Cisco Container Platform 9.0.0 Installation Guide

First Published: 2021-08-23

Last Modified: 2021-11-11

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

CONTENTS

CHAPTER 1

CHAPTER 2

CHAPTER 1

Cisco Container Platform

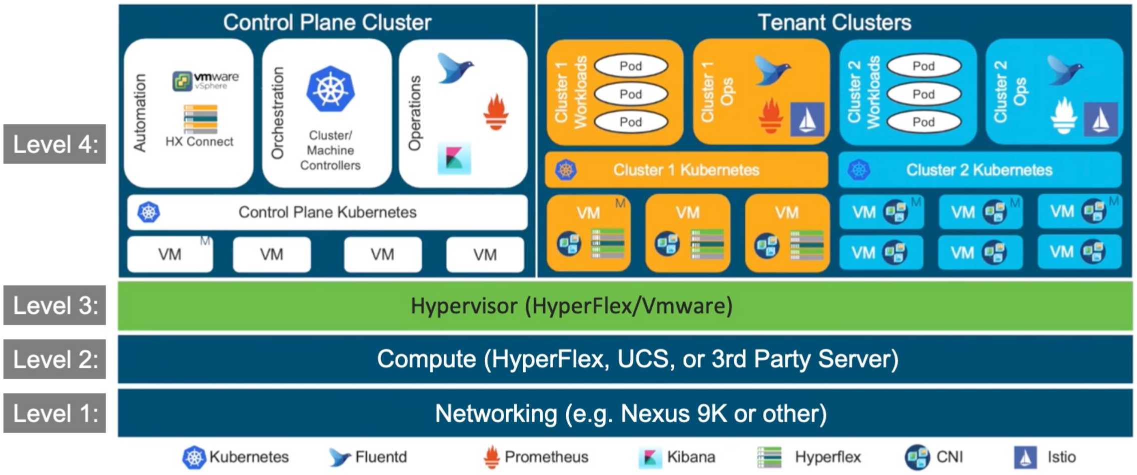
Cisco Container Platform is a turnkey, production grade, extensible platform to deploy and manage multiple Kubernetes clusters. It runs on 100% upstream Kubernetes. Cisco Container Platform offers seamless container networking, enterprise-grade persistent storage, built-in production-grade security,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

This chapter contains the following topics:

Cisco Container Platform Architecture Overview

The following figure shows the architecture of Cisco Container Platform.

Figure 1: Cisco Container Platform Architecture Overview



At the bottom of the stack is level 1, the **Networking** layer that can consist of Nexus switches, Application Policy Infrastructure Controllers (APIC), and Fabric Interconnects (FIs).

Note Cisco Container Platform can run on top of an ACI networking fabric as well as on a non-ACI networking fabric that performs standard L3 switching.

Level 2 is the **Compute** layer that consists of HyperFlex, UCS, or thrid-party servers that provide virtualized compute resources through VMware and distributed storage resources.

Level 3 is the **Hypervisor** layer that is implemented using HyperFlex or VMware.

Level 4 consists of the **Cisco Container Platform Control Plane** and **Data Plane (or tenant clusters)**. In the above figure, the left side shows the Cisco Container Platform Control Plane that runs on four control plane VMs, and the right side shows the tenant clusters. These tenant clusters are preconfigured to support Persistent Volumes using vSphere Cloud Provider and Container Storage Interface (CSI) plugin.

Components of Cisco Container Platform

The following table describes the components of Cisco Container Platform.

Function	Component
Container Runtime	Docker CE
Operating System	Ubuntu
Orchestration	Kubernetes
IaaS	vSphere
Infrastructure	HyperFlex, UCS
Container Network Interface (CNI)	ACI, Contiv, Calico
SDN	ACI
Container Storage	HyperFlex Container Storage Interface (CSI) plugin
Load Balancing	NGINX, Envoy
Service Mesh	Istio, Envoy
Monitoring	Prometheus, Grafana
Logging	Elasticsearch, Fluentd, and Kibana (EFK) stack

Sample Deployment Topology

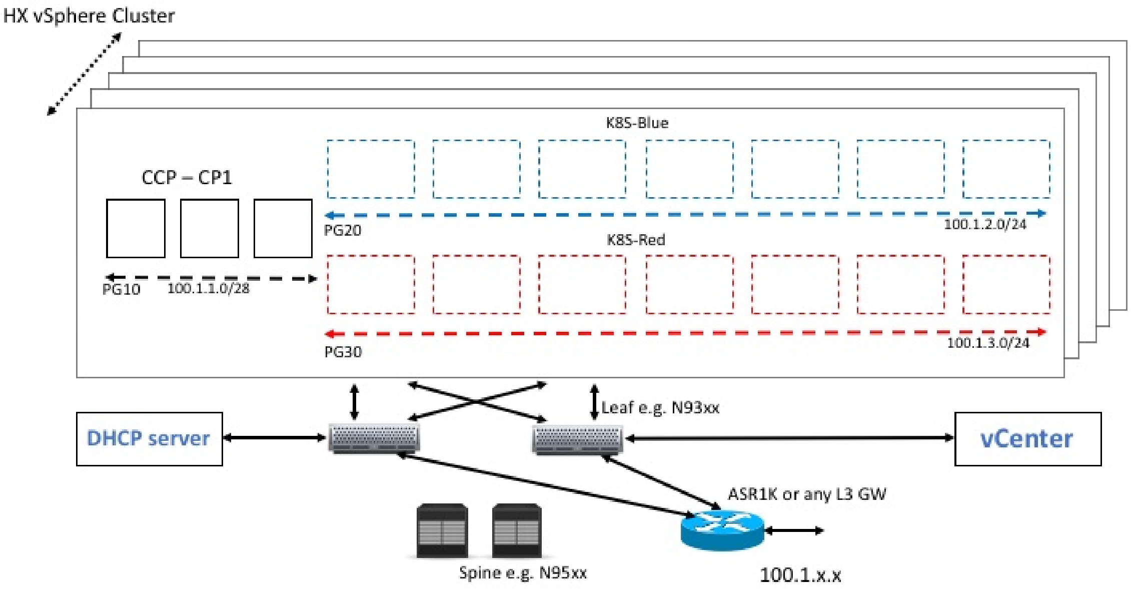
This section describes a sample deployment topology of the Cisco Container Platform and illustrates the network topology requirements at a conceptual level. Future sections of the document such as System Requirements, on page 5 and Installing Cisco Container Platform, on page 17 provide additional configuration details based on these concepts.

