Cisco Container Platform 9.0.0 User Guide

First Published: 2021-08-23

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

CHAPTER 1

Cisco Container Platform

Cisco Container Platform is a turnkey, production-grade, extensible platform that enables you to deploy and manage multiple Kubernetes clusters. It runs on 100% upstream Kubernetes. Seamless container networking, enterprise-grade persistent storage, built-in production-grade security, integrated logging, monitoring, and load balancing are the key benefits of Cisco Container Platform.

Cisco Container Platform provides authentication and authorization, security, high availability, networking, load balancing, and operational capabilities to effectively operate and manage Kubernetes clusters. Cisco Container Platform also provides a validated configuration of Kubernetes and can integrate with underlying infrastructure components such as Cisco UCS, Cisco HyperFlex, and Cisco ACI. You can use HyperFlex or VMware on UCS without HyperFlex, as the storage solution for Cisco Container Platform.

Using the Cisco Container Platform web interface, you can create Kubernetes clusters on which you can deploy containerized applications. The clusters are created on the infrastructure provider platform.

The two user personas in Cisco Container Platform are as follows:

- The **Administrator** persona, which is associated with the **Administrator** role.
- The **User** persona, which is associated with the **User** role.

This chapter contains the following topics:

Administrator Workflow

The following table lists the workflow for Cisco Container Platform administrators.

Task	Related Section
Access the Cisco Container Platform web interface using *Administrator* credentials.	Accessing Cisco Container Platform Web Interface, on page 3

Task	Related Section
Set up the Cisco Container Platform infrastructure configuration.	Setting Up an ACI Profile, on page 5
Configure Cisco Smart Software Licensing for your Cisco Container Platform instance.	Configuring Cisco Smart Software Licensing, on page 7
Manage the Cisco Container Platform infrastructure configurations using which clusters are created.	Managing Cisco Container Platform Infrastructure Configuration, on page 17
Create Kubernetes clusters.	Creating Clusters on vSphere, on page 31 Creating AWS EKS Clusters, on page 44 Creating OpenStack Clusters, on page 47 Creating AKS Clusters, on page 52
Add users, assign appropriate roles, and associate the new users to the Kubernetes clusters that you have created.	Managing Users and RBAC, on page 17
Monitor Kubernetes clusters.	Monitoring Health of Cluster Deployments, on page 65 Monitoring Logs from Cluster Deployments, on page 66
Manage Kubernetes clusters using the Kubernetes Dashboard.	Managing Kubernetes Clusters, on page 61
Manage the lifecycle of Kubernetes clusters by scaling or upgrading the clusters.	Scaling Clusters on vSphere, on page 36 Upgrading Clusters on vSphere, on page 36 Scaling AWS EKS Clusters, on page 46 Upgrading AKS Clusters, on page 54

User Workflow

The following table lists the workflow for developers assigned with the *User* role.

Task	Related Section
Access the Cisco Container Platform web interface using user credentials.	Accessing Cisco Container Platform Web Interface, on page 3
Monitor Kubernetes clusters that are assigned to the user.	Monitoring Health of Cluster Deployments, on page 65 Monitoring Logs from Cluster Deployments, on page 66

Task	Related Section
Manage the assigned Kubernetes clusters using the Kubernetes Dashboard or CLI.	Managing Kubernetes Clusters, on page 61
Deploy applications on the assigned Kubernetes clusters.	Deploying Applications on Kubernetes Clusters, on page 91

Accessing Cisco Container Platform Web Interface

Before you begin

Ensure that the prerequisite configurations for integrating ACI with Cisco Container Platform are complete.

For more information, see the following documents:

- *ACI Integration Requirements* section of the *Cisco Container Platform Installation Guide*
- Planning and Prerequisites section of the Cisco ACI and Kubernetes Integration page

Ensure that you have powered on the installer VM on vCenter. The URL of the installer appears on the vCenter **Web console**.

Step 1 Obtain the URL to access the Cisco Container Platform web interface from the vCenter **Web console**.

Step 2 Access the URL using your web browser.

```
https://<Cisco Container Platform IP Address>
```

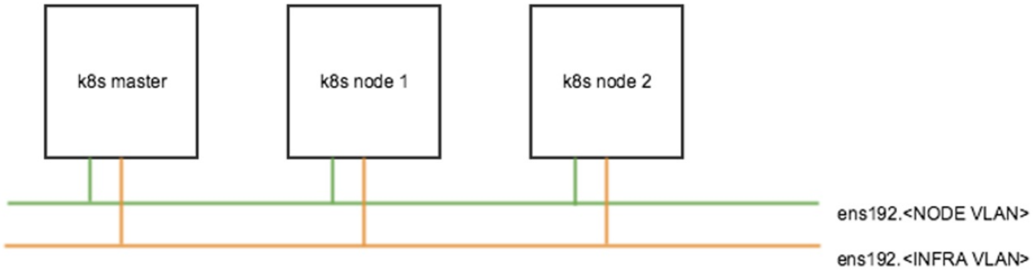
Note We recommend that you use the Chrome, Safari, or Firefox browser to access the URL.

Step 3 Log in to the web interface as an *admin* user using the passphrase given during the Cisco Container Platform installation.

Tenant Cluster with ACI Deployment

With an ACI deployment, each tenant cluster is required to have its own routable subnet. The node VLAN, pod subnet, and multicast subnet range should not overlap between clusters. Cisco Container Platform ensures that the VLAN and subnet do not overlap.

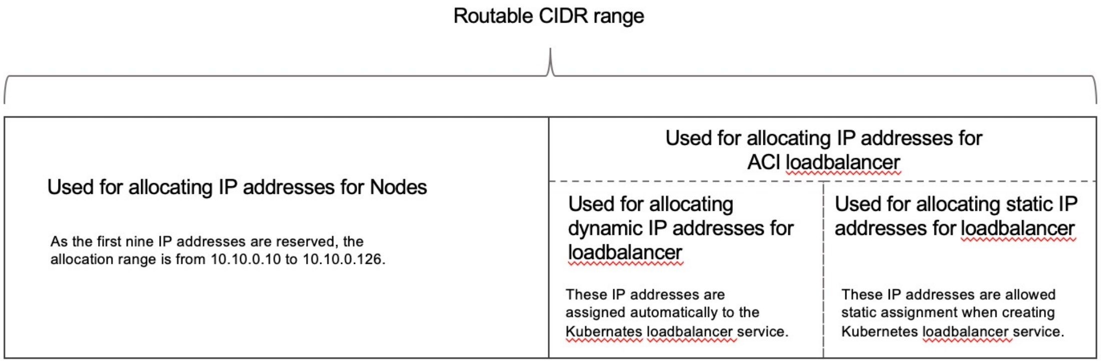
Unlike other CNI, an ACI tenant cluster requires two VLAN subinterfaces, one for the Node VLAN, and another for the Infra VLAN. As shown in the following figure, Cisco Container Platform assigns unique Node VLAN IDs. You need to assign a unique Infra VLAN ID for clusters during cluster creation.



When creating a tenant cluster with ACI, you need to select an ACI profile. Cisco Container Platform uses the ACI profile to allocate the VLAN, pod subnet, service subnet, and multicase subnet.

In addition to the ACI profile, you also need to set a Routable CIDR, which is a pre-configured routable CIDR at your router.

The following figure describes the IP address allocation from the routable CIDR range.


Routable CIDR range

Used for allocating IP addresses for Nodes

As the first nine IP addresses are reserved, the allocation range is from 10.10.0.10 to 10.10.0.126.

Used for allocating IP addresses for ACI loadbalancer

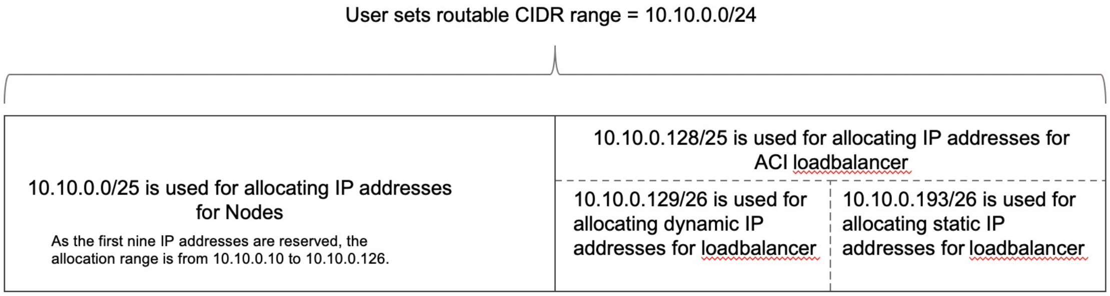
Used for allocating dynamic IP addresses for loadbalancer

These IP addresses are assigned automatically to the Kubernates loadbalancer service.

Used for allocating static IP addresses for loadbalancer

These IP addresses are allowed static assignment when creating Kubernetes loadbalancer service.

Example:


User sets routable CIDR range = 10.10.0.0/24

10.10.0.0/25 is used for allocating IP addresses for Nodes

As the first nine IP addresses are reserved, the allocation range is from 10.10.0.10 to 10.10.0.126.

10.10.0.128/25 is used for allocating IP addresses for ACI loadbalancer

10.10.0.129/26 is used for allocating dynamic IP addresses for loadbalancer

10.10.0.193/26 is used for allocating static IP addresses for loadbalancer

For more information on creating tenant clusters, see Creating Clusters on vSphere, on page 31.

For more information on the ACI and CNI plugin, see the latest documentation on Cisco ACI and Kubernetes Integration.

Setting Up an ACI Profile

Note This topic applies to an ACI environment. In a non-ACI environment, the IP address range of the default VIP pool must be expanded to include the additional VIPs for tenant clusters. For more information, see Managing Networks, on page 29.

When you log in to Cisco Container Platform for the first time, you need to configure the Cisco Container Platform initial setup using the **Cisco Container Platform Setup** wizard.

Step 1 On the **Welcome** page, click **START THE SETUP**.

Step 2 In the **ACI Credentials** screen, specify information such as IP address, username, and passphrase of the APIC instance, click **CONNECT**, and then click **NEXT**.

Step 3 In the **ACI Configuration** screen, perform these steps:

a) In the **NAMESERVERS** field, enter the IP address of all the DNS servers that the ACI fabric can access.
b) From the **VMM DOMAIN** drop-down list, choose the Virtual Machine Manager Domain (VMMD) that you want to use.
c) In the **INFRASTRUCTURE VLAN ID** field, enter the VLAN number for layer 2 networking.
d) From the **VRF** drop-down list, choose the Virtual Routing and Forwarding (VRF) IP address.
e) From the **L3OUT POLICY NAME** drop-down list, choose the ACI object for allowing external internet connectivity.
f) From the **L3OUT NETWORK NAME** drop-down list, choose the external network that is reachable through the L3OUT object.
g) From the **AAEP NAME** drop-down list, choose an Attachable Access Entity Profile (AAEP) name to associate the VMM domain with an AAEP.
h) In the **STARTING SUBNET FOR PODS** field, enter the starting IP address of the pod subnet.

   The IP addresses for the pods are allocated from the pod subnet.
i) In the **STARTING SUBNET FOR SERVICE** field, enter the starting IP address of the service subnet.

   The IP addresses for services are allocated from the service subnet.
j) In the **CONTROL PLANE CONTRACT NAME** field, enter the name of the contract that is provided by the endpoint group of the control plane.

   This setting allows traffic from the control plane cluster to the tenant cluster.
k) In the **NODE VLAN START ID** field, enter the starting VLAN ID of the node network.

   The IP addresses for the VLAN are allocated from the node network.
l) In the **NODE VLAN END ID** field, enter the ending VLAN ID of the node network.

   Note Ensure that you configure two VLANs for each cluster.
m) In the **OPFLEX MULTICAST RANGE** field, enter a range for the Opflex multicast.
n) Click **CONNECT**.

Step 4 In the **Summary** screen, verify the configuration, and then click **FINISH**.
For more information on adding, modifying, or deleting an ACI profile, see Managing ACI Profile, on page 27.

For more information on integrating Cisco Container Platform with ACI, see Cisco ACI and Kubernetes Integration.

Viewing Version and License Information

To view the version and license information of the product:

Step 1 Log in to the Cisco Container Platform web interface.

Step 2 Click the **Welcome** drop-down list in the top-right corner of the title bar.
The version and license information are displayed.

Changing User Interface Language

The user interface language is the language in which the page titles, menus, dialog boxes, and error messages are displayed on the web interface.

To change the user interface language:

Step 1 Log in to the Cisco Container Platform web interface.

Step 2 Click the **Welcome** drop-down list in the top-right corner of the title bar.

Step 3 In the **LANGUAGE** area, choose **English** or **Japanese** as the user interface language.
The user interface language on the web interface is changed.

CHAPTER 2

Configuring Cisco Smart Software Licensing

You need to configure Cisco Smart Software Licensing on Cisco Smart Software Manager (Cisco SSM) to easily procure, deploy, and manage licenses for your Cisco Container Platform instance. The number of licenses required depends on the number of VMs necessary for your deployment scenario.

Cisco SSM enables you to manage your Cisco Smart Software Licenses from one centralized website. With Cisco SSM, you can organize and view your licenses in groups called virtual accounts. You can also use Cisco SSM to transfer the licenses between virtual accounts as needed.

You can access Cisco SSM from the Cisco Software Central homepage, under the Smart Licensing area.

Cisco Container Platform is initially available for a 90-day evaluation period after which, you need to register the product.

Based on the level of security required for your environment, you can use one of the following licensing models:

- Connected Model, on page 7
- Disconnected Model, on page 8

You need to choose the licensing model when you register Cisco Container Platform.

This chapter contains the following topics:

Connected Model

In a connected deployment model, the license usage information is directly sent over the Internet or through an HTTP proxy server to Cisco SSM.

For a higher degree of security, you can opt to use a partially connected deployment model, where the license usage information is sent from Cisco Container Platform to a locally installed VM-based satellite server (Cisco SSM satellite). Cisco SSM satellite synchronizes with Cisco SSM on a daily basis.

Note Ensure that you use Cisco SSM Satellite version 5.0 or later. For more information, see Installing and Configuring Cisco SSM Satellite.

For more information, see Workflow in a Connected Model, on page 8.

Disconnected Model

For the highest degree of security, you can opt to use Specific License Reservations in a fully disconnected model. In this model, you do not need to have access to the Cisco SSM or Cisco SSM satellite. All license changes are processed manually.

For more information, see Workflow in a Disconnected Model, on page 12.

License Usage and Compliance

Once you register Cisco Container Platform with Cisco SSM, you will receive the **Cisco Container Platform License with Support** license.

Cisco SSM totals the license requirements for all your Cisco Container Platform instances and compares the total license usage to the number of licenses purchased. After data synchronization, your Cisco Container Platform instance displays one of the following status indicators:

- **Authorized**, when the number of licenses purchased is sufficient
- **Out of Compliance**, when the number of licenses is insufficient

  For more information, see Updating Reserved Licenses, on page 13.
- **Authorization Expired**, when the product has not communicated with Cisco SSM for a period of 90 days.

Workflow in a Connected Model

The following table describes the workflow of Cisco Smart Software Licensing in a connected model.

Task	Related Section
Configure the transport settings using which Cisco Container Platform connects to Cisco SSM or Cisco SSM satellite	Configuring Transport Settings, on page 9
Register the Cisco Container Platform instance	Registering Using Registration Token, on page 9
Manage licenses	Renewing Authorization, on page 10 Reregistering Cisco Container Platform License, on page 11 Deregistering Registration, on page 11

For more information, see Connected Model, on page 7.

Configuring Transport Settings

By default, Cisco Container Platform directly communicates with the Cisco SSM. You can modify the mode of communication by configuring the transport settings.

Before you begin

Ensure that you have obtained the registration token for the Cisco Container Platform instance.

Step 1 Log in to the Cisco Container Platform web interface.

Step 2 In the left pane, click **Licensing**.

If you are running Cisco Container Platform in the Evaluation mode, a license notification is displayed on the **Smart Software Licensing** pane.

Step 3 If a license notification is displayed, click the **edit the Smart Call Home Transport Settings** link.

Alternatively, click the **Licensing Status** tab, and then click the **View/Edit** link that appears under **Transport Settings**.

Step 4 Perform one of the following steps in the **Transport Settings** dialog box:

a) If you want to configure Cisco Container Platform to send the license usage information to Cisco SSM using the Internet:

1. From the **TRANSPORT MODE** drop-down list, choose **DIRECT**.
2. From the **GATEWAY URL** drop-down list, choose the default URL or enter a custom URL.
3. To configure a proxy server such as Cisco Transport Gateway or Apache:

b) If you want to configure Cisco Container Platform to send the license usage information to Cisco SSM using the Cisco SSM satellite:

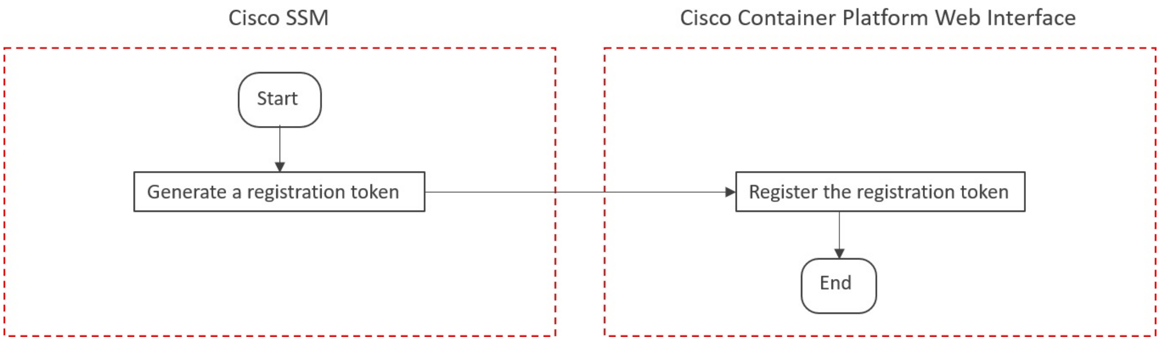
1. From the **TRANSPORT MODE** drop-down list, choose **Gateway**.
2. In the **GATEWAY URL** field, enter a custom URL.
3. Click **SAVE**.

Registering Using Registration Token

You need to register your Cisco Container Platform instance with Cisco SSM or Cisco SSM satellite before the 90-day evaluation period expires.

The following figure shows the workflow for registering Cisco Container Platform using a registration token.

Figure 1: Workflow of Registering Using Registration Token


Cisco SSM
Cisco Container Platform Web Interface
Start
Generate a registration token
Register the registration token
End

Before you begin

Ensure that you have configured the transport settings. For more information, see Configuring Transport Settings.

Step 1 Perform these steps on Cisco SSM or Cisco SSM satellite to generate a registration token:

a) Go to **Inventory** > **<Choose your virtual account>** > **General**, and then click **New Token**.

b) If you want to enable higher levels of encryption for the products registered using the registration token, check the **Allow export-controlled functionality on the products registered with this token** check box.

Note This option is available only if you are compliant with the Export-Controlled functionality.

c) Download or copy the token.

Step 2 Perform these steps in the Cisco Container Platform web interface to register the registration token and complete the license registration process:

a) In the left pane, click **Licensing**.

b) In the license notification, click **Register**.
The **Smart Software Licensing Product Registration** dialog box appears.

c) In the **Product Instance Registration Token** field, enter, copy and paste, or upload the registration token that you generated in Step 1.

d) Click **REGISTER** to complete the registration process.

If registering the token fails, you can reregister the Cisco Container Platform instance using a new token. For more information, see Reregistering Cisco Container Platform License.

Renewing Authorization

By default, the authorization is automatically renewed every 30 days. However, Cisco Container Platform allows a user to manually initiate the authorization renew in case the automatic renewal process fails. The authorization expires if Cisco Container Platform is not connected to Cisco SSM or Cisco SSM satellite for 90 days and the licenses consumed by Cisco Container Platform are reclaimed and put back to the license pool.

Before you begin

Ensure that the Cisco Container Platform instance is registered with Cisco SSM or Cisco SSM satellite.

Step 1 Log in to the Cisco Container Platform web interface.

Step 2 In the left pane, click **Licensing**.

Step 3 From the **ACTIONS** drop-down list, choose **Renew Authorization Now**.

Step 4 Click **OK** in the **Renew Authorization** dialog box.
Cisco Container Platform synchronizes with Cisco SSM or Cisco SSM satellite to check the license authorization status and Cisco SSM or Cisco SSM satellite reports back the status to Cisco Container Platform, on a daily basis.

Reregistering Cisco Container Platform License

You can reregister Cisco Container Platform with Cisco SSM or Cisco SSM satellite by deregistering it and registering it again, or by using a register force option.

Before you begin

Ensure that you have obtained a new registration token from Cisco SSM or Cisco SSM satellite.

Step 1 Log in to the Cisco Container Platform web interface.

Step 2 In the left pane, click **Licensing**.

Step 3 From the **ACTIONS** drop-down list, choose **Reregister**.

Step 4 In the **Product Instance Registration Token** field of the **Smart Software Licensing Product Reregistration** dialog box, enter the registration token that you generated using Cisco SSM or Cisco SSM satellite.

Step 5 Click **REGISTER** to complete the registration process.
Cisco Container Platform sends a request to Cisco SSM or Cisco SSM satellite to check the registration status and Cisco SSM or Cisco SSM satellite reports back the status to Cisco Container Platform, on a daily basis.

Deregistering Registration

You can deregister the Cisco Container Platform instance from Cisco SSM or Cisco SSM satellite to release all the licenses from the current Virtual account and the licenses are available for use by other products in the virtual account. Deregistering disconnects Cisco Container Platform from Cisco SSM or Cisco SSM satellite.

Before you begin

Ensure that the Cisco Container Platform instance is registered with Cisco SSM or Cisco SSM satellite.

Step 1 Log in to the Cisco Container Platform web interface.

Step 2 In the left pane, click **Licensing**.

Step 3 From the **ACTIONS** drop-down list, choose **Deregister**.

Step 4 Click **DEREGISTER** in the confirmation dialog box.

Cisco Container Platform sends a request to Cisco SSM or Cisco SSM satellite to check the deregistration status and Cisco SSM or Cisco SSM satellite reports back the status to Cisco Container Platform, on a daily basis.

Workflow in a Disconnected Model

The following table describes the workflow of Cisco Smart Software Licensing in a disconnected model.

