type	Select one of the following: • **Boolean** - A property whose value is True or False. • **List** - A property whose value is a list of String values. • **Integer** - A property whose value is a positive or negative whole number or zero. • **String** - A property whose value is a sequence of characters. You cannot change the type of a property after it is created.

Item	Description
	Configure the following items for each property type: ***For Boolean properties:*** • **Name** - A unique name for the property. • **Display Name** - The display name for the property. • **Description** - A description of the property. • **Property Value** - Select **True** or **False**. ***For list properties:*** • **Name** - A unique name for the property. • **Display Name** - The display name for the property. • **Description** - A description of the property. • **Property Value** - Click the **Add Item** icon to add a new value. Click the **Edit Item** icon to edit the selected value. Or click the **Remove Item** icon to remove a selected value. ***For integer properties***: • **Name** - A unique name for the property. • **Display Name** - The display name for the property. • **Description** - A description of the property. • **Property Value** - Select or type a positive or negative whole number or zero. If you enter a decimal number, the value will be truncated to the nearest integer. The maximum allowed integer value is 2147483647 and the minimum is -2147483648; if you enter a value outside these bounds the value will be automatically converted to the closest maximum or minimum value. ***For string properties:*** • **Name** - A unique name for the property. • **Display Name** - The display name for the property. • **Description** - A description of the property.

Item	Description
	• **Property Value** - Type a string of characters. • **Confidential Data** - Select this box to mask the values so that they cannot be read in the user interface; no encryption of the value is performed.

Provider environments

You can perform the following tasks:

- **View environments associated with the selected provider.** See the list of environments.
- **Select a resource environment.** Click **Select**. In the dialog box, add or remove resource environments to or from the resource provider.

Components (per provider instance)

> **Note:** Provider components are applicable only to topology designs and are not applicable to sequenced designs.

Tasks

You can perform the following tasks in this area:

- **View components associated with the selected provider instance.** See the list of components in the right pane.
- **Launch the component management area.** Click a component. Or click the **Manage** button (or the **Manage Components** button if no components exist). You must have Service Designer role access to perform this task.

Environments

Environments are optional and provide a mechanism for grouping providers. The most common grouping patterns include grouping by geographical location, organizational structure, or production readiness. For example, you may want to group providers geographically and create environments such as *East Coast* or *South America*. Or you may want to group providers by their production readiness and create environments such as *Production*, *Development*, and *Test*.

In addition, environments can be linked to a service catalogs to support distribution of resource provisioning.

Resource environments restrict the set of resource providers that can be chosen at subscription time. When provider selection occurs during service provisioning, only providers belonging to one or more of the environments associated with the service catalog will be eligible for selection. If no environments are associated with a service catalog, provider selection is not restricted based on environment membership.

Tasks

You can perform the following tasks in this area:

- **View providers by environment.** In the drop-down box, select to view providers **By Environment**. Environments are listed in the left pane. The list of providers contained by the environment is displayed in the right pane. Disabled providers are indicated by the disabled icon and will not be selected when provisioning new services.
- **Manage environments.** In the drop-down box, select **By Environment**. Select the manage icon. For more information, see "Manage environments" below.
- **Select providers to include in an environment.** In the drop-down box, select **By Environment**. Select the environment in which you want to include a provider. Click the **Select** button. In the dialog box, add or remove resource providers to or from the environment. If desired, use the drop-down box in the dialog to filter the resource provider list by provider type.

Best practices

- Locked items cannot be deleted.
- You can select whether to see the content in card view or table view.

Manage environments

You can perform the following tasks in this area:

- **Create a resource environment.** Provide the information listed in the table below.
- **Edit a resource environment.** See the table below for the items you can edit.
- **Delete a resource environment.** An environment can be deleted only if it is not associated with a service catalog. When an environment is deleted, its associations to providers are automatically removed.

Item	Description
Display Name	The display name you provide for the environment.
Description	The description you provide for the environment.

Item	Description
Image	An image that displays for the environment. Click **Change Image**. Choose the image you want, and click **Select**. Click **Upload Image** to add your own image. Supported file extensions include `.jpg`, `.jpeg`, `.gif`, and `.png`. The recommended image size is 256 by 256 pixels, and the image will be scaled to the appropriate size. The images are stored in the `%CSA_HOME%\jboss-as\standalone\deployments\csa.war\images\library` folder of the Codar server.

Managing packages

Packages represent a snapshot of an application design and allow properties to be parameterized within the design. We can also say that the package represents a particular build of an application.

A package is the smallest unit that can be deployed for an application. It represents both the implementation artifacts (the manner in which an application should be deployed) and deployment artifacts(the location of libraries like war, ear, etc., that should be deployed).

Packages are associated with a lifecycle stage. A package can belong to Development, Testing, Staging, or Production stages.

Packages are associated with pipeline management. They can be managed across lifecycle stages, such as promotion or rejection in a given stage. For example, a user with the QA role can reject a package. See "Lifecycle stages and actions" on page 75.

Tasks

- **Create a package from a specific application version.** An application version can consist of multiple packages. See "Create a package" on the next page.
- **Deploy or redeploy a package.** In this case the corresponding state of an application design along with the properties of the design specified in the package will be fulfilled. See "Deploy a package" on page 28.
- **Delete a package.** Go to the Release Pipeline tab, hold Ctrl to select multiple packages, and click **Delete**.

  You cannot delete a package that has an instance associated with it.

View packages

For more information about packages, see "Managing packages" above.

To view packages

1. Click the Release Pipeline tile.
2. Select a package.

3. Use the search box on the right to find a specific package. Filter packages by stage and state using the drop-down lists on the left.

Packages are grouped into stages, with three packages displayed per stage initially. Click More to the right of the stage heading to view all packages for a certain stage. For more information about stages, see "Lifecycle stages and actions" on page 75.

Create a package

For more information about packages, see "Managing packages" on the previous page.

Note: You may not have access to some or all of this functionality, depending on your role. See "Roles" on page 79.

To create or edit a package

1. Click the **Release Pipeline** tile.
2. Click **Create**.
3. Enter a name and description.

   The application design and version are displayed, but cannot be changed.
4. Click **Create**. A package will be created in the Development stage.

For more information about the package lifecycle, see

Edit a package

For more information about packages, see "Managing packages" on the previous page"Managing packages" on the previous page

You can edit a package name and description. For information about changing a package state or stage, see "Lifecycle stages and actions" on page 75.

Note: You may not have access to some or all of this functionality, depending on your role. See "Roles" on page 79.

To edit a package

1. Click the Release Pipeline tile.
2. Select a package.

3. Click on a package in the list. If you don't see the package you want to edit, see "Managing packages" on page 26.

4. Click **Edit**.

5. Change the package name or description.

6. Click **Save**.

Configure package properties

The Configure tab for a package allows you to change all required properties and properties that can be modified for all components used in the design.

To configure package properties

1. Click the Release Pipeline tile.

2. Select a package.

3. Select the Configure tab.

4. Change the desired properties.

5. Click **Save**.

Deploy a package

For more information about packages, see "Managing packages" on page 26.

> Note: You may not have access to some or all of this functionality, depending on your role. See "Roles" on page 79.

To deploy a package

1. Click the Release Pipeline tile.

2. Select a package.

3. Click **Deploy** to the right of the package.

4. Enter a name for the deployment.

5. Select an environment for the deployment.

6. Click **Next** and select a service design from the list of available designs.

Select a design from the list that, together with the partial design, will form a composite design that can be provisioned. If you select **Custom Selection**, you can select a microservice for each requirement in the partial design. Only service designs or microservices that fulfill the requirements are available.

7. (Optional) Click **Next** and change modifiable properties for this deployment.
8. Click **Deploy**.

Deployments tab

After a package has been deployed, an entry is made on the Deployment tab for the package. This tab lists all deployments, their status, and the user who created the deployment. Click a deployment for quick access to the test results for a deployment.

Redeploy a package

You may want to redeploy a package in order to test a deployment with different properties, such as a different location or different credentials. You can redeploy a package instead of recreating the design.

For more information about packages, see "Managing packages" on page 26.

Note: You may not have access to some or all of this functionality, depending on your role. See "Roles" on page 79.

To redeploy a package

1. Click the Release Pipeline tile.
2. Select a package.
3. Create a new package.
4. Click **Redeploy** to the right of the new package.
5. Enter a name for the deployment.
6. Select an existing deployment that you want to redeploy.

   After you select a deployment, its details will be displayed so you can verify that you selected the desired deployment.
7. Click **Next** and change modifiable properties as needed for the redeployment.
8. Click **Deploy**.

Deployments tab

After a package has been deployed, an entry is made on the Deployment tab for the package. This tab lists all deployments, their status, and the user who created the deployment. Click a deployment for quick access to the test results for a deployment.

Scaling a deployment

You can create multiple instances of a group for a deployment. Each component in the group will be duplicated by the number of instances you select.

To scale out

1. Open a package in the Release Pipeline area.
2. Go to the Deployments tab.
3. Click **Scale Out**.

   This button is only available for successful deployments.
4. Select an instance size.

   This is the number of instances you want of each component in the scalable group.
5. Click **Scale Out** in the dialog to complete the operation.

Save a deployed design

You can save a deployed design so you can access it from Topology Designs.

This option is only available for a topology design with a package in the Production stage and one active deployment.

To save a deployed design

1. Click the Release Pipeline tile.
2. Select a package.
3. Click **Save Design**.
4. Enter a **Name** for the design.
5. (Optional) Enter a **Description** for the design.
6. Go to Topology Designs to view the saved design.

Topology designs

Topology designs are typically used for Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) deployments that are enabled via Chef, Puppet, Server Automation and Operations Orchestration flow-based components.

Topology designs are declarative in nature and do not include explicit actions or sequencing. The provisioning sequence is inferred by the relationships that exist between components in a topology design.

A topology design is composed of components. Each component can have a relationship with other components. For example, you might define a relationship between a server component and a networking interface component. Each component can also have properties, such as an IP address, password, or name. See "Components " on page 46.

When a topology design is associated with a release pipeline process, it is referred to as an application design. When it is associated to microservices, it is referred to as a service design.

Component types

A component represents one service design element that is required to deploy a service. A topology design can include concrete components and capabilities.

- A *concrete component* is defined in the design as a specific service, such as a vCenter server.

  A concrete component is deployed as defined and does not require composition.

- A *capability* is a component that can be fulfilled by any service that meets its criteria, such as a web server that could be fulfilled by either Apache HTTP Server or Apache Tomcat. Capabilities are identified in the designer by angled brackets, such as <<Server>>.

  A capability must be matched to an service design or a microservice to be fulfilled.

See "Topology composition" on page 33

Topology design versions

A version of a topology design is where you add components, define relationships, and enter property values for a design. Each version can be edited and configured as needed. Throughout the topology designs area, the version identifier is shown at the top of the window under the display name.
See "Topology design versions" on page 37

Manage topology designs

Use the Topology Designs area to browse, view, and manage topology design.

Note: Only the admin user and Application Architect role have access to the Design tile. Users with other roles can access designs in the Release Pipeline tile if they have been granted access. See "Access control" on page 44.