Note In this example, the deployment target is a VMware vSphere virtualization platform, and Cisco Container Platform is using a non-ACI CNI such as Calico or Contiv. Other deployment environments are conceptually similar but with some slight differences appropriate to those environments.

In this case, it is expected that the vSphere based cluster is set up, provisioned and fully functional for virtualization and Virtual Machine functionality before any installation of Cisco Container Platform. You can refer to the standard VMware documentation for details on vSphere installation.

The following figure illustrates an example vSphere cluster on which Cisco Container Platform is to be deployed.

Figure 2: Example vSphere Cluster



Once the vSphere cluster is ready to provision VMs, the admin then provisions one or more VMWare port groups (for example PG10, PG20 and PG30 in the figure) on which virtual machines will subsequently be provisioned as container cluster nodes. Basic L2 switching using VMWare vswitch functionality can be used to implement these port groups. IP subnets should be set aside for use on these port groups and the VLANs used to implement these port groups should be terminated on an external L3 gateway (such as the ASR1K shown in the figure). The control plane cluster and tenant plane Kubernetes clusters of Cisco Container Platform can then be provisioned on these port groups.

All provisioned Kubernetes clusters may choose to use a single shared port group or separate port groups may be provisioned (1 per Kubernetes cluster) depending on the isolation needs of the deployment. Layer 3 network isolation may be used between these different port groups as long as the following conditions are met:

- There is L3 IP address connectivity among the port group that is used for the Control Plane cluster and the tenant cluster port groups
- The IP address of the vCenter server is accessible from the Control Plane cluster

- A DHCP server is provisioned for assigning IP addresses to the installer and upgrade VMs, and it must be accessible from the Control Plane port group cluster of the cluster

The simplest functional topology would be to use a single shared port group for all clusters with a single IP subnet to be used to assign IP addresses for all container cluster VMs. This IP subnet can be used to assign one IP per cluster VM and up to four virtual IP addresses per Kubernetes cluster, but would not be used to assign individual Kubernetes pod IP addresses. Hence a reasonable capacity planning estimate for the size of this IP subnet is as follows:

(The expected total number of container cluster VMs across all clusters) + 3 x (The total number of expected Kubernetes clusters)

Container Network Interface Plugins

Cisco Container Platform supports multiple Kubernetes CNI plugins such as:

- ACI is the recommended plugin for use with an ACI fabric. It is optimized for use with an ACI fabric. ACI is fully supported by Cisco.
- Calico is recommended when an ACI fabric is not used.

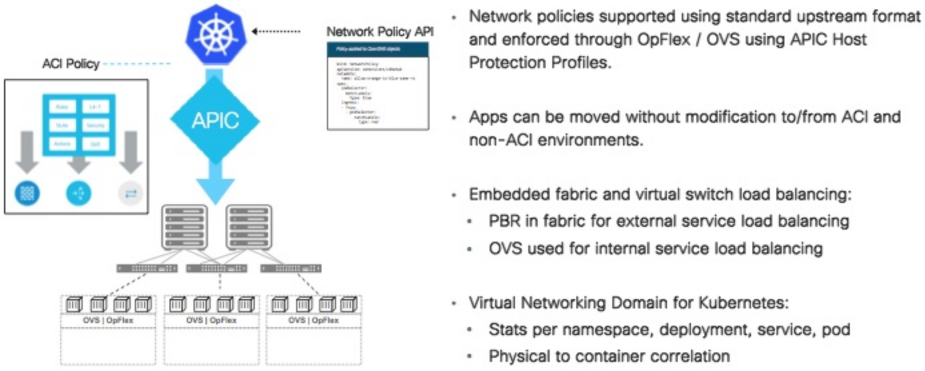
Operationally, all the CNI plugins offer the same experience to the customer. The container network connectivity is seamless and network policies are applied using Kubernetes NetworkPolicies. Under-the-hood, both ACI and Contiv offer advanced feature support. ACI allows you to map CNI NetworkPolicies to an ACI fabric and supports richer underlay policies such as common policies for containers/virtual machines/physical servers and inter-Kubernetes cluster policies. Additionally, ACI supports Kubernetes Type LoadBalancer using PBR policies in the ACI fabric.

