Cisco Container Platform 4.0.0 User Guide

First Published: 2019-06-03

Last Modified: 2019-07-09

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

CONTENTS

CHAPTER 1

Cisco Container Platform

Cisco Container Platform is a turnkey, production grade, extensible platform to deploy and manage multiple Kubernetes clusters. It runs on 100% upstream Kubernetes. Cisco Container Platform offers seamless container networking, enterprise-grade persistent storage, built-in production-grade security, integrated logging, monitoring and load balancing.

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 HyperFlex and Cisco ACI. The infrastructure provider for Cisco Container Platform is Hyperflex.

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 with *Administrator* credentials.	Accessing Cisco Container Platform Web Interface, on page 3
Set up the Cisco Container Platform infrastructure configuration.	Setting Up Cisco Container Platform, on page 3

Task	Related Section
Configure Cisco Smart Software Licensing for your Cisco Container Platform instance.	Configuring Cisco Smart Software Licensing, on page 4
Manage the Cisco Container Platform infrastructure configurations using which clusters are created.	Managing Cisco Container Platform Infrastructure Configuration, on page 9
Create Kubernetes clusters.	Creating Kubernetes Clusters on vSphere On-prem Clusters, on page 17 Creating AWS EKS Clusters, on page 27
Add users, assign appropriate roles, and associate the new users to the Kubernetes clusters that you have created.	Managing Users and RBAC, on page 9
Monitor Kubernetes clusters.	Monitoring Health of Cluster Deployments, on page 32 Monitoring Logs from Cluster Deployments, on page 34
Manage Kubernetes cluster using the Kubernetes Dashboard.	Managing Kubernetes Clusters, on page 31
Manage the lifecycle of Kubernetes clusters by scaling or upgrading the clusters.	Scaling vSphere Clusters, on page 19 Upgrading vSphere Clusters, on page 19 Scaling AWS EKS Clusters, on page 28

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 with 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 32 Monitoring Logs from Cluster Deployments, on page 34
Manage the assigned Kubernetes clusters using the Kubernetes Dashboard or CLI.	Managing Kubernetes Clusters, on page 31
Deploy applications on the assigned Kubernetes clusters.	Deploying Applications on Kubernetes Clusters, on page 53

Accessing Cisco Container Platform Web Interface

Before you begin

Ensure that you have configured the prerequisites for integrating ACI with Cisco Container Platform.

For more information, refer to 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.

Setting Up Cisco Container Platform

Note This topic is applicable only for 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 14.

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 for the IP pool that is used to allocate IP addresses to the pods.
i) In the **STARTING SUBNET FOR SERVICE** field, enter the starting IP address for the IP pool that is used to allocate IP addresses to the service VLAN.
j) In the **CONTROL PLANE CONTRACT NAME** field, enter the name of the contract that is provided by the Control Plane endpoint group to allow traffic from the Control Plane cluster to the tenant cluster.
k) In the **NODE VLAN START ID** field, enter the starting VLAN ID that is used to allocate VLAN to the node.
l) In the **NODE VLAN END ID** field, enter the ending VLAN ID that is used to allocate VLAN to the node.
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 13.

Configuring Cisco Smart Software Licensing

You need to configure Cisco Smart Software Licensing 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.

A Cisco Container Platform instance is available for a 90-day evaluation period after which, you need to register with Cisco Smart Software Manager (Cisco SSM). 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 at software.cisco.com, under the **Smart Licensing** area.

If you do not want to manage licenses using Cisco SSM, either for policy reasons or network availability reasons, you can choose to install Cisco SSM satellite at your premises. Cisco Container Platform registers and reports license consumption to the Cisco SSM satellite as it does to Cisco SSM.

Note Ensure that you use Cisco SSM Satellite version 5.0 or later. For more information on installing and configuring Cisco SSM satellite, refer to http://www.cisco.com/go/smartsatellite.

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 or Cisco SSM satellite totals the license requirements for all your Cisco Container Platform instances and compares the total license usage to the number of licenses purchased, on a daily basis. After the 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
- **Authorization Expired**, when the product has not communicated with Cisco SSM or Cisco SSM satellite for a period of 90 days.

Workflow of Cisco Smart Software Licensing

The following table describes the workflow of Cisco Smart Software Licensing.

Task	Related Section
Generate a product instance registration token in your virtual account	Generating Registration Token, on page 5
Configure the transport settings using which Cisco Container Platform connects to Cisco SSM or Cisco SSM satellite	Configuring Transport Settings, on page 6
Register the Cisco Container Platform instance with Cisco SSM or Cisco SSM satellite	Registering Cisco Container Platform License, on page 7
Manage licenses	Renewing Authorization, on page 7 Reregistering Cisco Container Platform License, on page 8 Deregistering Registration, on page 8

Generating Registration Token

You need to generate a registration token from Cisco SSM or Cisco SSM satellite to register the Cisco Container Platform instance.

Before you begin

Ensure that you have set up a Smart Account and a Virtual account on Cisco SSM or Cisco SSM satellite.

Step 1 Log in to your Smart Account on Cisco SSM or Cisco SSM satellite.

Step 2 Navigate to the Virtual account using which you want to register the Cisco Container Platform instance.

Step 3 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.

Step 4 Click **New Token** to generate a registration token.

Step 5 Copy and save the token for using it when you register your Cisco Container Platform instance.

For more information on registering your Cisco Container Platform instance, see Registering Cisco Container Platform License, 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 From 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 In the **Transport Settings** dialog box, perform one of these steps:

- To configure Cisco Container Platform to send the license usage information to Cisco SSM using the Internet:
  1. Click the **DIRECT** radio button.
  2. Configure a DNS on Cisco Container Platform to resolve *tools.cisco.com*.

  This is the default setting.
- To configure Cisco Container Platform to send the license usage information to Cisco SSM using the Cisco SSM satellite:
  1. Click the **TRANSPORT GATEWAY** radio button.
  2. Enter the URL of the Cisco SSM satellite.
- To configure Cisco Container Platform to send the license usage information to Cisco SSM using a proxy server. For example, an off-the-shelf proxy, such as Cisco Transport Gateway or Apache:
  1. Click the **HTTP/HTTPS PROXY** radio button.
  2. Enter the IP address and port number of the proxy server.