Tasks

In the Topology Designs area you can perform the following tasks:

- **Manage design tags.** Click the **Manage Design Tags** icon in the lower left pane. See "Manage design tags" on page 35.
- **View designs.** In the left pane, select **All Designs** or select a user-defined tag, to display a list of the corresponding topology designs.
- **Create a new version.** Click **New** under the design name to create a new version. Each design may have multiple versions, and the most recently edited version is shown in bold. The most recent five versions of a design are shown beneath the design name, with links to individual versions. For designs with more than five versions, click **All** to see the complete list. See "Topology design versions" on page 37.
- **Create a new design.** See "Create or edit a topology design" below.
- **Import a design.** See "Import or export a topology design" on page 35.

Create or edit a topology design

To create a topology design

1. Go to Topology Designs.
2. Click **Create** to create a new design.

To edit a topology design

1. Go to Topology Designs.
2. Select a design to view its details.
3. Click **Edit**.

Toplogy design properties

You can change the following properties for a new or existing design:

- **Palette** - For a new design only, select the palette for the design. See "Palettes" on page 58
- **Display Name** - Provide the display name for the design.
- **Description** - Provide the description for the design.
- **Initial Version** - Provide the version number you want to assign to the design.
- **Image** - Select an image that displays for the design. If you upload your own image, the supported file

extensions include .jpg, .jpeg, .gif, and .png. The recommended image size is 256 by 256 pixels, and the image will be scaled to the appropriate size.

- **Tags** - Click **Select Tags** to choose from a list of tags that you can define to provide a structure for organizing and grouping the service designs. You can select more than one tag for a design. To remove a tag, hover the cursor over the tag you want to remove, and click the **Delete** icon.

Topology composition

Components are the building blocks for your designs. Some components are created when you install Codar, and you can add new components in the Components tile in the Designs area.

A component represents one service design element that is required to deploy a service. A topology design can include concrete components and capabilities.

- A *concrete component* is defined in the design as a specific service, such as a vCenter server.

  A concrete component is deployed as defined and does not require composition.

- A *capability* is a component that can be fulfilled by any service that meets its criteria, such as a web server that could be fulfilled by either Apache HTTP Server or Apache Tomcat. Capabilities are identified in the designer by angled brackets, such as <<Server>>.

  A capability must be matched to an service design or a microservice to be fulfilled.

See "Components " on page 46.

You can use these components to create the following types of topology designs:

- A *concrete design* is any design that contains no capabilities. It may include components that support a capability, but it cannot directly contain capability components.

- A *partial design* is any design that contains capabilities. Such a design cannot be provisioned on its own and requires the selection of a compatible service design.

- A *service design* is a design that can fulfill all capability requirements of a partial design.

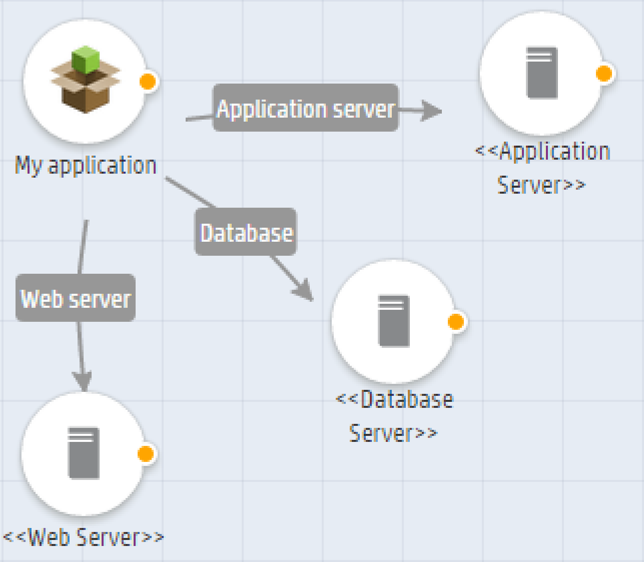
- A *microservice* is a design that can fulfill one part of a capability.

- The combination of a partial design with a service design creates a *composite design* that is itself a concrete design. A composite design is normally hidden from view in the list of topology designs, but it can be made visible by saving the design from a test run. See "Test a topology design" on page 42.

When only concrete components are included in a design, those specific services are provisioned as specified. When a design contains capabilities, the design requires a separate design that satisfies each capability.

The separate design can be a service design with concrete components or a set of microservices that together fulfill the capability requirements. If a microservice satisfies more than one capability, the topology instance can be shared between two or more requirements, or each requirement can use a new instance of the microservice.

For example, this design contains a My application component that has relationships to <<Application Server>>, <<Database Server>>, and <<Web Server>> (each of which is a capability):



A suitable service design might contain PHP, MySQL, and Apache components. A microservice would contain only one of those components, such as a MySQL server.

Designs with a capability cannot be published but can be provisioned as a test run. See "Test a topology design" on page 42.

When you deploy a package from a partial design, you select an service design or microservices that meet the requirements of the design.

Requirements (required characteristics)

You can specify requirements for capability components. A requirement allows you to refine which concrete designs should be selected when a partial design is provisioned. For example, Apache HTTP Server may be a required characteristic that is specified on a Web Server capability, so only designs that include Apache Web Server as a supported characteristic of the component will be selected when a design is provisioned. See "Component characteristics" on page 57.

Testing a partial design

During the test run of a partial design, you will be prompted to select from the list of candidate service designs, each of which meets the following criteria:

- Is a concrete design.
- Contains concrete components that support each of the capabilities used in the partial design, each of which supports all of the required characteristics specified on the capability in the partial design.
- Has property values configured for all required properties in the design.

- Is not a composite design that was created by the combination of a partial design with another concrete design, unless that composite design was subsequently saved as its own design (see the description of **Composite design** below for more information).

- Has no validation errors.

  For example, consider a partial design that includes a concrete component My App with relationships to Web Server, Application Server, and Database Server components.

  Assume the Web Server, Application Server, and Database Server components have required characteristics of Apache 2.4, PHP 8, and MySQL 5.5, respectively.

  The partial design could be provisioned with a that contains Apache Web Server, PHP, and MySQL components, respectively supporting the Web Server, Application Server, and Database Server capabilities, which support the characteristics Apache 2.4, PHP 8, and MySQL 5.5, with each component configured to be provisioned on a server.

See "Test a topology design" on page 42.

Manage design tags

Tags are user-defined, color-coded labels and images used to organize and group topology designs. A tag can be assigned to multiple designs.

To manage design tags

1. Go to Topology Designs.

2. Click the **Manage Design Tags** icon in the lower left pane.

3. Click the **Create Tag** icon to add a new tag.

4. Click the **Edit Tag** icon to change the selected tag.

5. Click the **Delete Tag** icon to remove a tag.

6. Click **Done** when you are finished.

Import or export a topology design

The export process creates either a zip file with all the topology files, design, and dependencies as either a zip or JSON file. This file can then be imported to Codar.

The import process imports archives of designs and their supported artifacts. Designs with the same internal name are considered to be functionally equivalent and are not imported. When you import a version of an existing design, the imported design is saved as a version of the existing design.

During the update process, identical artifacts that exist on the target system are updated (overwritten) with the changes from the archive. Artifacts are created if they do not exist on the target system.

This process imports all artifacts present in an archive, whether or not they exist on the target system. During this operation, if there is an artifact with the same internal name in the system, the name, the display name, and the description of the artifact are modified internally; the display name and the description are appended with "Superseded on" and the date. The internal name, display name, and description of the artifact being imported remain intact.

Exporting a design creates a zip or JSON file for the design you are exporting, which contains associated artifacts, icons for customizing the artifacts, and the Manifest document, which contains meta-information about the archive files.

When you import a design, any tags you have selected in the designs area do not affect the imported design. The imported design will contain the tags that were included when the design was exported, and new tags will be created on the system, as necessary, to match what was exported.

If you import a design that already exists, access control for the existing design is retained. If you try to import a design and you do not have access to the design, then the import will not be permitted.

Note: Importing and exporting JSON files is recommended for infrastructure as a service (IaaS) applications. For other uses, we recommend that you export designs as a zip file.

Prerequisites

Before you import a design archive, complete the following prerequisites:

- You can only import JSON files using the API.
- If you import a JSON archive and the design has a dependency on any custom component types, then these custom component types must be imported before the design can be imported.
- When you export a zip file, the export can only reference files or content contained within the file itself, or that are already contained in the csa.war file. By default, all images in the archive files must end in one of the following suffix values. For information on adding additional suffix values, see the *Codar Configuration Guide*.

  jpg, jpeg, jpe, jfif, svg, tif, tiff, ras, cmx, ico, pnm, pbm, pgm, ppm, rgb, xbm, xpm, xwd, png, gif, bmp, cod, ief

To import a design archive

1. In the lower, right pane of the Topology Designs area, click **Import**.
2. Select or specify the zip file that contains the design you want to import. Archive file names for designs begin with SERVICE_DESIGN_.
3. Select an option:

   Import - imports new designs; does not update existing service designs. Note that you cannot import a design with the same internal name as an existing design.

Update - imports new designs (and associated resource offerings) and updates (overwrites) existing designs. Check Preserve Originals to create backup copies of the original items, appending "Superseded on" and the date to the artifact display names and descriptions.

4. Click **Preview** to see a report of prospective results for the import process, including information about the artifacts and their status.
5. Click **Import**.
6. Click **View Detailed Report** to see a summary and details of the import process, including information about the artifacts and their status.
7. Click **Close**.

To export a design archive

1. Open the version of a design that you want to export.
2. In the Overview tab, click **Export**.
3. Save the exported design, if required by your browser.

When you export a design archive, the design and all referenced resource offerings are packaged in an archive file with the following name:

SERVICE_DESIGN_<service_design_display_name>_<service_design_id>.zip

To export a design JSON

1. Open the version of a design that you want to export.
2. In the Overview tab, click **Export**.
3. Select JSON and click **Export**
4. Save the exported design JSON, if required by your browser.

When you export a design JSON, the design and all its components values are added in the JSON file with the following name:

<service_design_display_name>.json

Topology design versions

A version of a topology design is where you add components, define relationships, and enter property values for a design. Each version can be edited and configured as needed. Throughout the topology designs area, the version identifier is shown at the top of the window under the display name.

For general information about topology designs, see "Topology designs" on page 31.

Tasks

- **View versions of a design.** The versions list is sorted alphanumerically. You can click a version identifier to open that version of the design. See "Manage a topology design version" below.

- **Create a new version.** While viewing the list of designs, click **New** under the design name. While viewing a design, open the Versions tab and click **New Version**. While viewing a version of a design, click **New Version** to create a new version based on the current design. Each design may have multiple versions.

  You may select a palette for the version. See "Palettes" on page 58.

  You can enter any version number or string and an optional description.

Manage a topology design version

The Overview tab for a version of a design provides a summary view of the topology design details.

Tasks

In the **Overview** tab, you can perform the following tasks, although some tasks may not be available depending on your role:

- **Edit this version.** See "Manage topology designs" on page 31.