ACI

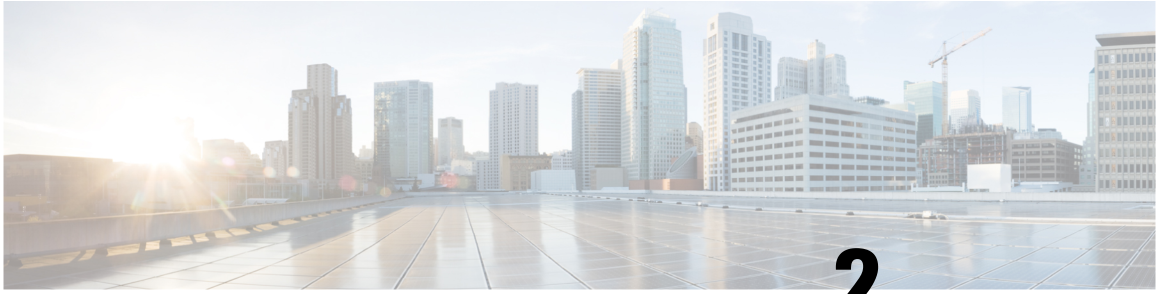
ACI is tightly integrated with the ACI fabric. It supports underlay integration with the ACI fabric and hardware accelerated load balancing.

The following figure shows the architecture of ACI.

Figure 3: Architecture of ACI



- Network policies supported using standard upstream format and enforced through OpFlex / OVS using APIC Host Protection Profiles.
- Apps can be moved without modification to/from ACI and non-ACI environments.
- Embedded fabric and virtual switch load balancing:
  - PBR in fabric for external service load balancing
  - OVS used for internal service load balancing
- Virtual Networking Domain for Kubernetes:
  - Stats per namespace, deployment, service, pod
  - Physical to container correlation



CHAPTER 2

System Requirements

This section describes the requirements that are necessary to deploy Cisco Container Platform.

Note Cisco Container Platform does not support installing virtual machines in nested datacenter, vCenter cluster or virtual machine folders. Cisco Container Platform does not support moving the virtual machines or changing their configuration from vCenter directly.

It contains the following topics:

Supported Version Matrix

Cisco Container Platform uses various software and hardware components.

For more information on the validated versions of each component, refer to the latest *Cisco Container Platform Release Notes*.

Software Requirements

Ensure that the following software applications are installed in your deployment environment:

- VMware vCenter server 6.7 Update 3 and later
- VMware client integration plugin
- vSphere Flash client

Hardware Requirements

- In the Cisco Container Platform Control Plane VM, each master and worker node requires 2 vCPUs, 8 GB memory, and 40 GB HDD.
- In the Cisco Container Platform Tenant Cluster VM, each master and worker node requires 2 vCPUs, 16 GB memory, and 40 GB HDD. You can modify the vCPU and memory configurations when you deploy a new tenant cluster.

Resource Management Requirements

The following topics provide information on the necessary resource management requirements:

Enabling DRS and HA on Clusters

Note You must use the Enterprise Plus license to set up VMware clusters with HA and DRS enabled. For more information on the supported versions of VMware, see Supported Version Matrix, on page 5.

It is required that you enable DRS and HA on vCenter for the following reasons:

- DRS continuously monitors resource utilization across vSphere servers and intelligently balances VMs on the servers.
- HA provides easy to use, cost-effective high availability for applications running on virtual machines.

Step 1 In the vSphere Web Client, navigate to the host or cluster on which you want to deploy Cisco Container Platform.

Step 2 Click the **Configure** tab.

Step 3 Under **Services**, click **vSphere DRS**, and then click **Edit**.

Step 4 In the right pane of the **Edit Cluster Settings** window, check the **Turn ON vSphere DRS** check box, and then click **OK**.

Step 5 Under **Services**, click **vSphere Availability**, and then click **Edit**.

Step 6 In the right pane of the **Edit Cluster Settings** window, check the **Turn ON vSphere HA** check box, and then click **OK**.

Enabling NTP Services

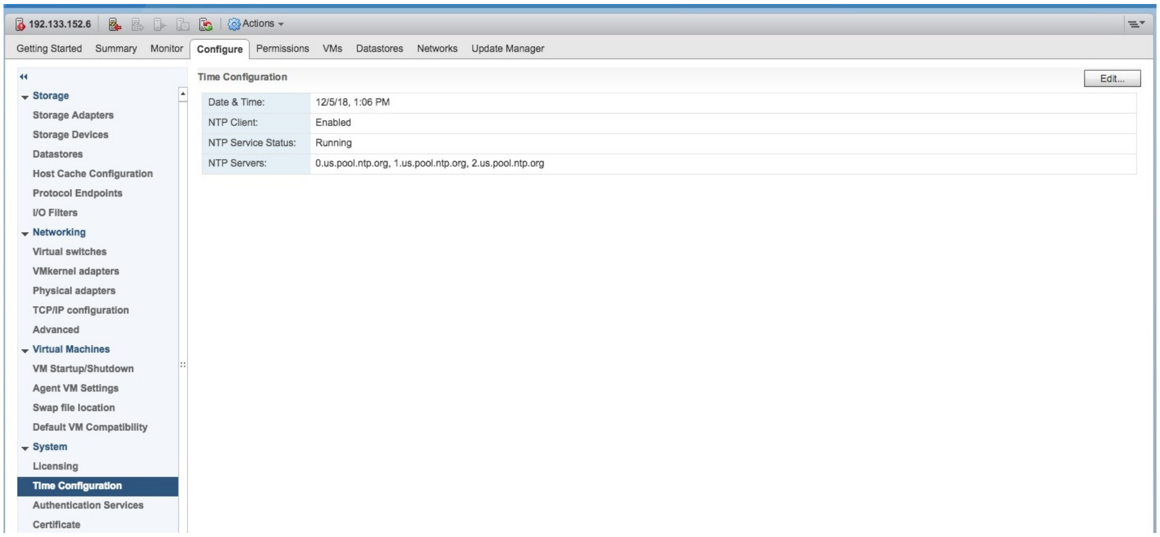
You need to enable the Time Synchronization services on each host within your vSphere environment. If you do not enable this service, errors due to timing differences between hosts may cause installation of the Cisco Container Platform to fail.