Step 5 Click **SAVE**.

Registering Cisco Container Platform License

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

Before you begin

Ensure that you have configured the transport settings.

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

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

Step 3 In the license notification, click **Register**.
The **Smart Software Licensing Product Registration** dialog box appears.

Step 4 In the **Product Instance Registration Token** field, copy and paste the registration token that you generated using the Cisco SSM or Cisco SSM satellite.

For more information on generating a registration token, see Generating Registration Token, on page 5.

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.

If registering the token fails, you can reregister the Cisco Container Platform instance using a new token.

For more information on reregistering Cisco Container Platform, see Reregistering Cisco Container Platform License, on page 8.

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 From 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 From 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.

For more information on generating a registration token, see Generating Registration Token, on page 5.

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 From 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.

CHAPTER 2

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 10.

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

Step 2 Click **NEW USER**.

Step 3 Specify information such as first name, last name, username, passphrase, and role for the user.

Step 4 Click **SUBMIT**.

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.

Changing Login Passphrase

Step 1 From 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 **Edit** corresponding to your name.

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

Step 3 Change the passphrase and role assigned as necessary, and click **SUBMIT**.

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.

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 From 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, specify the domain name of the AD server for all the accounts that you have.

Step 6 In the **ACCOUNT USERNAME** field, specify the service account name that is used for accessing the LDAP server.

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

Step 8 Click **SUBMIT**.

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 10.

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

Step 2 Click **ADD GROUP**.

Step 3 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 4 From the **CLUSTERS** drop-down list, choose the names of the cluster that you want to assign to the AD group.

Step 5 Click **SUBMIT**.

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. Hence, you need to ensure that this user has the necessary privileges.

For more information on the vSphere user privileges, see User Privileges on vSphere, on page 59.

Step 1 From 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

Step 1 From 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 Amazon account.
b) In the **ACCESS KEY ID** field, enter the key ID for the related Amazon account.
c) In the **SECRET ACCESS KEY** field, enter the access key for the related Amazon account.
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 on administering AWS EKS clusters, see Administering AWS EKS Clusters, on page 23.

Modifying Provider Profile

This section contains the following topics:

Modifying vSphere Provider Profile

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

Step 2 Click the **vSphere** tab.

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**.

Modifying Amazon Provider Profile

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

Step 2 Click the **AWS** tab.

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 From the left pane, click **Infrastructure Providers**.

Step 2 Click the **vSphere** or **AWS** 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 From 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 for the IP pool that is used to allocate IP addresses to the pods.

j) In the **STARTING SUBNET FOR SERVICE** field, enter the starting IP address for the IP pool that is used to allocate IP addresses to the service VLAN.

k) In the **CONTROL PLANE CONTRACT NAME** field, enter the name of the contract that is provided by the Control Plane endpoint group to allow traffic from the Control Plane cluster to the tenant cluster.

l) In the **NODE VLAN START ID** field, enter the starting VLAN ID that is used to allocate VLAN to the node.

m) In the **NODE VLAN END ID** field, enter the ending VLAN ID that is used to allocate VLAN to the node.

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

Step 3 Click **SUBMIT**.

Modifying ACI Profile

Step 1 From 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 From 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 is applicable only for 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 From 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 From 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 From 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 From 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 From 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 3

Administering vSphere On-prem Clusters

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

This chapter contains the following topics:

Creating Kubernetes Clusters on vSphere On-prem Clusters

Step 1 From the left pane, click **Clusters**, and then click the **vSphere** 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 your Kubernetes cluster. For more information, see Adding vSphere Provider Profile , on page 12.

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

c) In the **VERSION** drop-down list, choose the version of Kubernetes that you want to use for creating the cluster.

d) In the **CNI** field, enter the Container Network Interface (CNI) that you want to use.

e) In the **DESCRIPTION** field, add a description.

f) If you are using ACI, specify the ACI profile, see Adding ACI Profile, on page 13.

g) Click **NEXT**.

Step 4 In the **Provider Settings** screen, specify the data center, cluster, resource pool, storage class, network, HyperFlex local network, datastore, and VM template that you want to use, and then click **NEXT**.

Note

- Ensure that DRS and HA are enabled on the cluster that you choose in this step. For more information on enabling DRS and HA on clusters, refer to the *Cisco Container Platform Installation Guide*.
- Ensure that the datastore that you choose in this step is accessible to the hosts in the cluster.
- For Network, select a subnet with an adequate number of free IP addresses. For more information, see Managing Networks, on page 14. The selected network must have access to vCenter.
- The following steps are only required on 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.

Step 5 In the **GPU Configuration** screen, specify the type and number of GPUs that you want to use for worker nodes in the cluster.

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

Step 6 In the **Node Configuration** screen, specify the following information, and then click **NEXT**:

- The number of worker and master nodes, and their VCPU and memory configurations.
- If you have GPUs in your HyperFlex cluster, you can select the type of GPU you want to use for cluster creation. This is an optional field.
- The SSH public key that you want to use for creating the cluster. 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.
- The VM username that you want to use as the login for the VM.
- The subnet that you want to use for this cluster.
- The number of load balancer IP addresses for this cluster.

  For more information, see Load Balancer Services, on page 43.
- The IP addresses in CIDR notation that you want to use as the pod subnet.
- Whether or not you want to enable Istio
- A root CA certificate to allow tenant clusters to securely connect to additional services

Step 7 In the **Harbor Registry** screen, specify if you want to enable Harbor. If no, click **NEXT**. If yes, you must specify the following information, and then click **NEXT**:

a) Ensure the switch to enable Harbor is activated
b) A password for Harbor server admin
c) The immutable registry size in gigabits

Step 8 In the **Summary** screen, verify the configuration, and then click **FINISH**.

The cluster deployment takes 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 53.

Upgrading vSphere Clusters

Before you begin

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

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

Step 1 From 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**.

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 vSphere Clusters

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 19.