- **Publish this version of the design.** Published designs are indicated in the Overview tab by a Published icon in the upper right corner. See Publish a Design.
  - You can publish a design only if it has no validation errors.
  - A published design cannot be modified, but you can use **Save As** or **Create New Version** to create editable designs based on the published design.
  - Partial designs contain capabilities and require composition with a compatible concrete design before they can be provisioned. Partial designs cannot be directly published. These designs are indicated in the Overview tab by a Requires Composition icon in the upper right of the window.
  - Published designs cannot be unpublished.

- **Save this version with a new name.** Use **Save As** to create copy of the current version of the design.

- **Create a new version.** This will create a new version based on the current version.

- **Export a version.** See "Import or export a topology design" on page 35.

- **Delete this version.** Deletes the selected version of the design.

- **Manage packages.** This button takes you to the design in the Release Pipeline area. See "Release pipeline" on page 58.

Best practices

- Perform a **Test Run** before creating a package.
- You must use **Save As** or **New Version** to modify a design that has a package.
- Deleting a design cannot be undone.

Topology designer

A version of a design has a Designer tab, where you create a layout of components and their relationships.

A single design normally contains components of multiple provider types (for example, VMware vCenter and HPE SiteScope components); however, if you include Helion OpenStack® in a design, then you can only use HPEHelion OpenStack® components.

Tasks

Designs can be created from the set of topology components that are defined in the Components area of the Cloud Service Management Console. Many components are included with Codar. If necessary, import any additional components that you need prior to creating a design. See "Import components" on page 48.

> **Note**: Designs are saved automatically every five seconds after a change or immediately after a significant and meaningful change is made to the design.

- **Add components to your design.** From the set of components (based on the palette you selected for the design) in the left pane of the Designer, drag-and-drop or double-click the component you want to add. The components you select can be concrete components or capabilities. Capabilites are identified by angle brackets, such as <<Server>>.
- **Add a relationship between components.** Click the orange dot on the right side of a component and drag to establish a relationship with another component. Components that are not appropriate will be grayed out.

  You can also click a component select the **Add** icon to see a list of compatible component relationships, then select a component to add to the design.

  You can create relationships only between components that have been configured to allow such relationships. In addition, the direction of the relationship (outgoing or incoming), and the number of allowable relationships are configured per component. See"Component relationships" on page 52.

  After the connection is made, the arrow automatically points in the proper direction, as defined by the component relationship configuration. Note that capabilities do not support outgoing relationships when included in a design; capabilities support only incoming relationships.

- **Remove a relationship between components.** Click a relationship arrow, then click the delete icon to delete the selected relationship.
- **Edit a component.** Select a component and then click the **Edit** icon. You can change its name and the group it belongs to.
- **Delete a component.** Select a component and click the **Delete** icon.
- **Set property values for components and relationships.** Select a component and its properties will be displayed in the panel on the right.

  Properties on a component can be set to modifiable during package deploy or redeploy. All properties can be modified during a test run.

  Properties on a component or relationship can set to modifiable during service modification to indicate that the property value can be modified during service modification. This check box is disabled if Modifiable during service creation is not selected for the property in the Properties tab of the Components area.

  Click **Save** to save your changes.
- **Select requirements for a capability.** Select a capability and then click Requirements in the panel on the right.

  Click **Select** to add or remove characteristics. If you filter by applicable characteristics, the list includes all characteristics supported by all concrete types that support the selected capability. For example, if there are concrete components Apache 2.2 Web Server and Apache 2.4 Web Server that support the Web Server capability, all characteristics supported by those concrete types will be displayed.
- **Create, edit, or delete a group.** See "Topology design groups" on the next page.
- **Assign components to a group.** See "Topology design groups" on the next page.
- **View validation errors.** Validation is a continuing process during design creation, configuration, and editing. Designs with validation errors can be saved but not published. Hover over an error icon to see more information about the validation error. Common sources of validation errors include required properties on components that are undefined, undefined groups, and required relationships that are not configured.

Best practices

- Partial designs contain capabilities and require composition with a compatible concrete design before they can be published and deployed.
- Use the icons at the bottom center of the Designer tab to use automatic layout or to zoom in and out.
- Click the arrow to open and close the component drawer if you need more space in the Designer window.
- Some features are not available for published designs; published designs are displayed in read-only mode.

Topology design groups

Groups are used to create scalable stacks of components in the Designer tab. The scalable stacks are made up of related components that can be scaled out together. For example, you might create a web server group that contains the following three components in its scalable stack:

- A web application
- An Apache web server
- The server component on which the other two components are hosted

The groups created for a topology design are unique to that design; they are not shared across designs. You can create a relationship between two components inside a group or between a component outside a group and a component inside a group, but you can't create a relationship between components that belong to different groups.

When you deploy a package from a partial design with a scalable group, you select a service design that has a group which meets the requirements of the design.

Manage groups

In the Designer tab, click **Manage Groups** in the lower left, which allows you to perform the following tasks for groups:

- **Create a group.** Click the create icon, and provide the following information for the group:

  Display Name - Type a unique display name for the group.

  Image - Select an image that will be displayed for the group.

  Color - Select a color that appears as the background for the group when it displays in the Designer tab.

- **Edit a group.** Click the edit icon and edit the display name, image, or color for a group.
- **Delete a group.** Select the group you want to delete and click the delete icon.
- **Move a group.** Click inside a group and drag the group to a new location.
- **View unused groups.** Information for unused groups is displayed in the lower left corner of the designer when a topology design contains an unused group. Unused groups are groups that have been defined but that do not contain components. Click **Show** to see the list of unused groups and click **Hide** to close the list.

Add components to a group

1. In the Designer tab, click a component in the design to access the menu that slides out to the right of the component. Click the edit icon, and select a group to which the selected component will be added.

2. Create relationships between components in the group. A group is displayed in the designer as a colored rectangle, which contains all the components in the group.

   You can create relationships between components that are part of the same group but not between components in different groups. See "Topology designer" on page 39.

3. Configure properties for the group. Click in the group's colored background or title bar, and do the following:

   a. In the Properties tab, in the right pane of the designer, set **Instance Count** to indicate the number of instances of the group that will be provisioned for the topology design.

   b. Set properties to **Modifiable only during package deploy** to indicate that the property value can be modified when a package is deployed or redeployed. All properties can be modified during a test run.

   c. Set properties to **Modifiable during package redeploy** to indicate that the property value can be modified during package redeploy.

   d. Click **Save**.

Test a topology design

Use the **Test** tab to test and manage topology design provisioning. You can test the following types of topology designs:

- **Partial design** - During the test run of a partial design, you will be prompted to select from the list of designs or microservices.

- **Concrete design**

See "Topology composition" on page 33 for information about these types of designs.

Tasks

You can view the following information in the **Test** tab:

- The name of the test run.

- Date and time of the test run.

- Current status of the test run.
- The user who initiated the test run.

You can perform the following tasks in the **Test** tab:

- **Cancel** - Cancel the realized design so that the instance is terminated. You must cancel before you can delete.
- **Delete** - Deletes a canceled or failed deployment instance. You must cancel before you can delete.
- **Test Run** - See Test Run Wizard below.
- **Refresh** - Refreshes the data in this tab.

You can link to detailed results of test provisioning by clicking the name of the test run. You can see the following information and perform the following tasks:

- **Overview** tab shows details of the test run.
  - **Cancel** the realized design so that the instance is terminated. You must cancel before you can delete.
  - **Delete** a canceled or failed deployment instance.
- **Events** tab shows:
  - **Event Time** - indicates the time the event occurred.
  - **Lifecycle State** - indicates the state of the event execution, for example, Deploying or Undeploying.
  - **Action** - events, such as deployment, server restart, etc., that are executed during the test run.
  - **Source** - the design component that is the source of the event.
  - **Status** - current state of the test run. If you click on an event you see details about the event in the Overview tab and property values used in the test run in the **Properties** tab.
- **Topology** tab shows:
  - The topology diagram, properties, status of the test, and an instance summary.
  - Select a server component in this tab to see properties in the right panel and available actions below. You can invoke an action to verify that it is working.
- **Providers** tab shows which provider instances have been used for deployment.

Test run wizard

Use **Test Run** on a topology design's Test tab.

The **Test Run** wizard allows you to specify the following:

- **General** - Set properties that are common to all designs.
  - Display Name - Type a display name or use the system proposed name of `Test run of <design name>`.
  - Environment - You can select an environment to restrict provider selection to only those providers located in the selected environment. Optional.
- **Service Designs** - Select a design from the list that, together with the partial design, will form a composite design that can be provisioned. If you select **Custom Selection**, you can select a microservice for each requirement in the partial design. Only service designs or microservices that fulfill the requirements are available.
- **Group Properties** - Properties for groups in the design.
- **Component Properties** - Properties for components in the design.
- **Relationship Properties** - Properties for partial designs. The list includes concrete designs that support all the capabilities and required characteristics in the partial design.

Best practices

- If you cancel a deployment, the test run is still listed in the tab. If you no longer need the entry, you can delete it.
- Select a new name for every test run so you can distinguish between them.
- Test your designs using various deployment scenarios and property configurations to ensure design stability before you create a package.

Access control

You can add users and groups to access control for a topology design.

- If a design is created by a user with the administrator role, then all users will have access to the design if you do not configure access control.
- If a design is created by a user with the Application Architect role, then only that user will have access to the design if you do not configure access control.
- If you add users or groups to access control, then only those users and groups can access the design and other users cannot see it.

Users with the Application Developer, Application QA, or Application Release Manager role have read-only access to designs and can view designs in the Release Pipeline area.

The users and groups are from an LDAP server and are configured in the Provider organization. See "Configure an organization" on page 11.

An external group is an LDAP group that includes users with different roles. The external group is not configured in the organization tab, but the user belongs to another group configured for the organization. When the external group is added to access control, the users who are part of other groups configured in the organization can log in and access the design. Users who are not part of the organization cannot log in.

Note: The admin user can create and access all designs, but is not listed in the Access Control tab.

To change access control for a design

1. Select the Access Control tab for a design.
2. Click **Manage**.
3. Select either:

   Do not restrict access to this design to allow all users to access the design. This is the default.

   Restrict access to this design to the following users and groups and select the users and groups who will have access.
4. Add or remove users and groups if access control is enabled.

   To search for a user or group, enter the user or group name in the dialog. You can use an asterisk (*) as a wildcard when you search for a user or group. Use Shift or Ctrl to select multiple users or groups.

   To search for a user with a specific role, that role must first be configured as a DN on the LDAP server and then the DN must be configured in the Provider organization. Roles are listed in the Add Users dialog even if they are not configured on the LDAP server or the Provider organization.

   To search for a group, the base DN of the configured LDAP server is searched.
5. Click **Save**.

Tasks

You can perform the following tasks in the Access Control tab if you have the correct permission:

- **Sort the list of users and groups.** You can sort the list by selecting By Users or By Roles in the upper left side of the tab. When you sort by role, groups from LDAP are sorted into External Groups, while groups from the organization are sorted by role.
- **Search for users and groups.** Use the search box in the upper left side of the tab.