Step 1 In the vSphere Web Client, navigate to the host or cluster on which you want to deploy Cisco Container Platform.

Step 2 Click the **Configure** tab.

Step 3 From the left pane, expand **System**, and then click **Time Configuration**.

Figure 4: Time Configuration pane



Time Configuration

Date & Time:	12/5/18, 1:06 PM
NTP Client:	Enabled
NTP Service Status:	Running
NTP Servers:	0.us.pool.ntp.org, 1.us.pool.ntp.org, 2.us.pool.ntp.org

Step 4 In the right pane, click **Edit**.

Step 5 In the **Edit Time Configuration** window, check the **Turn ON vSphere DRS** check box, and then click **OK**.

Note You must ensure that each host has DNS access to enable NTP services.

Network Requirements

The following topics provide information on the necessary network requirements:

If you have chosen Contiv as the CNI, the pod-to-pod traffic across nodes is tunneled by the VXLAN protocol.

Provisioning a Port Group for Cisco Container Platform VM Deployment

Cisco Container Platform creates VMs that are attached to a Port Group on either a vSphere Standard Switch (VSS) or a Distributed Virtual Switch (DVS). The HyperFlex installer creates VSS switches in vSphere for the networks that are defined during installation. You need to create either VSS or DVS Switches for managing the VM traffic.

The following topics provide information on configuring a VSS or a DVS.

Configuring vSphere Standard Switch

Step 1 In the vSphere Web Client, navigate to the host or cluster on which you want to deploy Cisco Container Platform.

Step 2 Click the **Configure** tab.

Step 3 Expand **Networking**, and then select **Virtual switches**.

Step 4 Click **Add host networking**.

Step 5 Choose **Virtual Machine Port Group for a Standard Switch** as the connection type for which you want to use the new standard switch and click **Next**.

Step 6 Select **New standard switch** and click **Next**.

Step 7 Add physical network adapters to the new standard switch.

Step 8 Under **Assigned adapters**, click **Add adapters**.

Step 9 Select one or more physical network adapters from the list.

Step 10 From the **Failover order group** drop-down list, choose from the Active or Standby failover lists.

Step 11 For higher throughput and to provide redundancy, configure at least two physical network adapters in the Active list.

Step 12 Click **OK**.

Step 13 Enter connection settings for the adapter or the port group as follows:

a) Enter a network Label or the port group, or accept the generated label.
b) Set the VLAN ID to configure VLAN handling in the port group.

Step 14 On the **Ready to Complete** screen, click **OK**.

Configuring Distributed Virtual Switch

Step 1 In the **Navigation** pane, click the DVS switch.

Step 2 In the right pane, click the **Hosts** tab.

Step 3 Click the **Actions** icon and click the **Add and Manage Hosts** radio button.
The **Add and Manage Hosts** wizard appears.

Step 4 In the **Select tasks** screen, click the **Add Hosts** radio button, and then click **Next**.

Step 5 In the **Select hosts** screen, click the **Add Hosts** icon.

Step 6 In the **Select new hosts** screen, check the check box next to the hosts that you want to add, and then click **OK**.

Step 7 Click **Next** in the **Select network adapter tasks** screen.

Step 8 In the **Manage physical network adapters** screen, click the network switch that you want to configure, and then click the **Assign** uplink.

Step 9 Repeat Step 8 for all the networks, and click **Next**.

Step 10 In the **Manage VMKernel network adapters** screen, click **Next**.

Step 11 In the **Analyze impact** screen, click **Next**.

Step 12 In the **Ready to complete** screen, click **Finish**.

Configuring DHCP Server

Cisco Container Platform requires a DHCP server to be present. The Cisco Container Platform installer VM and upgrade VM get their primary interface IP addresses from the DHCP server. You must ensure that you have configured a DHCP server.

If the DHCP server does not provide the location of the NTP service, enter the NTP address in the Installer UI, under **Control Plane Settings** > **Advanced Settings**.

Reserving IP Addresses for Static Allocation

Cisco Container Platform uses static IP addresses for all cluster nodes and the **CCP Control Plane master node VIP**, which provides worker nodes with a consistent IP address. Additionally, a load balancer VIP is used as an external IP address for NGINX Ingress in each Kubernetes cluster. These VIPs are configured using IP pools. The static IP addresses are assigned from the same subnet as the load balancer VIP addresses, and you must ensure that the static IP address pools for the subnet do not overlap with a DHCP pool.

Static and DHCP IP Address Requirements

You must ensure that the following conditions are met:

- The subnet is routable to and from the VMware vCenter server.
- The client install machine is routable to the network during the Cisco Container Platform control plane install.
- The network allows communication between Cisco Container Platform VM instances. You must not use a private LAN.