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 From 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 14.

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 14.

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 vSphere Clusters

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 From 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 4

Administering AWS EKS Clusters

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 AWS EKS Clusters

The prerequisites for configuring AWS EKS clusters are as follows:

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 Virtual Private Cloud (Amazon VPC) for each AWS account	3	Each tenant cluster requires a separate Amazon VPC.

Amazon Resource	Default Limit	Description
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.

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 From 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.

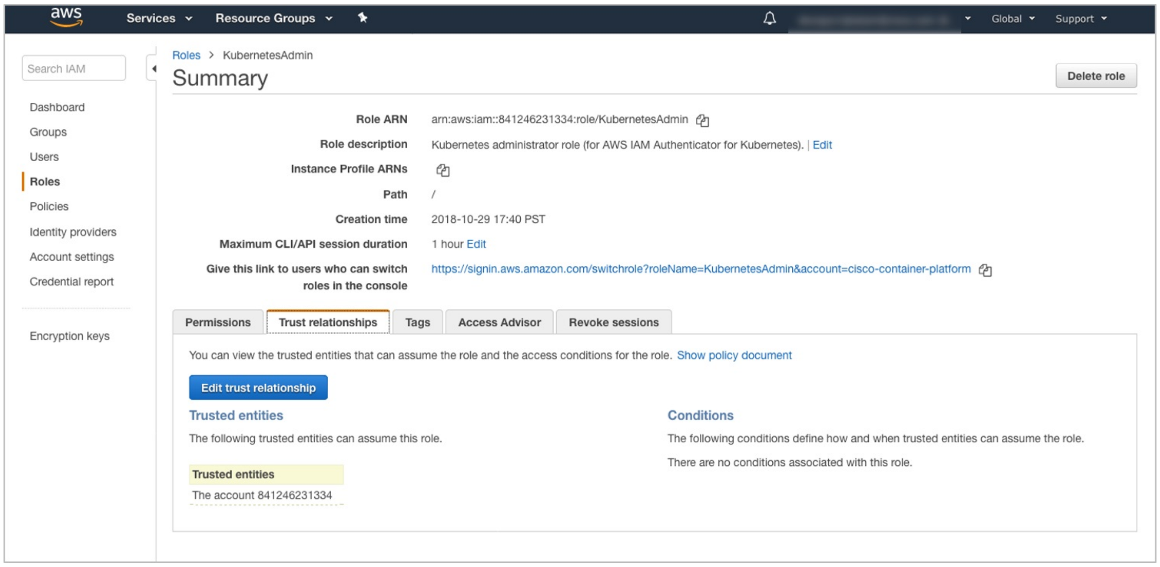
Step 8 In the **Description** field, enter a description of your choice and click **Create role**.

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 1: 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 841246231334

Conditions

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

There are no conditions associated with this role.

AWS Account Policy Requirements

Provider permissions

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

The minimum permissions needed are included in the Sample aws-provider-policy.json File. You can 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": "*"
        }
    ]
}
```

Amazon IAM Authentication

By default, the AWS IAM identity is used to authenticate EKS clusters to establish connection. 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 AWS EKS clusters.

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 Replace <*Proxy_IP_address*> with the IP address of your proxy server.

```
    kubectl patch deployment kaas-api --patch
'{"spec":{"template":{"spec":{"$setElementOrder/containers":[{"name":"api"}],"containers":[{"$setElementOrder/env":[{"name":"https_proxy"},{"name":"no_proxy"}],"env":[{"name":"https_proxy","value":"<Proxy_IP_address>"},{"name":"no_proxy","value":"kaas-*,mysql*,kube*,localhost,127.0.0.1,10.96.0.1"}],"name":"api"}]}}}}'


    kubectl patch deployment kaas-ccp-eks-operator --patch
'{"spec":{"template":{"spec":{"$setElementOrder/containers":[{"name":"ccp-eks-operator"}],"containers":[{"$setElementOrder/env":[{"name":"https_proxy"},{"name":"no_proxy"}],"env":[{"name":"https_proxy","value":"<Proxy_IP_address>"},{"name":"no_proxy","value":"kaas-*,mysql*,kube*,localhost,127.0.0.1,10.96.0.1"}],"name":"ccp-eks-operator"}]}}}}'


    kubectl patch daemonset aws-iam-authenticator -n kube-system --patch
'{"spec":{"template":{"spec":{"$setElementOrder/containers":[{"name":"aws-iam-authenticator"}],"containers":[{"$setElementOrder/env":[{"name":"https_proxy"},{"name":"no_proxy"}],"env":[{"name":"https_proxy","value":"<Proxy_IP_address>"},{"name":"no_proxy","value":"kaas-*,mysql*,kube*,localhost,127.0.0.1,10.96.0.1"}],"name":"aws-iam-authenticator"}]}}}}'
```

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 12.
- 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 24.
- 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 24.

Step 1 From 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, provide 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 24.

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, provide 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 24.

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 From 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 24.

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 From the left pane, click **Clusters**, and then click the **EKS Clusters** 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 5

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:

Setting up Kubernetes Dashboard to Access Clusters

The steps to set up the Kubernetes dashboard differ based on the method used for cluster creation.

This section contains the following topics:

Setting up Kubernetes Dashboard for vSphere On-prem Clusters

For a vSphere on-prem cluster, you can access the Kubernetes dashboard using the `Kubeconfig` file or the Kubernetes default token. The steps to use the default token are same as those provided in [Setting up Kubernetes Dashboard for On-prem AWS IAM Enabled Clusters].

Follow these steps to set up the Kubernetes dashboard using the `Kubeconfig` file.

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) From 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 set up the Kubernetes dashboard access, perform these steps in the Kubernetes dashboard:

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

b) Select the `Kubeconfig` file that you got in Step 1-b.

Setting up Kubernetes Dashboard for On-prem AWS IAM Enabled

For an on-prem AWS cluster that has IAM enabled, you can access the Kubernetes dashboard only by using 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) From 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 that you got in Step 2-c.

Setting up Kubernetes Dashboard for AWS EKS Clusters

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) From 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.

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.

The following example shows a use case in which Grafana is available in the `ccp` namespace.

Step 1 Access the Kubernetes cluster master node using ssh.