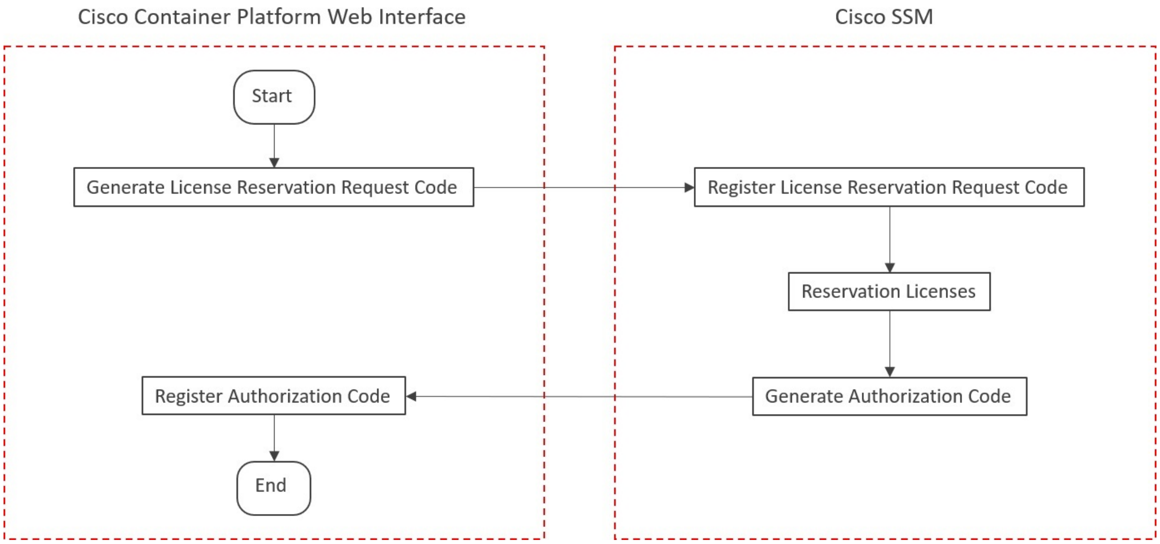
Task	Related Section
Register the Cisco Container Platform instance	Registering Using Specific License Reservations, on page 12
Manage licenses	Updating Reserved Licenses, on page 13 Deregistering Registration by Returning Reserved Licenses, on page 14

For more information, see Disconnected Model, on page 8.

Registering Using Specific License Reservations

The following figure shows the workflow for registering Cisco Container Platform using Specific License Reservations (SLRs).

Figure 2: Workflow of Registering Using SLRs


Cisco Container Platform Web Interface
Cisco SSM
Start
Generate License Reservation Request Code
Register License Reservation Request Code
Reservation Licenses
Register Authorization Code
Generate Authorization Code
End

Before you begin

Ensure that your smart account is authorized for License Reservation.

Step 1 Perform these steps in the Cisco Container Platform web interface:

a) In the left pane, click **Licensing**.
b) In the license notification, click **Register**.
The **Smart Software Licensing Product Registration** dialog box appears.
c) Click the **start here** link to initiate **Specific License Reservation**.
d) In the **Ensure Smart Account Can Use License Reservation** dialog box, check the **YES, MY SMART ACCOUNT IS LICENSE RESERVATION ENABLED** check box, and then click **CONTINUE**.
e) In the **Smart License Reservation** dialog box, click **GENERATE REQUEST CODE**.
f) Copy and save the **License Reservation Request Code**.

Step 2 Perform these steps on Cisco SSM:

a) Go to **Inventory** > **<Choose your virtual account>** > **Licenses**, and then click **License Reservation**.

Note If you cannot find the **License Reservation** button, contact Cisco support to enable this feature.

The **Smart License Reservation** dialog box appears.
b) In the **Reservation Request Code** field, enter, paste, or upload the **License Reservation Request Code** from Step 1, and then click **Next**.
c) In the **Licenses to Reserve** area, check the **Reserve a specific license** check box.
d) Under the **Quantity To Reserve** column, type the number of licenses to reserve for your product instance, and then click **Next**.
e) Click **Generate Authorization Code**.
f) Copy or download the authorization code.

Step 3 Perform these steps in the Cisco Container Platform web interface:

a) In the left pane, click **Licensing**.
b) In the license notification, click **ENTER RESERVATION AUTHORIZATION CODE**.
The **Enter Reservation Authorization Code** dialog box appears.
c) In the **RESERVATION AUTHORIZATION CODE** field, enter, paste, or upload the authorization code from Step 2.
d) Click **RESERVE**.
The number of licenses reserved is displayed in **Licensing** > **Smart Licensing Usage**. When your license usage is out of compliance, you can Updating Reserved Licenses.

For more information, see License Usage and Compliance, on page 8.

Updating Reserved Licenses

If your license usage is out of compliance, or you want to increase the number of licenses for your Cisco Container Platform instance, you can update the number of reserved licenses.

Step 1 Perform these steps in Cisco SSM:

a) Go to **Inventory** > **<Choose your virtual account>** > **Product Instances**.
b) From the **ACTIONS** drop-down list of your Cisco Container Platform instance, choose **Update reserved licenses**.
The **Update License Reservation** dialog box appears.
c) In the **Licenses to Reserve** area, check the **Reserve a specific license** check box.

d) Under the **Quantity To Reserve** column, modify the number of reserved licenses for your product instance, and then click **Next**.
e) Click **Generate Authorization Code**.
f) Copy or download the authorization code.

Step 2 Perform these steps in the Cisco Container Platform web interface:

a) In the left pane, click **Licensing** > **Smart Licensing Usage**.
b) From **ACTIONS** drop-down list, click **Update Reservation**.
The **Enter Reservation Authorization Code** dialog box appears.
c) In the **RESERVATION AUTHORIZATION CODE** field, enter, paste, or upload the reservation code from Step 1.
d) Click **RESERVE**.
A license reservation confirmation code is generated.

Step 3 Perform these steps in Cisco SSM:

a) In the **Update License Reservation** dialog box, click **Enter Confirmation Code**.
b) In the **Reservation Confirmation Code** field, enter, paste, or upload the confirmation code that you received in Step 2.
c) Click **OK**.
The number of reserved licenses for your product instance is updated on Cisco SSM.

For more information, see License Usage and Compliance, on page 8.

Deregistering Registration by Returning Reserved Licenses

You can deregister your Cisco Container Platform instance from Cisco SSM by returning the reserved licenses. Consequently, these licenses are available for other product instances.

Step 1 Perform these steps in the Cisco Container Platform web interface:

a) In the left pane, click **Licensing**.
The **Smart Software Licensing** page appears.
b) From **ACTIONS** drop-down list, click **Return reserved licenses**.
The **Confirm Return Licenses** dialog box appears.
c) Click **GENERATE RESERVATION RETURN CODE**.
If you cannot generate a **Reservation Return Code**, contact Cisco support.

Caution The licenses remain reserved in Cisco SSM and cannot be used for other product instances until they are released on Cisco SSM.

Step 2 Follow these steps in the Cisco SSM to release the licenses:

a) Go to **Inventory** > **<Choose your virtual account>** > **Product Instances**.
b) From the **ACTIONS** drop-down list of your Cisco Container Platform instance, choose **Remove**.
The **Remove Product Instance** dialog box appears.
c) In the **Reservation Return Code** field, enter, paste, or upload the reservation return code from Step 1.
d) Click **Remove Product Instance**.

The reserved licenses are released, and you can use them for other product instances. Your Cisco Container Platform instance goes back to the evaluation mode.

CHAPTER 3

Managing Cisco Container Platform Infrastructure Configuration

This chapter contains the following topics:

Managing Users and RBAC

Cisco Container Platform provides Role-based Access Control (RBAC) through built-in static roles, namely the *Administrator* and *User* roles. Role-based access allows you to use local accounts and LDAP for authentication and authorization.

Configuring Local Users

Cisco Container Platform allows you to manage local users. An administrator can add a user, and assign an appropriate role and cluster(s) to the user.

Caution Use of local authentication is not recommended and is considered less secure for production data.

Before you begin

Ensure that you have configured LDAP Server for authentication of Cisco Container Platform users.

For more information, see Configuring AD Servers, on page 21.

Step 1 In the left pane, click **User Management**, and then click the **Users** tab.
The **Add User** screen appears.

Step 2 Click **ADD USER**.

Step 3 In the **USERNAME** field, enter a username.

Step 4 From the **ROLE** drop-down list, choose one of the following roles:

- Administrator
- User

Step 5 If you want to generate a passphrase automatically:

a) Click the **AUTOMATICALLY GENERATE PASSPHRASE** toggle button.
The **User Details** screen appears.

b) Click **COPY PASSPHRASE** to copy the passphrase to your clipboard.

c) Click **CLOSE**.

Step 6 If you want to type a passphrase of your own, enter a passphrase in the **PASSPHRASE** field.

a) In the **FIRST NAME** field, enter the first name of your user.

b) In the **LAST NAME** field, enter the last name of your user.

c) Click **ADD**.

The new user is displayed on the **User Management** page.

Note You can edit or delete a user by using the options available under the **ACTIONS** column.

Modifying Local Authentication Policy

Caution There will be a temporary downtime for the Cisco Container Platform API during this procedure.

Follow these steps to modify the local authentication policies for the local accounts.

Step 1 SSH to a control plane master node.

Step 2 Edit the API auth configmap `kaas-api-auth`.

```
kubectl edit cm kaas-api-auth
```

Step 3 Modify the local authentication parameters under `data.authentication_settings.py`.

For example:

```
apiVersion: v1
    data:
    authentication_settings.py: |-
        PASSWORD_MIN_STRENGTH=0.40
        PASSWORD_LIFETIME_DAYS=365
        PASSWORD_WARNING_DAYS=20
        PASSWORD_GRACE_DAYS=1
        VALIDATOR_STRENGTH_ENABLE=False
    kind: ConfigMap
    metadata:
    name: api-auth
    namespace: default
```

For more information on the local authentication parameters, see Local Authentication Parameters, on page 19.

Step 4 Delete the pod to restart the API service.

```
kubectl delete pod -l app=api
```

Local Authentication Parameters

The following table describes the parameters used for local authentication.

Parameter	Default Setting	Description
VALIDATOR_MIN_LEN	8	Minimum character length of passphrase.
PASSWORD_LIFETIME_DAYS	0 (Forever)	Number of days for which a passphrase is valid before requiring a change
PASSWORD_WARNING_DAYS	14	Number of days for which a warning is sent to the user to warn expiry of passphrase
PASSWORD_GRACE_DAYS	0	Number of days after a passphrase has expired during which you are allowed to continue to login
PASSWORD_HISTORY_COUNT	0	Passphrase reuse limitation value
PASSWORD_HISTORY_DAYS	0	Number of days after which passphrase reuse is allowed
VALIDATOR_FORBIDDEN_WORDS	{"cisco123", "ccp123"}	Explicit list of restricted passphrases
VALIDATOR_COMMON_ENABLE	True	Restrict passphrases based on a common dictionary
VALIDATOR_STRENGTH_ENABLE	True	Enable passphrase complexity requirement
PASSWORD_MIN_STRENGTH	0.20	Passphrase complexity requirement (range 0.00..0.99)
LOGIN_THROTTLE_ENABLED	True	Enable rate-limiting on login endpoint
THROTTLE_ANON_LOGIN_BURST	'60/min'	Rate-limiting burst limit for unsuccessful login attempts
THROTTLE_ANON_LOGIN_SUSTAINED	False	Rate-limiting sustained limit for unsuccessful login attempts, for example: '100/day'

Note Rate limit applies to each worker process of an API service. An API service is backed by 10 worker processes, which serve requests in a round-robin fashion. The overall number of requests before throttle occurs is calculated using the formula: Rate value x 10

For example: The default allowed overall requests for login attempts is calculated as follows: (THROTTLE_ANON_LOGIN_BURST) * 10

= (60) x 10

= 600 requests/mins

Changing Login Passphrase

Step 1 In the left pane, click **User Management**, and then click the **Users** tab.

Step 2 From the drop-down list displayed under the **ACTIONS** column, choose **Change passphrase** corresponding to your name.

Note Administrators can change passphrase and role for other users as well.

Step 3 If you want to generate a passphrase automatically:

a) Click the **AUTOMATICALLY GENERATE PASSPHRASE** toggle button.

b) Click **CHANGE**.
The **User Details** screen appears.

c) Click **COPY PASSPHRASE** to copy the passphrase to your clipboard.

d) Click **CLOSE**.

Step 4 If you want to use a passphrase of your own:

a) Enter a passphrase in the **PASSPHRASE** field.

b) Click **CHANGE**.

The passphrase is changed successfully.

Recovering Login Passphrase for Local Admin

Step 1 Perform one of the following steps:

a) If you have SSH access to the Cisco Container Platform Control Plane nodes, log in to a Cisco Container Platform Control Plane node.

b) If you have the Kubeconfig file, save it in the `$HOME/.kube` directory. You can specify other kubeconfig files by setting the KUBECONFIG environment variable or by setting the `--kubeconfig` flag.

Step 2 List the available pods.

```
kubectl get pods
```

Step 3 Search for the pod that has the following format:

```
kaas-corc-xxxxxxxx-xxxx
```

Step 4 Reset the login passphrase for the admin user.

```
kubectl exec kaas-corc-7df5d76f87-55n7b ./password_reset
Password reset for 'admin' user : <50-char-long-random-string>
```

The local admin passphrase is reset to a 50-character random string. You can choose to continue using this passphrase, or reset the passphrase by Changing Login Passphrase.

Restoring Login Access Using a Breakglass Account

If you are an admin user and you are unable to log in to Cisco Container Platform, you can use a breakglass account to restore your login access.

Follow these steps to restore your login access:

Step 1 SSH in to the Cisco Container Platform control plane master node.

Step 2 Create a breakglass account.

```
 $ python /opt/ccp/bin/user.py --enable-user breakglass --password <password> --controlplane
https://<CCP_Control_Plane_IP:Port>
```

An emergency breakglass account is created and the ccp-api pod is restarted.

Step 3 Log in to the Cisco Container Platform web interface using the newly created breakglass account credentials and make the required LDAP configuration changes to restore your login access.

Step 4 Disable the breakglass user account.

```
 $ python /opt/ccp/bin/user.py --disable-user breakglass -password <password> --controlplane
https://<CCP_Control_Plane_IP:Port>
```

Configuring AD Servers

LDAP authentication is performed using a service account that can access the LDAP database and query for user accounts. You will need to configure the AD server and service account in Cisco Container Platform.

Step 1 In the left pane, click **User Management**, click the **Active Directory** tab, and then click **EDIT**.

Step 2 In the **SERVER IP ADDRESS** field, type the IP address of the AD server.

Step 3 In the **PORT** field, type the port number for the AD server.

Step 4 For improved security, we recommend that you check **STARTTLS**.

Step 5 In the **BASE DN** field, type the domain name of the AD server for all the accounts that you have.

For example:

CN=Users,DC=example,DC=com

Note You can use a comma-separated list to enter multiple domain names.

Step 6 2. In the **ACCOUNT USERNAME** field, enter an LDAP CN.

For example:

CN=UserName,OU=Folder,DC=example,DC=cisco,DC=com

Step 7 In the **PASSPHRASE** field, type the passphrase of the AD account.

Step 8 Click **SUBMIT**.

Troubleshooting AD User Credentials

Step 1 Run the `ldapwhoami` command-line tool to validate the AD service account credentials.

Command:

```
ldapwhoami -x -W -D <ACCOUNT USERNAME> -H ldap://<SERVER IP ADDRESS>/
```

Example:

```
ldapwhoami -x -W -D cn=admin,dc=example,dc=org -H ldap://10.10.10.100/
```

Step 2 When prompted, type the passphrase of the AD account.

If the user credential validation fails, an `Invalid Credentials` message is displayed.

Configuring AD Groups

Cisco Container Platform allows you to manage users using AD groups. An administrator can add users to AD groups, and then assign appropriate roles and clusters to the groups.

Before you begin

Ensure that you have configured the AD server that you want to use.

For more information on configuring AD servers, see Configuring AD Servers, on page 21.

Step 1 In the left pane, click **User Management**, and then click the **Groups** tab.

Step 2 Click **ADD GROUP**.

Step 3 In the **ACTIVE DIRECTORY GROUP** field, type the list of distinguished names for all the accounts that you have.

For example, type **`CN=CCP-Cluster1-Admin,CN=Users,DC=aervacan-lab,DC=local`**, where the distinguished names are entered using a comma-separated list.

Step 4 Specify information such as the name of the AD group and the role you want to assign to the group.

Note If the AD group is associated with the *Administrator* role, by default, access is provided to all clusters. But, if the AD group is associated with the *User* role, you need to assign a cluster.

Step 5 From the **CLUSTERS** drop-down list, choose the names of the cluster that you want to assign to the AD group.

Step 6 Click **SUBMIT**.

Troubleshooting AD Groups

Consider an AD group with the following parameters:

- **SERVER IP ADDRESS:** 10.10.10.100
- **PORT:** 389
- **BASE DN:** dc=example,dc=org
- **ACCOUNT USERNAME:** cn=admin,dc=example,dc=org

Step 1 Run the `ldapsearch` command-line tool to view users who belong to the `cn=Admin Users,dc=example,dc=org` group.

```
ldapsearch -x -D "cn=admin,dc=example,dc=org" -W -b "dc=example,dc=org" -h 10.10.10.100
\'(&(memberOf=cn=Admin Users,dc=example,dc=org))' 'sAMAccountName'
```

Step 2 Run the `ldapsearch` command-line tool to view the list of groups to which the `cn=user,dc=example,dc=org` user belongs.

```
ldapsearch -x -D "cn=admin,dc=example,dc=org" -W -b "cn=user,dc=example,dc=org" -h 10.10.10.100
'memberOf'
```

The group information that is configured in the AD service is displayed.

Managing Provider Profile

Cisco Container Platform enables you to define the provider profile on which clusters can be created.

You can configure multiple provider profiles in an instance of Cisco Container Platform and use the same provider profile for multiple clusters.

Adding Provider Profile

After your Cisco Container Platform control plane is available, log in to the Cisco Container Platform web interface, and then add the required provider profiles.

This section contains the following topics:

Adding vSphere Provider Profile

Before you begin

Cisco Container Platform interacts with vSphere through the user that you configure when you add a provider profile. You need to ensure that this user has the necessary privileges. For more information, see Minimum User Privileges on vSphere, on page 99.

Step 1 In the left pane, click **Infrastructure Providers**.
The **Infrastructure Providers** screen appears.

Step 2 Click **NEW PROVIDER** and enter information such as name, description, address, port, username and passphrase of the provider profile.

Step 3 Click **ADD**.

The vSphere provider profile that you added is displayed on the **Infrastructure Providers** > **vSphere** screen.

Adding Amazon Provider Profile

Cisco Container Platform interacts with Amazon through the user that you configure when you add a provider profile. You need to ensure that this user has the necessary privileges. For more information on minimum user privileges on AWS, see Minimum User Privileges on AWS, on page 108.

Before you begin

Ensure that you have completed the prerequisites for configuring clusters on AWS EKS. For more information, see Prerequisites for Configuring Clusters on AWS EKS, on page 39.

Step 1 In the left pane, click **Infrastructure Provider**.

The **Infrastructure Providers** screen appears.

Step 2 Click **NEW PROVIDER** and specify the following information:

a) In the **PROVIDER NAME** field, enter a name for the related Amazon account.
b) In the **ACCESS KEY ID** field, enter the access key ID for the related Amazon account.

   For more information on creating an access key ID, see Creating Access Keys, on page 42.
c) In the **SECRET ACCESS KEY** field, enter the access key for the related Amazon account.

   For more information on creating a secret access key, see Creating Access Keys, on page 42.
d) Click **ADD**.

Note The access key and secret must not be from your AWS root user account.

The Amazon provider profile that you added is displayed on the **Infrastructure Providers** > **AWS** screen.

For more information, see Administering Clusters on Amazon Web Services (AWS) EKS, on page 39.

Adding OpenStack Provider Profile

Step 1 In the left pane, click **Infrastructure Provider**.

The **Infrastructure Providers** screen appears.

Step 2 Click the **NEW PROVIDER** and enter the following information:

a) In the **PROVIDER NAME** field, enter a name for the related OpenStack account.
b) From the **PROTOCOL** drop-down list, choose the protocol that you want to use.
c) In the **AUTH URL** field, enter the URL that is used to authenticate against an Identity Server.
d) In the **REGION** field, enter an appropriate OpenStack region.
e) In the **DOMAIN NAME** field, enter the domain name account that is used for accessing the OpenStack server.
f) In the **PROJECT NAME** field, enter project name that you want to use.
g) In the **USERNAME** field, enter the username for the OpenStack provider profile.
h) In the **PASSPHRASE** field, enter a passphrase for the OpenStack provider profile.

i) In the **CA CERTIFICATE** field, add a root CA certificate to allow tenant clusters to securely connect to additional services.

j) Click **ADD**.

The OpenStack provider profile that you added is displayed on the **Infrastructure Providers** > **Openstack** screen. For more information on administering clusters on OpenStack, see Administering Clusters on OpenStack, on page 47.

Adding Azure Provider Profile

Cisco Container Platform interacts with Azure through the user that you configure when you add a provider profile. You need to ensure that this user has the necessary privileges. For more information on minimum user privileges on Azure see, Minimum User Privileges on AKS, on page 110.

Before you begin

Ensure that you have created a service principal in your Azure account and noted down the values of the `id`, `appID`, `password`, and `tenant` parameters. For more information, see Creating Service Principals, on page 25.

Step 1 In the left pane, click **Infrastructure Provider**.
The **Infrastructure Providers** screen appears.

Step 2 Click the **NEW PROVIDER** and specify the following information:

a) In the **NAME** field, enter a name for your Azure account.

b) If you want to use Virtual Kubelet to provision pods on Azure Container Instance in your clusters on AKS, in the **APPLICATION NAME** field, enter the name of the service principal that you have created for your Azure cluster.

c) In the **CLIENT ID** field, enter the value of the `appID` parameter from Creating Service Principals, on page 25.

d) In the **CLIENT SECRET** field, enter the value of the `password` parameter from Creating Service Principals, on page 25.

e) In the **TENANT ID** field, enter the value of the `tenant` parameter from Creating Service Principals, on page 25.

f) In the **SUBSCRIPTION ID** field, enter the value of the `id` parameter from Creating Service Principals, on page 25.

g) Click **ADD**.

The Azure provider profile is displayed on the **Infrastructure Providers** > **Azure** screen.

For more information on administering Azure Kubernetes Service (AKS) clusters, see Administering Clusters on Azure Kubernetes Service (AKS), on page 51.

Creating Service Principals

Step 1 Login to the Azure Portal.

Step 2 Install the Azure CLI.

Step 3 Follow these steps to configure the Azure CLI to use the Azure account that you want to use with Cisco Container Platform:

a) Log in to the Azure CLI.

```
az login
```

The URL to the device login page and an authentication code is displayed.

b) Use a browser to access the device login page, enter the code that you have received, and then click **Continue**.

c) Choose your Azure account.

Step 4 From the command output on the Azure CLI, note down the value of the `id` parameter. This value is required while Adding Azure Provider Profile to Cisco Container Platform.

For example:

```
{
    "cloudName": "AzureCloud",
    "id": "aaaaaaaa-bbbb-1111-cc22-ddddd3333444ddd",
    "isDefault": true,
    "name": "Microsoft Azure Enterprise",
    "state": "Enabled",
    "tenantId": "xxxxx-yyyy-999z-9090-uuuuu999uuuu",
    "user": {
    "name": "user@org.com",
    "type": "user"
    }
```

Step 5 Create a service principal using the Azure CLI.

```
az ad sp create-for-rbac -n myserviceprincipal
```

Where, `myServicePrincipal` is the name of the service principal. You may give any name for your service principal.