The following table summarize the static and DHCP IP address requirements for the Cisco Container Platform components:

Component	Static IP for Calico	Static IP for ACI-CNI	DHCP IP
Installer VM	0	0	1
Tenant clusters	2 + Number of masters + Number of load balancer VIPs desired for applications + Number of workers	See also, Cisco ACI and Kubernetes Integration	0
Control Plane and Cisco Container Platform web interface	6 **Note** 1 IP address for the Kubernetes master VIP, 1 IP address for the Ingress LoadBalancer, and 1 IP address for the master node, and 3 IP addresses for the worker nodes.	6 **Note** 1 IP address for the Kubernetes master VIP, 1 IP address for the Ingress LoadBalancer, and 1 IP address for the master node, and 3 IP addresses for the worker nodes.	0

By default, the Cisco Container Platform Control Plane pod network uses the 192.168.0.0/16 subnet for Calico. If you have routed IP addresses in that space, you must assign another RFC1918 range for your VXLAN network. It does not need to be a full /16 subnet, a /22 subnet is adequate for the Cisco Container Platform control plane.

HyperFlex Integration Requirements

Note This section is applicable only if you want to use HyperFlex environment. It is not required for running VMware on UCS.

Cisco Container Platform is supported on all hardware configurations that are supported by the required HyperFlex software versions. For more information on HyperFlex hardware configurations, refer to the UCS HyperFlex product documentation.

The following topics provide information on the necessary HyperFlex integration requirements:

Configuring Shared Datastore

After HyperFlex is installed, you need to configure a shared datastore. The datastore must be accessible to hosts such as NFS or iSCSI or FC in the cluster.

The datastore is required for the following purposes:

- Provisioning persistent volume storage
- Deploying the Cisco Container Platform tenant base VM

Step 1 Log in to the **HX Connect UI** using the VMware vCenter SSO administrator credentials.

For more information on installing HyperFlex and accessing the HyperFlex Connect UI, refer to the latest HperFlex documentation.

Step 2 In the left pane, click **Manage** > **Datastores**.

Step 3 Perform these steps to create a datastore for provisioning the Kubernetes persistent volume storage and deploying the Cisco Container Platform tenant base VM:

a) In the right pane, click **Create Datastore**.

b) In the **Name** field, enter `ds1`, and then enter a size and block size for the datastore.

Note We recommend that you use `1TB` size and `8K` block size.

c) Click **Create Datastore**.

The newly created datastore is available on vCenter.

Configuring Link-local Network for HyperFlex iSCSI Communication

The FlexVolume plug-in requires a host-only link between each VM that runs Kubernetes and the Internet Small Computer System Interface (iSCSI) target on the ESX host.

For HyperFlex 3.5+

Step 1 Log in to the **HX Connect UI**.

Step 2 Choose **Settings** > **Integrations** > **Kubernetes**.

Step 3 Click **Enable All Node** and wait until the **KUBERNETES STORAGE PROVISIONING** option is enabled.

The HyperFlex infrastructure is configured and ready to use for Cisco Container Platform with Kubernetes persistent volume support.

For HyperFlex 3.0.x

Step 1 Open an SSH session to the HyperFlex 3.0 Platform Installer VM or one of the HyperFlex Controller VMs and log in as a root user.

Step 2 Perform these steps to get the vCenter details that you need to enter when you run the `add_vswitch.py` script.

a) Run the following command to get the vCenter datacenter name and vCenter cluster name.

```
stcli cluster info | grep -i vcenter
```

b) Run the following command to validate the reachability of vCenter IP address.

```
ping <vcenter URL>
```

Step 3 Navigate to the following location:

`/usr/share/springpath/storfs-misc/hx-scripts/`

Step 4 Run the `add_vswitch.py` script.

```
python add_vswitch.py --vcenter-ip <vCenter IP address>
```

When prompted, specify the vCenter credentials, datacenter name, and cluster name that you got from the output of Step 2.

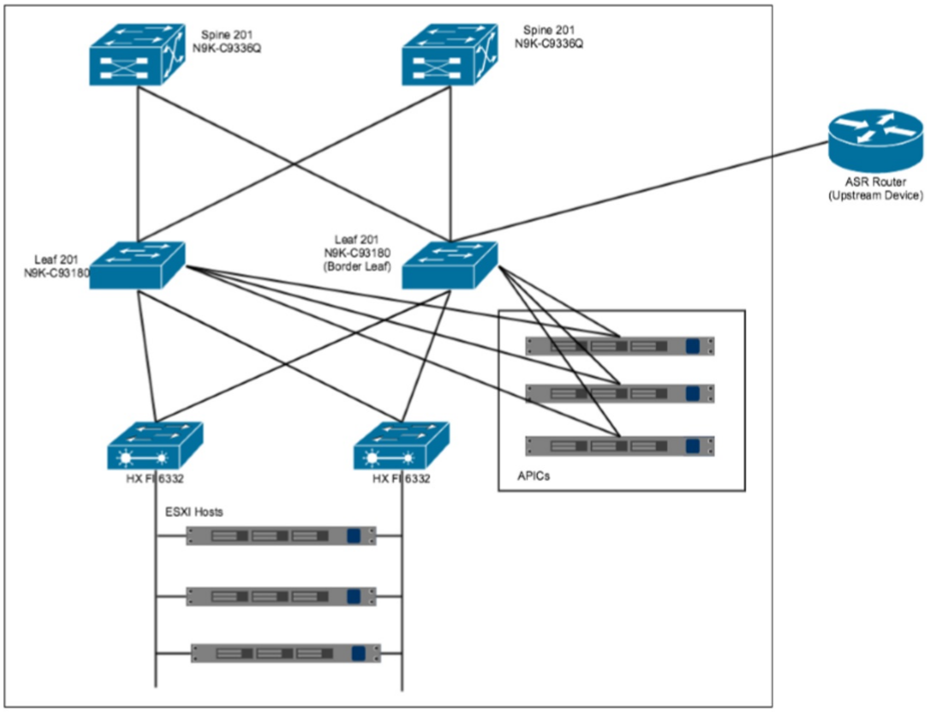
The HyperFlex infrastructure is configured and ready to use for Cisco Container Platform with Kubernetes persistent volume support.