```
ssh -l <username> <IP address of master node>
```

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 Follow these steps to access Grafana.

Note On the Control Plane, Grafana is installed in the default namespace, therefore you must skip `-n ccp` in the commands.

```
export INGRESS_IP=$(kubectl get svc nginx-ingress-controller -n ccp
-o=jsonpath='{.spec.loadBalancerIP}')
export GRAFANA_USER=$(kubectl get secret ccp-monitor-grafana -n ccp -o=jsonpath='{.data.admin-user}'
 | base64 --decode)
export GRAFANA_PASSWORD=$(kubectl get secret ccp-monitor-grafana -n ccp
-o=jsonpath='{.data.admin-password}' | base64 --decode)
echo "Login to Grafana at http://${INGRESS_IP}/grafana as user ${GRAFANA_USER} with password
${GRAFANA_PASSWORD}" && unset GRAFANA_PASSWORD GRAFANA_USER
```

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 3 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: /
name: ccp-monitor-prometheus-server
namespace: ccp
spec:
rules:
- http:
    paths:
    - backend:
        serviceName: ccp-monitor-prometheus-server
        servicePort: 9090
        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 53.

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 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
```

Step 4 Access the Kibana UI and view the data from the target tenant cluster using a web browser.

http://localhost:5601/app/kibana

For more information on customizing the Kibana UI, refer to 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 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
```

Step 4 Access the Kibana UI and view the data from the target tenant cluster using a web browser.

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

For more information on customizing the Kibana UI, refer to the latest Kibana documentation.

Forwarding Logs to External Elasticsearch Server

Use the following Curl commands to configure forwarding of logs to an external Elasticsearch server:

Step 1 Open a terminal that has a curl client installed.

Step 2 Configure Cisco Container Platform login credentials.

```
export MGMT_HOST=https://<Cisco Container Platform IP address>:<Port>
export CCP_USER=<Username>
export CCP_PASSPHRASE=<Passphrase>
```

Step 3 Login to Cisco Container Platform and save the session cookie for future requests into the cookies.txt local file.

```
curl -k -j -c cookies.txt -X POST -H "Content-Type:application/x-www-form-urlencoded" -d
"username=$CCP_USER&password=$CCP_PASSWORD" $MGMT_HOST/2/system/login/
```

Step 4 Get the list of cluster names.

```
curl -s -k -b cookies.txt -H "Content-Type: application/json"
$MGMT_HOST/2/clusters/ | jq -r '.[].name'
```

Step 5 Set the CLUSTER_NAME environment variable to the cluster that you are working on.

```
export CLUSTER_NAME="<A cluster name from Step 2>"
```

Step 6 Configure the cluster UUID.

```
export CLUSTER_UUID=$(curl -s -k -b cookies.txt -H "Content-Type: application/json"
$MGMT_HOST/2/clusters/$CLUSTER_NAME | jq -r '.uuid')
```

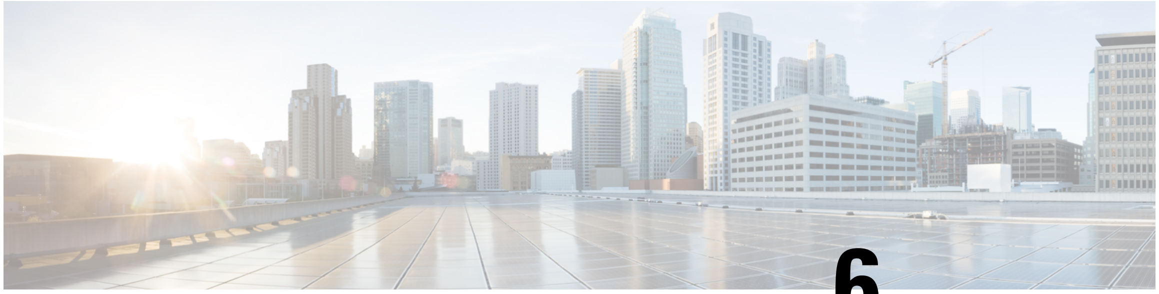
Step 7 Configure the Elasticsearch server IP address and port number.

```
export EFK_SERVER=<IP address of Elasticsearch server>
export EFK_PORT=<Port number of Elasticsearch server>
```

Step 8 Install the helm chart to configure the custom Elasticsearch server.

```
curl -s -k -b cookies.txt -X POST --header 'Content-Type: application/json' --header 'Accept:
application/json' -d '{"chart_url": "/opt/ccp/charts/ccp-agent.tgz", "name": "ccpagent", "options":