Environment groups

You can assign different environment groups based on the lifecycle stage for a design. When the package is deployed, an environment is chosen from the group.

For example, you may want to restrict deployments in the Testing stage to certain environments. You would create an environment group that includes only the environments that are appropriate for this lifecycle stage. You would exclude environments where you don't want a test deployment created.

By default, a package can be deployed to all environments unless the design contains an environment association.

To change the environment groups for a lifecycle stage

1. Go to the Environments tab for a design.
2. Click **Manage Environments** to the right of the lifecycle stage.
3. Add or remove resource environments.
4. Click **Save**.

Components

Each component binds to a single provider for fulfillment automation. The component lifecycle provisioning is delegated to the providers.

Use the **Components** area in topology designs to import, create, and configure components. Codar provides a number of out-of-the-box components you can use for creating topology designs.

Concrete components

Concrete components are the most common components that you will interact with.

- Concrete components can contain properties, relationships, and operations.
- Concrete components can express support for capabilities, such as server, application server, database server, and web server, and can also support characteristics.
- Concrete components can be included in a design.

Abstract components

Abstract components provide a base type for other components and can be the target of relationships configured on a component.

- Abstract components can contain properties and relationships.
- Abstract components cannot express support for capabilities or characteristics.
- Abstract components cannot be included in a design.

Capability components

A capability component is used in a partial design and only contains properties or relationships. For example, a capability component might contain the properties required for an application server. It is associated with a concrete component that supports the capability. When a concrete component supports a capability, the concrete component inherits the capability's relationships and must provide property mappings from the concrete component properties to the capability properties.

- Capabilities can be the target of relationships configured on a component.
- Capabilities can be included in a design, but for such a design to be successfully provisioned, another design must exist that contains a concrete component supporting the capability.

Tasks

You can perform the following tasks in this area:

- **Search for components.** Enter your search string in the **Search** field.
- **Select how to view components.** Select to view: **By Palette**, **By Tag**, or **By Provider Type**. Click an item in the left pane to see its components displayed in the right pane.
- **Manage tags.** When you are in the **By Tag** view, click the **Manage Component Tags** icon in the lower left pane. Tags are labels that provide a structure for organizing and grouping related items. A component can be assigned to multiple tags.
- **Manage palettes.** When you are in the **By Palette** view, click the Manage Palette icon in the lower left pane. See "Palettes" on page 58.
- **Navigate to component details.** Select a component in the list to see more information about it.
- **Import components.** See "Import components" on the next page.
- **Create a topology component.** Click **Create** and provide the following information:

  Display Name - Type a unique display name for the component.

  Description - Type a description of the component.

  Version - Type a version number for the component.

  Functional Type - Select a functional type for the component. The functional type can be Abstract, Concrete, or Capability. See the descriptions in the Concepts section above for more information about the functional types.

  Provider Type - Select a provider type for the component.

  Image - Select an image that will display for the component.

  Tags - Select one or more tags that will include the component. Tags are user-defined, color-coded labels and images used to provide a structure for organizing and grouping topology components.

Best practices

If you have a large number of components, use tags to organize them into meaningful groups.

Import components

Importing provides you with the ability to use content from sources external to Codar. Importing is the only way to add additional components, as you cannot create components manually.

Importing components is done through an import wizard, with options based on the selected **Import Source** (provider type). The provider types are:

- Puppet
- Chef
- Operations Orchestration
- Server Automation

Importing components

From the Topology Components area, click **Import**, and provide the following information:

Item	Description
General	• **Import Source** - Select the provider type to use as the import source for the component. • **Provider Instance** - Select the provider instance that contains the component you want to import. Provider instances must first be configured in the Providers area to enable import for Chef, Server Automation, and Puppet provider types. For Operations Orchestration, content can be imported from the instance that is configured to integrate with Cloud Service Automation. • **Image** - Select an Image that will display for the component. • **Tags** - Select one or more tags that will include the component. Tags are user-defined, color-coded labels and images used to provide a structure for organizing and grouping topology components.

Select Content	Depending on your selections in the General tab, select the content for the component: • **Chef** - Select one or more Chef cookbooks to be imported as new Cloud Service Automation topology components. Each selected cookbook will create a separate component. • **Operations Orchestration** - Browse the Operations Orchestration library and select one or more standard component elements that will be imported as new topology components. Each selected item will create a separate component. • **Server Automation** - Select one or more Server Automation policy to be imported as new topology components. Each selected policy will create a separate component. • **Puppet** - Select one or more Puppet classes to be imported as new topology components. Each selected class will create a separate component.
Summary	Review your import selections, and click **Import**. The imported components will appear in the Topology Components list.

Component overview

You can view a summary of information about the selected component in the **Overview** tab.

Tasks

You can perform the following tasks in this area:

- **View a component.** See the following component information:
  - **Display Name** - The name displayed for the component.
  - **Description** - The description configured for the component.
  - **Version** - The version configured for the component.
  - **Provider Type** - The provider type configured for the component.
  - **Functional Type** - The functional type can be abstract, concrete, or capability. See "Components " on page 46.
  - **Image** - The image configured for the component.
  - **Tags** - The tags configured for the component.
- **Edit a component.** The following restrictions apply to the editing of components:
  - For some out-of-the-box components, you can edit the **Display Name**, **Description**, **Image**, and **Tags**.
- **Save a component with a new name.** Save the component with a new **Display Name** and **Description**. **Save As** cannot be used for abstract components or capabilities; it can be used only for concrete components.

- **Delete a component.** You cannot delete a component that is being used in a topology design. Some out-of-the-box components cannot be deleted.

Component properties

Properties provide a base set of attributes that can be used and edited when creating components in a service design. They represent configuration settings to be applied to the component during service design provisioning. The value defined for a component property is the default value exposed in the service design.

Tasks

View the following information about the properties:

- **Display Name** - The display name of the property. You can click the property name to open the **Edit** window. An icon next to the **Display Name** indicates that the property can be modified during the modify lifecycle transition phase.

- **Value** - The default value for the property.

Use the corresponding icon or button to:

- See whether the property is **Visible/Invisible** to the subscriber.

- **Show/Hide** the description.

- **Create or Edit a property**
  - You cannot create or edit properties of a component that is an abstract component or a capability.
  - Newly created properties do not automatically affect service designs that have been provisioned. You must manually update any design that uses the component by re-creating the component in the design.
  - For components that are being used in a topology design, you can only add additional properties, and the new properties cannot be **Required**. You cannot modify existing properties, except to change the default values for **Display Name**, **Description**, **Default Value**, and **Can Be Modified**.

Provide the following information for a property:

Item	Description
Type	• **Boolean** - True or False. • **Integer** - A positive or negative whole number or zero. • **List** - A list of string values. • **String** - A sequence of characters.
Name	A unique name for the property. This value cannot be changed after the component is created.

Item	Description
Display Name	The name that displays for the property.
Description	The description that displays for the property.
Resource Type and Unit for a Measurable Property	Available for **Integer** properties only. Select a resource type and unit (other than **None**) to make this property measurable and to enable automatic accounting of resource usage in a provider's resource pool. This field is available for all integer properties on concrete components.
Default Value	The default value exposed in the service design.
Confidential Data	Select this box to mask the values so that they cannot be read in the user interface; no encryption of the value is performed.
Visible	Use the check box to indicate if the property is visible in the topology design.
Required	Use the check box to indicate if the property is required in the topology design.
Can be Modified	• Select **No** to specify that the property cannot be modified in the topology design. • Select **Yes (checked by default)** to allow the property to be modified in the topology design and for the **Modifiable during service creation** box to be checked by default in the topology designer. • Select **Yes (unchecked by default)** to allow the property to be modified in the topology design and for the **Modifiable during service creation** box to be unchecked by default in the topology designer. An icon below the **Can be Modified** field indicates that the property can be modified during the modify lifecycle transition phase. This indicates that subscribers can change the property value after the component has been instantiated, and allows the change to be properly reflected when the modify lifecycle transition occurs. For example, the **vCenter Server** component’s **cpuCount** and **memorySize** properties can be modified after initial provisioning, allowing a subscriber to change these values during a subscription modify operation. You cannot configure whether a given property has the capacity to be modified during the modify lifecycle transition phase. Out-of-the-box components have the appropriate setting configured automatically for each of their properties. For components imported from Operations Orchestration, the signature of the Operations Orchestration flow dictates which properties are marked as being able to be modified during a modify operation.

- **Delete** the property. You cannot delete properties that are being used in a saved design.

  Note that if you delete a property that supports the modify lifecycle action (indicated by an icon next to the **Display Name** of the property), and you later want to add the property back into the component, you must re-import the corresponding modify lifecycle operation to ensure that the support for the modifiable lifecycle action is restored to the property.

Component relationships

Relationships in topology designs define dependencies between components and also impact how a design is provisioned. For example, imported Chef components require a server in order to be provisioned. Therefore, all imported Chef components are created with an outgoing relationship to the server capability, ensuring that a server is provisioned before the Chef component.

When importing new components, you may need to define new relationships. Some relationships are created for you automatically at import time, such as the Chef component dependency on server discussed above. If there are additional dependencies, define them manually in this tab.

Tasks

View the following information about the relationships:

- **Direction** - Outgoing or Incoming.
- **Target or Source Component** - Click the component name of the target or source component to open the **Overview** tab for that component.
- **Display Name** - The display name of the relationship.

Use the corresponding icon or button to:

- **Create or edit a relationship.**
  - You cannot create or edit incoming relationships.
  - You cannot edit a component that is an abstract component or a capability.
  - After a component is used in a design, you cannot add a new *required* relationship or make an existing relationship required. You can only change its display name and description.

    Provide the following information for a relationship:

Item	Description
Target Component	Select the target component for the relationship from the **Select Component** dialog, which provides a list of available components.
Display Name	The name for the relationship.
Description	The description for the relationship.
Maximum Outgoing Relationships From Source	The number of instances of the target component that a single instance of the source component can be connected to.

Item	Description
Maximum Incoming Relationships To Target	The number of source components that can be connected to a single instance of the target component.
Required Relationship	Use the check box to indicate if the relationship is required in topology designs.

- **Delete a relationship.** Click the delete icon. You cannot delete incoming relationships.

Component operations

View the following information about the operations:

- **Display Name** - The display name of the operation.
- **Parameters** - The parameters associated with the operation.

Use the corresponding icon or button to:

- See whether the operation is **Visible/Invisible** to the subscriber.
- **Show/Hide** the description.
- **Edit** an operation.