ACI Integration Requirements

Cisco ACI enables you to group your application into End Point Groups (EPGs), define policies for the EPGs, and then deploy network policies on the ACI fabric. The policy enforcement is implemented using the spine and leaf architecture of the ACI fabric.

The following figure shows the components of a Cisco Container Platform ACI integrated network topology.

Figure 5: Cisco Container Platform ACI Integrated Network Topology



The main components of the network topology are as follows:

- **ACI Fabric** includes two spine nodes, two leaf nodes, and three APIC controllers. You can choose the number of the spine and leaf nodes and APIC controllers as per your network requirement.
- **HyperFlex Fabric Interconnect (FI)** includes two fabric interconnect switches connected between the ESXi hosts and the ACI leaf switches.
- **ESXi Hosts** includes a UCS server such as UCS C220 M4.
- **ASR router** is connected to an ACI border leaf for external internet access.

APIC Controller Requirements

If you are using ACI, ensure that you have configured the following settings on the APIC controller:

- Assign a port number other than 4094 for Infra VLAN as 4094 is reserved for provisioning HyperFlex fabric interconnect
- Create a common tenant
- Create a Virtual Route Forwarder (VRF) in the common tenant
- Create at least one L3OUT
- Create an Access Entity Profile (AEP) for the ACI tenant physical domain

- Create an AEP for L3OUT
- Create a Virtual Machine Manager (VMM) domain which connects to vSphere

For more information on configuring an APIC controller, refer to the latest ACI documentation.

HyperFlex FI Requirements

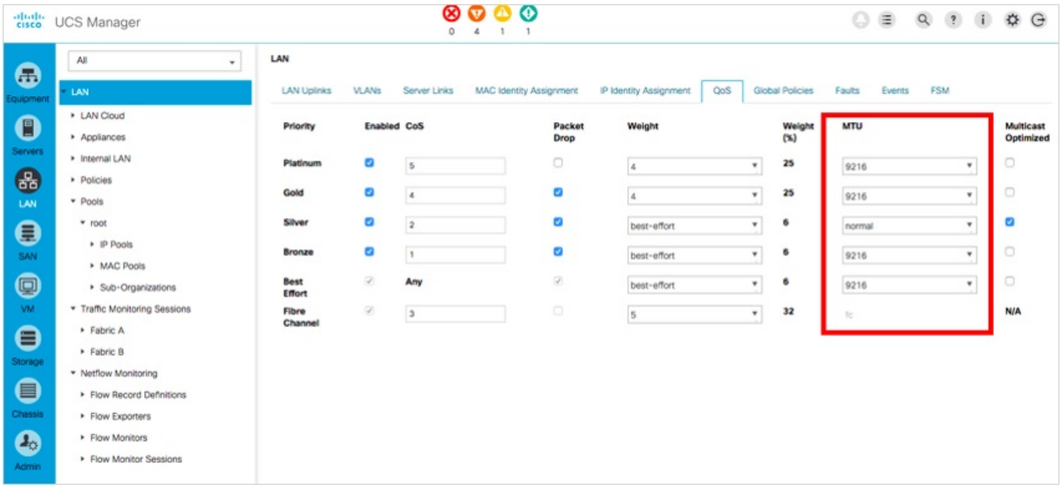
Ensure that you have configured the following settings on HyperFlex FI:

- Configure QOS
  1. From the left pane, click **LAN**.
  2. From the right pane, click the **QoS** tab, and then configure QoS.

Note

- Using the **MTU** configuration, you must set the priority that is associated with the QoS policy of the vNIC template.
- To support Jumbo Frames, you must set the **MTU** for **Best Efforts** to `9216` as shown in the following figure.

Figure 6: QoS Tab



Priority	Enabled	CoS	Packet Drop	Weight	Weight (%)	MTU	Multicast Optimized
Platinum	☑	5	□	4	25	9216	□
Gold	☑	4	☑	4	25	9216	□
Silver	☑	2	☑	best-effort	6	normal	☑
Bronze	☑	1	☑	best-effort	6	9216	□
Best Effort	☑	Any	☑	best-effort	6	9216	□
Fibre Channel	☑	3	□	5	32	fc	N/A

- Ensure that the tenant VLAN is allowed

Once Cisco Container Platform Control Plane and management node networking are configured, you can access the HyperFlex cluster on vSphere and install Cisco Container Platform. Each time you create a tenant cluster, the ACI constructs such as L3OUT, VRF, and AEP stored in the common tenant cluster are reused.