"ccp-efk.localLogForwarding.enabled=False,ccp-efk.localLogForwarding.elasticsearchHost='$EFK_SERVER',ccp-efk.localLogForwarding.elasticsearchPort='EFK_PORT'"}'
 $MGMT_HOST/2/clusters/$CLUSTER_UUID/helmcharts
```



CHAPTER 6

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 56.

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

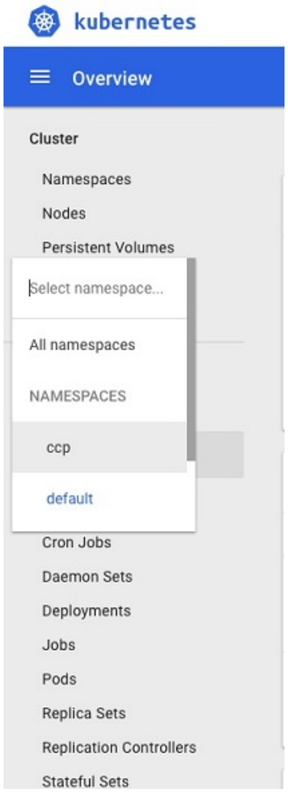
Cisco Container Platform by default creates an L7 Ingress service in order to support Monitoring Health of Cluster Deployments,Monitoring Logs from Cluster Deployments, and Setting up Kubernetes Dashboard for vSphere On-prem Clusters. All of these services are exposed with TLS enabled, and the certificate 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. The following section describes how to obtain the CA certificate.

Step 1 Log in to the Kubernetes dashboard from browser as described in Setting up Kubernetes Dashboard for vSphere On-prem Clusters 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.

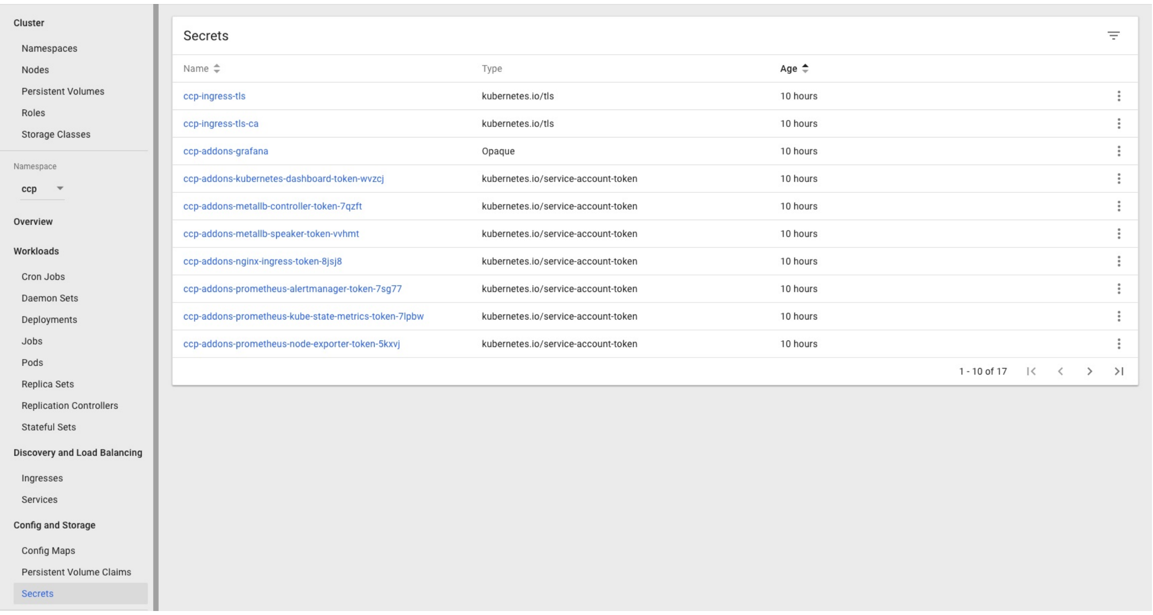
Figure 2: Kubernetes Dashboard



Step 3 Click the **Secrets** tab.

The **Secrets** pane appears.

Figure 3: Secrets Pane



Secrets

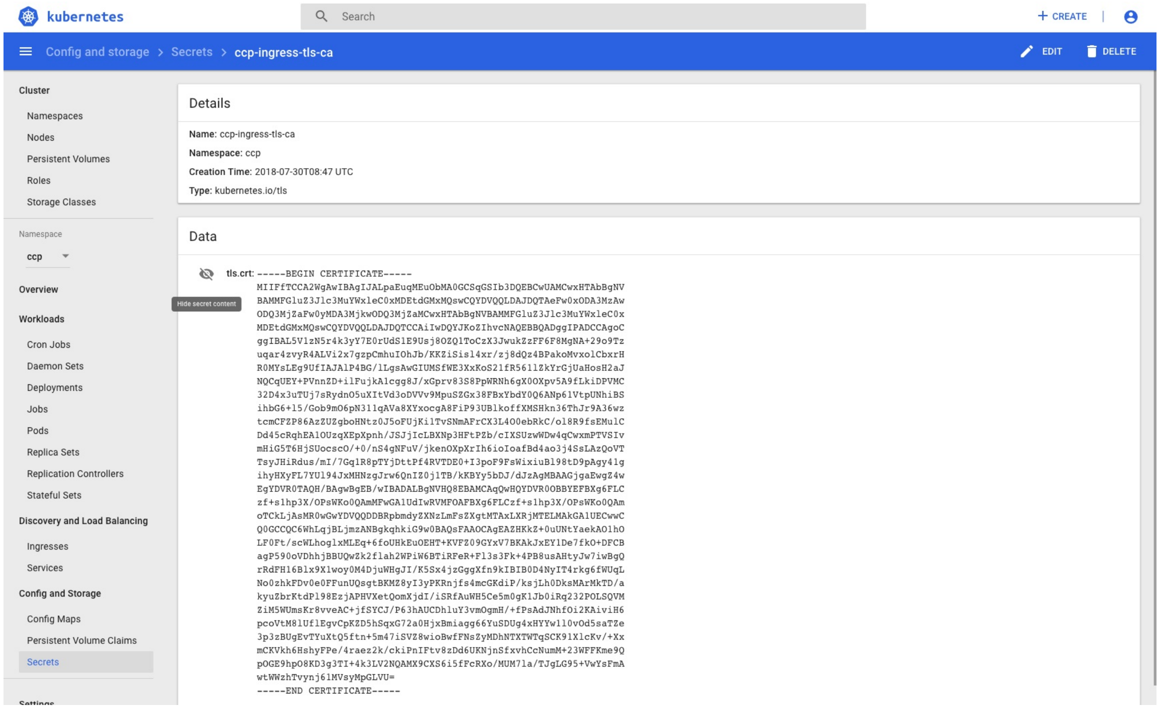
Name	Type	Age
ccp-ingress-tls	kubernetes.io/tls	10 hours
ccp-ingress-tls-ca	kubernetes.io/tls	10 hours
ccp-addons-grafana	Opaque	10 hours
ccp-addons-kubernetes-dashboard-token-wvzcj	kubernetes.io/service-account-token	10 hours
ccp-addons-metallb-controller-token-7qzft	kubernetes.io/service-account-token	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 4: 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
```

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 Kubernetes Clusters on vSphere On-prem Clusters, on page 17.