You cannot edit a component that is an abstract component or a capability. Provide the following information for an operation:

Item	Description
General	• **ID** - A unique identifier for the operation. You cannot edit this field. • **Display Name** - The name that displays for the operation. • **Description** - The description that displays for the operation. • **Visible** - Use the check box to indicate if the property is visible. • **Lifecycle Action** - Select a lifecycle action that will be used during provisioning or de-provisioning: ■ **Deploy** – Actions specified in this lifecycle phase realize a component during the provisioning of a design that is using the component. ■ **Deploy Failure Handler** – Actions specified in this lifecycle phase handle failures that occur during the execution of the **Deploy** lifecycle phase action on a component as part of provisioning a design. ■ **Undeploy** – Actions specified in this lifecycle phase de-provision a component during the de-provisioning of a service instance that is using the component. ■ **Undeploy Failure Handler** – Actions specified in this lifecycle phase handle failures that occur during the execution of the **Undeploy** lifecycle phase action on a component as part of de-provisioning a design. ■ **Modify** – Actions specified in this lifecycle phase are executed on a component during the modification of a realized component in a service instance, when the service instance is modified by the **Modify Subscription** operation. ■ **Modify Failure Handler** – Actions specified in this lifecycle phase are executed for the cleanup of failed **Modify** actions on a component during the modification of a service instance. ■ **Unmodify** – Actions specified in this lifecycle phase revert the effect of modification on a successfully modified component. This action occurs during the rollback of an unsuccessful overall modification transition of the service instance due to failure occurring on another component that is also subject to modification. The result is that the service instance reverts to the state it was in prior to the modification operation.
Parameters	• **Input Parameter Mappings** - The list of input parameters and their configured parameter mappings for this operation. Click the appropriate icon to edit or delete an input parameter. See below for more information about editing. • **Output Parameter Mappings -** The list of output parameters and their configured parameter mappings for this operation. Click the appropriate icon to edit or delete an output parameter. See below for more information about editing.

Item	Description
Create or Edit Parameter Mapping	**Create or Edit Parameter Mapping** To create a parameter mapping, click the **Add Parameter** button just below the **Input Parameter Mappings** or **Output Parameter Mappings** list. Typically, you will add a parameter if the operation is tied to an Operations Orchestration flow and you have modified the flow to have new inputs or outputs. To edit an existing parameter mapping, click the **Edit Parameter Mapping** icon. • **Name** - A unique name for the parameter. • **Display Name** - The name that displays for the parameter. • **Description** - The description for the parameter. • **Required** - Use the check box to indicate if the parameter is required. • **Mapping Type** - Select the mapping type for the parameter mapping. ■ **Not Mapped** - Select to provide no parameter mapping. ■ **User Value** - Select to prompt the user for a value during design execution. This mapping is valid for public actions only. When a public action is executed, the user is asked to provide values for all parameters mapped to the **User Value**. ■ **Component Property** - Select to map the parameter to or from a property on this component. Select the component property in the **Value** list. ■ **Constant Value** - Select to provide a constant value for the parameter. Enter the constant value in the **Value** field. This is not available for output parameters. ■ **Multiple Properties** - Map to multiple properties. Click **Add Parameter** to create additional parameters. ■ **Provider Property** - Select to map the parameter from a property on the resource provider that is used to provision the component. Enter the name of the resource provider property in the **Value** input field. ■ **Relationship Target Property** - Select to map the parameter from a property on a different component that this component has a relationship to. Select the proper relationship in the **Relationship** list, and then select the component property in the **Value** list.

- **Delete** the operation.
- **Import** an operation. You can import an operation, assign a lifecycle action (for example, Deploy or Undeploy), and edit operation parameter mappings. For example, you can import a new Operations Orchestration operation into an existing Chef component. You can import an operation either from an Operations Orchestration live instance or from a content pack.

Component capability

See "Components " on page 46 for a definition of capability components.

Concrete components can claim support for a capability component. A supported capability includes a reference to the capability component, as well as property mappings between the capability component and the concrete component. A single capability, such as Web Server, can be supported by multiple concrete components; however, a concrete component can only support a single capability.

A capability component can be included in a design to indicate that a concrete component supporting that capability is required in order to successfully provision the design. Designs that include capabilities are referred to as partial designs and require composition with another design in order to be provisioned. For additional information on partial designs, see "Test a topology design" on page 42.

Tasks

View the following information about the capability:

- **Supported Capability** - The display name of the supported capability, if any. Supported capabilities are provided out-of-the-box for some concrete components.

- **Property mappings** - A list of each **Capability Property** and the **Component Property** from which it gets its value.

Perform the following tasks:

- **Refresh the data in this tab.**

- **Add or Edit a supported capability.** You cannot add or edit a capability if the component is being used by a topology design. Click **Add Supported Capability** or select a supported capability and click **Edit**. Provide or modify the following information:

Item	Description
Selected Capability	Choose a capability to add to this concrete component. You cannot edit this field after a supported capability has been added.
Property Mappings	For each **Capability Property** in the list, do one of the following: • Select a **Component Property** from which the **Capability Property** gets its value. • Leave the **Capability Property** as **Not Set**. Note that any capability property that has the same 'property Name' as a concrete component property (using a case-sensitive string match) is automatically mapped.

- **Remove the selected capability.** You cannot remove a capability from a component if the component is being used by a topology design.

Component characteristics

The **Characteristics** tab is visible for concrete components.

Characteristics describe a component and must be used in combination with capabilities. For example, an Apache 2.4 Web Server concrete component may define support for the following characteristics: Apache Web Server, Apache Web Server 2.x, or Apache Web Server 2.4.

If this component supports the Web Server capability, a service designer can include a Web Server capability in a design and specify that the design requires the Apache Web Server 2.4 characteristic. This allows the Apache 2.4 Web Server component (and any other Web Server components supporting this characteristic) to be chosen as part of provisioning the component.

A component should be associated with a capability when supported characteristics are configured. For more information, see "Component capability" on the previous page.

Tasks

View the following information about the characteristics:

- **Supported Characteristics** - See a list of the characteristics supported by the component.

Perform the following tasks:

- **Select characteristics** - Click **Select**. Or, if no characteristics are configured for the component, click **Select characteristics for this component**. See the following table for selection information:

Item	Description
Available characteristics	Select one or more characteristics in the list, and click **Add**.
Selected characteristics	View the list of selected characteristics, or select one or more and click **Remove**.
Manage characteristics	**Create a characteristic** - If this is your first characteristic, click **Create your first characteristic now**. • Provide a **Display Name** for the characteristic. • Provide a **Description** for the characteristic. When characteristics already exist, you can: • Click the create icon to create a new characteristic. • Click the edit icon to edit a selected characteristic. • Click the delete icon to delete a selected characteristic. You cannot delete a characteristic that is a required characteristic in a partial design.

Palettes

A palette is a group of related components that can be used in a topology design.

Tasks

After a palette is created, you can do the following from the Overview tab:

- **Save a palette.**
- **Edit a palette.** You can edit global properties, such as name and description.
- **Delete a palette.**

From the Components tab, you can click a component name to open it for editing. You can also:

- **Remove selected components from the palette.**
- **Add a component to the palette.** You can filter the list of available components by tag, palette, or provider type.

Release Automation

The Release Automation tile defines the configuration tasks required to move a design across lifecycle stages. This tile is further divided into three tiles:

- Release Pipeline - used to manage packages, create release gate actions, manage environments, and view pipeline statistics
- Pipeline Configurations - used to create and configure lifecycle stages, create roles, associate roles with a fixed set of permissions, and associate lifecycle stages with the roles.
- Approvals - used to manage approval requests for approval actions. The approver can view the approval requests for stage transition and approve or reject these requests.

Release pipeline

This tile displays information for application designs and microservices. Only designs that have been marked as application or microservice are displayed.

Application designs

To view application designs, select **Application Designs** in the list on the left area of the page. The designs that you can see may be limited by access control that is configured for each design.

The **Versions** tab lists all versions of an application design and the number of packages in each lifecycle stage.

The **Pipeline Statistics** tab displays a graphical representation of package data and equips users to make informed decisions based on the statistics.

The **Team** tab shows all users and groups who have been granted access to the design. This is a read-only view and can only be changed in the Design area.

Microservices

When a new lifecycle stage is created, a new microservice option is added. To view microservices, select a lifecycle stage microservice in the upper right area of the page.

View different versions for a single application design

The **Release Automation** > **Release Pipeline** > **Versions** tab displays different versions of a selected application design. It also displays all the lifecycle stages present in the design and the number of packages present in each stage. You can drill-down further and view the package overview and workflow by clicking a version number.

View the package workflow

The Package Workflow screen displays all the lifecycle stages in an application design and the state of the packages in each lifecycle stage. To view the Package Workflow screen:

1. Navigate to **Release Automation** > **Release Pipeline**.
2. Click the version for which you want to view the package workflow.

   The Package Workflow screen is displayed. This screen has several views:

   - View by Packages: Select **View by Packages** from the drop-down list on the right side of the screen to view all the packages in the design. Packages are displayed in their respective lifecycle stages. You can further filter by package states by selecting a package state from the drop-down list next to the **View by Packages** list.
   - View by Gate Actions: Select **View by Gate Actions** to view Release Gate Actions configured in respective lifecycle stages.
   - View by Environments: Select **View by Environments** to view the selected Resource Environments in respective lifecycle stages.
3. You can "Release gate actions" on the next page for each lifecycle stage.
4. You can associate an environment with each lifecycle stage.
5. You can configure lifecycle stages by adding or removing a lifecycle stage. The lifecycle stage you add or remove applies only to the specific version you select. To configure lifecycle stages, select the gear icon on the right side of the screen and click **Configure Lifecycle Stages**.

Release gate actions

Codar allows you to configure release gate actions at each lifecycle stage. These actions act as a gate between two lifecycle stages. A package has to pass through each action that is defined; hence they are called release gate actions. During promotion of the package, the gate actions defined in each are executed. After all the actions complete successfully, the package transitions to the next lifecycle stage.

Note: You need permission to configure release gate action.

Release gate actions are of three types:

- Deploy - this action deploys the Codar application design on the specified target environment. If the design is a partial design, the microservice designs that meet the requirements of the application design must also be selected.
- Custom - this action executes the specified flow in Operations Orchestration. Typically this action can be used to execute specific tests on a deploy instance.
- Approval - this action is used to configure approvers who will approve the package promotion. It is also used to configure the minimum number of approvals, automatic approvals, and the number of days after which the automatic approval process is triggered.

Deploy action versus custom action

Deploy actions are typically created to test your environment and to perform a test deployment before promoting your design to the production stage. Deploy actions also serve to be input to custom actions. You cannot create a custom action without first creating a deploy action.

Custom actions are typically used to execute operations or tests on a deployed instance. You must specify the deploy action while creating a custom action. This enables instance properties such as host name of a deployed environment to be passed as input to the Operations Orchestration workflows so that specific actions can be executed on the deployed instance.

Create a deploy release gate action

You can create a deploy release gate action during any stage in a lifecycle. To create a deploy release gate action, for example, in the development stage:

1. Navigate to **Release Automation** > **Release Pipeline**.

2. Select the application and the version for which you want to create a release gate action.

3. Click the gear icon next to the lifecycle stage in which you want create the action and click **Configure Gate Actions**.

4. Click **Add Action** > **Deploy Action** to create a deploy action.

5. In the Add Deploy Action dialog box, fill in the following fields:

Field	Description
Display Name	Name of the custom action
Description	Description of the action you are creating
Environment	Select the environment in which you want to create the deployment.