GPU Integration Requirements

Cisco Container Platform supports GPU devices in passthrough mode to enable AI/ML workloads.

This section describes the requirements on the ESXi and vCenter hosts to integrate the GPU devices with Cisco Container Platform.

Step 1 Follow these steps to enable GPU Passthrough for the devices that you want to use:

a) Access the ESXi host by typing its IP address in a web browser.
b) From the left pane, click **Manage**.
c) In the right pane, click **Hardware** > **PCI Devices** .

   The list of available passthrough devices is displayed.

d) Select the device, and then click **Toggle Passthrough**.

Step 2 Follow these steps to enable shared direct passthrough for the GPU device:

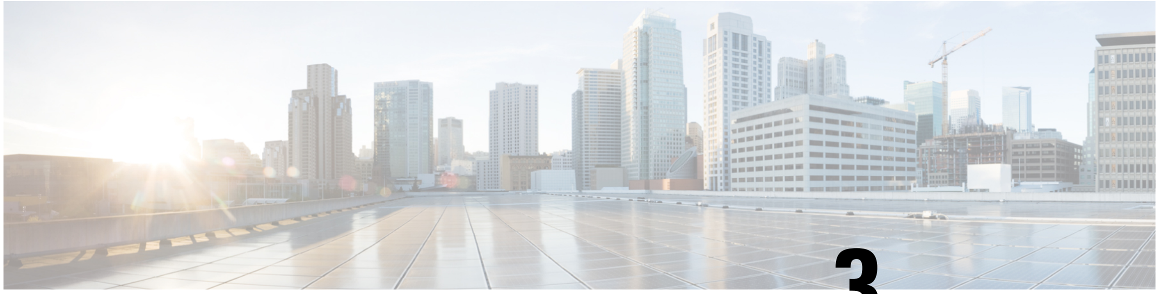
a) Access the vCenter server by typing its IP address in a web browser.
b) From the right pane, click **Configure** > **Graphics** > **Graphics Devices**.
c) Select the device for which you want to enable shared direct passthrough.
d) In the **Edit Graphics Device Settings** dialog box, click the **Shared Direct** radio button.
e) Click **Ok**.

Step 3 Follow these steps to allow VM access to the GPU device:

a) From the right pane, click **Configure** > **PCI Devices**.
b) Click the **Edit** icon.

   The **Edit PCI Device Availability** dialog box appears.

c) Select the device and check the checkbox next to the device.
d) Click **OK**.



CHAPTER 3

Getting Cisco Container Platform Software

This chapter contains the following topics:

Downloading the Software

Before you begin the installation, you need to download the required software assets.

Step 1 Go to the Product Support Page of Cisco Container Platform.

Step 2 Under **Support Documentation And Software**, click **Download Software**.

The **Software Download** page appears displaying the latest release assets.

Step 3 Log in using your Cisco username and password that is associated with a valid service contract.

Step 4 Download the Installer and Tenant images.

Unpacking the Software

Step 1 Browse to the directory where you have downloaded the software.

Step 2 Open the Shell command prompt and extract each `tar.gz` file.

Example

```
$ tar -zxvf kcp-vm-$VERSION.tar.gz
kcp-vm-$VERSION/
kcp-vm-$VERSION/ee.pem
kcp-vm-$VERSION/ccp_image_signing_release_v1_pubkey.der
kcp-vm-$VERSION/root_ca.pem
kcp-vm-$VERSION/kcp-vm-$VERSION.ova.signature
kcp-vm-$VERSION/kcp-vm-$VERSION.ova
kcp-vm-$VERSION/verify
kcp-vm-$VERSION/sub_ca.pem
kcp-vm-$VERSION/README
```

The `.ova` file contains the Cisco Container Platform image.

Verifying the Software

Before you begin

Ensure that your system has python 3.5.2 or later and OpenSSL installed.

Step 1 Browse to the directory where you have unpacked the software.

Step 2 Open the Shell command prompt and run the script to verify the software.

Note You must run the verification steps for each release image.

Example

```
$ ./verify --type release --signature kcp-vm-$VERSION.ova.signature --image kcp-vm-$VERSION.ova
Verifying sha512 hash of ./root_ca.pem
Successfully verfied sha512 hash of ./root_ca.pem
Verifying sha512 hash of ./sub_ca.pem
Successfully verfied sha512 hash of ./sub_ca.pem
Verifying root(./root_ca.pem) and subca(./sub_ca.pem)
Successfully verified root and subca.
Verifying cert(./ee.pem) against root(./root_ca.pem) and subca(./sub_ca.pem)
Successfully verified end entity cert.
Extracting pubkey(kcp-vm-$VERSION/ee.pubkey) from ./ee.pem
Successfully extrated public key to kcp-vm-$VERSION/ee.pubkey.
Verifying signature(kcp-vm-$VERSION.ova.signature) of kcp-vm-$VERSION.ova using
kcp-vm-$VERSION/ee.pubkey
Successfully verified signature.
```

CHAPTER 4

Installing Cisco Container Platform

Installing Cisco Container Platform is a three-step process:

- Importing Cisco Container Platform Tenant Base VM

  The Cisco Container Platform tenant base VM contains the container image and the files that are necessary to create the tenant Kubernetes clusters that are used for configuring monitoring, logging, container network interfaces (CNI), and persistent volumes.

- Deploying Installer VM, on page 19

  The Installer VM contains the VM image and the files for installing other components such as Kubernetes and the Cisco Container Platform application.