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 7

Istio Service Mesh

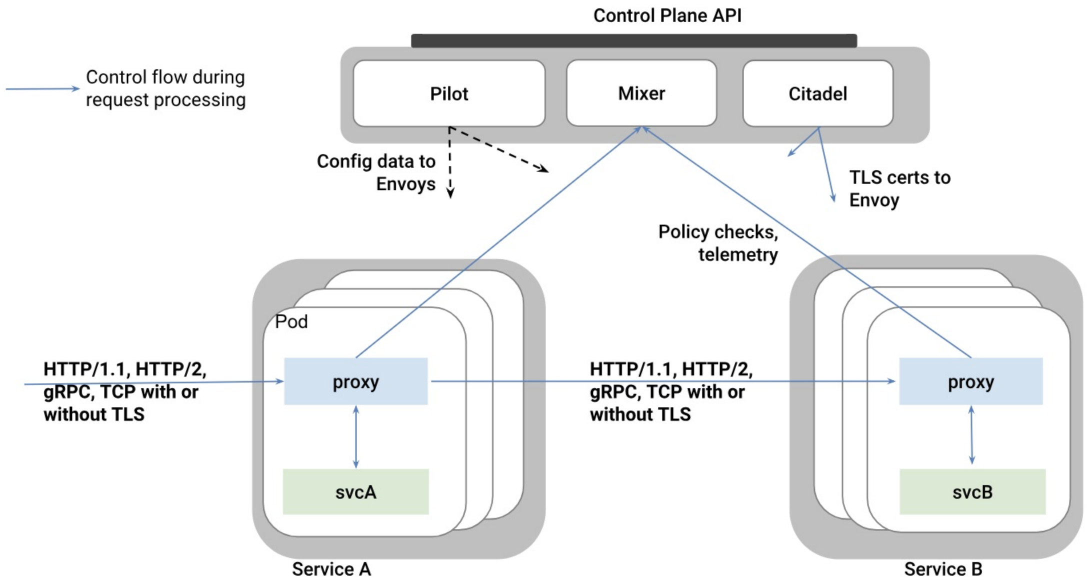
This chapter contains the following topics:

Introduction to Istio Service meshes

Cisco Container Platform includes support for Istio service meshes. 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 meshes. 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, refer to 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 v1.0. You need to contact your service representative for support on the version of Cisco Container Platform you have deployed.

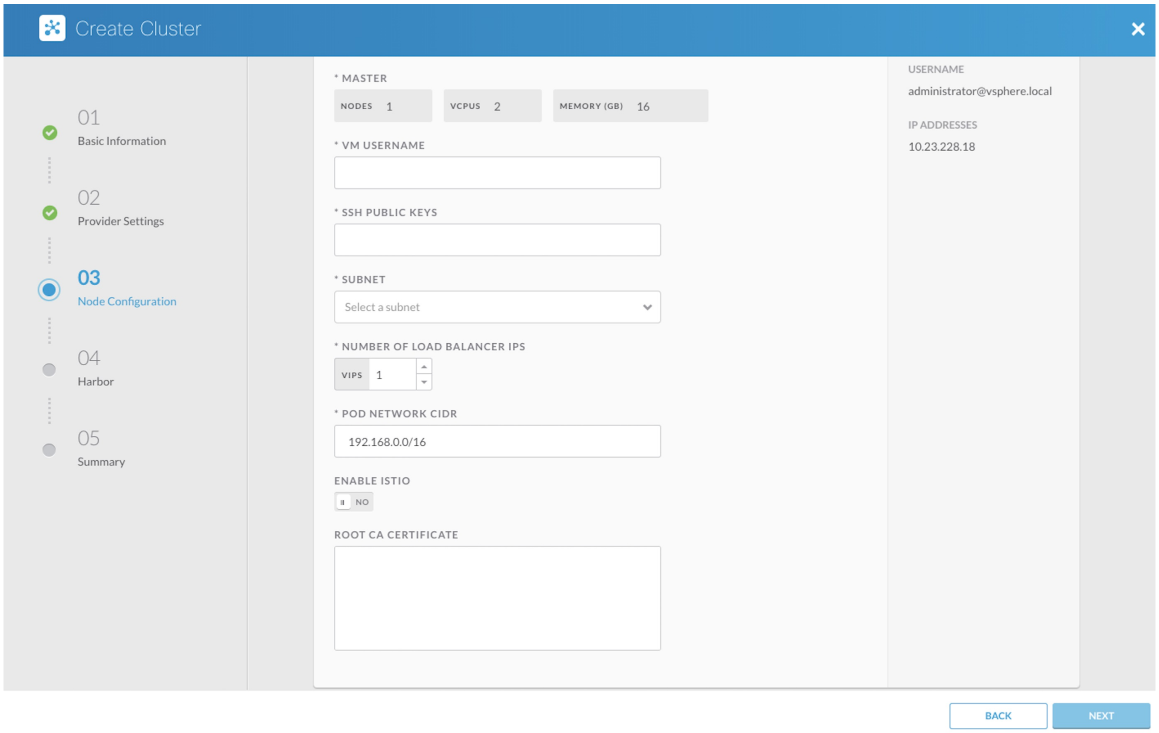
Configuring Istio Service meshes

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, refer to the Virtual IP address section.

The following figure shows the **Node Configuration** screen, using which you can enable the Istio service mesh on a tenant cluster of the Cisco Container Platform.