Step 6 From the command output on the Azure CLI, note down the values of the `appID`, `password`, and `tenant` parameters. These values are required while Adding Azure Provider Profile to Cisco Container Platform.

For example:

```
{
    "appId": "qqqqqqqq-a1a1-2b2b-9z9z-wwww1111vvvv",
    "displayName": "myserviceprincipal",
    "name": "http://myserviceprincipal",
    "password": "mmmmmm-n0n0-p1p1-q3q3-uuuu0000vvvv",
    "tenant": "xxxxx-yyyy-999z-9090-uuuuu999uuuu"
}
```

Adding Google Kubernetes Engine Provider Profile

Before you begin

Ensure that you have completed the prerequisites for configuring clusters on Google Kubernetes Engine (GKE). For more information, see Prerequisites for Configuring Clusters on GKE, on page 55.

Step 1 In the left pane, click **Infrastructure Provider**.
The **Infrastructure Providers** screen appears.

Step 2 Click **NEW PROVIDER** and specify the following information:

a) In the **PROVIDER NAME** field, enter a name for the related GKE account.

b) In the **CREDENTIALS** field, copy and paste the content from the credentials.json file that you created on Google Cloud Platform (GCP).

For more information, see Creating User Credentials on GCP, on page 56.

c) From the **PROJECT ID** drop-down list, choose the project ID in which you want to create the GKE clusters.

Step 3 Click **ADD**.

For more information on administering clusters on Google Cloud Platform (GCP), see Administering Clusters on Google Kubernetes Engine (GKE), on page 55.

Modifying Provider Profile

Step 1 In the left pane, click **Infrastructure Providers**.
The **Infrastructure Providers** screen appears.

Step 2 Click the **vSphere**, **AWS**, **OpenStack**, **Azure**, or **GKE** tab as necessary.

Step 3 From the drop-down list displayed under the **ACTIONS** column, choose **Edit** for the provider profile that you want to modify.

Step 4 Change the provider details as necessary and click **SUBMIT**.

Deleting Provider Profile

Step 1 In the left pane, click **Infrastructure Providers**.

Step 2 Click the **vSphere**, **AWS**, **OpenStack**, **Azure**, or **GKE** tab as necessary.

Step 3 From the drop-down list displayed under the **ACTIONS** column, choose **Delete** corresponding to the provider profile that you want to delete.

Step 4 Click **DELETE** in the confirmation dialog box.

Managing ACI Profile

Cisco Container Platform enables you to define ACI profiles using which tenant clusters can be created.

You can define multiple ACI profiles and use the same profile for multiple clusters.

Adding ACI Profile

Step 1 In the left pane, click **ACI Profiles**.

Step 2 Click **Add New ACI Profile** and perform these steps:

a) Specify information such as profile name, IP address, username, and passphrase of the ACI instance.

Note If there is more than one host, use a comma-separated host list in the **APIC IP ADDRESSES** field.

b) In the **NAMESERVERS** field, enter the IP address of all the DNS servers that the ACI fabric can access.
c) From the **VMM DOMAIN** drop-down list, choose the Virtual Machine Manager Domain (VMMD) that you want to use.
d) In the **INFRASTRUCTURE VLAN ID** field, enter the VLAN number for layer 2 networking.

e) From the **VRF** drop-down list, choose the Virtual Routing and Forwarding (VRF) IP address.

f) From the **L3OUT POLICY NAME** drop-down list, choose the ACI object for allowing external internet connectivity.

g) From the **L3OUT NETWORK NAME** drop-down list, choose the external network that is reachable through the L3OUT object.

h) From the **AAEP NAME** drop-down list, choose an Attachable Access Entity Profile (AAEP) name to associate the VMM domain with an AAEP.

i) In the **STARTING SUBNET FOR PODS** field, enter the starting IP address of the pod subnet.

The IP addresses for the pods are allocated from the pod subnet.

j) In the **STARTING SUBNET FOR SERVICE** field, enter the starting IP address of the service subnet.

The IP addresses for services are allocated from the service subnet.

k) In the **CONTROL PLANE CONTRACT NAME** field, enter the name of the contract that is provided by the endpoint group of the control plane.

This setting allows traffic from the control plane cluster to the tenant cluster.

l) In the **NODE VLAN START ID** field, enter the starting VLAN ID of the node network.

The IP addresses for the VLAN are allocated from the node network.

m) In the **NODE VLAN END ID** field, enter the ending VLAN ID of the node network.

Note Ensure that you configure two VLANs for each cluster.

n) In the **OPFLEX MULTICAST RANGE** field, enter a range for the Opflex multicast.

Step 3 Click **SUBMIT**.

Modifying ACI Profile

Step 1 In the left pane, click **ACI Configuration**.

Step 2 From the drop-down list displayed under the **ACTIONS** column, choose **Edit** for the ACI profile that you want to modify.

Step 3 Change the ACI profile details as necessary and click **SUBMIT**.

Deleting ACI Profile

Step 1 In the left pane, click **ACI Configuration**.

Step 2 From the drop-down list displayed under the **ACTIONS** column, choose **Delete** for the ACI profile that you want to delete.

Step 3 Click **DELETE** in the confirmation dialog box.

Managing Networks

Note This section applies to a non-ACI environment.

Based on the information that you provided during installation, Cisco Container Platform creates a network, subnet, and an IP pool. Cisco Container Platform requires a minimum of six IP addresses. After installation, you can add or modify the IP pool range, subnet, or network by using the Cisco Container Platform web interface. The IP address pools define the IP address ranges that are managed by Cisco Container Platform.

Note You must ensure that the range of IP addresses in the VIP pools is outside of the IP addresses that are assigned by DHCP.

The IP addresses that are managed by Cisco Container Platform are used for the following purposes:

- A VIP for the Cisco Container Platform Kubernetes Master
- A VIP for the external Ingress access of Cisco Container Platform
- Static Interface IP addresses for master and worker nodes in each tenant cluster
- A VIP for the Kubernetes master of each tenant cluster
- A VIP for the external NGINX Ingress Controller of each tenant cluster
- VIPs for any LoadBalancer type Kubernetes Service of a tenant cluster

To create tenant clusters, you need to configure a subnet during cluster creation. The total number of free IP addresses across all the pools for that subnet must be at least:

3 + (Number of tenant worker nodes)

Modifying Networks

Step 1 In the left pane, click **Networks**.
The **Networks** page displays the default network.

Step 2 From the drop-down list displayed under the **ACTIONS** column, choose **Edit** for the network that you want to modify.

Alternatively, click the **SUBNETS** tab or the **POOLS** tab, and then click **EDIT** from the right pane to view the **Edit** dialog box.

Step 3 Modify the network name as necessary and click **SUBMIT**.

Adding Subnets

If you want to allocate VIP from a different subnet CIDR you need to add the subnet.

Step 1 In the left pane, click **Networks**, and then click the network to which you want to add a subnet.

Step 2 From the right pane, click **NEW SUBNET**.

Step 3 Enter a name and CIDR for the subnet.

Step 4 Enter a gateway IP address that you want to use.

A gateway IP address allows a cluster to acess other networks.

Step 5 Enter the IP address of the necessary DNS nameserver.

You can click **+NAMESERVER** to enter IP addresses of additional nameservers.

Step 6 Click **SUBMIT**.

Modifying Subnets

Step 1 In the left pane, click **Networks**, and then click the network that contains the subnet you want to modify.

Step 2 Click the **SUBNETS** tab.

Step 3 From the drop-down list displayed under the **ACTIONS** column, choose **Edit** for the subnet that you want to modify.

Step 4 Modify the subnet name, CIDR, gateway IP or list of nameservers as necessary.

Step 5 Click **SUBMIT**.

Adding VIP Pool

Step 1 In the left pane, click **Networks**, and then click the network to which you want to add a VIP pool.

Step 2 From the right pane, click **NEW POOL**.

Step 3 Specify a name, subnet and IP address range for the VIP pool.

Step 4 Click **SUBMIT**.

Modifying VIP Pool

Step 1 In the left pane, click **Networks**, and then click the network that contains the VIP pool you want to modify.

Step 2 Click the **POOLS** tab.

Step 3 From the drop-down list displayed under the **ACTIONS** column, choose **Edit** for the VIP pool that you want to modify.

Step 4 Change the pool name and the IP address as necessary, and then click **SUBMIT**.

CHAPTER 4

Administering Clusters on vSphere

You can create, upgrade, modify, or delete vSphere on-prem Kubernetes clusters using the Cisco Container Platform web interface.

Cisco Container Platform supports v2 and v3 clusters on vSphere. The v2 clusters use a single master node for its Control Plane, whereas the v3 clusters can use 1 or 3 master nodes for its control plane. The multi-master approach of v3 clusters is the preferred cluster type as this approach ensures high availability for the Control Plane.

Note The UI differences between v2 and v3 clusters are called out in the cluster creation task.

This chapter contains the following topics:

Creating Clusters on vSphere

Before you begin

Ensure that your subnets do not overlap. The list of subnets to consider are:

- 172.17.0.0/16: Docker Bridge uses this default subnet for networking. You can change the Default Docker Bridge IP address during cluster deployment.
- 10.96.0.0/12: This subnet is defined as `--service-cluster-ip-range` in each guest cluster. Kubernetes allocates and assigns IP addresses from this subnet for Kubernetes services. You cannot change the Kubernetes cluster IP address range.
- Routable CIDR subnet: The routable CIDR subnet for node and load balancer services. The subnet is defined during cluster deployment.

- Pod subnet: This subnet applies to the ACI-CNI and Calico options. The IP addresses for the pods are assigned from this subnet. The subnet is defined either in the ACI-CNI profile for the ACI-CNI option or during cluster deployment for the Calico CNI option.
- Service subnet: This subnet applies to the ACI-CNI option. Cisco Container Platform assigns each Kubernetes node an IP address from this subnet. It is used by ACI as PBR target for Kubernetes services of type load balancer. By default, this subnet is part of the same ACI VRF as the pod subnet and the Routable CIDR subnet.
- ACI infrastructure network: This subnet applies to the ACI-CNI option.

Step 1 In the left pane, click **Clusters**, and then click the **vSphere** tab.

Step 2 Click **NEW CLUSTER**.

Step 3 In the **Basic Information** screen:

a) From the **INFRASTRUCTURE PROVIDER** drop-down list, choose the provider related to your Kubernetes cluster. For more information, see Adding vSphere Provider Profile , on page 23.
b) In the **KUBERNETES CLUSTER NAME** field, enter a name for your Kubernetes tenant cluster.
c) In the **DESCRIPTION** field, enter a description for your cluster.
d) In the **KUBERNETES VERSION** drop-down list, choose the version of Kubernetes that you want to use for creating the cluster.
e) If you are using ACI, specify the ACI profile.

   For more information, see Adding ACI Profile, on page 27.

f) Click **NEXT**.

Step 4 In the **Provider Settings** screen:

a) From the **DATA CENTER** drop-down list, choose the data center that you want to use.
b) From the **CLUSTERS** drop-down list, choose a cluster.

   Note Ensure that DRS and HA are enabled on the cluster that you choose. For more information on enabling DRS and HA on clusters, see *Cisco Container Platform Installation Guide*.

c) From the **DATASTORE** drop-down list, choose a datastore.

   Note Ensure that the datastore is accessible to the hosts in the cluster.

d) From the **VM TEMPLATE** drop-down list, choose a VM template.
e) From the **NETWORK** drop-down list, choose a network.

   Note
   - Ensure that you select a subnet with an adequate number of free IP addresses. For more information, see Managing Networks, on page 29. The selected network must have access to vCenter.
   - For v2 clusters that use HyperFlex systems:
     - The selected network must have access to the HypexFlex Connect server to support HyperFlex Storage Provisioners.
     - For HyperFlex Local Network, select **k8-priv-iscsivm-network** to enable HyperFlex Storage Provisioners.

f) From the **RESOURCE POOL** drop-down list, choose a resource pool.

g) Click **NEXT**.

Step 5 In the **Node Configuration** screen:

a. From the **GPU TYPE** drop-down list, choose a GPU type.

Note GPU Configuration applies only if you have GPUs in your HyperFlex cluster.

b. For v3 clusters, under **MASTER**, choose the number of master nodes, and their VCPU and memory configurations.

Note You may skip this step for v2 clusters. You can configure the number of master nodes only for v3 clusters.

c. Under **WORKER**, choose the number of worker nodes, and their VCPU and memory configurations.

d. In the **SSH USER** field, enter the ssh user name.

e. In the **SSH KEY** field, enter the SSH public key that you want to use for creating the cluster.

Note Ensure that you use the Ed25519 or ECDSA format for the public key. As RSA and DSA are less secure formats, Cisco prevents the use of these formats.

f. In the **ROUTABLE CIDR** field, enter the IP addresses for the pod subnet in the CIDR notation.

For more information on the routable CIDR, see Tenant Cluster with ACI Deployment, on page 3.

g. From the **SUBNET** drop-down list, choose the subnet that you want to use for this cluster.

h. In the **POD CIDR** field, enter the IP addresses for the pod subnet in the CIDR notation.

i. In the **DOCKER HTTP PROXY** field, enter a proxy for the docker.

j. In the **DOCKER HTTPS PROXY** field, enter an https proxy for the docker.

k. In the **DOCKER BRIDGE IP** field, enter a valid CIDR to override the default Docker bridge.

Note If you want to install the HX-CSI addon, ensure that you set the CIDR network prefix of the **DOCKER BRIDGE IP** field to `/24`.

l. Under **DOCKER NO PROXY**, click **ADD NO PROXY**, and then specify a comma-separated list of hosts that you want to exclude from proxying.

m. In the **VM USERNAME** field, enter the VM username that you want to use as the login for the VM.

n. Under **NTP POOLS**, click **ADD POOL** to add a pool.

o. Under **NTP SERVERS**, click **ADD SERVER** to add an NTP server.

p. Under **ROOT CA REGISTRIES**, click **ADD REGISTRY** to add a root CA certificate to allow tenant clusters to securely connect to additional services.

q. Under **INSECURE REGISTRIES**, click **ADD REGISTRY** to add docker registries created with unsigned certificates.

r. For v2 clusters, the Istio add-on is deprecated.

s. Click **NEXT**.

Step 6 For v2 clusters, to integrate Harbor with Cisco Container Platform:

Note Harbor is currently not available for v3 clusters.

a) In the **Harbor Registry** screen, click the toggle button to enable Harbor.
b) In the **PASSWORD** field, enter a password for the Harbor server administrator.

c) In the **REGISTRY** field, enter the size of the registry in gigabits.
d) Click **NEXT**.

Step 7 In the **Summary** screen, verify the configuration, and then click **FINISH**.

The cluster deployment takes a few minutes to complete. The newly created cluster is displayed on the **Clusters** screen.

For more information on deploying applications on clusters, see Deploying Applications on Kubernetes Clusters, on page 91.

Configuring Add-ons for Clusters on vSphere

Note This section applies to v3 clusters.

In v3 clusters, the monitoring, logging, Istio, Harbor, and Kubernetes dashboard functions are available as configurable add-ons.

In v2 clusters, the monitoring, logging, Harbor, and Kubernetes dashboard add-ons are installed by default. The Istio add-on has been deprecated.

Step 1 In the left pane, click **Clusters**, and then click the **vSphere** tab.

Step 2 From the **VERSIONS** drop-down list, choose **VERSION 3** to view the v3 clusters.

Step 3 Choose the cluster for which you want to configure add-ons.

Step 4 Click the **ADD-ONS** tab.
The **Installed Add-ons** page appears.

Step 5 Click **INSTALL ADD-ON**.
The **Install Add-on** page appears.

Step 6 In the **Select an Add-on** area, click one of the following add-ons:

- Monitoring: For monitoring clusters
- Logging: For logging
- Dashboard: For deploying and managing the applications that are deployed on the clusters
- Kubeflow: For deploying machine learning (ML) workloads
- HyperFlex Storage (CSI): For deploying HyperFlex storage
- Istio Operator: For deploying the Istio operator service, which is required for running Istio
- Istio: For deploying the Istio services, which requires the Istio Operator to be running beforehand
- Harbor Operator: For deploying the Harbor operator service, which is required for running Harbor

Note The default registry size of a Harbor instance is 20Gi. You can modify the default size using the **REGISTRY SIZE** field in the **Configure the Add-on** area. Customizing the Chartmuseum size using the Cisco Container Platform web interface is not currently supported. As a workaround, see Customizing Chartmuseum Size of Harbor Instance, on page 35.

- Harbor: For deploying the Harbor service, which requires the Harbor Operator to be running beforehand

Step 7 Click **INSTALL**.

Customizing Chartmuseum Size of Harbor Instance

Note This section applies to v3 clusters.

The default chartmuseum size of a Harbor instance is 5Gi. Customizing the Chartmuseum size using the Cisco Container Platform web interface is not currently supported. As a workaround, you can use the following steps:

Step 1 Install the Harbor operator add-on as described in Configuring Add-ons for Clusters on vSphere, on page 34.

Step 2 SSH into the master of the tenant cluster.

Step 3 Customize the Chartmuseum size.

For example, to set the size of the chartmuseum to `40Gi`, run the following command:

```
helm install -n harbor-cr /opt/ccp/charts/ccp-harbor-cr.tgz --set chartmuseumSize=40Gi
```

Deleting Add-ons for v3 Clusters

Note This section applies to v3 clusters.

In v3 clusters, the monitoring, logging, Istio, Harbor, and Kubernetes dashboard functions are removable through the Cisco Container Platform web interface.

In v2 clusters, you cannot delete these add-ons through the Cisco Container Platform web interface.

Step 1 In the left pane, click **Clusters**, and then click the **vSphere** tab.

Step 2 From the **VERSIONS** drop-down list, choose **VERSION 3** to view the v3 clusters.

Step 3 Choose the cluster for which you want to delete add-ons.

Step 4 Click the **ADD-ONS** tab.
The **Installed Add-ons** page appears.

Step 5 From the drop-down list displayed under the **ACTIONS** column, click **Delete** for the add-on that you want to delete.

Step 6 Click **Close**.

Upgrading Clusters on vSphere

Before you begin

Ensure that you have imported the latest tenant cluster OVA to the vSphere environment.

Ensure that an adequate number of free IP addresses are available. For more information, see Managing Networks, on page 29.

For more information on importing the tenant cluster OVA, see the *Cisco Container Platform Installation Guide*.

Step 1 In the left pane, click **Clusters**, and then click the **vSphere** tab.

Step 2 From the drop-down list displayed under the **ACTIONS** column, choose **Upgrade** for the cluster that you want to upgrade.

Step 3 In the **Upgrade Cluster** dialog box, choose a Kubernetes version and a new template for the VM, and then click **Submit**.
It may take a few minutes for the Kuberenetes cluster upgrade to complete.

Scaling Clusters on vSphere

You can scale clusters by adding or removing worker nodes to them based on the demands of the workloads you want to run. You can add worker nodes in a default or custom node pool.

For more information on adding worker node pools, see Configuring Node Pools, on page 36.

Configuring Node Pools

Node pools allow the creation of worker nodes with varying configurations. Nodes belonging to a single node pool have identical characteristics.

In the Cisco Container Platform vSphere implementation, a node pool has the following properties:

- vcpus
- memory
- template
- labels
- taints

Labels and taints are optional parameters. All nodes that belong to a nodepool are tagged with labels and they are tainted. Taints are key-value pairs, which are associated with an *effect*.

The following table describes the available *effects*.

Effect	Description
NoSchedule	Ensures that the pods that do not contain this taint are not scheduled on the node.
PreferNoSchedule	Ensures that Kubernetes avoids scheduling pods that do not contain this taint on the node.
NoExecute	Ensures that a pod is removed from the node if it is already running on the node, and is not scheduled on the node if it is not yet running on the node.

During cluster creation, each cluster is assigned a default node pool. Cisco Container Platform supports the ability for different master and worker configurations. Upon cluster creation, the master node is created in the default-master-pool and the worker nodes are created in the default-pool.

Cisco Container Platform supports the ability to create multiple node pools and customize each pool characteristics such as vCCPUs, memory, labels, and taints.

Adding Node Pools

Cisco Container Platform allows you to add custom node pools to an existing cluster.

Step 1 Click the cluster for which you want to add a node pool.
The **Cluster Details** page displays the node pools of the cluster that you have selected.

Step 2 In the right pane, click **ADD NODE POOL**.
The **Add Node Pool** page appears.

Step 3 Under **POOL NAME**, enter a name for the node pool.

Step 4 Ensure that an adequate number of free IP addresses is available in the subnet that you have selected during tenant cluster creation. For more information, see Managing Networks, on page 29.

Step 5 Under **Kubernetes Labels**, enter the key-value pair of the label.

You can click the **Delete** icon to delete a label and the **+LABEL** icon to add a label.

Step 6 Under **Kubernetes Taints**, enter the key-value pair and the effect you want to set for the label.

You can click the **Delete** icon to delete a taint and the **+TAINT** icon to add a taint.

Step 7 Click **ADD**.

The **Cluster Details** page displays the node pools. You can point the mouse over the **Labels** and **Taints** to view a summary of the labels and taints that are assigned to a pool.

Modifying Node Pools

Cisco Container Platform allows you to modify the worker node pools.

Step 1 Click the cluster that contains the node pool that you want to modify.
The **Cluster Details** dialog box appears displaying the node pools of the cluster that you have chosen.

Step 2 From the drop-down list next to the name of the node pool, click **Edit**.
The **Update Node Pool** page appears.

Step 3 Ensure that an adequate number of free IP addresses is available in the subnet that you have selected during tenant cluster creation. For more information, see Managing Networks, on page 29.

Step 4 Under **Kubernetes Labels**, modify the key-value pair of the label.

Step 5 Under **Kubernetes Taints**, modify the key-value pair and the effect you want to set for the label.

Step 6 Click **UPDATE**.

Deleting Node Pools

Cisco Container Platform allows you to delete the worker node pools. You cannot delete the default master pool.

Step 1 Click the cluster that contains the node pool that you want to delete.
The **Cluster Details** page displays the node pools of the cluster that you have chosen.

Step 2 From the drop-down list next to the worker pool that you want to delete, choose **Delete**.
The worker pool is deleted from the **Cluster Details** page.

Deleting Clusters on vSphere

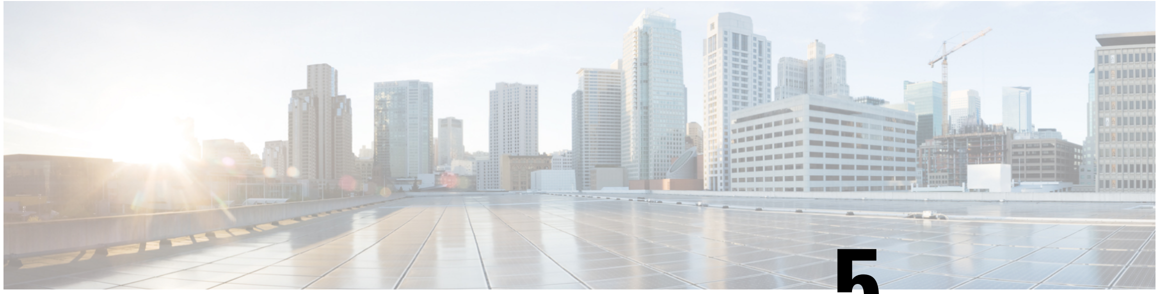
Before you begin

Ensure that the cluster you want to delete is not currently in use, as deleting a cluster removes the containers and data associated with it.