6. In the Success and Failure Configuration area, select the check boxes as applicable:

Check box	Description
Notify on Success	Select this check box for a notification message to be sent if the deploy action executes successfully. The notification message is sent to the user who initiated the deploy action.
Notify on Failure	Select this check box for a notification message to be sent if the deploy action fails to execute. The notification message is sent to the user who initiated the promotion request.
Reject on Failure (Blocking)	Select this check box to reject the package if the deploy action fails to execute.
Cleanup on Promote Success	Select this check box undeploy or clean up the deployed instances if the package is successfully promoted to the next stage.
Cleanup on Promote Failure	Select this check box to undeploy or clean up or remove the deployed instance if the package has failed to promote to the next stage.

7. Click **Add Action**.

   The action is listed in the Configure Development Gate Actions dialog box.

Edit or delete a deploy release gate action

You can edit or delete the details of a release gate action during any stage in a lifecycle.

Edit a release gate action

To edit a release gate action:

1. In the Configure <*lifecycle stage*> Gate Actions dialog box, click the gear icon next to the release gate action that you want to edit and click **Edit**.

2. In the Edit Deploy Action dialog box, edit the details of the action. You can edit the name, description, environment, or select/deselect any of the check boxes in the Success and Failure Configuration area. For information about these fields, see Create a release gate action.

3. Click **Save Action**.

   The action has been edited.

Delete a release gate action

To delete a release gate action:

1. In the Configure <*lifecycle stage*> Gate Actions dialog box, click the gear icon next to release gate action that you want to delete and click **Delete**.

2. Click **Yes** in the confirmation box.

> If the deploy action is associated with a custom action, you cannot delete the deploy action without first deleting the custom action.

The action is deleted.

Create a custom release gate action

You can create a custom release gate action during any stage in a lifecycle.

Ensure that you have created a deploy action before creating a custom action.

To create a custom release gate action:

1. Navigate to **Release Automation** > **Release Pipeline**.
2. Select the application and the version for which you want to create a custom release gate action.
3. Click the gear icon next to the stage in which you want to create the custom action and click **Configure Gate Actions**.
4. Click **Add Action** > **Custom Action** to create a custom action.
5. a. In the Add Custom Action dialog box, fill in the following fields:

Field	**Description**
Display Name	Name of the custom action
Description	Description of the action you are creating
Select Deployment	Select the deploy action that must be a part of the custom action. All the deploy actions that have been created for the selected package are displayed. You cannot create a custom action without selecting a deploy action.

   b. In the Success and Failure Configuration area, select the check boxes as applicable:

Check box	**Description**
Notify on Success	Select this check box for a notification message to be sent if the custom action executes successfully. The notification message is sent to the user who initiated the promotion request.
Notify on Failure	Select this check box to receive a notification email if the custom action fails to execute. The notification message is sent to the user who initiated the promotion request.

Check box	Description
Reject on Failure (Blocking)	Select this check box to reject the package if the custom action fails to execute.
Ignore on Failure	Select the check box to mark the specified action optional (non-blocking). The execution will move to the next release gate action even if the action fails to execute.

> You can select more than one check box. However, for the **Reject on Failure** and the **Ignore on Failure** options, you can select only one of them at the same time.

6. Click **Next**.

7. In the Custom Action dialog box, select an OO workflow for the custom action. The OO workflow defines the course of your custom action and is invoked by passing the relevant parameters during the execution of the custom action.

8. Click **Next**.

9. The Parameters dialog box displays the parameters present in the OO workflow. You need to map these parameters to the components and component properties present in the design. To map the parameters:
   a. Select the gear icon next to the parameter you want to map to and click **Edit**.

   b. In the Edit Parameter Mapping dialog box, fill in the following fields:

Field	Description
Display Name	Name of the parameter. You can change the default name to a name of your choice.
Description	Description of the parameter
Mapping Type	Object that you want to map the parameter to. You can either map the parameter to a component property, context property or constant value. If you select Context Property as the Mapping Type, you must select any of the following Context properties from the **Value** field. If you select Constant Value as the Mappying Type, you ,must provide a constant value in the **Value** field.
Components	Displayed only if you selected Component Property in the **Mapping Type** field. Select the component that you want to map the parameter to. All the components that are displayed are a part of the Codar design.

Field	Description
Component Property	Displayed only if you selected Component Property in the **Mapping Type** field. Select the component property for the component you selected. This is a dynamic list whose values change depending on the component you selected. The value of the property is obtained from the deployed instance selected for this custom action.

> Mapping of parameters marked with an asterisk is mandatory. If you do not map such parameters, an error message is displayed when you try to add a custom action.

c. Click **Save**.

10. Click **Add Action**.

The action that you added is listed in the Configure <*stage*> Gate Actions dialog box.

Edit or delete a custom release gate action

You can edit or delete the details of a custom release gate action during any stage in a lifecycle.

Edit a custom release gate action

To edit a custom release gate action:

1. In the Configure <*lifecycle stage*> Gate Actions dialog box, click the gear icon next to release gate action that you want to edit and click **Edit**.

2. In the Edit Custom Action dialog box, edit the details of the action. You can edit the name, description, deployment, or select/deselect any of the check boxes in the Success and Failure Configuration area. For information about these fields, see "Create a custom release gate action" on page 64.

3. You can also edit the name, description, mapping type, components, or component type of any parameter.

4. Click **Save Action**.

   The action has been edited.

Delete a custom release gate action

To delete a custom release gate action:

1. In the Configure <*lifecycle stage*> Gate Actions dialog box, click the gear icon next to release gate action that you want to delete and click **Delete**.

2. Click **Yes** in the confirmation box.

The action is deleted.

Create an approval release gate action

An approval action acts as a final gate before you promote your package to the next lifecycle stage. You can choose the users who must approve the promotion to the next stage. Choosing users that belong to administrator or similar roles and granting the right to approve to a limited set of users minimize the chances of accidental promotion to the next stage.

When an approval action is executed, an email is sent to the approvers notifying them of a pending approval request. They can then approve or deny the promotion using the **Release Automation** > **Approvals** tile.

When an approval action is executed, an email is sent to the approvers notifying them of a pending approval request.

To create an approval action

1. In the **Release Pipeline** tile, click **Packages**, select the application and the version for which you want to create an approval action.
2. Click the gear icon next to the Development stage and click **Configure Gate Actions**.
3. Click **Add Action** > **Approval Action** to create an approval action.
4. In the Add Approval Policy dialog box:
    a. In the **Display Name** filed, enter the name of the approval action.

    b. In the **Description** filed, enter a description for the action you are creating.

    c. In the **Assigned Approvers** field, click the + button to search for and add approvers who must approve the promotion. When you click the + button, the Add Users dialog box is displayed.

    d. In the Add Users dialog box, the drop-down list displays all the roles that are present in the design. You can select a particular role and then type the user name or the starting alphabets of the user name that belongs to the role.

    > Only user names configured in LDAP are displayed in the drop-down list.

    e. Click **Search**. All user names that belong to the role you selected are displayed. If you are not sure which role the user name belongs to, select **All Roles**. You can select multiple user names.

    f. In the **Minimum Approvals/Denials** field, enter the minimum number of approvers who must approve or reject the promotions. For example, if you enter 2, at least 2 of the approvers you added in the **Assigned Approvers** field must approve for the package to be promoted to the next lifecycle stage.

g. Select the **Automatic Approval** check box for the package to be automatically promoted or denied if the assigned approvers do not approve or reject within the days specified in the **Wait time for Automatic Approval** field.

h. In the **Automatic Approve/Deny** drop-down list, select whether the package must be approved or rejected during the approval process.

   For example, if the minimum number of approvers is 3, and you select **Approved** from the **Automatic Approve/Deny** drop-down list and specify a wait time of 2 days, then even if only 1 approver approves and the other 2 approvers do not respond, the package is automatically promoted to the next lifecycle stage after 2 days.

   > The **Automatic Approve/Deny** and **Wait time for Automatic Approval** fields are displayed only if the **Automatic Approval** check box is selected.

i. In the **Wait time for Automatic Approval** field, enter the number of days after which the package is automatically automatically promoted to the next lifecycle stage.

j. Click **Add**.

An email is now sent to all the approvers named in the **Assigned Approvers** field. Approvers have the right to "Approve or deny an approval request" on page 82.

Edit or delete an approve release gate action

You can edit or delete the details of an approve release gate action during any stage in a lifecycle.

Edit an approve release gate action

To edit an approve release gate action:

1. In the Configure <*lifecycle stage*> Gate Actions dialog box, click the gear icon next to release gate action that you want to edit and click **Edit**.

2. In the Edit Approve Action dialog box, edit the details of the action. You can edit the name, description, assigned approvers, and so on. For information about these fields, see "Create an approval release gate action" on page 68.

3. Click **Save Action**.

   The action has been edited.

Delete an approve release gate action

To delete an approve release gate action:

1. In the Configure <*lifecycle stage*> Gate Actions dialog box, click the gear icon next to release gate action that you want to delete and click **Delete**.

2. Click **Yes** in the confirmation box.

The action is deleted.

Notifications

Different types of notifications are sent during the execution of release gate actions. They are:

- Stage transition notifications

  By default, notification messages are sent to all users or groups added in the Manage Access Control dialog box of the application design when a package is promoted successfully from one lifecycle stage to another. This notification is sent only if users or groups are added to the application design under access control.

- Action complete notification

  The notification is sent to the user who promoted the package. This notification is governed by the **Notify on Success** and **Notify on Failure** options available in each action.

- Approval request notification

  This notification is sent to approvers informing them of an approval request that is triggered during the execution of a release gate approval action.

.

Associate an environment with a lifecycle stage

You can select resource environments that already exist and associate it with lifecycle stages. This helps you choose a subset of deployment environments to be associated with the lifecycle stage. By default all resource environments are associated with all lifecycle stages.

To select an environment to associate

1. Navigate to **Release Automation** > **Release Pipeline**.

2. Select the application design and version in which you select the environment.

3. Select the ⚙ icon in the lifecycle stage with which you want to associate the environment and click **Select Environments**.

4. The Select Environments for <stage> dialog box displays all available resource environments. Select those that you want to associate and click **Add**. Selected environments are displayed in the Selected Environments box.

5. You can even search for environments using the Search fields. Type the name of the environment that

you want to search for and press the Enter key.

6. Click **Save** to add the selected environments.

View and cancel deployments

You can view all the deployments that you triggered during any lifecycle stage and choose to remove any of those deployments that are no longer relevant.

To view and remove deployments

1. Navigate to **Release Automation** > **Release Pipeline**.
2. Select the application design and the version for which you want to view the deployments.
3. Click the **Deployments** tab.

   All the deployments that you triggered are displayed in a tabular format along with details such as the packages in each deployment, the lifecycle stage in which the deployment was triggered, and the status of the deployment.
4. You can filter the deployments based on lifecycle stages to view deployments pertaining to a specific lifecycle stage.
5. To remove a deployment, click the gear icon and click **Cancel**.

   All the packages in the deployment are listed in the Cancel Deployments dialog box. You can choose to remove one package or multiple packages.

   > To delete a deployment, you must first cancel it.
6. Click **Cancel Deployments** and then **Yes** in the Confirmation Required confirmation box.
7. Click the gear icon and click **Delete**.
8. Select the packages to delete and click **Delete Deployments** in the Delete Deployments dialog box.