* MASTER

NODES 1 VCPUS 2 MEMORY (GB) 16

* VM USERNAME

* SSH PUBLIC KEYS

* SUBNET

Select a subnet

* NUMBER OF LOAD BALANCER IPS

VIPS 1

* POD NETWORK CIDR

192.168.0.0/16

ENABLE ISTIO

NO

ROOT CA CERTIFICATE

USERNAME

administrator@vsphere.local

IP ADDRESSES

10.23.228.18

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 meshes

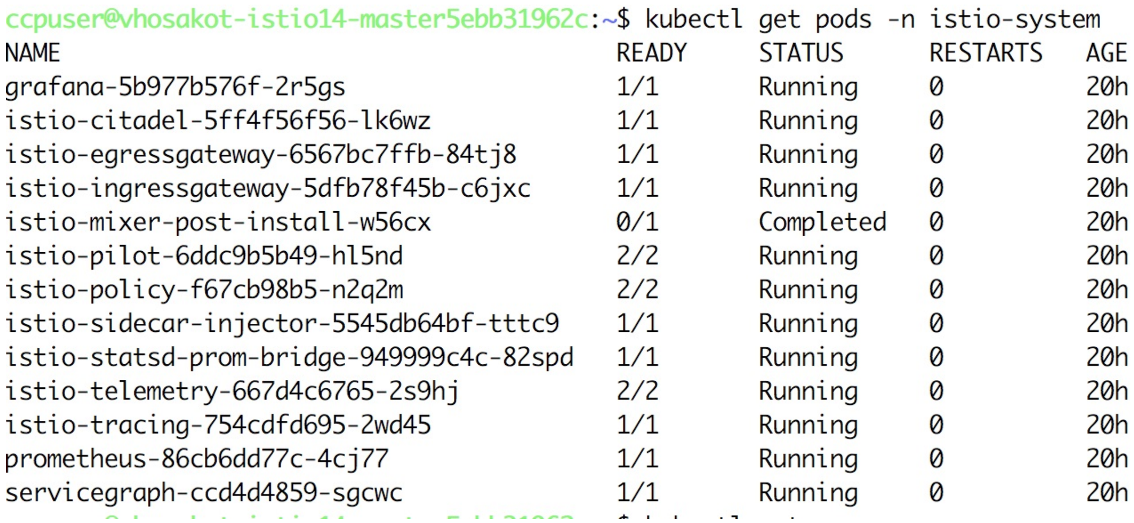
On Cisco Container Platform, the Istio Control Plane is deployed in a special **istio-system** namespace of a tenant Kubernetes cluster. This is similar to how other add-on services such as Prometheus based monitoring or NGINX based Kubernetes ingress are provided. In a production deployment, a tenant 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.

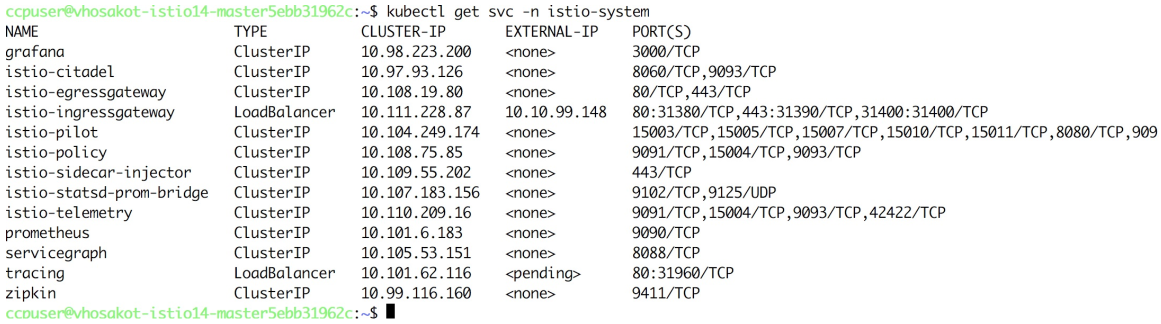
2. Confirm that all pods are running in the istio-system namespace of the tenant Kubernetes cluster. The following figure shows a sample CLI output indicating that all Istio control pods are running correctly in a tenant Kubernetes cluster. If one or more pods continuously fails to run, use **kubectl 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, refer to the Istio upstream documentation.

CHAPTER 8

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:

- Use the steps in Ingress CA , on page 40.
- Run the following command on the tenant cluster where Harbor registry is installed.

```
kubectl get secrets -n ccp ccp-ingress-tls-ca -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 Kubernetes Clusters on vSphere On-prem Clusters, on page 17.

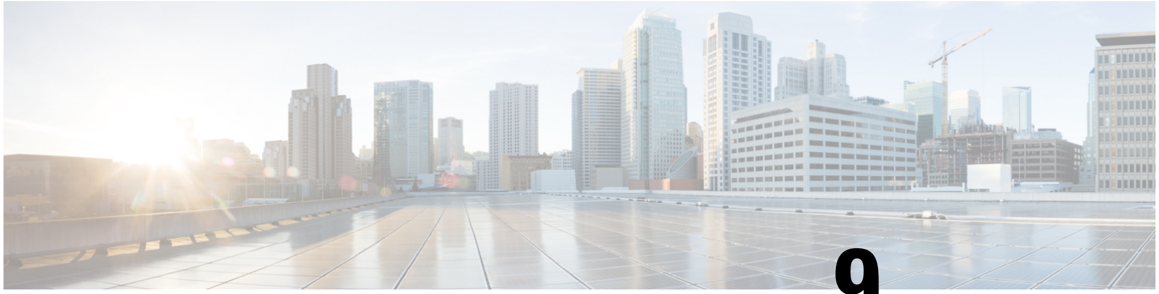
Step 3 In the **Node Configuration** screen, copy and paste the Root CA certificate obtained in Step 1.

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 After tenant cluster creation, SSH to one of the VMs in the cluster and login to the Harbor registry with the password you provided during the installation of Harbor.