Step 1 In the left pane, click **Clusters**, and then click the **vSphere** tab.

Step 2 From the drop-down list displayed under the **ACTIONS** column, choose **Delete** for the cluster that you want to delete.

Step 3 Click **DELETE** in the confirmation dialog box.



CHAPTER 5

Administering Clusters on Amazon Web Services (AWS) EKS

Integrating Cisco Container Platform with Amazon Web Services (AWS) allows you to deploy and run containerized applications across both Cisco-based on-prem environments and the AWS cloud.

This chapter contains the following topics:

Prerequisites for Configuring Clusters on AWS EKS

The prerequisites for configuring clusters on AWS EKS are as follows:

See also Adding Amazon Provider Profile, on page 24.

Amazon Resource Requirements

The following table describes the default limits for the Amazon resources that you may need to increase depending on your Cisco Container Platform deployment requirements.

Note To increase the limits for a specific resource, you need to contact Amazon support.

Amazon Resource	Default Limit	Description
Network Address Translation (NAT) gateway for each AWS account	14	Each EKS cluster uses three NAT gateways. With the default setting, you are limited to four clusters.

Amazon Resource	Default Limit	Description
Amazon Virtual Private Cloud (Amazon VPC) for each AWS account	3	Each tenant cluster requires a separate Amazon VPC.
Amazon Elastic Container Service for Kubernetes (Amazon EKS) cluster for each AWS account	3	**Note** Changes to the Amazon EKS cluster limit are updated only on Thursdays.
Elastic IP address for each region	5	Each EKS cluster uses three elastic IP addresses. For more information, see Amazon VPC Limits.
Internet gateway for each region	5	Each EKS cluster uses one internet gateway.

Configuring Storage Class for EKS Clusters

You can configure additional storage classes to allow Kubernetes clusters running on AWS to manage the lifecycle of Amazon EFS file systems by installing the Amazon EFS Container Storage Interface (CSI) driver.

Note Cisco TAC support is not available for the AWS EFS CSI storage.

For more information, see Amazon EFS CSI driver.

Adding AMI Files to your Amazon Account

Cisco Container Platform generates a specific AMI (Amazon Machine Image) file with each product release. The AMI file ensures that compatible packages are available for successful tenant cluster creation.

To make the AMI file available to your Amazon account, you must submit a support case that includes your 12 digit Amazon account ID. You will be notified when the AMI is available within your Amazon account.

Creating AWS Roles

Step 1 Log in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.

Step 2 In the navigation pane of the IAM console, click **Roles**, and then click **Create role**.

Step 3 Under **Select type of trusted entity**, click **Another AWS account**.

Step 4 In the **Account ID** field, enter your **AWS Account ID**, and then click **Next**.

The AWS account number must be a trusted entity so that Cisco Container Platform can use the Role ARN during EKS cluster creation.

Step 5 Skip the screen to choose permission policies and permission boundary and click **Next**.

Step 6 Add metadata to the role by attaching tags of your choice as key–value pairs and click **Next**.

Step 7 In the **Role name** field, enter the name for the role as `k8s-ccp-user` or any other name of your choice.

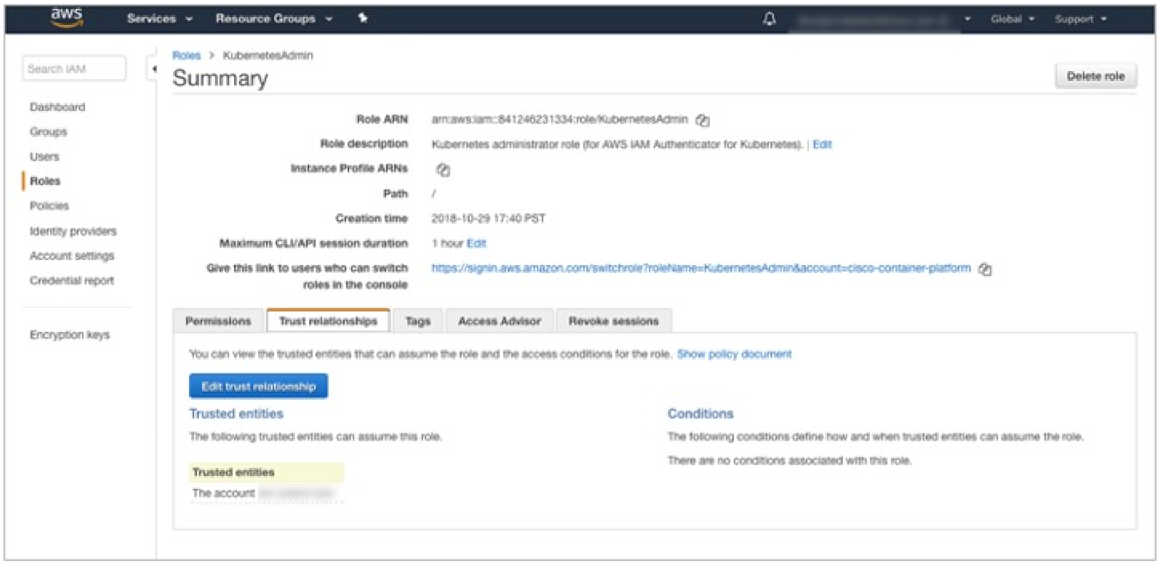
Step 8 In the **Description** field, enter a description of your choice and click **Create role**.

Step 9 After the role is created, navigate to the created role and verify the following details of the role:

a) Click the **Permissions** tab to verify that permissions are not set.

b) Click the **Trust Relationships** tab to verify that a trust relationship exists for the AWS account that you entered during creation of the Role ARN.

Figure 3: AWS Management Console-Trust Relationships Tab



Summary

Role ARN arn:aws:iam::841246231334:role/KubernetesAdmin

Role description Kubernetes administrator role (for AWS IAM Authenticator for Kubernetes). | Edit

Instance Profile ARNs

Path /

Creation time 2018-10-29 17:40 PST

Maximum CLI/API session duration 1 hour Edit

Give this link to users who can switch roles in the console https://signin.aws.amazon.com/switchrole?roleName=KubernetesAdmin&account=cisco-container-platform

You can view the trusted entities that can assume the role and the access conditions for the role. Show policy document

Trusted entities

The following trusted entities can assume this role.

Trusted entities
The account [redacted]

Conditions

The following conditions define how and when trusted entities can assume the role.

There are no conditions associated with this role.

Configuring Permissions for AWS Account

If the AWS provider account is not a root account, you must ensure that the account has the permissions needed to create the EKS and EC2 resources.

The following Sample aws-provider-policy.json File shows configuring the minimum permissions required for your AWS account. You need to create and import this file to configure the necessary permissions.

Sample aws-provider-policy.json File

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudformation:*",
                "elasticloadbalancing:*",
                "autoscaling:*",
                "ec2:*",
                "eks:*",
                "ecr:*",
                "ecs:*",
                "s3:*"
                ],
                "Resource": "*"
            },
            {
                "Effect": "Allow",
                "Action": [
                    "iam:List*",
                    "iam:Get*",
                    "iam:PassRole",
                    "iam:AddRoleToInstanceProfile",
                    "iam:RemoveRoleFromInstanceProfile",
                    "iam:CreateRole",
                    "iam:CreateInstanceProfile",
                    "iam:DeleteInstanceProfile",
                    "iam:DeleteRole",
                    "iam:DeleteRolePolicy",
                    "iam:AttachRolePolicy",
                    "iam:DetachRolePolicy",
                    "iam:PutRolePolicy",
                    "iam:*AccessKey*",
                    "iam:*MFA*"
                ],
                "Resource": "*"
            }
        ]
    }
```

For more information on user privileges on AWS, see Minimum User Privileges on AWS, on page 108.

Creating Access Keys

Access keys are required to authenticate your requests to the AWS Provider. An access key consists of two parts — an access key ID and a secret access key.

You can use the AWS IAM system in one of the following ways:

- Using a single user or personal account

  See Creating Access Keys for a Single User Account, on page 42, for creating access keys to allow access to AWS resources.

- Using a federated login account, for enterprises or corporate entities

  See Creating Access Keys for Federated Login Accounts, on page 43, for creating access keys to allow programmatic access to AWS resources.

Creating Access Keys for a Single User Account

Step 1 Log in to the AWS Management Console at https://console.aws.amazon.com.

Step 2 From the **Username** drop-down list on the top-right corner, choose **My Security Credentials**.
The **My Security Credentials** page appears.

Step 3 Expand the **Access keys** section.

Step 4 Click **Create New Access Key**.
A popup appears displaying the new access key ID and the secret access key.

Step 5 Click **Download Credentials**, and download the CSV file that contains the access keys and save it on your computer.

You can use this access key while adding your Amazon provider profile. For more information, see Adding Amazon Provider Profile, on page 24.

Creating Access Keys for Federated Login Accounts

Step 1 Log in to the AWS Management Console at https://console.aws.amazon.com.

Step 2 In the left pane, click **Add User** to create a new user, which Cisco Container Platform will use to login.

Step 3 In the **Set user details** section, enter a username in the **Username** field.

For example, **ccp-user**.

Step 4 In the **Select AWS access type** section, set **Programmatic Access** as the **Access Type**, and then click **Next**.

This setting provides the access key ID and secret access key for the Cisco Container Platform AWS provider.

Step 5 In the **Set Permissions** table, click **Add User to Group**, and in the lower section, select the group that you created previously (ccp-user), and then click **Next**.

Step 6 In the **Add tags** page, click **Next**.

In the next screen, the new access key ID and secret access key are displayed. Stay on this screen as it is only shown once.

Step 7 Click **Download Credentials**, and download the CSV file that contains the access keys and save it on your computer. You can use this access key while adding your Amazon provider profile. For more information, see Adding Amazon Provider Profile, on page 24.

Amazon IAM Authentication

By default, the AWS IAM identity is used to authenticate and connect with clusters on EKS clusters. Cisco Container Platform uses AWS IAM Authenticator to authenticate on-prem cluster using the AWS IAM identity. This authentication provides a consistent, unified identity scheme across both on-premise and clusters on AWS EKS.

The AWS IAM Authenticator fulfills both a client and server function. On the client side, the authenticator generates, tokenizes and transmits a pre-signed URL to the server-side for identity validation. The client is a Go binary, installed on your workstation, which is transparently invoked by kubectl each time you interact with your Kubernetes cluster. The server-side is a containerized instance of AWS IAM Authenticator running as a DaemonSet on the Kubernetes master nodes. This interacts with the AWS Secure Token Service (STS) to perform identity validation. Cisco Container Platform takes care of the initial server-side configuration and provides a preconfigured `Kubeconfig` file for admin users to download.

Note You need to ensure that the AWS IAM Authenticator is available within your `$PATH` while using kubectl to interact with the clusters.

Enabling Common Identity

Within the Cisco Container Platform web interface, users are able to select a common identity scheme for clusters. After the clusters are provisioned, you can apply a shared RBAC policy.

Note The use of IAM Authentication is implicitly enabled for EKS clusters. Cisco Container Platform can map a user supplied IAM role to the EKS cluster and configuring IAM auth for on-premises clusters.

Configuring Control Plane Proxy for EKS Access

If your Control Plane VMs need proxy configuration to access the internet, specifically AWS API endpoints, you need to configure Cisco Container Platform application deployments with the proxy information.

Step 1 SSH to the Control Plane cluster master VM.

Step 2 Run the following commands to specify the proxy information:

Note You need to replace <*Proxy_URL_or_IPAddress:Port*> with the URL/IP address of your proxy server and port and the no_proxy list with a list of your internal IP addresses.

```
kubectl patch deployment kaas-api --patch
[illegible]

kubectl patch deploy kaas-ccp-eks-operator --patch
[illegible]

kubectl patch daemonset aws-iam-authenticator -n kube-system --patch
[illegible]
```

Creating AWS EKS Clusters

Before you begin

- Ensure that you have added your Amazon provider profile. For more information, see Adding Amazon Provider Profile, on page 24.
- Ensure that you have added the required AMI files to your account. For more information, see Adding AMI Files to your Amazon Account, on page 40.
- Ensure that you have created an AWS IAM Role for the Cisco Container Platform usage to create AWS EKS Clusters. For more information, see Creating AWS Roles, on page 40.

Step 1 In the left pane, click **Clusters**, and then click the **AWS** tab.

Step 2 Click **NEW CLUSTER**.

Step 3 In the **Basic Information** screen, enter the following information:

a) From the **INFRASTUCTURE PROVIDER** drop-down list, choose the provider related to the appropriate Amazon account.

b) From the **AWS REGION** drop-down list, choose an appropriate AWS region.

Note Not all regions support EKS. Ensure that you select a supported region. Currently, Cisco Container Platform supports the **ap-northeast-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, eu-central-1, eu-north-1, eu-west-1, eu-west-2, eu-west-3, us-east-1, us-east-2,** and **us-west-2** regions.

c) In the **KUBERNETES CLUSTER NAME** field, enter a name for your cluster.

d) Click **NEXT**.

Step 4 In the **Node Configuration** screen, specify the following information:

a) From the **INSTANCE TYPE** drop-down list, choose an instance type for your cluster.

b) From the **MACHINE IMAGE** drop-down list, choose an appropriate Cisco Container Platform Amazon Machine Image (AMI) file.
To add AMI files to your Amazon account, see Adding AMI Files to your Amazon Account, on page 40.

c) In the **WORKER COUNT** field, enter an appropriate number of worker nodes.

d) In the **SSH PUBLIC KEY** drop-down field, choose an appropriate authentication key.
This field is optional. It is needed if you want to ssh to the worker nodes for troubleshooting purposes. Ensure that you use the Ed25519 or ECDSA format for the public key.

Note: As RSA and DSA are less secure formats, Cisco prevents the use of these formats.

e) In the **IAM ACCESS ROLE ARN** field, enter the Amazon Resource Name (ARN) information.

Note By default, the AWS credentials specified at the time of Amazon EKS cluster creation, that is the credentials configured in the Infrastructure Provider, are mapped to the `Kubernetes cluster-admin ClusterRole`. A default `ClusterRoleBinding` binds the credentials to the `system:masters` group, thereby granting super-user access to the holders of the IAM identity. The **IAM ACCESS ROLE ARN** field allows you to specify the ARN of an additional AWS IAM role or IAM user who is also granted administrative control of the cluster.

f) Click **NEXT**.

Step 5 In the **VPC Configuration** screen, specify the following information:

a) In the **SUBNET CIDR** field, enter a value of the overall subnet CIDR for your cluster.

b) In the **PUBLIC SUBNET CIDR** field, enter values for your cluster on separate lines.

c) In the **PRIVATE SUBNET CIDR** field, enter values for your cluster on separate lines.

Step 6 In the **Summary** screen, review the cluster information and then click **FINISH**.

Cluster creation can take up to 20 minutes. You can monitor the cluster creation status on the **Clusters** screen.

Note If you receive the Could not get token: AccessDenied error message, it indicates that the AWS account is not a trusted entity for the Role ARN.

For information on adding your AWS account as a trusted entity, see Creating AWS Roles, on page 40.

Scaling AWS EKS Clusters

You can scale EKS clusters by adding or removing worker nodes to them based on the demands of the workloads you want to run.

Step 1 In the right pane, click **EDIT**.
The **Edit Cluster** dialog box appears.

Step 2 From the **INSTANCE TYPE** drop-down list, choose an instance type for your cluster.

Step 3 From the **MACHINE IMAGE** drop-down list, choose an appropriate Cisco Container Platform Amazon Machine Image (AMI) file.
To add AMI files to your Amazon account, see Adding AMI Files to your Amazon Account, on page 40.

Step 4 In the **WORKER COUNT** field, change the number of work nodes as necessary.

Step 5 Click **UPDATE**.

Deleting AWS EKS Clusters

Before you begin

Ensure that the AWS EKS cluster that you want to delete is not currently in use, as deleting a cluster removes the containers and data associated with it.

Step 1 In the left pane, click **Clusters**, and then click the **EKS** tab.

Step 2 From the drop-down list displayed under the **ACTIONS** column, choose **Delete** for the cluster that you want to delete.

Step 3 Click **DELETE** in the confirmation dialog box.
Upon deleting an AWS EKS cluster, it takes about 10 minutes for the cluster resources to be released.

CHAPTER 6

Administering Clusters on OpenStack

Note This section is applicable only for OpenStack clusters deployed on Cisco Virtualized Infrastructure Manager (Cisco VIM).

You can create and delete OpenStack Kubernetes clusters using the Cisco Container Platform web interface.

This chapter contains the following topics:

Creating OpenStack Clusters

Before you begin

Ensure that you have added your OpenStack provider profile.

For more information, see Adding OpenStack Provider Profile, on page 24.

Step 1 In the left pane, click **Clusters**, and then click the **OPENSTACK** tab.

Step 2 Click **NEW CLUSTER**.

Step 3 In the **Basic Information** screen, enter the following information:

a) From the **INFRASTRUCTURE PROVIDER** drop-down list, choose the provider related to the appropriate OpenStack account.
b) In the **KUBERNETES CLUSTER NAME** field, enter a name for your cluster.
c) From the **CONTAINER NETWORK INTERFACE (CNI)** drop-down list, choose the CNI that you want to use.
d) From the **KUBERNETES VERSION** drop-down list, choose the version of Kubernetes that you want to use for creating the cluster.

Step 4 In the **Provider Settings** screen, enter the following information:

a) From the **PROTOCOL** drop-down list, choose the **IPv4** or **IPv6** protocol.
b) From the **NETWORK TYPE** drop-down list, choose the type of network that you want to use.

c) From the **PUBLIC NETWORK** drop-down list, choose the network that you want to use.

d) In the **DNS SERVERS** field, enter the IP address of the necessary DNS servers.

You can use a comma-separated list to enter the IP addresses of additional DNS servers.

e) From the **IMAGE** drop-down list, choose an appropriate image file.

f) From the **CINDER AVAILABILITY ZONE** drop-down list, choose a suitable Cinder availability zone.

g) From the **NOVA AVAILABILITY ZONE** drop-down list, choose a suitable Nova availability zone.

h) Click **NEXT**.

Step 5 In the **Cluster Configuration** screen, enter the following information:

a) Under **MASTER NODES**, specify the number of master nodes.

b) Under **WORKER NODES**, specify the number of worker nodes.

c) From the **FLAVOR** drop-down list, choose a flavor to configure each node.

d) In the **POD NETWORK CIDR** field, enter a CIDR for your pod network.

e) From the **SSH KEY NAME** drop-down list, choose an appropriate key name.

f) If you want to allow connection to an external router, from the **ROUTER** drop-down list, choose the router that you want to use.

g) Under **ROOT CA REGISTRIES**, click **ADD REGISTRY**, and then specify root CA certificates to allow tenant clusters to securely connect to additional services.

You can use a comma-separated list to include multiple values.

h) Under **SELF-SIGNED REGISTRIES**, click **ADD REGISTRY**, and then specify self-signed certificates for Docker registries.

You can use a comma-separated list to include multiple values.

i) Under **NTP POOLS**, click **ADD REGISTRY**, and then specify IP address ranges for NTP pools.

You can use a comma-separated list to include multiple values.

j) Under **NTP SERVERS**, click **ADD REGISTRY**, and then specify IP addresses for NTP servers.

You can use a comma-separated list to include multiple NTP servers.

k) Click **NEXT**.

Step 6 In the **Summary** screen, review the cluster information, and then click **FINISH**.
Cluster creation can take up to 20 minutes. You can monitor the cluster creation status on the **Clusters** screen.

Configuring Add-ons for Clusters on Openstack

Note This section applies to v3 clusters.

In v3 clusters, the Harbor, and monitoring functions are available as configurable add-ons.

In v2 clusters, these add-ons are installed by default.

Step 1 In the left pane, click **Clusters**, and then click the **Openstack** tab.

Step 2 From the **VERSIONS** drop-down list, choose **VERSION 3** to view the v3 clusters.

Step 3 Choose the cluster for which you want to configure add-ons.

Step 4 Click the **ADD-ONS** tab.
The **Installed Add-ons** page appears.

Step 5 Click **INSTALL ADD-ON**.
The **Select an Add-on** page appears.

Step 6 Select one of the following add-ons:

- Harbor Operator: For deploying the Harbor operator service, which is required for running Harbor
- CVIMMON Plugin: For monitoring clusters

Step 7 Click **Close**.

For more information, see Customizing Size of Registry and Chartmuseum of Harbor Instance, on page 49.

Customizing Size of Registry and Chartmuseum of Harbor Instance

Note This section applies to v3 clusters.

The default size of the registry and chartmuseum of a Harbor instance is 20Gi and 5Gi respectively.

Customizing this size using the Cisco Container Platform web interface is not currently supported. As a workaround, you can use the following steps:

Step 1 Install the Harbor operator add-on.

For more information, see Configuring Add-ons for Clusters on Openstack, on page 48.

Step 2 Ssh into the master of the tenant cluster.

Step 3 Customize the size of the registry and Chartmuseum.

For example, to set the size of the registry and chartmuseum to `40Gi`, run the following command:

```
helm install -n harbor-cr /opt/ccp/charts/ccp-harbor-cr.tgz --set registrySize=40Gi --set
chartmuseumSize=40Gi
```

Deleting Add-ons for Clusters on Openstack

Note This section applies to v3 clusters.

In v3 clusters, you can delete the monitoring, logging, Istio, Harbor, and Kubernetes dashboard functions through the Cisco Container Platform web interface.

In v2 clusters, you cannot delete these add-ons through the Cisco Container Platform web interface.

Step 1 In the left pane, click **Clusters**, and then click the **Openstack** tab.

Step 2 From the **VERSIONS** drop-down list, choose **VERSION 3** to view the v3 clusters.

Step 3 Choose the cluster for which you want to delete add-ons.

Step 4 Click the **ADD-ONS** tab.
The **Installed Add-ons** page appears.

Step 5 From the drop-down list displayed under the **ACTIONS** column, click **Delete** for the add-on that you want to delete.

Step 6 Click **Close**.

Deleting OpenStack Clusters

Before you begin

Ensure that the OpenStack cluster that you want to delete is currently not in use, as deleting a cluster removes the containers and data associated with it.

Step 1 In the left pane, click **Clusters**, and then click the **Openstack** tab.

Step 2 From the drop-down list displayed under the **ACTIONS** column, choose **Delete** for the cluster that you want to delete.

Step 3 Click **DELETE** in the confirmation dialog box.
Upon deleting an OpenStack cluster, it takes about 10 minutes for the cluster resources to be released.

CHAPTER 7

Administering Clusters on Azure Kubernetes Service (AKS)

Integrating Cisco Container Platform with Azure Kubernetes Service (AKS) allows you to deploy and run containerized applications on the Azure cloud.

This chapter contains the following topics:

Prerequisites for Configuring AKS Clusters

The prerequisites for configuring AKS clusters are as follows:

Azure User Account Requirements

The following roles are necessary for your Azure user account that you want to use with Cisco Container Platform:

- An **App administrator** within the directory
- An **Owner** within the Azure subscription

You can contact your Azure administrator to set up these roles for your Azure account.

Creating Resource Groups

A resource group is a logical grouping of the resources that are required to deploy Cisco Container Platform on Azure. They allow you to manage your resources efficiently.

Step 1 Log in to the Microsoft Azure portal.

Step 2 In the left navigation pane, click **All services** > **Resource groups**.

The **Resource groups** pane is displayed.

Step 3 Click **Add** to create a new resource group.

Step 4 In the **Create an empty resource group** pane, specify the following information:

a) In the **Resource Group name** field, enter a unique name.

b) From the **Subscription** drop-down list, choose a subscription type.

c) From the **Resource group location** drop-down list, choose the region in which you want to create your AKS cluster.

Note Not all regions support AKS. Ensure that you select a supported region. Currently, Cisco Container Platform supports only the **eastus** and **westus** regions.

For more information on a resource group, see Resource Group Overview.

Configuring Control Plane Proxy for AKS Access

If your Control Plane VMs need proxy configuration to access the internet, specifically Azure API endpoints, you must configure the proxy information on Cisco Container Platform.

Step 1 SSH to the master node of the control plane.

Step 2 Run the following commands to specify the proxy information:

Note You need to replace *<Proxy_URL_or_IPAddress:Port>* with the URL/IP address of your proxy server and the no_proxy list with a list of your internal IP addresses.