Configure lifecycle stages

You can configure lifecycle stages by adding or removing a lifecycle stage to the existing subset of lifecycle stages. The lifecycle stage you add or remove applies only to the specific version you select.

To configure lifecycle stages:

1. Go to **Release Automation** > **Release Pipeline**.
2. Select the application design and version for which you want to configure the lifecycle stages.

3. Select the gear icon on the right side of the screen and click **Configure Lifecycle Stages**.

4. Click **Add Lifecycle Stage** to add a lifecycle stage to the existing set of lifecycle stages.

5. To remove a lifecycle stage, click the bin icon next to the lifecycle stage that you want to remove.

6. To change the order of the lifecycle stages, drag and drop them above or below each other in the order you want.

7. Click **Save**.

View package statistics

The Pipeline Statistics tab displays detailed information about packages and includes graphical representations of package summary, trends, states, deployment status and so on. It provides a holistic view of all packages and deployments and enables you to make informed decisions with respect to package deployment.

To view package statistics

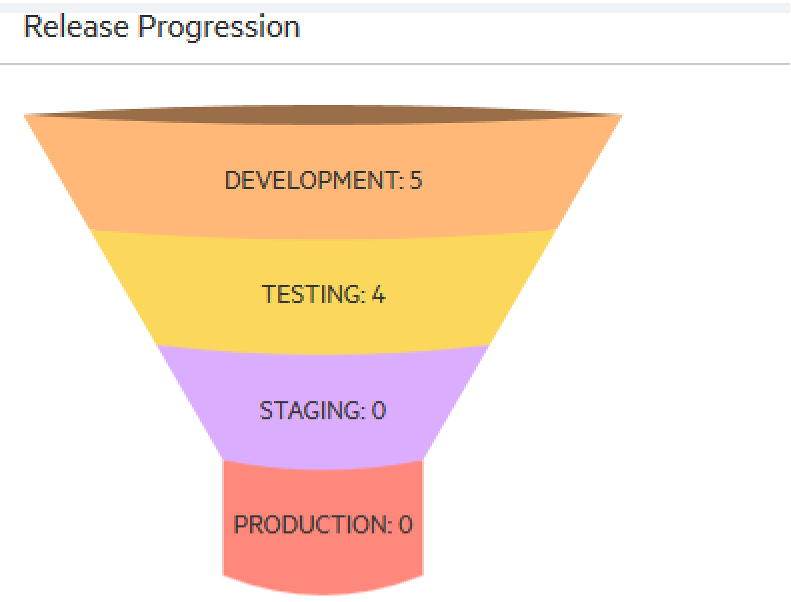
1. Navigate to **Release Automation** > **Release Pipeline**.

2. Select the application design and version in which you to view the package information.

3. Click the **Pipeline Statistics** tab.

   The following table lists the metrics that comprise pipeline statistics:

Metric	**Description**
Package Trend	Displays in line graph format, the days during which packages were created. The X-axis represents the dates and the Y-axis represents the number of packages. Hover your cursor on the X-axis to view the number of packages created on any date.

Release Progression	The mouth or the top layer of the funnel represents the first lifecycle stage and the bottom layer of the funnel represents the last lifecycle stage. Each layer displays the total number of packages that are not only present in the stage but also those that have successfully transitioned to the next stage. The colors that represent each lifecycle stage in the funnel are the colors that were set when the lifecycle stage was created.  For example, in the figure above, a total of 5 packages were in the Development stage, out of which 4 were sent to the Testing stage. From the Testing stage, 0 were sent to the Staging stage and from there 0 were sent to the Production stage. This implies that out of 5 packages, 4 are in the Testing stage and 1 is still in the Development stage.
Package Summary	Displays in pie chart format the number of packages in each lifecycle stage. The colors used for each lifecycle stage are the colors set when the lifecycle stage was created. The number of packages in each state is displayed below the pie chart.
Deployment Summary	Displays in pie chart format the number of deployments in each lifecycle stage. The colors used for each lifecycle stage are the colors set when the lifecycle stage was created. The number of deployments in each state is displayed below the pie chart.

Package Status	Displays in bar graph format, in each lifecycle stage the number of packages in each state. The X-axis represents the lifecycle stages and the Y-axis represents the number of packages. The different colors represent the different package states. Select **Grouped** to view different bars for each package state or **Stacked** to view all package states in one bar. See "Lifecycle stages and actions" below for details about package states.
Deployment Status	Displays in bar graph format, in each lifecycle stage the number of deployments in each state. The X-axis represents the lifecycle stages and the Y-axis represents the number of deployments. The different colors represent the different deployments. This graph depicts the number of deployments that are not canceled in the system and how many resources are consumed. Select **Grouped** to view different bars for each deployment or **Stacked** to view all deployments in one bar.

View access control for a design

The **Team** tab shows all roles that have been granted access to the application design. Under the roles, it also displays individual users that belong to the role. This is a read-only view and can only be changed in the Design area. See Access Control.

You can filter the page either by roles or by users. If you filter by roles, all the roles are displayed and users that belong to each role are displayed under that role. If you filter by users, all users are displayed and all the roles that a user belongs to are displayed under that user.

The Team tab is a read-only view and can only be changed in the Design area. See Access control.

Pipeline Configurations

Use the Pipeline Configurations tile to create and configure lifecycle stages and roles. You can also assign permissions to the roles.

Lifecycle stages and actions

Every lifecycle comprises a stage and every stage has roles associated with it. It means that all users who belong to the roles in a particular lifecycle stage can perform operations defined in the role during that stage. For example, if the Development lifecycle stage has the Application Architect role associated with it, then users belonging to the Application Architect role can perform tasks as per the assigned permissions.

The following are the out-of-the-box lifecycle stages available in Codar:

- Development: This is usually the first stage in which the code is developed and application artifacts are created.
- Testing: Stage in which test cases are executed against the code developed in the Development stage.

- Staging: Pre-production stage that replicates the production environment; used to test the code and artifacts.
- Production: This is usually the final stage in which the application is deployed in a live environment.

Apart from the out-of-the-box lifecycle stages, there are custom stages that you can create. For information about creating, editing, and deleting custom stages, see Create a lifecycle stage and Edit or delete a lifecycle stage.

The following table lists the actions pertaining to a package that take place in each lifecycle stage:

Stage	Promote	Deploy, Redeploy	Edit	Delete	Reject
First stage (usually the Development stage)	Yes	Yes	Yes	Yes	Yes
Intermediate	Yes	Yes	Yes	Yes	Yes
Final stage (usually the Production stage)	No	Yes	Yes	Yes	Yes

You can access lifecycle stages by using the **Release Automation** > **Pipeline Configurations** tile.

Use the following actions to deploy or move a package through the stages. These actions describe the flow when release gate actions are not defined.

- **Promote**: Moves the package to the next lifecycle stage. The package state remains Active.
- **Deploy, Redeploy**: Deploys the package. See "Deploy a package" on page 28.
- **Edit**: Change the properties of a package. See "Edit a package" on page 27.
- **Reject**: Stops the package from advancing to another stage. The package will remain in its current stage, its state will be set to Rejected, and the action buttons will no longer be available.
- **Delete**: Delete a package. The package will be removed permanently from the system. A package can only be deleted if all associated deployed instances are canceled and deleted.
- **Refresh**: Retrieves current package status.

Package states

Packages have the following states:

- Active
- Rejected
- Transitioned
- Failed

If you reject a package, then it remains in its current stage, its state is set to Rejected, and no further actions can be applied; however, it can be deleted and will be removed from the system.

When a package is promoted, it moves to the next stage and remains in the active state. Packages are always created in the first stage. If the Codar Jenkins plug-in is configured, then after a successful build the Jenkins plug-in talks to Codar and creates a package.

Grouping service designs by lifecycle stage

A partial design with an active package requires a selection of a service design for provisioning in the deploy package wizard. These service designs can be grouped for different lifecycle stages so that the package deployment in the lifecycle stage lists only those grouped service designs from the respective lifecycle stage.

To group the service designs for a lifecycle stage, create a tag with the name of the lifecycle stage in each topology design. For example, you can create a Development tag and associate it with all required designs in the Development lifecycle stage.

> The Test Run wizard in the Test lab lists all the designs and does not group by tag.

Create a lifecycle stage

Apart from the out-of-the-box lifecycle stages available in Codar, you can also create your own lifecycle stage (referred to as a custom lifecycle stage in this document) and associate both custom and out-of-the box roles with the lifecycle stage.

To create a custom lifecycle stage

Use the Pipeline Configurations tile to create a custom lifecycle stage.

1. Navigate to **Release Automation** > **Pipeline Configurations** > **Lifecycle Stages**.
2. Select the ⚙ icon next to **Lifecycle Stages** and click **Create Lifecycle Stage**.
3. In the Create Lifecycle Stage dialog, fill in the following fields:

Field	Description
Display Name	Name of the lifecycle stage
Description	Description of the lifecycle stage
Roles	Roles that must be associated with the lifecycle stage. Custom roles if any are also displayed.
Image	Select an image that can serve as an icon for the lifecycle stage.
Color	Select a color for the lifecycle stage. The color is useful if you want the differentiate various lifecycle stages on the basis of color.

4. Click **Create**.

The lifecycle stage is listed in the Lifecycle Stages page along with the roles associated with it.

Edit or delete a lifecycle stage

You can edit or delete not only custom lifecycle stages but also the out-of-the-box lifecycle stages available in Codar.

To edit a custom lifecycle stage

1. Navigate to **Release Automation** > **Pipeline Configurations** > **Lifecycle Stages**.
2. Select the ⚙ icon next to the lifecycle stage you want to update and click **Edit**.
3. In the Edit Lifecycle Stage dialog, you can edit the name, description, icon, or color of the lifecycle stage. You can also add or remove roles associated with the lifecycle stage.
4. Click **Save**.

The edits are reflected in the Lifecycle Stages page.

To delete a custom lifecycle stage

1. Navigate to **Release Automation** > **Pipeline Configurations** > **Lifecycle Stages**.
2. Select the ⚙ icon next to the lifecycle stage you want to update and click **Delete**.
3. Click **Yes** in the Confirmation Required dialog to delete the lifecycle stage.

The lifecycle stage is deleted.

> If the lifecycle stage that you are deleting is associated with a design, then when deleting the lifecycle stage, an error message is displayed.

Set default lifecycle stages for pipeline management

You can select lifecycle stages (both out-of-the-box and custom) and group these stages together to create a default or defined lifecycle stage. Any new application design version that you create is then associated with these default or defined lifecycle stages.

Advantages of setting a default lifecycle stage

The main advantage of setting a default set of lifecycle stages to a new design version is that you do not need to manually define a lifecycle flow each time a new version of an application design is created.

To set default lifecycle stages

Use the Pipeline Configurations tile to set the default lifecycle stage.