```
docker login -u admin -p *****
 https://<LOAD_BALANCER_IP>:443
```



CHAPTER 9

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 Kubernetes Clusters on vSphere On-prem Clusters, on page 17
Download the kubeconfig file that contains the cluster information and the certificates required to access clusters.	Downloading Kubeconfig File, on page 53
Use the kubectl utility to deploy the application and test the scenario.	Sample Scenarios, on page 54

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 From 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+

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 describes deploying and configuring the *Cafe application* with Ingress rules to manage incoming HTTP requests. It uses a **Simple fanout with SSL termination Ingress**.

For more information on Ingress, see Load Balancing Kubernetes Services using NGINX, on page 37.

Step 1 Go to the following URL:

https://github.com/nginxinc/kubernetes-ingress/tree/master/examples/complete-example

Step 2 Download the following yaml files:

- `tea-rc.yaml`
- `tea-svc.yaml`
- `coffee-rc.yaml`
- `coffee-svc.yaml`
- `cafe-secret.yaml`
- `cafe-ingress.yaml`

Step 3 Open the **kubectl** utility.

Step 4 Obtain the IP address of the L7 NGINX load balancer that Cisco Container Platform automatically installs:

```
kubectl get pods --all-namespaces -l app=nginx-ingress -o wide
```

```
NAMESPACE     NAME              READY  STATUS   RESTARTS AGE  IP            NODE
ingressnginx  nginx-             1/1   Running  0        3d   10.10.45.235  test-clusterwc5729f9ce2
              ingresscontroller
              -66974b775-jnmpl
```

Step 5 Deploy the Cafe application.

a) Create the coffee and the tea services and replication controllers:

```
kubectl create -f tea-rc.yaml<br>
kubectl create -f tea-svc.yaml<br>
kubectl create -f coffee-rc.yaml<br>
kubectl create -f coffee-svc.yaml
```

Step 6 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 7 Verify that the Cafe application is deployed.

```
kubectl get pods -o wide
```

```
NAMESPACE       READY STATUS     RESTARTS  AGE   IP                NODE
coffee-rc-jb9sx 1/1   Running    0         3d    192.168.151.134   test-cluster-wb3d42afeff
coffee-rc-tjwgj 1/1   Running    0         3d    192.168.44.133    test-cluster-wc5729f9ce2
tea-rc-6qmvm    1/1   Running    0         3d    192.168.44.132    test-cluster-wc5729f9ce2
tea-rc-ms46j    1/1   Running    0         3d    192.168.151.132   test-cluster-wb3d42afeff
tea-rc-tnftv    1/1   Running    0         3d    192.168.151.133   test-cluster-wb3d42afeff
```

Step 8 Verify if the coffee and tea services are deployed.

```
kubectl get svc
```

```
NAME         TYPE        CLUSTER-IP    EXTERNAL-IP     PORT(S)     AGE
coffee-svc   ClusterIP   10.105.139.1  80/TCP          3d
kubernetes   ClusterIP   10.96.0.1     443/TCP         4d
tea-svc      ClusterIP   10.109.34.129 80/TCP          3d
```

Step 9 Verify if the Ingress is deployed.

```
kubectl describe ing
```

```
Name: cafe-ingress
Namespace: default
Address:
Default backend: default-http-backend:80 (<none>)
TLS: cafe-secret terminates cafe.example.com
Rules:
```

```
Host               Path       Backends
cafe.example.com
                   /tea       tea-svc:80 (<none>)
                   /coffee    coffee-svc:80 (<none>)
```

```
Annotations:
Events: <none>
```

Step 10 Test the application.

a) Access the load balancer IP address `10.10.45.235`, which is obtained in Step2.

b) Test if the Ingress controller is load balancing as expected.

```
curl --resolve cafe.example.com:443:10.10.45.235 https://cafe.example.com/coffee --insecure
<!DOCTYPE html>
...
<p><span>Server&nbsp;address:</span> <span>192.168.151.134:80</span></p>
...
curl --resolve cafe.example.com:443:10.10.45.235 https://cafe.example.com/coffee --insecure
<!DOCTYPE html>
...
<p><span>Server&nbsp;address:</span> <span>192.168.44.133:80</span></p>
...
```

APPENDIX A

User Privileges on vSphere

This appendix contains the following topic:

User Privileges on vSphere

The following table provides the minimal set of privileges that are required by the vSphere user to execute the relevant operations in vCenter.

Roles	Privileges	Entities	Propagate to Children
manage-k8s-node-vms	Resource.AssignVMToPool System.Anonymous System.Read System.View VirtualMachine.Config.AddExistingDisk VirtualMachine.Config.AddNewDisk VirtualMachine.Config.AddRemoveDevice VirtualMachine.Config.RemoveDisk VirtualMachine.Inventory.Create VirtualMachine.Inventory.Delete	Cluster, Hosts, VM folder	Yes
manage-k8s-volumes	Datastore.AllocateSpace Datastore.FileManagement System.Anonymous System.Read System.View	Datastore	No

Roles	Privileges	Entities	Propagate to Children
k8s-system-read-and-spbmprofile-view	StorageProfile.View System.Anonymous System.Read System.View	vCenter	No
ReadOnly	System.Anonymous System.Read System.View	Datacenter, Datastore cluster, Datastore storage folder	Yes
ccp-register-extension	Extension.Register Extension.Unregister Extension.Update	vCenter	No

Roles	Privileges	Entities	Propagate to Children
CCP_Admin		Cluster, Hosts, Vcenter, Datastore, Datastore cluster	Yes

Roles	Privileges	Entities	Propagate to Children
	Extension.Register Extension.Unregister Extension.Update Resource.AssignVMToPool Network.Assign StorageProfile.View System.Anonymous System.Read System.View VirtualMachine.Config.AddExistingDisk VirtualMachine.Config.AddNewDisk VirtualMachine.Config.AddRemoveDevice VirtualMachine.Config.RemoveDisk VirtualMachine.Config.CPUCount VirtualMachine.Config.AdvancedConfig VirtualMachine.Config.Resource VirtualMachine.Config.ManagedBy VirtualMachine.Config.DiskExtend VirtualMachine.Config.Memory VirtualMachine.Config.Settings VirtualMachine.Config.RawDevice VirtualMachine.Inventory.Create VirtualMachine.Inventory.Remove VirtualMachine.Provisioning.Clone VirtualMachine.Provisioning.CreateTemplateFromVM VirtualMachine.Provisioning.DeployTemplate VApp.Import VApp.PowerOn VApp.PowerOff VApp.Suspend VApp.ResourceConfig VApp.InstanceConfig		

Roles	Privileges	Entities	Propagate to Children
	VApp.ApplicationConfig VApp.ManagedByConfig		

For more information on adding a provider profile, see Adding vSphere Provider Profile , on page 12.

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, refer to the latest documentation from your storage vendor.

APPENDIX B

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.