```
kubectl patch deploy kaas-api
[illegible]

kubectl patch deploy kaas-ccp-aks-operator --patch
[illegible]
```

Creating AKS Clusters

Before you begin

- Ensure that you have added your Azure provider profile.

  For more information, see Adding Azure Provider Profile, on page 25.

- Ensure that your Azure account that is used with Cisco Container Platform has the necessary roles.

  For more information, see Azure User Account Requirements, on page 51.

- Ensure that you have created the necessary resource groups in your Azure account.

  For more information, see Creating Resource Groups, on page 51.

Step 1 In the left pane, click **Clusters**, and then click the **Azure** tab.

Step 2 Click **NEW CLUSTER**.

Step 3 In the **Basic Information** screen, specify the following information:

a) From the **INFRASTUCTURE PROVIDER** drop-down list, choose the provider related to the appropriate Azure account.

b) From the **AZURE REGION** drop-down list, choose an appropriate AKS region.

Note Not all regions support AKS. Ensure that you select a supported region. Currently, Cisco Container Platform supports only the **eastus** and **westus** regions.

c) In the **KUBERNETES CLUSTER NAME** field, enter a name for your cluster.

d) From the **KUBERNETES VERSION** drop-down list, choose the Kubernetes version for your cluster.

Note Not all Kubernetes versions are supported in all Azure regions. Ensure that you select a supported Kubernetes version.

e) Click **NEXT**.

Step 4 In the **Cluster settings** screen, specify the following information:

a) In the **NODE POOL NAME** field, enter an alphanumeric name for the primary node pool of your cluster.

b) From the **RESOURCE GROUP** drop-down list, enter the name of the resource group that you have created in your Azure account.

See also Creating Resource Groups, on page 51.

c) To create and use a virtual network subnet for an AKS cluster, from the **NETWORK PLUGIN** drop-down list, choose **Azure** or **Kubenet**.

See also Using Kubenet Networking.

d) If you want to use Virtual Kubelet to provision pods on the Azure Container Instance in your AKS clusters, under **VIRTUAL KUBELET**, click the toggle button to enable virtual kubelet.

Note The provider must have an application name to enable Virtual Kubelet.

e) Click **NEXT**.

Step 5 In the **Node Configuration** screen, specify the following information:

a) From the **WORKER INSTANCE TYPE** drop-down list, choose an instance type that provides an appropriate combination of CPU, memory, storage, and networking capacity for your cluster.

See also Amazon EC2 Instance Types.

b) In the **WORKER COUNT** field, enter an appropriate number of worker nodes.

c) In the **POD CIDR** field, enter the desired CIDR value.

Note This is an optional field. So you can leave the field empty or leave the default as is.

d) In the **SERVICE CIDR** field, enter the desired CIDR value.

Note This is an optional field. So you can leave the field empty or leave the default as is.

e) In the **VNET SUBNET ID** field, enter a subnet within the virtual network where you want to deploy the Azure resources.

f) In the **DOCKER BRIDGE CIDR** field, enter a valid CIDR to override the default Docker bridge.

g) In the **DNS SERVICE IP** field, enter an IP address within the Kubernetes service address range that will be used by cluster service discovery (kube-dns).

See also Configure Azure CNI Networking in AKS.

Note You must not use the first IP address in your address range, such as .1. The first address in your subnet range is used for the kubernetes.default.svc.cluster.local address.

h) Click **NEXT**.

Step 6 In the **Summary** screen, review the cluster information, and then click **FINISH**.

Cluster creation can take up to 20 minutes. You can monitor cluster creation status on the **Clusters** screen.

Upgrading AKS Clusters

The upgrade path for an AKS cluster is allowed based on the AKS upgrade map, which is completely controlled by AKS.

For example:

If the current version of your AKS cluster is 1.12.x, you must first upgrade 1.12.x to 1.13.x, and then upgrade from 1.13.x to 1.14.x. You cannot directly upgrade from 1.12.x to 1.14.x.

See also Upgrade an AKS cluster.

Before you begin

Ensure that the AKS cluster is in the **READY** state.

Step 1 In the left pane, click **Clusters**, and then click the **Azure** tab.

Step 2 From the drop-down list displayed under the **ACTIONS** column, choose **Upgrade** for the cluster that you want to upgrade.

Step 3 In the **Upgrade Azure Cluster** dialog box, choose a Kubernetes version for the VM, and then click **Upgrade**.
The cluster status changes from **READY** to **UPGRADING** to **READY**.

It may take a few minutes for the Kubernetes cluster upgrade to complete.

Deleting AKS Clusters

Before you begin

Ensure that the AKS cluster that you want to delete is not currently in use, as deleting a cluster removes the containers and data associated with it.

Step 1 In the left pane, click **Clusters**, and then click the **Azure** tab.

Step 2 From the drop-down list displayed under the **ACTIONS** column, choose **Delete** for the cluster that you want to delete.

Step 3 Click **DELETE** in the confirmation dialog box.

Upon deleting an AKS cluster, it takes about 15 minutes for the cluster resources to be released.

CHAPTER 8

Administering Clusters on Google Kubernetes Engine (GKE)

Integrating Cisco Container Platform with Google Kubernetes Engine (GKE), allows you to deploy and run containerized applications on Google Cloud Platform (GCP).

This chapter contains the following topics:

Prerequisites for Configuring Clusters on GKE

The prerequisites for configuring clusters on GKE are as follows:

See also Adding Google Kubernetes Engine Provider Profile, on page 26.

Creating New Project on GCP

You need to create a new project on Google Cloud Platform (GCP) to use the GKE services.

Step 1 Go to the New Project page on GCP.

Step 2 In the **Project name** field, enter a name for your project, and then click **CREATE**.

For more information, see Creating Your Project.

Creating Service Account

A service account represents a Google Cloud service identity. You need to create a service account to interact with the google APIs.

Step 1 Open the GCP console:

https://console.cloud.google

Step 2 In the left pane, click **IAM & Admin** > **Service Accounts**.

The **Service Accounts** page appears on the GCP console.

Step 3 Click **SELECT PROJECT**, choose your project, and then click **OPEN**.

Step 4 Click **CREATE SERVICE ACCOUNT**.
The **Create service account** page appears.

Step 5 In the **Service account details** area, enter a service account name, a description, and then click **CREATE**.

Step 6 In the **Service account permissions** page, add the following roles:

- Compute Viewer
- Kubernetes Engine Admin
- Service Account User
- Viewer

Step 7 Click **CONTINUE**.

Step 8 Grant users access to this service account and click **DONE**.

For more information, see Creating and managing service accounts.

Creating User Credentials on GCP

Step 1 Open the GCP console:

https://console.cloud.google.com

Step 2 In the left pane, click **IAM & Admin** > **Service Accounts**.

The **Service accounts** page appears on the GCP console.

Step 3 In the left pane, click **Service accounts**.

Step 4 In the row of the service account for which you want to create a key, click **Actions** button, and then click **Create key**.

Step 5 Select a **JSON** key type and click **CREATE**.
The credentials are saved to your computer in the `credentials.json` file. You need to copy and paste the contents of this file when adding a GKE provider. For more information, see Adding Google Kubernetes Engine Provider Profile, on page 26.

For more information on creating credentials in GCP, see Creating and managing service account keys.

Enabling GCP APIs

You need to enable the GCP APIs to allow querying and responding to the Google APIs.

Step 1 Open the GCP console:

https://console.cloud.google

Step 2 In the left pane, click **APIs & Services** > **Dashboard**.
The **APIs & Services** page appears on the GCP console.

Step 3 Click **ENABLE APIS AND SERVICES**.

Step 4 In the search box, enter **Kubernetes Engine API**, and then click **ENABLE**.

Step 5 In the search box, enter **Cloud Resource Manager API**, and then click **ENABLE**.

Configuring Control Plane Proxy for GKE Access

If your Control Plane VMs need proxy configuration to access the internet, specifically GKE API endpoints, you must configure the proxy information on Cisco Container Platform.

Step 1 SSH to the master node of the control plane.

Step 2 Run the following commands to specify the proxy information:

Note You need to replace <*Proxy_URL_or_IPAddress:Port*> with the URL/IP address of your proxy server and the no_proxy list with a list of your internal IP addresses.

```
kubectl patch deploy kaas-api
[illegible]

kubectl patch deploy kaas-ccp-gke-operator --patch
[illegible]
```

Creating Clusters on GKE

Before you begin

Ensure that you have completed the prerequisites for configuring clusters on GKE. For more information, see Prerequisites for Configuring Clusters on GKE, on page 55.

Step 1 In the left pane, click **Clusters**, and then click the **GKE** tab.

Step 2 Click **NEW CLUSTER**.

Step 3 In the **Basic Information** screen, specify the following information:

a) From the **INFRASTRUCTURE PROVIDER** drop-down list, choose the provider related to the appropriate GKE account.
b) In the **KUBERNETES CLUSTER NAME** field, enter a name for your cluster.
c) From the **LOCATION** drop-down list, choose a GKE region.

Note For more information on the supported regions, see Cloud Locations.

d) The **KUBERNETES VERSION** drop-down list, choose the Kubernetes version for your cluster.
e) Click **NEXT**.

Step 4 In the **Node Configuration** screen, specify the following information:

a) In the **WORKER NODES** field, enter an appropriate number of worker nodes.
b) Use the **AUTOSCALING** toggle button to enable autoscaling, and then set the minimum and maximum number of worker nodes.
c) In the **NODE POOL NAME** field, enter an alphanumeric name for the primary node pool of your cluster.
d) From the **MACHINE TYPE** drop-down list, choose an appropriate machine type for your VMs.

For more information, see Machine types.

e) From the **IMAGE TYPE** drop-down list, choose an appropriate image type for your VMs.
f) Use the **USE PREEMPTIBLE VMS** toggle button to enable or disable preemptible VMs.

For more information, see Running preemptible VMs.

g) Click **NEXT**.

Step 5 In the **Summary** screen, review the cluster information, and then click **FINISH**.
Cluster creation can take up to 20 minutes. You can monitor cluster creation status on the **Clusters** screen.

Scaling Clusters on GKE

Cisco Container Platform supports both manual scaling and autoscaling of GKE cluster node pools. Autoscaling is enabled or disabled during cluster creation. If autoscaling is disabled, you can manually scale the node pool by modifying the number of worker nodes. Whereas, if autoscaling is enabled, in addition to modifying the number of worker nodes, you can modify the minimum and maximum limits of a node pool range.

Follow these steps to scale the node pool in the cluster:

Before you begin

Ensure that the cluster is in READY state.

Step 1 Click on the GKE cluster that you want to scale.
The **Cluster Details** screen appears displaying the node pools of the cluster that you have chosen.

Step 2 From the drop-down list next to the name of the node pool, click **Edit**.
The **Edit node pool** screen appears.

Step 3 If autoscaling is disabled, increase or decrease the number of worker nodes.

Step 4 If autoscaling is enabled, you can perform these actions:

- Increase or decrease the number of worker nodes.
- Modify the node pool range, by changing the minimum and maximum number of nodes configured for the node pool.

Step 5 Click **UPDATE**.
In the **Cluster Details** screen, the status of the cluster is indicated as **OPERATION_IN_PROGRESS**.

After the scaling operation is complete, the status of the cluster is changed to **READY** and the updated number of worker nodes is displayed in the **Cluster Details** page.

Deleting Clusters on GKE

Before you begin

Ensure that the GKE cluster that you want to delete is not currently in use, as deleting a cluster removes the containers and data associated with it.

Step 1 In the left pane, click **Clusters**, and then click the **GKE** tab.

Step 2 From the drop-down list displayed under the **ACTIONS** column, choose **Delete** for the cluster that you want to delete.

Step 3 Click **DELETE** in the confirmation dialog box.

Upon deleting a GKE cluster, it takes about 15 minutes for the cluster resources to be released.

CHAPTER 9

Managing Kubernetes Clusters

The Cisco Container Platform web interface allows you to manage Kubernetes clusters by using the **Kubernetes Dashboard**. Once you set up the **Kubernetes Dashboard**, you can deploy applications on the authorized Kubernetes clusters, and manage the application and the cluster itself.

This chapter contains the following topics:

Accessing Kubernetes Clusters

The steps to access the Kubernetes clusters differ based on the method used for cluster creation.

This section contains the following topics:

Accessing Kubernetes Clusters on vSphere

You can access clusters on vSphere using the Kubernetes dashboard and the Kubernetes default token.

Step 1 To download the Kubeconfig file that provides you access to the Kubernetes cluster, perform these steps on the Cisco Container Platform web interface:

a) In the left pane, click **Clusters**.

b) From the drop-down list displayed under the **ACTIONS** column, choose **Download Token** to get the `Kubeconfig` file of the vSphere cluster.

The `Kubeconfig` file is downloaded to your local system.

Step 2 To get the Kubernetes default token, perform these steps in the kubectl utility:

a) SSH into the master of the tenant cluster.

b) List the Kubernetes secrets in the kube-system namespace.

```
kubectl get secrets -n kube-system
```

c) Search for the secret that has the following format:

```
default-token-XXXXX
```

d) Get the default token in one of the following ways:

- `kubectl describe secret default-token-XXXXX -n kube-system`
- `kubectl get secret default-token-XXXXX -n kube-system -o jsonpath='{.data.token}' | base64 -d`

Step 3 To set up the Kubernetes dashboard access, perform these steps:

a) In the left pane of the Cisco Container Platform web interface, click **Clusters**.

b) From the drop-down list displayed under the **ACTIONS** column, choose **Kubernetes Dashboard**.
The Kubernetes Dashboard login screen is displayed.

c) Log in to the Kubernetes Dashboard in one of the following ways:

- Using the **TOKEN**: Click the **TOKEN** radio button and enter the token from Step 2-c.
- Using the **kubeconfig** file: Click the **Kubeconfig** radio button and select the Kubeconfig file from Step 1-b.

Accessing Kubernetes Clusters for On-prem AWS IAM Enabled Clusters

For an on-prem AWS cluster that has IAM enabled, you can access clusters using the Kubernetes dashboard and the Kubernetes default token.

Step 1 To download the `Kubeconfig` file that provides you access to the Kubernetes cluster, perform these steps on the Cisco Container Platform web interface:

a) In the left pane, click **Clusters**.

b) From the drop-down list displayed under the **ACTIONS** column, choose **Download Token** to get the `Kubeconfig` file of the on-prem AWS cluster.
The `Kubeconfig` file is downloaded to your local system.

Step 2 To get the Kubernetes default token, perform these steps in the kubectl utility:

a) List the Kubernetes secrets in the `kube-system` namespace.

```
kubectl get secrets -n kube-system
```

b) Search for the secret that has the following format:

```
default-token-XXXXX
```

c) Get the default token.

```
kubectl describe secret default-token-XXXXX -n kube-system
```

Step 3 To set up the Kubernetes dashboard access, perform these steps in the Kubernetes dashboard:

a) Click the **Token** radio button.

b) In the **Enter token** field, enter the Kubernetes default token from Step 2-c.

Accessing Kubernetes Clusters on AWS EKS

You can use the `Kubeconfig` file along with the kubectl command line tool to access the clusters on AWS EKS.

Step 1 To download the `Kubeconfig` file that provides you access to the AWS EKS cluster, perform these steps on the Cisco Container Platform web interface:

a) In the left pane, click **Clusters**.

b) From the drop-down list displayed under the **ACTIONS** column, choose **Download Token** to get the `Kubeconfig` file of the AWS EKS cluster.
The `Kubeconfig` file is downloaded to your local system.

Step 2 To set up the Kubernetes dashboard access, follow the steps provided on the Dashboard Tutorial page.

Accessing Kubernetes Clusters on AKS

You can use the `Kubeconfig` file along with the kubectl command line tool to access the clusters on AKS.

Step 1 To download the `Kubeconfig` file that provides you access to the AKS cluster, perform these steps on the Cisco Container Platform web interface:

a) In the left pane, click **Clusters** and then click the **Azure** tab.

b) For the cluster whose dashboard you would like to access, from the drop-down list displayed under the **ACTIONS** column, choose **Download Kubeconfig** to get the `kubeconfig` file of the AKS cluster.
The `kubeconfig` file is downloaded to your environment.

Step 2 To set up Kubernetes dashboard access, follow these steps:

a) Create the necessary cluster role binding.

```
kubectl --kubeconfig=[location_of_kubeconfig_downloaded_from_ccp_dashboard] create
clusterrolebinding kubernetes-dashboard --clusterrole=cluster-admin
--serviceaccount=kube-system:kubernetes-dashboard
```

b) Connect to the Kubernetes dashboard using the Azure CLI.

```
az aks browse --resource-group [name_of_your_resource_group] --name [name_of_your_aks_cluster]
```

The Kubernetes dashboard is displayed in a new browser tab.

For more information, see Dashboard Tutorial.

Note In the Dashboard Tutorial page, you must follow the steps in the RBAC enabled cluster section because currently, in Cisco Container Platform, RBAC is enabled by default for all clusters on AKS.

Accessing Kubernetes Clusters on GKE

Before you begin

- Install Google Cloud SDK on your computer. For more information, see Installing Google Cloud SDK.

  Note You can choose a suitable installation option depending on the OS installed on your computer.

- After the SDK is installed, add the `GOOGLE_CLOUD_SDK_INSTALLATION_DIR/bin` path to the environment variable `PATH`.
- Setup the GCP credentials.

  For more information, see Configuring cluster access for kubectl.

You can configure your computer to access clusters on GKE in one of the following ways:

Accessing Clusters on GKE using GKE Dashboard

We recommend that you use the GKE dashboard that is available on the GCP console to view, inspect, manage, and delete resources in your clusters.

For more information, see GKE Dashboard.

Note The open-source Kubernetes dashboard is deprecated for clusters on GKE.

Accessing Clusters on GKE using Kubeconfig and Kubectl

You can use the `Kubeconfig` file along with the kubectl command line tool to access the clusters on GKE.

Generate the `kubeconfig` file in one of the following ways:

- Use GCP to generate the `kubeconfig` file in your environment.

  When you create a cluster using GCP, an entry is automatically added to the `kubeconfig` in the `$HOME/.kube/config` file.

  For more information, see Setting up kubeconfig on your machine.

- Alternatively, use Cisco Container Platform to generate the `kubeconfig` file:

  a. In the left pane, click **Clusters**, and then click the **GKE** tab.

  b. To access the cluster, from the drop-down list displayed under the **ACTIONS** column, choose **Download Kubeconfig** to get the `kubeconfig` file of the cluster.

  The `kubeconfig` file is downloaded to your environment.

  c. Edit the `kubeconfig` file to replace `$GCLOUD_SDK_PATH` with the local path where google-cloud-sdk is installed.

  d. A separate `kubeconfig` file is generated to allow access to the GKE clusters.

Monitoring Health of Cluster Deployments

It is recommended to continuously monitor the health of your cluster deployment to improve the probability of early detection of failures and avoid any significant impact from a cluster failure.

Cisco Container Platform is deployed with Prometheus and Grafana, which are configured to start monitoring and logging services automatically when a Kubernetes cluster is created.

Prometheus is an open-source systems monitoring and alerting toolkit and Grafana is an open source metric analytics and visualization suite.

Prometheus collects the data from the cluster deployment, and Grafana provides a general purpose dashboard for displaying the collected data. Grafana offers a highly customizable and user-friendly dashboard for monitoring purposes.

Note A user with *Administrator* role can view all the cluster deployments, but a user with *User* role can view only those clusters for which the user has permission to view.

Step 1 Install the **Monitoring** add-on in the tenant cluster.

For more information, see Configuring Add-ons for Clusters on vSphere, on page 34.

Step 2 In the left pane, click **Clusters**, and then click on the name of the tenant cluster.

Step 3 Click on the **ADD-ONS** tab.

Step 4 Under **Monitoring**, click on **DASHBOARD**.
The Grafana login page is displayed.

Step 5 Create an SSH connection to the tenant master node and follow these steps to access Grafana. Use values of `GRAFANA_USER` and `GRAFANA_PASSWORD` to login to Grafana.

```
    export GRAFANA_USER=$(kubectl get secret ccp-monitor-grafana -n ccp -o=jsonpath='{.data.admin-user}'
  | base64 --decode)

     export GRAFANA_PASSWORD=$(kubectl get secret ccp-monitor-grafana -n ccp
-o=jsonpath='{.data.admin-password}' | base64 --decode)

     echo $GRAFANA_USER
     echo $GRAFANA_PASSWORD
```

Note It is important to either change or retain the original login credentials since the secret that was used to initialize the Grafana login may be lost or changed with future upgrades.

Step 6 On the upper-left corner of the Grafana UI, click **Home**, and then click **Kubernetes cluster monitoring (via Prometheus)** to monitor the health of the tenant cluster.

Step 7 Add Prometheus as the data source and configure the Grafana dashboard to monitor the health of your cluster deployments.

Example of Monitoring Multiple Prometheus Instances

To monitor multiple Prometheus instances you must expose Prometheus as an Ingress resource so that you can access it from a Grafana instance that is running in a different cluster.

Note The following example is valid only if Harbor is not installed.

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/add-base-url: "true"
    nginx.ingress.kubernetes.io/rewrite-target: /$1
    nginx.ingress.kubernetes.io/backend-protocol: "HTTPS"
name: ccp-monitor-prometheus-server
namespace: ccp
spec:
rules:
- http:
    paths:
    - backend:
        serviceName: ccp-monitor-prometheus-server
        servicePort: 443
        path: /
```

After Promethus is accessible externally from the cluster, you can add it as a new datasource in Grafana.

Monitoring Logs from Cluster Deployments

The Elasticsearch, Fluentd, and Kibana (EFK) stack enables you to collect and monitor log data from containerized applications for troubleshooting or compliance purposes. These components are automatically installed when you install Cisco Container Platform.