1. Navigate to **Release Automation** > **Pipeline Configurations** > **Lifecycle Stages**.
2. Select the gear icon next to **Lifecycle Stages** and click **Set Default Lifecycle Stage**.
3. When you log on to Codar for the first time, all out-of-the-box lifecycle stages are displayed. As and when custom lifecycle stages are created, those stages are also displayed. You can remove any lifecycle stage that you do not want to include as part of the default by clicking the bin icon at the right of the each lifecycle stage.
4. Click **Add Lifecycle Stage** and select the lifecycle stages that you want to be a part of the default set.
5. Click **Save**.

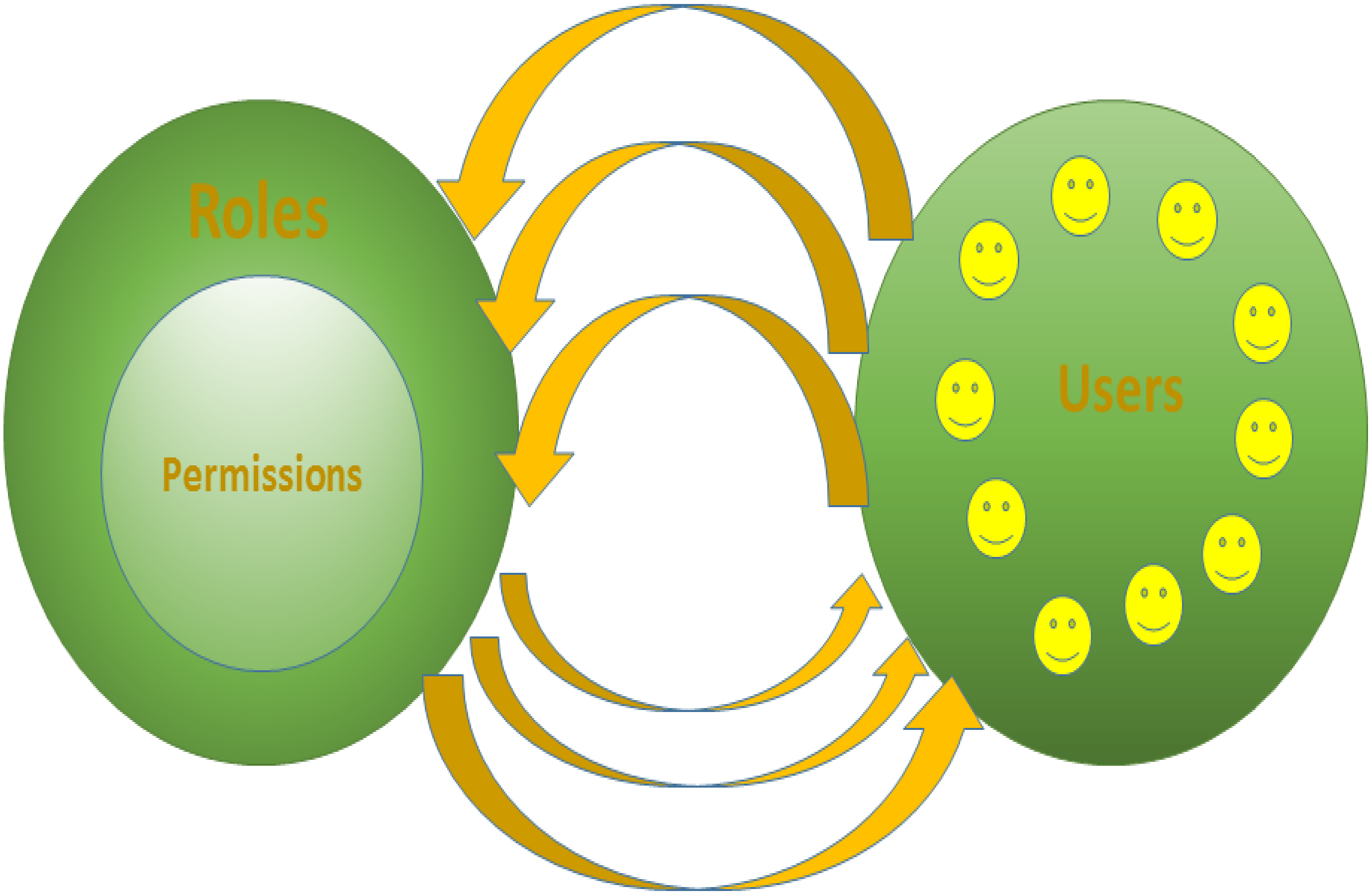
Roles

Every user in Codar is assigned one or multiple roles. Roles determine the permissions or activities that a Codar user can perform on the product. Codar contains some out-of-the-box roles; however, users can also create their own roles and then assign permissions to the roles they create. Only Codar users belonging to the following roles have the right to create roles:

- Administrator
- Application Architect
- Application Release Manager

Out-of-the-box roles can be edited but not deleted. This allows administrators or architects to add or remove permissions to these roles. Roles that are created can be edited or deleted. For more information about creating, editing, and deleting roles, see "Create, edit, and delete roles " on page 81. For more information about out-of-the-box roles, see "Out-of-the-box roles in Codar" on the next page.

All roles (out-of-the-box roles and roles created by Codar users) are displayed in the Roles page. Roles and users follow a many-to-many relationship. That is, multiple users can have multiple roles and multiple roles can be assigned to one or more users. The following diagram depicts the many-to-many relationship between users and roles.



Out-of-the-box roles in Codar

Out-of-the-box roles provide authorization for members to perform tasks. These roles can be edited but not deleted.

Users with the Administrator role have access to all areas.

Summary of access by role

The following table shows the predefined roles and the tasks to which each has access.

Permissions	Application Architect	Application Developer	Application QA	Release Manager	Application Operations Manager	Codar Integration User
Design create	Yes					
Package create	Yes	Yes				Yes
Package update	Yes	Yes	Yes	Yes	Yes	Yes

Package delete	Yes	Yes	Yes	Yes	Yes	Yes
Package deploy	Yes	Yes	Yes	Yes	Yes	Yes
Package promote		Yes	Yes	Yes	Yes	Yes
Package reject		Yes	Yes	Yes	Yes	Yes
Dashboard view	Yes			Yes		
Custom role and stage create	Yes			Yes		
Application release gate	Yes			Yes		
Application release gate approve	Yes			Yes		

Create, edit, and delete roles

To create a role in Codar:

1. In the Codar Console, click the **Release Automation** > **Pipeline Configurations** tile.
2. Click **Roles**.
3. Click the **Create Role** button in the Roles page. This button is enabled only for users belonging to the Administrators, Application Architect, Application Release Manager, and other roles that have the "Custom role and stage create" permission.
4. Enter information as per the following table:

Field	**Description**
Display Name	Name of the role
Description	Description of the role
Permission	Permissions that you want to assign to the role. You can select more than one permission.
Image	Browse and select an image that you want to associate with the role. The image is displayed next the role in the Roles page.

5. Click **Create**.

The role that you created is displayed in the Roles page and the permissions assigned to the role are displayed below the role. You can update or delete the role by clicking the wheel icon next to the role.

Edit a role

To edit an existing role in Codar:

1. Click the wheel icon next to the role you want to edit and click **Update Role**.
2. In the Update Role dialog box, edit the role name, description, or the permissions associated with the role. You can add or remove permissions. You can also change the image associated with the role.
3. Click **Save**.

The updated role with your changes is reflected in the Roles page.

> If you update a role by modifying its permissions, you must relogin to Codar to see the updated permissions.

Delete a role

To delete an existing role in Codar:

1. Click the wheel icon next to the role you want to edit and click **Delete Role**.
2. Click **Yes** in the Confirmation Required dialog box.

The role is removed from the Roles page.

> If you delete a role that is associated with a lifecycle stage, the role is removed from the lifecycle stage.

Approve or deny an approval request

When an approval request is created, an email is sent to the assigned approvers who can then approve or deny the promotion of a package to the next lifecycle stage.

To approve or deny an approval request

1. Navigate to **Release Automation** > **Approvals**.

   > The **Approvals** tile is displayed only to assigned approvers or users who have the "Application release gate approve" permission.

2. Click **Pending Requests** on the left panel. A list of all requests that require your approval is displayed.

   You can click on any request to drill-down and view the details of the request.

3. Select a request to approve or reject. You can also select multiple requests.

4. Click the **Approve** button to approve the request or the **Reject** button to reject it.

5. If you click the **Approve** button, entering a comment in the conformation box is optional but if you click the **Reject** button, you must mandatorily enter a comment.

6. Depending on whether you approve or reject a request, the request is moved to the **Approved Requests** area or the **Reject Requests** area.

Orchestration

This tile launches Operations Orchestration Central.

Migrate to Cloud

HPE Enterprise Maps (HPE EM) manages a centralized Business Model that links to Codar. To bring the highest cost savings, improved agility, and quality using Codar, the right applications and services need to be selected. The HPE Enterprise Maps Cloud Assessment process will identify the most suitable applications and services, and register them in Codar. Management can continuously evaluate the actual cloud migration progress to ensure that the IT infrastructure capabilities and cloud providers are optimally used to meet the cloud transformation goals.

Cloud migration process

HPE Enterprise Maps consolidates information about the existing application portfolio and sends out surveys to appropriate stakeholders using data from tools such as Universal CMDB, HPE PPM, HPE APM, or spreadsheets. Based on the collected information, HPE Enterprise Maps calculates scores showing suitability of the systems from business, technical and financial points of view. The results are visualized using a set of predefined reports. For selected services and applications, HPE Enterprise Maps creates initial service designs in Codar using information consolidated in the first phase.

Prerequisites

- HPE Enterprise Maps must be installed in the same domain as Codar.

- To ensure seamless navigation between the products, make sure that the HPE Single Sign-On (SSO) for HPE Enterprise Maps is configured to enable logging on to Codar.

- For SSO between Codar and HPE Enterprise Maps to work successfully, both products have to be installed on machines that are in the same domain. The value of domain and protected domain parameters specified for SSO configuration must be the same.

- See the *HPE Enterprise Maps Administrator Guide* for more information.

Enable Migrate to Cloud

Users with the Administrator and Application Architect roles have access to the migration feature.

To enable HPE Enterprise Maps tiles in the Management Console:

1. Make a backup of the %CSA_HOME%/jboss-as/standalone/deployments/csa.war/dashboard/config.json file (where %CSA_HOME% is the directory in which Codar is installed).
2. Edit the %CSA_HOME%/jboss-as/standalone/deployments/csa.war/dashboard/config.json file (where %CSA_HOME% is the directory in which Codar is installed).
3. Search for a tile called enterprise_maps. You can search for this text: "id": "enterprise_maps".
4. Under the tiles node, enable the first two tiles by changing "enabled": false to "enabled": true, and disable the third tile by changing "enabled": true to "enabled": false.
5. In the data section for each of the tiles, change <EM_HOST_NAME> to match the host name of your HPE Enterprise Maps installation.
6. Save and close the file.

Note: The changes do not require you to restart Codar.

After HPE Enterprise Maps is enabled (as described above), click the Migrate to Cloud tile to see the next level of tiles:

- **Application Catalog tile** – starts and manages data collection and surveys.
- **Reports tile** – displays the cloud migration dashboard.