Fluentd is an open source data collector. It works at the backend to collect and forward the log data to Elasticsearch.

Kibana is an open source analytics and visualization platform designed to work with Elasticsearch. It allows you to create rich visualizations and dashboards with the aggregated data.

Note A user with the *Administrator* role can view all logs, but a user with *User* role can view logs for only those clusters for which the user has permission to view.

This section contains the following topics:

Viewing EFK Logs Using Kibana (Tenant Cluster)

Before you begin

Ensure that you have installed the `kubectl` utility.

Step 1 Download the Kubeconfig file of the cluster whose logs you want to view, see Downloading Kubeconfig File, on page 91.

Step 2 Copy the contents of the downloaded Kubeconfig file to:

- Your local host `~/.kube/config`
- A local file and export KUBECONFIG=<*Downloaded Kubeconfig file*>

Step 3 Perform one of the following steps to access Kibana from outside a cluster:

- Create a port forward using kubectl to access Kibana from outside a cluster.

  a. Determine the pod.

  ```
   kubectl -n ccp get pods
  ```

  Example

  ```
  ccp-efk-kibana-6d7c97575c-9qxbf
  ```

  b. Open a port forward.

  Example

  ```
  kubectl port-forward -n ccp
   ccp-efk-kibana-6d7c97575c-9qxbf 5601:5601
  ```

  c. Access the Kibana UI and view the data from the target tenant cluster using a web browser.

  ```
  http://localhost:5601/app/kibana
  ```

- Run a bash script to create a Kibana login and access Kibana through Ingress from outside a cluster.

  a. Create ingress resource yaml for kibana.

  ```
      cat > kibana_ingress.yaml <<EOF
      apiVersion: extensions/v1beta1
      kind: Ingress
      metadata:
        annotations:
          kubernetes.io/ingress.class: nginx
          nginx.ingress.kubernetes.io/add-base-url: "false"
          nginx.ingress.kubernetes.io/auth-realm: Kibana
          nginx.ingress.kubernetes.io/auth-secret: ccp-kibana-basic-auth
          nginx.ingress.kubernetes.io/auth-type: basic
          nginx.ingress.kubernetes.io/rewrite-target: /
        name: ccp-kibana
        namespace: ccp
      spec:
        rules:
        - http:
            paths:
            - path: /kibana/?(.*)
              backend:
                serviceName: ccp-efk-kibana
                servicePort: 5601
      EOF
  ```

  b. Create a bash script.

  ```
   cat > ccp_kibana_ingress_setup.sh << EOF

   #!/bin/bash

   # Kibana username
   KIBANA_USERNAME=<USER>
   KIBANA_PASSWORD=<PASSWORD>

   # Configure Kibana server.basePath param
  ```

```
kubectl -n ccp get cm ccp-efk-kibana -o json |
jq '.["data"]["kibana.yml"] += "server.basePath: \"/kibana\"\n"' |
kubectl replace -f -

# Generating password and creating secret
KIBANA_SECRET=`htpasswd -n -b $KIBANA_USERNAME $KIBANA_PASSWORD | head -n 1`
kubectl -n ccp create secret generic ccp-kibana-basic-auth --from-literal=auth=$KIBANA_SECRET


# Creating ingress
kubectl apply -f kibana_ingress.yaml

EOF
```

c. Run the script.

```
sudo apt-get install htpasswd
chmod 700 ccp_kibana_ingress_setup.sh
./ccp_kibana_ingress_setup.sh
```

d. Restart the Kibana pod.

```
kubectl -n ccp delete $(kubectl -n ccp get pods -o name | grep ccp-efk-kibana)
```

e. Access the Kibana UI and view the data using a web browser.

```
http://<INGRESS IP>/kibana
```

For more information on customizing the Kibana UI, see the latest Kibana documentation.

Viewing EFK Logs Using Kibana (Control Plane Cluster)

Before you begin

Ensure that you have installed the `kubectl` utility.

Step 1 Access the Kubernetes cluster master node using ssh.

```
ssh ccpuser@control plane master node
sudo cat /etc/kubernetes/admin.conf
```

Step 2 Copy the contents of the downloaded Kubeconfig file to:

- Your local host `~/.kube/config`
- A local file and export KUBECONFIG=<*Full path of the Kubeconfig local file*>

For more information on setting Kubeconfig, see Configure Access to Multiple Clusters.

Step 3 Perform one of the following steps to access Kibana from outside a cluster:

- Create a port forward using kubectl to access Kibana from outside a cluster.

a. Determine the pod.

```
kubectl get pods
```

Example

```
ccp-efk-kibana-6d7c97575c-9qxbf
```

b. Open a port forward.

Example

```
kubectl port-forward ccp-efk-kibana-6d7c97575c-9qxbf 5601:5601
```

c. Access the Kibana UI and view the data from the target tenant cluster using a web browser.

```
http://localhost:5601/app/kibana
```

- Run a bash script to create a Kibana login and access Kibana through Ingress from outside a cluster.

a. Create ingress resource yaml for kibana.

```
cat > kibana_ingress.yaml <<EOF
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/add-base-url: "false"
    nginx.ingress.kubernetes.io/auth-realm: Kibana
    nginx.ingress.kubernetes.io/auth-secret: ccp-kibana-basic-auth
    nginx.ingress.kubernetes.io/auth-type: basic
    nginx.ingress.kubernetes.io/rewrite-target: /
  name: ccp-kibana
  namespace: ccp
spec:
  rules:
  - http:
      paths:
      - path: /kibana/?(.*)
        backend:
          serviceName: ccp-efk-kibana
          servicePort: 5601
EOF
```

b. Create a bash script.

```
cat > ccp_kibana_ingress_setup.sh << EOF

#!/bin/bash

# Kibana username
KIBANA_USERNAME=<USER>
KIBANA_PASSWORD=<PASSWORD>

# Configure Kibana server.basePath param
kubectl -n ccp get cm ccp-efk-kibana -o json |
jq '.["data"]["kibana.yml"] += "server.basePath: \"/kibana\"\n"' |
kubectl replace -f -

# Generating password and creating secret
KIBANA_SECRET=`htpasswd -n -b $KIBANA_USERNAME $KIBANA_PASSWORD | head -n 1`
kubectl -n ccp create secret generic ccp-kibana-basic-auth --from-literal=auth=$KIBANA_SECRET


# Creating ingress
kubectl apply -f kibana_ingress.yaml

EOF
```

c. Run the script.

```
sudo apt-get install htpasswd
chmod 700 ccp_kibana_ingress_setup.sh
./ccp_kibana_ingress_setup.sh
```

d. Restart the Kibana pod.

```
kubectl -n ccp delete $(kubectl -n ccp get pods -o name | grep ccp-efk-kibana)
```

e. Access the Kibana UI and view the data using a web browser.

```
http://<INGRESS IP>/kibana
```

For more information on customizing the Kibana UI, see the latest Kibana documentation.

Forwarding Logs to External Elasticsearch Server

Step 1 SSH to the master node of the tenant cluster.

Step 2 Edit the `ccp-efk` helmchart custom resource.

a) Open the `ccp-efk` helmchart using the visual editor.

```
kubectl edit helmchart ccp-efk -n=ccp
```

b) In `ccp-efk` helmchart custom resource, add the following lines for the external Elasticsearch server to the `spec` section.

```
set:
  localLogForwarding.enabled: "False"
  localLogForwarding.elasticsearchHost: "<_IP address of Elasticsearch server_>"
  localLogForwarding.elasticsearchPort: "<_Port number of Elasticsearch server_>"
```

c) Save the `ccp-efk` helmchart.

Step 3 Verify the custom configurations for the external Elasticsearch server in the EFK helmchart.

```
helm get values ccp-efk --namespace ccp
```

Helm-operator and fluentd pod logs also show the IP and port of the external Elasticsearch server.

Renewing Kubernetes API Certificates

By default, the Kubernetes API certificates have a validity period of one year, after which you must manually renew these certificates.

As of Kubernetes 1.17, upgrading Kubernetes will automatically renew the certificates. If you perform regular upgrades, you should not need this procedure.

To manually renew the Kubernetes API certificates for a cluster:

Step 1 SSH to a cluster master node, and check for expired (or expiring) certificates using the following command:

```
sudo kubeadm alpha certs check-expiration
```

The command shows the expiration date and time of all the certificates in the node.

Step 2 On each of the master nodes of the cluster, back up the existing certificates and configuration files:

```
sudo cp -r /etc/kubernetes/pki "$HOME/previous-certs"
```

Step 3 On each of the master nodes of the cluster, renew the certificates:

a) Renew the certificates:

```
sudo kubeadm alpha certs renew all
```

b) Verify that the cluster has new certificates:

```
sudo kubeadm alpha certs check-expiration
```

c) Generate Kubernetes configuration files:

```
    sudo su
     kubeadm alpha kubeconfig user --org system:masters --client-name kubernetes-admin >
/etc/kubernetes/admin.conf
     kubeadm alpha kubeconfig user --client-name system:kube-controller-manager >
/etc/kubernetes/controller-manager.conf
     kubeadm alpha kubeconfig user --client-name system:kube-scheduler >
/etc/kubernetes/scheduler.conf
     kubeadm alpha kubeconfig user --org system:nodes --client-name system:node:$(hostname) >
/etc/kubernetes/kubelet.conf
     exit
```

d) If there is a `/etc/kubernetes/node.conf`, update its `client-certificate-data` and `client-key-data` values to match those in `/etc/kubernetes/admin.conf`.

```
sudo vi /etc/kubernetes/node.conf
```

e) If you have a file `$HOME/.kube/config`, update its `client-certificate-data` and `client-key-data` values to match those in `/etc/kubernetes/admin.conf`.

```
cp $HOME/.kube/config $HOME/.kube/config.bak vi $HOME/.kube/config
```

f) Reboot the cluster master node.

```
sudo shutdown -r now
```

g) Verify that the master node is back in the cluster.

```
kubectl get nodes
```

Step 4 Repeat Steps 2 and 3 for each cluster master node.

CHAPTER 10

Services and Networking

This chapter contains the following topics:

Load Balancing Kubernetes Services using NGINX

Cisco Container Platform uses NGINX to offer advanced layer 7 load balancing solutions. NGINX can handle a large number of requests and at the same time, it can be run on Kubernetes containers.

The NGINX load balancer is automatically provisioned as part of Kubernetes cluster creation. Each Kubernetes cluster is provisioned with a single L7 NGINX load balancer. You can access the load balancer using its virtual IP address, which can be found by running the command `kubectl get svc -n ccp`.

To use the NGINX load balancer, you must create an Ingress resource. Ingress is a Kubernetes object that allows you to define HTTP load balancing rules to allow inbound connections to reach the cluster services. You can configure Ingress to create external URLs for services, load balance traffic, terminate SSL, offer name-based virtual hosting, and so on.

L7 Ingress

Cisco Container Platform supports the following types of L7 Ingresses:

- **Simple fanout**

  It enables you to access the website using http.

  Example

  ```
  cafe.test.com ->   10.1.1.1   ->   /tea      tea-svc:80
                                                /coffee   coffee-svc:80
  ```

  For this type of Ingress, you need to create a yaml file that defines the Ingress rules.

  Sample yaml file

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
name: cafe-ingress
spec:
rules:
-host: cafe.test.com
http:
    paths:
    -path:/
    backend:
    serviceName: tea-svc
    servicePort: 80
    -path:/
    backend:
    serviceName: tea-svc
    servicePort: 80
```

- **Simple fanout with SSL termination**

  It enables you to access the website using https.

  Example

  ```
  https://cafe.test.com   ->   10.1.1.1   ->   /tea       tea-svc:80
                                               /coffee    coffee-svc:80
  ```

  For this type of Ingress, you need to create the following yaml files:

  - A yaml file that defines the Secret

    Sample yaml file

    ```
    apiVersion: v1
    kind: Secret
    metadata:
      name: cafe-secret
    type: Opaque
    data:
      tls.crt: base64 encoded cert
      tls.key: base64 encoded key
    ```

  - A yaml file that defines the Ingress rules

    Sample yaml file

    ```
    apiVersion: extensions/v1beta1
    kind: Ingress
    metadata:
      name: cafe-ingress
    spec:
      tls:
      -hosts:
      -cafe.test.com
       secretName: cafe-secret
      rules:
      -host: cafe.example.com
      http:
       paths:
       -path:/
       backend:
         serviceName: tea-svc
         sevicePort: 80
       -path:/
       backend:
    ```

```
         serviceName: coffee-svc
         servicePort: 80
```

- **Name based virtual hosting**

  It enables you to access the website using multiple host names.

  Example

  ```
  tea.test.com    --|          |-> tea.test.com      s1:80
                    |10.1.1.1  |
  coffee.test.com --|          |-> coffee.test.com  s2:80
  ```

  For this type of Ingress, you need to create a yaml file that defines the Ingress rules.

  Sample yaml file

  ```
  apiVersion: extensions/v1beta1
  kind: Ingress
  metadata
  name: cafe-ingress
  spec:
   rules:
   -host: tea.test.com
   http:
      paths:
      -path:/
      backend:
      serviceName: tea-svc
      servicePort: 80
  -host: coffee.test.com
  http:
  paths:
  -path:/
  backend:
  serviceName: coffee-svc
  servicePort: 80
  ```

Note You can download the yaml files that are shown in this topic from the following link:

https://github.com/nginxinc/kubernetes-ingress/tree/master/examples/complete-example

For more information on a sample scenario of implementing Ingress, see Deploying Cafe Application with Ingress, on page 94.

L4 Ingress

NGINX supports L4 TCP and UDP Ingress load balancing. It uses the NGINX helm chart that contains the TCP or UDP service mappings, instead of the Ingress resources as in the case of L7 support.

Configuring L4 Load Balancing

Note NGINX supports either TCP or UDP L4 load balancing, but not both simultaneously.

Step 1 Access the Kubernetes cluster master node using ssh.

```
ssh -l <username> <IP address of master node>
```

Note Once you create a Kubernetes cluster, it may take a few minutes for the necessary services to start. If ssh to a cluster fails, we recommend that you try again after a few minutes.

Step 2 Get the current helm configuration values.

```
helm get values --all nginx-ingress > l4.yaml
```

Step 3 Edit the `l4.yaml` file.

You can search for *tcp* or *udp* in the `l4.yaml` file, and then add your L4 services.

The following example shows adding the tcp-test-svc TCP service that uses port 3333.

```
tcp:
    "9000": default/tcp-test-svc:3333
```

The following example shows adding the udp-test-svc UDP service that uses port 5005.

```
udp:
    "9001": default/udp-test-svc:5005
```

Step 4 Update the NGINX helm chart with the L4 service mappings.

```
helm upgrade --install nginx-ingress /opt/ccp/charts/nginx-ingress.tgz -f l4.yaml
```

Note You need to restart the NGINX Ingress controller pods for the new configuration to take effect.

Step 5 Verify that ingress has successfully mapped the port.

```
kubectl get services -o wide -w nginx-ingress-controller
```

Ingress CA

Cisco Container Platform by default creates an L7 Ingress service in order to support Monitoring Health of Cluster Deployments,Monitoring Logs from Cluster Deployments, and Accessing Kubernetes Clusters on vSphere. All of these services are exposed with TLS enabled, and the certificate authority (CA) that is used to sign the Ingress controller server certificate is self-signed and per cluster based.

In order to reach the services without triggering SSL warning, you can either add the CA as part of your application that needs to interact with services behind Cisco Container Platform ingress (preferred), or add the CA to your system trusted CA list.

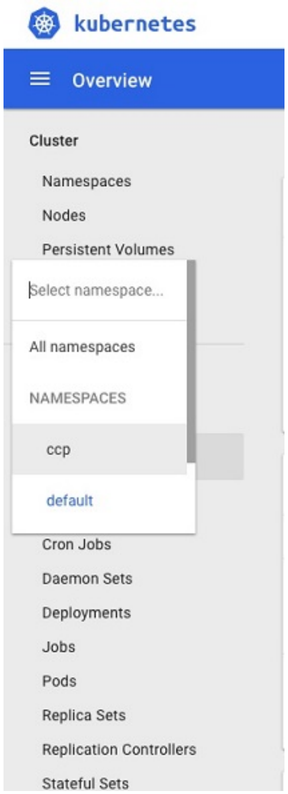
Obtaining CA Certificate for Nginx Ingress Controller

To obtain a CA certificate.

Step 1 Log in to the Kubernetes dashboard from browser as described in Accessing Kubernetes Clusters on vSphere section, download the kubeconfig file, and then use it to login to the Kubernetes dashboard.

Step 2 From the right pane, click the dropdown box under **Namespace**, click the **ccp** namespace.

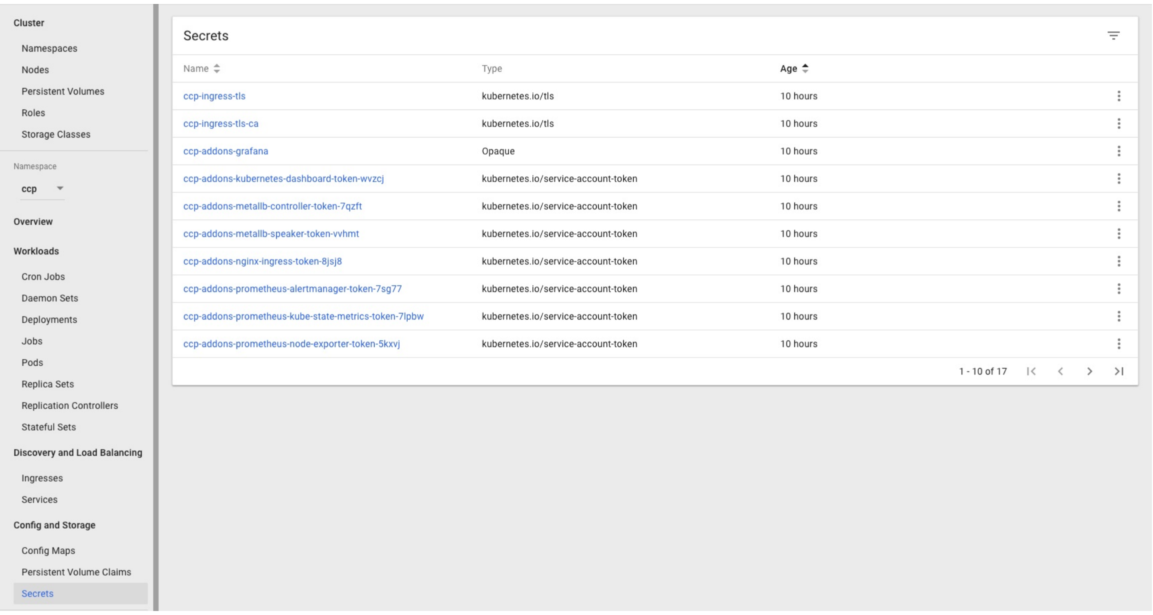
Figure 4: Kubernetes Dashboard



Step 3 Click the **Secrets** tab.

The **Secrets** pane appears.

Figure 5: Secrets Pane



Secrets

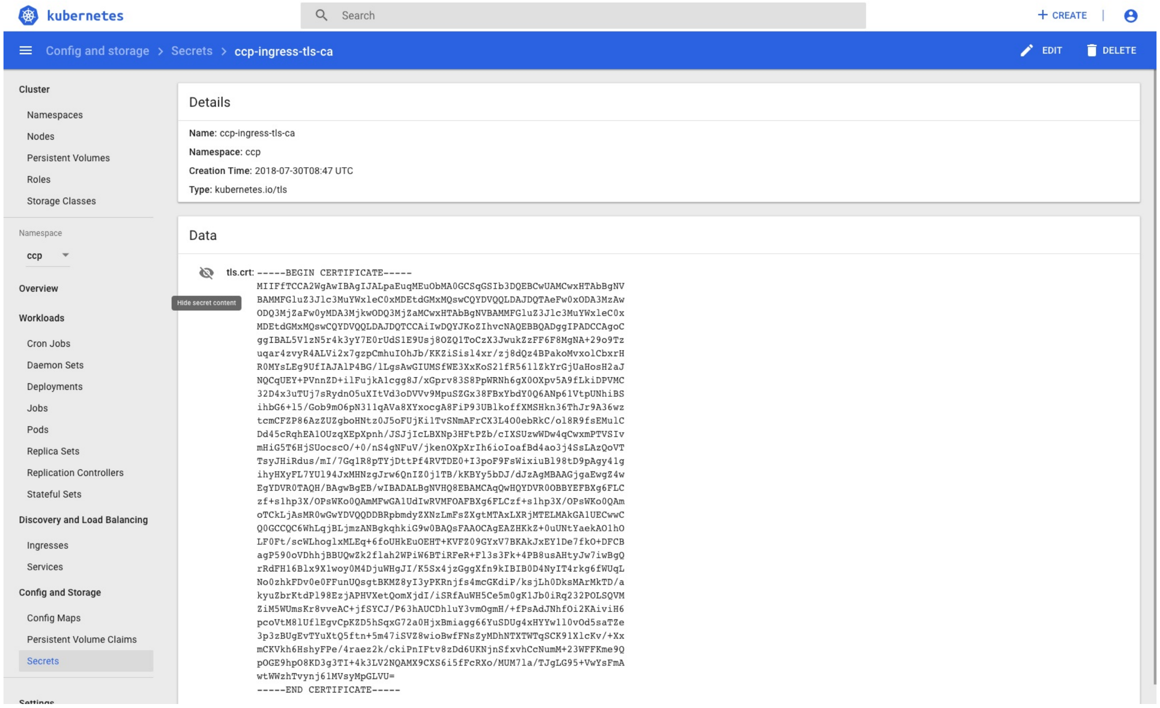
Name	Type	Age
ccp-ingress-tls	kubernetes.io/tls	10 hours
ccp-ingress-tls-ca	kubernetes.io/tls	10 hours
ccp-addons-grafana	Opaque	10 hours
ccp-addons-kubernetes-dashboard-token-wvzcj	kubernetes.io/service-account-token	10 hours
ccp-addons-metallb-controller-token-7qzft	kubernetes.io/service-account-token	10 hours
ccp-addons-metallb-speaker-token-vvhmt	kubernetes.io/service-account-token	10 hours
ccp-addons-nginx-ingress-token-8jsj8	kubernetes.io/service-account-token	10 hours
ccp-addons-prometheus-alertmanager-token-7sg77	kubernetes.io/service-account-token	10 hours
ccp-addons-prometheus-kube-state-metrics-token-7lpbw	kubernetes.io/service-account-token	10 hours
ccp-addons-prometheus-node-exporter-token-5kxvj	kubernetes.io/service-account-token	10 hours

1 - 10 of 17

Step 4 Open the `ccp-ingress-tls-ca` secret and find the data for `tls.crt`.

Step 5 Click the **Eye** icon to view the details of a `tls.crt`.

Figure 6: Secrets Pane Showing Details of tls.crt



Details

Name: ccp-ingress-tls-ca

Namespace: ccp

Creation Time: 2018-07-30T08:47 UTC

Type: kubernetes.io/tls

Data

tls.crt:

```
-----BEGIN CERTIFICATE-----
MIIFfTCCA2WgAwIBAgIJALpaEuqMEuObMA0GCSqGSIb3DQEBCwUAMCwxHTAbBgNV
BAMMFGluZ3Jlc3MuYWxleC0xMDEtdGMxMQswCQYDVQQLDAJDQTAeFw0xODA3MzAw
ODQ3MjZaFw0yMDA3MjkwODQ3MjZaMCwxHTAbBgNVBAMMFGluZ3Jlc3MuYWxleC0x
MDEtdGMxMQswCQYDVQQLDAJDQTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoC
ggIBAL5V1zN5r4k3yY7E0rUdS1E9Usj8OZQ1ToCzX3JwukZzFF6F8MgNA+29o9Tz
uqar4zvyR4ALVi2x7gzpCmhuIOhJb/KKZiSisl4xr/zj8dQz4BPakoMvxolCbxrH
R0MYsLEg9UfIAJAlP4BG/lLgsAwGIUMSfWE3XxKoS21fR561lZkYrGjUaHosH2aJ
NQCqUEY+PVnnZD+ilFujkA1cgg8J/xGprv83S8PpWRNh6gX0OXpv5A9fLkiDPVMC
32D4x3uTUj7sRydnO5uXItVd3oDVVv9MpuSZGx38FBxYbdY0Q6ANp61VtpUNhiBS
ihbG6+l5/Gob9mO6pN311qAVa8XYxocgA8FiP93UBlkoffXMSHkn36ThJr9A36wz
tcmCFZP86AzZUZgboHNtz0J5oFUjKi1TvSNmAFrCX3L4O0ebRkC/ol8R9fsEMulC
Dd45cRqhEA1OUzqXEpXpnh/JSJjIcLBXNp3HFtPZb/cIXSUzwWDw4qCwxmPTVSIv
mHiG5T6HjSUocscO/+0/nS4gNFuV/jkenOXpXrIh6ioIoafBd4ao3j4SsLAzQoVT
TsyJHiRdus/mI/7Gq1R8pTYjDttPf4RVTDE0+I3poF9FsWixiuBl98tD9pAgy41g
ihyHXyFL7YUl94JxMHNzgJrw6QnIZ0j1TB/kKBYy5bDJ/dJzAgMBAAGjgaEwgZ4w
EgYDVR0TAQH/BAgwBgEB/wIBADALBgNVHQ8EBAMCAqQwHQYDVR0OBBYEFBXg6FLC
zf+s1hp3X/OPsWKo0QAmMFwGA1UdIwRVMFOAFBXg6FLCzf+s1hp3X/OPsWKo0QAm
oTCkLjAsMR0wGwYDVQQDDBRpbmdyZXNzLmFsZXgtMTAxLXRjMTELMAkGA1UECwwC
Q0GCCQC6WhLqjBLjmzANBgkqhkiG9w0BAQsFAAOCAgEAZHKkZ+0uUNtYaekAO1hO
LF0Ft/scWLhoglxMLEq+6foUHkEuOEHT+KVFZ09GYxV7BKAkJxEY1De7fkO+DFCB
agP590oVDhhjBBUQwZk2flah2WPiW6BTiRFeR+Fl3s3Fk+4PB8usAHtyJw7iwBgQ
rRdFH16Blx9X1woy0M4DjuWHgJI/K5Sx4jzGggXfn9kIBIB0D4NyIT4rkg6fWUqL
No0zhkFDv0e0FFunUQsgtBKMZ8yI3yPKRnjfs4mcGKdiP/ksjLh0DksMArMkTD/a
kyuZbrKtdPl98EzjAPHVXetQomXjdI/iSRfAuWH5Ce5m0gK1Jb0iRq232POLSQVM
ZiM5WUmsKr8vveAC+jfSYCJ/P63hAUCDhluY3vmOgmH/+fPsAdJNhfOi2KAiviH6
pcoVtM8lUflEgvCpKZD5hSqxG72a0HjxBmiagg66YuSDUg4xHYYw1l0vOd5saTZe
3p3zBUgEvTYuXtQ5ftn+5m47iSVZ8wioBwfFNsZyMDhNTXTWTqSCK91XlcKv/+Xx
mCKVkh6HshyFPe/4raez2k/ckiPnIFtv8zDd6UKNjnSfxvhCcNumM+23WFFKme9Q
pOGE9hpO8KD3g3TI+4k3LV2NQAMX9CXS6i5fFcRXo/MUM7la/TJgLG95+VwYsFmA
wtWWzhTvynj61MVsyMpGLVU=
-----END CERTIFICATE-----
```

You can save the CA data into a file, and use it when a client is trying to connect to the Ingress service.

The following example uses **curl** to get to the dashboard using the saved CA certificate.

```
curl --cacert ./ca.crt -I https://10.10.99.185/dashboard
 HTTP/1.1 200 OK
 Server: nginx/1.13.12
 Date: Mon, 30 Jul 2018 19:08:11 GMT
 Content-Type: text/html; charset=utf-8
 Connection: keep-alive
 Vary: Accept-Encoding
 Accept-Ranges: bytes
 Cache-Control: no-store
 Strict-Transport-Security: max-age=15724800; includeSubDomains
```

Updating CA Certificates for Nginx Ingress Controller

To update certificates configured for the Nginx Ingress controller with custom certificates that are not provisioned using Cisco Container Platform:

Step 1 On the control plane node of your Cisco Container Platform instance, backup the existing daemonset ingress-nginx-controller resources.

```
kubectl get daemonset ingress-nginx-controller -o yaml > ds_ingress_nginx_controller_ccp.yaml
```

Step 2 Create or copy the TLS key file and TLS crt file of the new certificate to a known location on the control plane node.

Step 3 Create a new TLS secret in the default namespace configured for the new custom certificates.

```
kubectl create secret tls custom-certificate-secret --key=/path/to/tls.key --cert=/path/to/tls.crt
```

Step 4 Update the existing daemonset to use the new certificates.

```
     kubectl patch daemonset ingress-nginx-controller --type='json'
-p='[{"op":"replace","path":"/spec/template/spec/containers/0/args/6","value":"—default-ssl-certificate=default/custom-certificate-secret"}]'
```

Step 5 Check the status of the daemonset.

```
   ccpuser@ccp800-master80bcc3ccdc:~$ kubectl get ds ingress-nginx-controller

   NAME                       DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR
AGE

   ingress-nginx-controller   3         3         3       3            3           <none>
73d
```

Note Ensure that the **UP-TO-DATE** count and **READY** count are equal.

Step 6 Verify the certificates used by the Cisco Container Platform dashboard in one of the following ways:

- Using the browser.

  This depends on the type of browser you are using. Check the browser settings and instructions.

- Using the CLI.

  Run the following curl commands to verify if the new custom certificates have been installed for your Cisco Container Platform instance.

  ```
  $ curl --insecure -vvI https://<master_vip_address>
  * Rebuilt URL to: https://10.10.96.6/
  ...
  ```

```
        * Server certificate:
        *  subject: CN=ingress.ccp800; OU=server
        *  start date: Jan 15 20:27:17 2021 GMT
        *  expire date: Jan 15 20:27:17 2023 GMT
        *  issuer: CN=ingress.ccp800; OU=CA
        *  SSL certificate verify result: unable to get local issuer certificate (20), continuing
anyway.
        * Using HTTP2, server supports multi-use
        * Connection state changed (HTTP/2 confirmed)
        * Copying HTTP/2 data in stream buffer to connection buffer after upgrade: len=0
        ...
        * Connection #0 to host 10.10.96.6 left intact
```

Reverting CA Certificates for Nginx Ingress Controller

To revert to the original certificate settings that you configured when you installed Cisco Container Platform:

Step 1 When Cisco Container Platform is installed, it configures an ingress-nginx-controller with self-signed certificates that is stored in the `default/ccp-ingress-tls` Kubernetes secret. Ensure that this secret exists on the control plane.

```
     kubectl get secret ccp-ingress-tls
```

Step 2 Update the daemonset ingress-nginx-controller with this secret.

```
     kubectl patch daemonset ingress-nginx-controller --type='json'
-p='[{"op":"replace","path":"/spec/template/spec/containers/0/args/6","value":"--default-ssl-certificate=default/ccp-ingress-tls"}]'
```

Step 3 Wait for the daemonset to be ready.

```
     ccpuser@ccp800-master80bcc3ccdc:~$ kubectl get ds ingress-nginx-controller

     NAME                          DESIRED    CURRENT    READY    UP-TO-DATE    AVAILABLE    NODE SELECTOR
  AGE

     ingress-nginx-controller   3          3          3        3             3            <none>
  73d
```

Note Ensure that the **UP-TO-DATE** count and **READY** count are equal.

The certificate configuration will now be restored to the original certificate settings that you configured when you installed Cisco Container Platform.

Step 4 Verify the certificates used by the Cisco Container Platform dashboard in one of the following ways:

- Using the browser.

  This depends on the type of browser you are using. Check the browser settings and instructions

- Using the CLI.

  Run the following curl commands to verify if the new custom certificates have been installed for your Cisco Container Platform instance.

  ```
  $ curl --insecure -vvI https://<master_vip_address>
  * Rebuilt URL to: https://10.10.96.6/
  ...
  * Server certificate:
  *  subject: CN=ingress.ccp800; OU=server
  ```

```
     *  start date: Jan 15 20:27:17 2021 GMT
     *  expire date: Jan 15 20:27:17 2023 GMT
     *  issuer: CN=ingress.ccp800; OU=CA
     *  SSL certificate verify result: unable to get local issuer certificate (20), continuing
anyway.
     * Using HTTP2, server supports multi-use
     * Connection state changed (HTTP/2 confirmed)
     * Copying HTTP/2 data in stream buffer to connection buffer after upgrade: len=0
     ...
     * Connection #0 to host 10.10.96.6 left intact
```

Network Policies

Cisco Container Platform supports Kubernetes NetworkPolicies. The NetworkPolicies are independent of the underlying container network plugin.

Load Balancer Services

Cisco Container Platform supports load balancer services on tenant clusters.

While creating a tenant cluster, you need to choose the number of load balancer IP addresses that you want to allocate for a tenant cluster from a VIP pool that you want to use.

Note The cluster creation operation fails if the number of requested load balancer IP addresses is more than the available IP addresses in the pool.

For more information, see Creating Clusters on vSphere, on page 31.

Once load balancer IP addresses are allocated for a tenant cluster, externally reachable load balancer IP addresses are automatically provisioned for the load balancer services.

The following code provides an example of creating a service of type **LoadBalancer**.

```
apiVersion: v1
kind: Service
metadata:
     name: frontend
     labels:
             app: guestbook
             tier: frontend
     type: LoadBalancer
```

You can update the number of available load balancer IP addresses from the **Edit Cluster** screen. You need to be aware of the number of used addresses in order to update the number of allocated load balancer IP addresses.

For example:

Suppose the current tenant is allocated with five load balancer IP addresses. If there are three load balanced services running, you cannot reduce the number of load balancer IP addresses to three or less as there are services using those IP addresses already.

Note When you delete a tenant cluster, the allocated load balancer IP addresses are recycled to the VIP pool.

CHAPTER 11

Istio Service Mesh

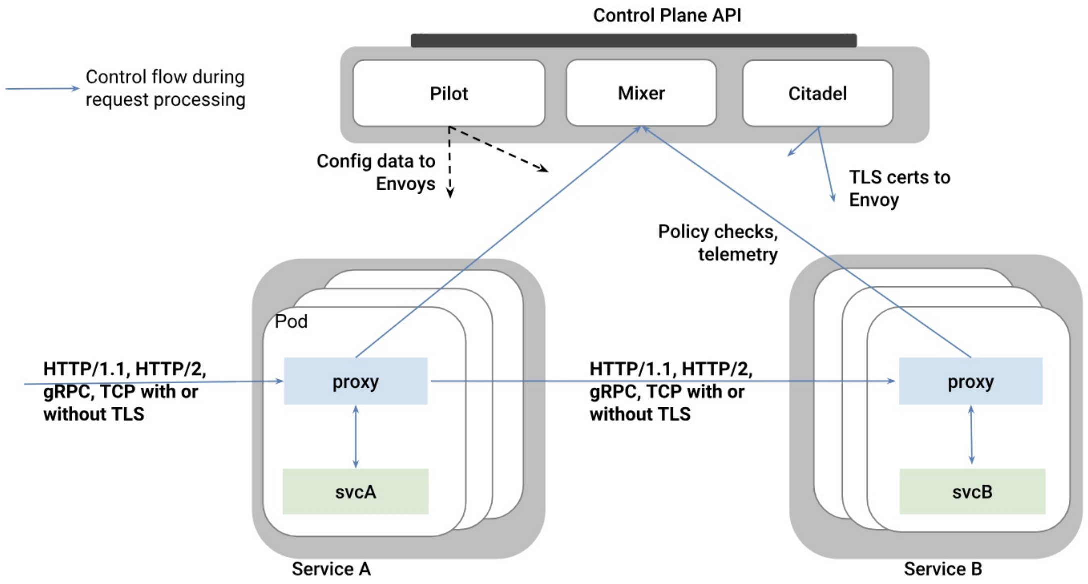
This chapter contains the following topics:

Introduction to Istio Service mesh

Cisco Container Platform includes support for an Istio service mesh. An Istio service mesh is logically split into a Data Plane and a Control Plane. The Data Plane includes a set of intelligent proxies (Envoy) and the Control Plane provides a reliable Istio framework. The term Istio is sometimes also used as a synonym to refer to the entire service mesh stack that includes the Control Plane and the Data Plane components.

The service mesh technology allows you to construct North-South and East-West L4 and L7 application traffic mesh. It provides containerized applications a language-independent framework that removes several common tasks related to L4 and L7 application networking from the actual application code. The common tasks include L4 and L7 service routing and load balancing, support for polyglot environments in a language-independent manner and advanced telemetry. The service mesh technology enhances operational capabilities such as monitoring, security, load balancing and troubleshooting for the applications. You can deploy a service mesh in a multi-cloud topology allowing these functions to operate with applications that run across multiple independent cloud deployments.

The following figure shows the high-level architecture of an Istio service mesh.



Istio Architecture

In Cisco Container Platform, the components of Istio and Envoy are supported in the upstream Istio community. The Control and Data Plane components of the solution, such as Pilot, Mixer, Citadel and the Data Plane Envoy proxy for both North-South and East-West load balancing, are supported on Cisco Container Platform.

For more information on these technologies, see the upstream community documentation pages for Istio and Envoy.

Note Currently, the Istio service mesh feature is marked as a Tech Preview feature and uses the Istio community version v1.3.6. You need to contact your service representative for support on the version of Cisco Container Platform you have deployed.

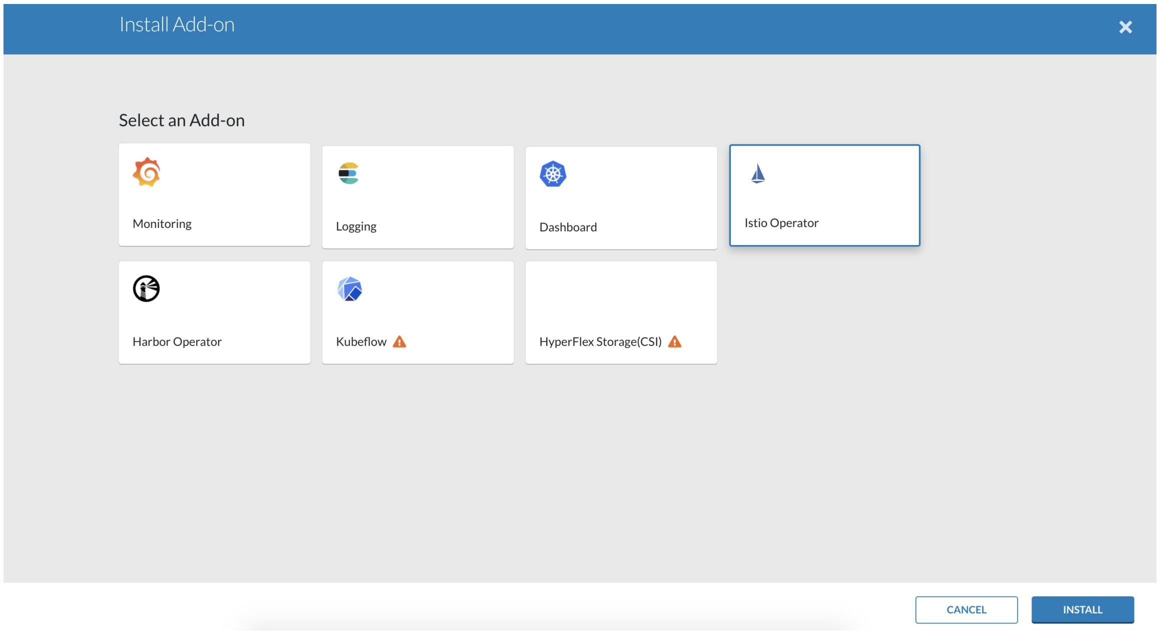
Configuring Istio Service mesh

An Istio service mesh is a configurable feature on Cisco Container Platform. You can configure a separate instance of the service mesh stack on each tenant cluster. Support for Istio must be configured at the time of creating a tenant Kubernetes cluster. You can perform this configuration using APIs or the Cisco Container Platform web interface.

Each instance of the Istio service mesh uses an IP address from the Virtual IP address pool that is associated with the tenant cluster. Consequently, you need to ensure that there is sufficient number of IP addresses free and available in the VIP pool before enabling Istio. Typically, at least three IP addresses are required, one each for the Kubernetes API, Kubernetes Ingress, and Istio Ingress gateway. This number may change in future when additional features require more virtual IP addresses.

For more information on the required number of virtual IP addresses for a given software version of Cisco Container Platform, see the Virtual IP address section.

The following figure shows the **ADD-ONS** tab for a v3 tenant cluster, using which you can enable the Istio Operator and Istio service mesh on a tenant cluster of the Cisco Container Platform.



In the current version of Cisco Container Platform, you can use a boolean flag to enable an Istio service mesh in a tenant Kubernetes cluster of Cisco Container Platform. If you enable the flag, a predetermined configuration of an Istio-based service mesh with Envoy as the Data Plane is configured in the tenant Kubernetes cluster. An internal instance of a service load balancer is automatically configured and a virtual IP address is automatically allocated for the Ingress gateway function of Istio.

Monitoring Service mesh

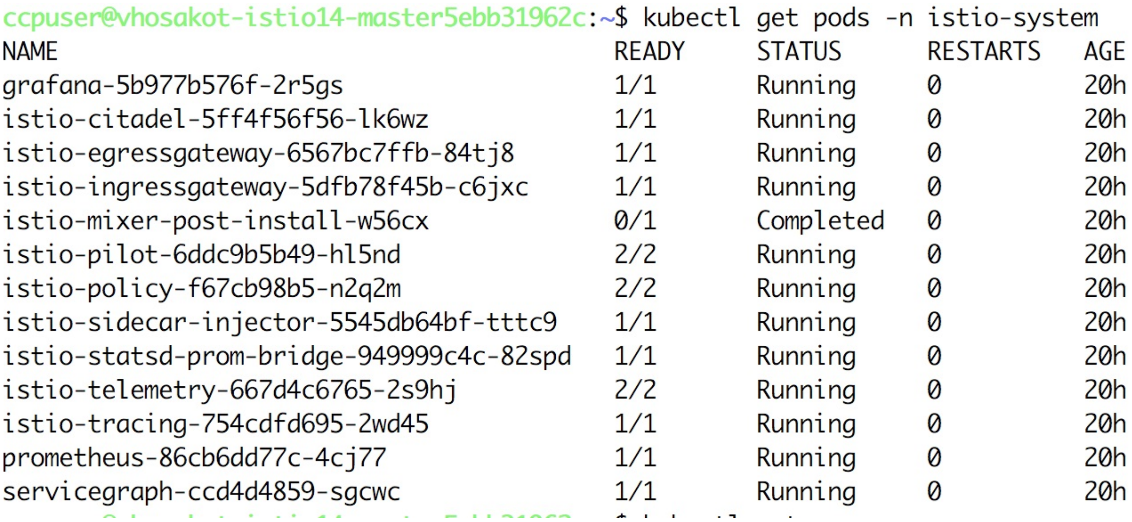
On Cisco Container Platform, the Istio Control Plane is deployed in a special **istio-system** namespace of a tenant Kubernetes cluster. This is similar to how other add-on services such as Prometheus based monitoring or NGINX based Kubernetes ingress are provided. In a production deployment, a tenant Kubernetes cluster administrator grants read-write access to your development namespaces but not to the namespaces of system add-on services such as Istio, thereby protecting the Control Plane of such services from getting over-written accidentally or maliciously by your application containers.

The following is a checklist of monitoring and troubleshooting steps when using Istio on Cisco Container Platform:

1. If Istio fails to be enabled on your tenant Kubernetes cluster, in addition to the usual troubleshooting steps for Cisco Container Platform, also ensure that there is a sufficient number of virtual IP addresses available in the pool configured for this Kubernetes tenant cluster. In the current version of Cisco Container Platform, at least three IP addresses need to be free and available for a tenant Kubernetes cluster that has Istio enabled.

2. Confirm that all pods are running in the istio-system namespace of the tenant Kubernetes cluster. The following figure shows a sample CLI output indicating that all Istio control pods are running correctly in

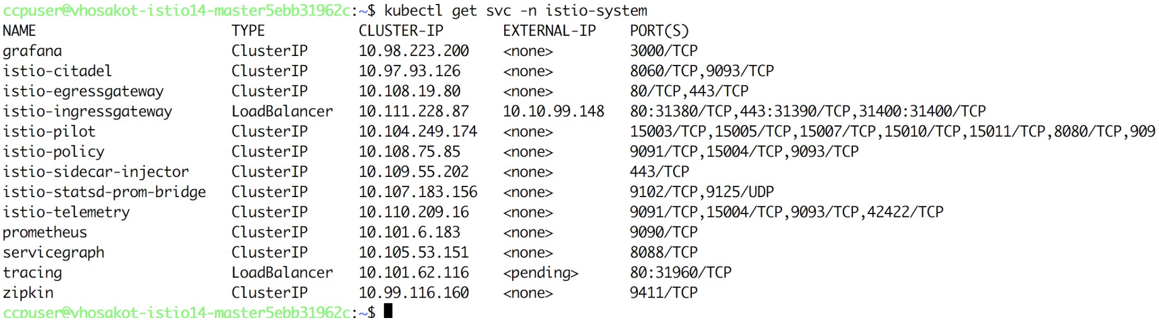
a tenant Kubernetes cluster. If one or more pods continuously fails to run, use **kubectl describe pod <name_of_pod>** to troubleshoot the issue.



```
ccpuser@vhosakot-istio14-master5ebb31962c:~$ kubectl get pods -n istio-system
NAME                                        READY   STATUS      RESTARTS   AGE
grafana-5b977b576f-2r5gs                    1/1     Running     0          20h
istio-citadel-5ff4f56f56-lk6wz              1/1     Running     0          20h
istio-egressgateway-6567bc7ffb-84tj8        1/1     Running     0          20h
istio-ingressgateway-5dfb78f45b-c6jxc       1/1     Running     0          20h
istio-mixer-post-install-w56cx              0/1     Completed   0          20h
istio-pilot-6ddc9b5b49-hl5nd                2/2     Running     0          20h
istio-policy-f67cb98b5-n2q2m                2/2     Running     0          20h
istio-sidecar-injector-5545db64bf-tttc9     1/1     Running     0          20h
istio-statsd-prom-bridge-949999c4c-82spd    1/1     Running     0          20h
istio-telemetry-667d4c6765-2s9hj            2/2     Running     0          20h
istio-tracing-754cdfd695-2wd45              1/1     Running     0          20h
prometheus-86cb6dd77c-4cj77                 1/1     Running     0          20h
servicegraph-ccd4d4859-sgcwc                1/1     Running     0          20h
```

3. Confirm that all Istio services are running in the **istio-system** namespace of the tenant Kubernetes cluster.

   The following figure shows a CLI output with the Istio services up and running.



```
ccpuser@vhosakot-istio14-master5ebb31962c:~$ kubectl get svc -n istio-system
NAME                       TYPE           CLUSTER-IP       EXTERNAL-IP    PORT(S)
grafana                    ClusterIP      10.98.223.200    <none>         3000/TCP
istio-citadel              ClusterIP      10.97.93.126     <none>         8060/TCP,9093/TCP
istio-egressgateway        ClusterIP      10.108.19.80     <none>         80/TCP,443/TCP
istio-ingressgateway       LoadBalancer   10.111.228.87    10.10.99.148   80:31380/TCP,443:31390/TCP,31400:31400/TCP
istio-pilot                ClusterIP      10.104.249.174   <none>         15003/TCP,15005/TCP,15007/TCP,15010/TCP,15011/TCP,8080/TCP,909
istio-policy               ClusterIP      10.108.75.85     <none>         9091/TCP,15004/TCP,9093/TCP
istio-sidecar-injector     ClusterIP      10.109.55.202    <none>         443/TCP
istio-statsd-prom-bridge   ClusterIP      10.107.183.156   <none>         9102/TCP,9125/UDP
istio-telemetry            ClusterIP      10.110.209.16    <none>         9091/TCP,15004/TCP,9093/TCP,42422/TCP
prometheus                 ClusterIP      10.101.6.183     <none>         9090/TCP
servicegraph               ClusterIP      10.105.53.151    <none>         8088/TCP
tracing                    LoadBalancer   10.101.62.116    <pending>      80:31960/TCP
zipkin                     ClusterIP      10.99.116.160    <none>         9411/TCP
ccpuser@vhosakot-istio14-master5ebb31962c:~$ ▌
```

4. Confirm that the Ingress gateway service has an external IP address allocated and that this IP address is one of the previously available IP addresses in the virtual IP address pool associated with this tenant Kubernetes cluster. An example of this CLI output is shown in the preceding figure.

5. Deploy the bookinfo example application provided in the Istio upstream community web site.

6. The **istioctl** CLI utility is not deployed in the current version of the Cisco Container Platform. Most of the Istio functionality is now available through the **kubectl** CLI, but if you want to use **istioctl**, run these steps to deploy **istioctl** on a tenant Kubernetes cluster of the Cisco Container Platform:

```
export ISTIO_VERSION=1.0
    curl -L https://git.io/getLatestIstio | sh -
    chmod +x istio-${ISTIO_VERSION}/bin/istioctl
    sudo mv istio-${ISTIO_VERSION}/bin/istioctl /usr/local/bin/
    istioctl version
```

For more information and operational guidelines, see Istio upstream documentation.

CHAPTER 12

Harbor Registry

Using a Harbor registry, you can host container images in a local, private Docker registry. Harbor is an extension of the basic Docker registry that implements access controls, identity management, and a graphical interface. Using imagePullSecrets, Kubernetes resources can connect to a Harbor Registry to retrieve container images on other systems.

This chapter contains the following topic:

Using Harbor Registry in Tenant Clusters

Follow these steps to create a new tenant cluster with access to the Harbor registry:

Step 1 Obtain the Ingress Root CA Certificate from the Kubernetes UI in one of the following ways:

- Follow the steps in the Ingress CA, on page 76 section.
- Run the following command on the tenant cluster where Harbor registry is installed.

```
kubectl get secrets -n ccp ccp-ingress-default-cert -o jsonpath='{.data.tls\.crt}' | base64
--decode
```

You can view the Harbor endpoint at `https://<LOAD_BALANCER_IP>:443` of the cluster where it is installed.

Step 2 Create a new tenant cluster.

For more information, see Creating Clusters on vSphere, on page 31.

Step 3 In the **Node Configuration** screen, copy and paste the Root CA certificate obtained in Step 1.

Adding CA certificates to the Root CA is the only supported method of enabling secure registries in Cisco Container Platform tenant clusters.

Note Do not enable **Harbor** in the **Harbor Registry** screen.

Step 4 Get the admin password of Harbor registry.

The admin password of Harbor registry is randomly generated by the Harbor operator and stored in a Kubernetes secret.

Run the following command on the master node of the tenant cluster to get the admin password of the Harbor registry from the Kubernetes secret.

```
$ kubectl get secret ccp-harbor-cr -n=ccp -o jsonpath='{.data.admin-password}' | base64 -d
```

Note We recommend that you change the admin password of the harbor registry after logging into the harbor GUI the first time.

Step 5 After tenant cluster creation, create an SSH connection to one of the VMs in the cluster and login to the Harbor registry with the admin password from Step 4.

```
docker login -u admin -p *****
 https://<LOAD_BALANCER_IP>:443
```

Using Harbor Chartmuseum in Tenant Clusters

You can configure tenant clusters to pull and push helm charts from and to a Harbor chartmuseum on a tenant.

Step 1 Add the helm repository to the tenant that you want to use for accessing the Chartmuseum:

a) Obtain the Ingress Root CA Certificate (ca-file), TLS Certificate (cert-file), and TLS Key (key-file), from the Kubernetes dashboard in one of the following ways:

- Use the steps in the Ingress CA, on page 76 section.
- Alternatively, on the tenant cluster where Harbor registry is installed, run the following commands:

```
  kubectl get secrets -n ccp ccp-ingress-default-cert -o jsonpath='{.data.ca\.crt}' | base64
--decode
  kubectl get secrets -n ccp ccp-ingress-default-cert -o jsonpath='{.data.tls\.crt}' | base64
--decode
  kubectl get secrets -n ccp ccp-ingress-default-cert -o jsonpath='{.data.tls\.key}' | base64
--decode
```

b) Save the ca-file, cert-file, and key-file files on your computer.

c) Add the helm repository.

```
 helm repo add --ca-file <ca file> --cert-file <cert file> --key-file <key file> --username
<username> --password <password> <repo name>
https://<LOAD_BALANCER_IP_OF_HARBOR_TENANT>/chartrepo/<library_name>
```

Note

- For `ca-file`, `cert-file`, and `key-file`, use the output from Step 1.
- For `username` and `password`, use the same credentials that you use to access the Harbor registry.
- For `repo-name`, use the name of the helm repository that you have chosen.
- For `library_name`, use the name of the project as displayed on the Harbor UI. The default project name is `library`.

d) Verify if the helm repository is added successfully.

```
helm repo list
```

Step 2 If you want to use the CLI to upload or download helm charts, follow these steps:

a) To push a helm chart to the Chartmuseum helm repository:

Run the following command on the tenant cluster from where you want to upload the helm chart.

```
     helm push <chart name>.tgz --ca-file <ca file> --cert-file <cert file> --key-file <key file>
 --username <username> --password <password> <repo name>
```

b) To pull a helm chart from the Chartmuseum helm repository:

Run the following command on the tenant cluster from where you want to download and install the helm chart:

```
      helm repo update
     helm install <helm-release-name> --ca-file <ca file> --cert-file <cert file> --key-file <key
 file> --username=<username> --password=<password> --version <chart version> <repo name>/<chart
name>
```

Step 3 If you want to use the Harbor endpoint to upload, view, or download helm charts, follow these steps:

a) In the left pane of the Harbor UI, click **Projects**, and then click on your project name.

Note The default project name is **Library**.

b) Click the **Helm Charts** tab.

c) To push the helm chart to the Harbor endpoint, click **UPLOAD**.

d) To view the available helm charts, click an existing chart name, and then click on the version of the chart that you want to view.

e) To pull the helm chart from a Harbor endpoint, click **DOWNLOAD**.

You can scroll down to the end of the page for instructions on *helm add* and *helm install*.

CHAPTER 13

Deploying Applications on Kubernetes Clusters

Once you have created Kubernetes cluster using the Cisco Container Platform web interface, you can deploy containerized applications on top of it.

This chapter contains the following topics:

Workflow of Deploying Applications

Task	Related Section
Create Kubernetes clusters using the Cisco Container Platform web interface.	Creating Clusters on vSphere, on page 31
Download the kubeconfig file that contains the cluster information and the certificates required to access clusters.	Downloading Kubeconfig File, on page 91
Use the kubectl utility to deploy the application and test the scenario.	Sample Scenarios, on page 92

Downloading Kubeconfig File

You must download the cluster environment to access the Kubernetes clusters using command line tools such as `kubectl` or using APIs.

Step 1 In the left pane, click **Clusters**.

Step 2 Click the **Download** icon corresponding to the cluster environment that you want to download.

The `kubeconfig` file that contains the cluster information and the certificates required to access clusters is downloaded to your local system.

Sample Scenarios

This topic contains a few sample scenarios of deploying applications.

Deploying a Pod with Persistent Volume

Tenant clusters are deployed with a default storage class named **standard**, and a default storage class provider named **vSphere provider**.

If you select a HyperFlex local network during cluster creation, HyperFlex storage class and storage class provisioner are created by default.

In Cisco container Platform 4.0+, when deployed with Hyperflex 4.0+, the following two HyperFlex provisioners are supported:

- **hyperflex**, the HyperFlex FlexVolume provisioner available with HyperFlex 3.5+
- **hyperflex-csi**, the HyperFlex Container Storage Interface (CSI) provisioner available with HyperFlex 4.0+

Note We recommended that you use the HyperFlex Container Storage Interface (CSI) plugin, which is a more robust framework.

Step 1 Configure a tenant Kubernetes cluster.

```
export KUBECONFIG=<Path to kubeconfig file>
```

Step 2 Verify if the storage cluster is created.

```
kubectl describe storageclass standard

  Name:                   standard
  IsDefaultClass:         Yes
  Provisioner:            kubernetes.io/vsphere-volume
  Parameters:             diskformat=thin
  ReclaimPolicy:          Delete
  VolumeBindingMode:      Immediate
```

On HyperFlex 4.0+, if you have selected a HyperFlex local network, additional storage classes are displayed when you run the following command:

```
kubectl get sc

  NAME                 PROVISIONER                    AGE
  hyperflex            hyperflex.io/hxvolume          22h
  hyperflex-csi        csi-hxcsi                      22h
  standard (default)   kubernetes.io/vsphere-volume   22h
```

Step 3 Create the persistent volume claim to request for storage.

```
cat <<EOF > pvc.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pv-claim
spec:
  storageClassName: standard
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
EOF
```

Note The `storageClassName` field is optional. For HyperFlex 4.0+, you must use **hyperflex-csi** as the storage class.

```
kubectl create -f pvc.yaml
persistentvolumeclaim "pv-claim" created
```

Note The HyperFlex storage class supports the ReadWriteOnce or ReadOnlyMany access modes and the vSphere storage class supports the ReadWriteOnce access mode.

Step 4 Verify if the persistent volume claim (pvc) is created.

```
kubectl describe pvc pv-claim
Name:          pv-claim
Namespace:     default
StorageClass:  standard
Status:        Bound
Volume:        hx-default-pv-claim-5c4e8978-cdd2-11e8-9a07-005056b8fd7b
Labels:
Annotations:   pv.kubernetes.io/bind-completed=yes
               pv.kubernetes.io/bound-by-controller=yes
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      3Gi
Access Modes:  RWO,ROX
Events:        \
```

Persistent Volume is automatically created and is bounded to this pvc.

Note When **VSPHERE** is used as the default storage class, a VMDK file is created inside the **kubevols** folder in the datastore which is specified during the creation of the tenant Kubernetes cluster.

Step 5 Create a pod that uses persistent volume claim with storage class.

```
cat <<EOF > pvc-pod.yaml
kind: Pod
apiVersion: v1
metadata:
  name: pvc-pod
spec:
  volumes:
    - name: pvc-storage
      persistentVolumeClaim:
       claimName: pv-claim
  containers:
    - name: pvc-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/usr/share/nginx/html"
          name: pvc-storage
EOF

kubectl create -f pvc-pod.yaml
pod "pvc-pod" created
```

Step 6 Verify if the pod is up and running.

```
kubectl get pod pvc-pod

NAME       READY      STATUS     RESTARTS    AGE
pvc-pod    1/1        Running    0           16s
```

When **VSPHERE** is used as the default storage class, you can access vCenter and view the dynamically provisioned VMDKs of the pod.

Deploying Cafe Application with Ingress

This scenario shows how to deploy the NGINX or NGINX Plus Ingress controller, the Cafe application, and then configure load balancing for the Cafe application using the Ingress resource.

For more information on Ingress, see Load Balancing Kubernetes Services using NGINX, on page 73.

Step 1 Download the required artifacts.

a) Go to the following URL:

https://github.com/nginxinc/kubernetes-ingress/tree/master/examples/complete-example

b) Download the following yaml files:

- cafe-ingress.yaml
- cafe-secret.yaml
- cafe.yaml

Step 2 Deploy the Ingress controller.

a) Save the public IP address of the Ingress controller in a shell variable.

```
$ IC_IP=XXX.YYY.ZZZ.III
```

b) Save the HTTPS port of the Ingress controller in a shell variable.

```
$ IC_HTTPS_PORT=<port number>
```

Step 3 Deploy the Cafe application by creating the coffee and the tea deployments and services.

```
$ kubectl create -f cafe.yaml
```

Step 4 Configure load balancing.

a) Create a secret with an SSL certificate and a key.

```
$ kubectl create -f cafe-secret.yaml
```

b) Create an Ingress resource.

```
$ kubectl create -f cafe-ingress.yaml
```

Step 5 Test the application.

a) To access the application, curl the coffee and the tea services.

Note You can use the curl `--insecure` option to turn off certificate verification of the self-signed certificate and the `--resolve` option to set the Host header of a request with cafe.example.com.

- Accessing the coffee application

```
  $ curl --resolve cafe.example.com:$IC_HTTPS_PORT:$IC_IP
https://cafe.example.com:$IC_HTTPS_PORT/coffee --insecure
  Server address: 10.12.0.18:80
  Server name: coffee-7586895968-r26zn
  ...
```

- Accessing the tea application

```
 $ curl --resolve cafe.example.com:$IC_HTTPS_PORT:$IC_IP
https://cafe.example.com:$IC_HTTPS_PORT/tea --insecure
  Server address: 10.12.0.19:80
  Server name: tea-7cd44fcb4d-xfw2x
  ...
```

b) Monitor the runtime activities of the application in one of the following ways:

1. If you are using NGINX, follow the instructions to access the NGINX status page.

2. If you are using NGINX Plus, go to the **Upstream** tab.

CHAPTER 14

Cisco Container Platform Operator Overview

Cisco Container Platform Operator Overview

You can use Cisco Container Platform to install, operate, and upgrade control plane and tenant clusters using Kubernetes Custom Resources. Cisco Container Platform has the following operators to do day-2 operations using custom resources:

- **CCP vSphere Operator** - used to operate Kubernetes nodes as virtual machines in a vSphere provider
- **CCP net tinker** - used to do network management tasks such as operating with IP addresses, IP pools, subnets, networks, load balancer IP addresses, metallb, ACI profiles, nginx ingress controller, and CNI

  For example, the Cisco Container Platform net tinker assigns static IP addresses from an IP pool for node IP addresses, master node VIP, and external load balancer IP addresses.

APPENDIX A

Minimum User Privileges

This appendix contains the following topics:

Minimum User Privileges on vSphere

The following tables provide the minimal set of privileges that are required by the vSphere user to execute the relevant operations in vCenter:

- When using vSphere with HyperFlex, on page 100
- When using vSphere without Hyperflex, on page 105

When using vSphere with HyperFlex

Roles	Privileges	Entities	Propagate to Children
Administrator		vCenter	No

Roles	Privileges	Entities	Propagate to Children
	Datastore.AllocateSpace Datastore.FileManagement Network.Assign Resource.AssignVMToPool StorageProfile.View System.Anonymous System.Read System.View VApp.ApplicationConfig VApp.Import VApp.InstanceConfig VApp.ManagedByConfig VApp.PowerOff VApp.PowerOn VApp.ResourceConfig VApp.Suspend VirtualMachine.Config.AddExistingDisk VirtualMachine.Config.AddNewDisk VirtualMachine.Config.AddRemoveDevice VirtualMachine.Config.AdvancedConfig VirtualMachine.Config.CPUCount VirtualMachine.Config.DiskExtend VirtualMachine.Config.ManagedBy VirtualMachine.Config.Memory VirtualMachine.Config.RawDevice VirtualMachine.Config.RemoveDisk VirtualMachine.Config.Rename VirtualMachine.Config.Resource VirtualMachine.Config.Settings VirtualMachine.Interact.PowerOff VirtualMachine.Interact.PowerOn VirtualMachine.Inventory.Create		

Roles	Privileges	Entities	Propagate to Children
	VirtualMachine.Inventory.CreateFromExisting VirtualMachine.Inventory.Delete VirtualMachine.Provisioning.Clone VirtualMachine.Provisioning.CreateTemplateFromVM VirtualMachine.Provisioning.DeployTemplate		

Roles	Privileges	Entities	Propagate to Children
ccp-datacenter		Datastore	Yes

Roles	Privileges	Entities	Propagate to Children
	Datastore.AllocateSpace Datastore.FileManagement Network.Assign Resource.AssignVMToPool StorageProfile.View System.Anonymous System.Read System.View VApp.ApplicationConfig VApp.Import VApp.InstanceConfig VApp.ManagedByConfig VApp.PowerOff VApp.PowerOn VApp.ResourceConfig VApp.Suspend VirtualMachine.Config.AddExistingDisk VirtualMachine.Config.AddNewDisk VirtualMachine.Config.AddRemoveDevice VirtualMachine.Config.AdvancedConfig VirtualMachine.Config.CPUCount VirtualMachine.Config.DiskExtend VirtualMachine.Config.ManagedBy VirtualMachine.Config.Memory VirtualMachine.Config.RawDevice VirtualMachine.Config.RemoveDisk VirtualMachine.Config.Rename VirtualMachine.Config.Resource VirtualMachine.Config.Settings VirtualMachine.Interact.PowerOff VirtualMachine.Interact.PowerOn VirtualMachine.Inventory.Create		

Roles	Privileges	Entities	Propagate to Children
	VirtualMachine.Inventory.CreateFromExisting VirtualMachine.Inventory.Delete VirtualMachine.Provisioning.Clone VirtualMachine.Provisioning.CreateTemplateFromVM VirtualMachine.Provisioning.DeployTemplate		

For more information on adding a vSphere provider profile, see Adding vSphere Provider Profile , on page 23.

When using vSphere without Hyperflex

Roles	Privileges	Entities	Propagate to Children
ccp-vcenter	Extension.Register Extension.Unregister Extension.Update StorageProfile.View System.Anonymous System.Read System.View	vCenter	No

Roles	Privileges	Entities	Propagate to Children
ccp-datacenter		Datastore	Yes

Roles	Privileges	Entities	Propagate to Children
	Datastore.AllocateSpace Datastore.FileManagement Network.Assign Resource.AssignVMToPool StorageProfile.View System.Anonymous System.Read System.View VApp.ApplicationConfig VApp.Import VApp.InstanceConfig VApp.ManagedByConfig VApp.PowerOff VApp.PowerOn VApp.ResourceConfig VApp.Suspend VirtualMachine.Config.AddExistingDisk VirtualMachine.Config.AddNewDisk VirtualMachine.Config.AddRemoveDevice VirtualMachine.Config.AdvancedConfig VirtualMachine.Config.CPUCount VirtualMachine.Config.DiskExtend VirtualMachine.Config.ManagedBy VirtualMachine.Config.Memory VirtualMachine.Config.RawDevice VirtualMachine.Config.RemoveDisk VirtualMachine.Config.Rename VirtualMachine.Config.Resource VirtualMachine.Config.Settings VirtualMachine.Interact.PowerOff VirtualMachine.Interact.PowerOn VirtualMachine.Inventory.Create		

Roles	Privileges	Entities	Propagate to Children
	VirtualMachine.Inventory.CreateFromExisting VirtualMachine.Inventory.Delete VirtualMachine.Provisioning.Clone VirtualMachine.Provisioning.CreateTemplateFromVM VirtualMachine.Provisioning.DeployTemplate		

For more information on adding a vSphere provider profile, see Adding vSphere Provider Profile , on page 23.

Minimum User Privileges on AWS

The following table provides the minimal set of privileges that are required by an AWS user to create the EKS and EC2 resources.

Roles	Privileges (* Indicates full access)	Entities	Propagate to Children
aws-role	cloudformation	*	No
	elasticloadbalancing	*	
	autoscaling	*	
	ec2	*	
	eks	*	
	ecr	*	
	ecs	*	
	s3	*	
	iam	List*	
		Get*	
		PassRole	
		AddRoleToInstanceProfile	
		RemoveRoleFromInstanceProfile	
		CreateRole	
		CreateInstanceProfile	
		DeleteInstanceProfile	
		DeleteRole	
		DeleteRolePolicy	
		AttachRolePolicy	
		DetachRolePolicy	
		PutRolePolicy	
		AccessKey	
		MFA	

For more information on configuring the necessary permissions, see Configuring Permissions for AWS Account, on page 41.

For more information on adding an Amazon provider profile, see Adding Amazon Provider Profile, on page 24.

Minimum User Privileges on AKS

For using Cisco Container Platform with Azure Kubernetes Service, you must use a Service Principal with an **Owner** role.

For more information on adding an Azure provider profile, see Adding Azure Provider Profile, on page 25

User Privileges on GKE

For using Cisco Container Platform with GKE, you must use a service account with the permissions as described in Creating Service Account, on page 55.

Erase User Data

You need to erase user data and return a cluster to a clean state when its physical media is replaced or removed. When working with **Virtual Volumes**, deleting or overwriting a file is not adequate for completely erasing user data. File systems do not overwrite the disk blocks that contain data. This means that deletion of a VM or datastore does not erase user data. In order to securely erase user data, you need to erase the physical storage underlying the datastore.

For more information on securely erasing user data from a cluster, see the latest documentation from your storage vendor.

APPENDIX B

Accessibility Features in Cisco Container Platform

This chapter contains the following topic:

Accessibility Features in Cisco Container Platform

The list of accessibility features in Cisco Cisco Container Platform is available on the Voluntary Product Accessibility Template (VPAT) page under the Cloud section. For further assistance, you can contact accessibility@cisco.com.

All product documents are accessible except for images, graphics, and some charts. If you would like to receive the product documentation in audio format, braille, or large print, contact accessibility@cisco.com.