- Deploying Cisco Container Platform, on page 22

  The Cisco Container Platform Control Plane is set up using an installer UI. After the installer VM is switched on, the URL of the installer appears on the vCenter **Web console**.

Importing Cisco Container Platform Tenant Base VM

Before you begin

- Ensure that you have configured the storage and networking requirements. For more information, see HyperFlex Integration Requirements, on page 10 and Network Requirements, on page 7.
- Ensure that vSphere has an Enterprise Plus license, which supports DRS and vSphere HA.
- Recommend to use the *vSphere Web Client (Flash)* version of the vSphere Web Client.

Step 1 Log in to the **VMware vSphere Web Client** as an administrator.

Step 2 In the **Navigation** pane, right-click the cluster on which you want to deploy Cisco Container Platform, and then choose **Deploy OVF Template**.
The **Deploy OVF Template** wizard appears.

Step 3 In the **Select template** screen, perform these steps:

a) Click the **URL** radio button, and enter the URL of the Cisco Container Platform Tenant OVA.

Alternatively, click the **Local file** radio button, and browse to the location where the Cisco Container Platform tenant OVA is saved on your computer.

Note The format of the Tenant OVA filename is as follows:

```
ccp-tenant-image-x.y.z-ubuntuXX-a.b.c.ova
```

Where `x.y.z` corresponds to the version of Kubernetes and `a.b.c` corresponds to the version of Cisco Container Platform.

The Version Mapping Table, on page 61 provides the Cisco Container Platform version, Kubernetes version and image names mapping for each release.

b) Click **Next**.

Step 4 In the **Select name and location** screen, perform these steps:

a) In the **Name** field, enter a name for the Cisco Container Platform tenant base VM.

Note You need to note down the Cisco Container Platform tenant base VM name as you will need to specify it while creating a cluster.

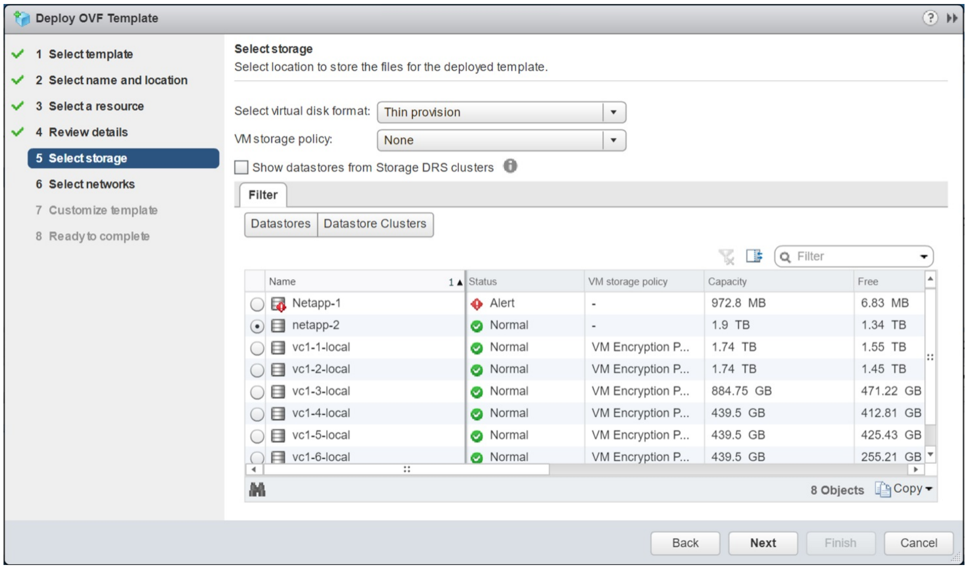
b) In the **Browse** tab, choose the data center where you want to deploy Cisco Container Platform.

c) Click **Next**.

Step 5 In the **Select a resource** screen, choose a cluster where you want to run the Cisco Container Platform tenant base VM, and then click **Next**.

Step 6 In the **Review details** screen, verify the Cisco Container Platform tenant base VM details, and then click **Next**.
The **Select storage** screen appears.

Figure 7: Select Storage Screen



Deploy OVF Template

- ✓ 1 Select template
- ✓ 2 Select name and location
- ✓ 3 Select a resource
- ✓ 4 Review details
- 5 Select storage
- 6 Select networks
- 7 Customize template
- 8 Ready to complete

Select storage
Select location to store the files for the deployed template.

Select virtual disk format: Thin provision
VM storage policy: None
□ Show datastores from Storage DRS clusters

Filter

Datastores | Datastore Clusters

	Name 1 ▲	Status	VM storage policy	Capacity	Free
○	Netapp-1	Alert	-	972.8 MB	6.83 MB
●	netapp-2	Normal	-	1.9 TB	1.34 TB
○	vc1-1-local	Normal	VM Encryption P...	1.74 TB	1.55 TB
○	vc1-2-local	Normal	VM Encryption P...	1.74 TB	1.45 TB
○	vc1-3-local	Normal	VM Encryption P...	884.75 GB	471.22 GB
○	vc1-4-local	Normal	VM Encryption P...	439.5 GB	412.81 GB
○	vc1-5-local	Normal	VM Encryption P...	439.5 GB	425.43 GB
○	vc1-6-local	Normal	VM Encryption P...	439.5 GB	255.21 GB

8 Objects

Step 7 In the **Select storage** screen, perform these steps:

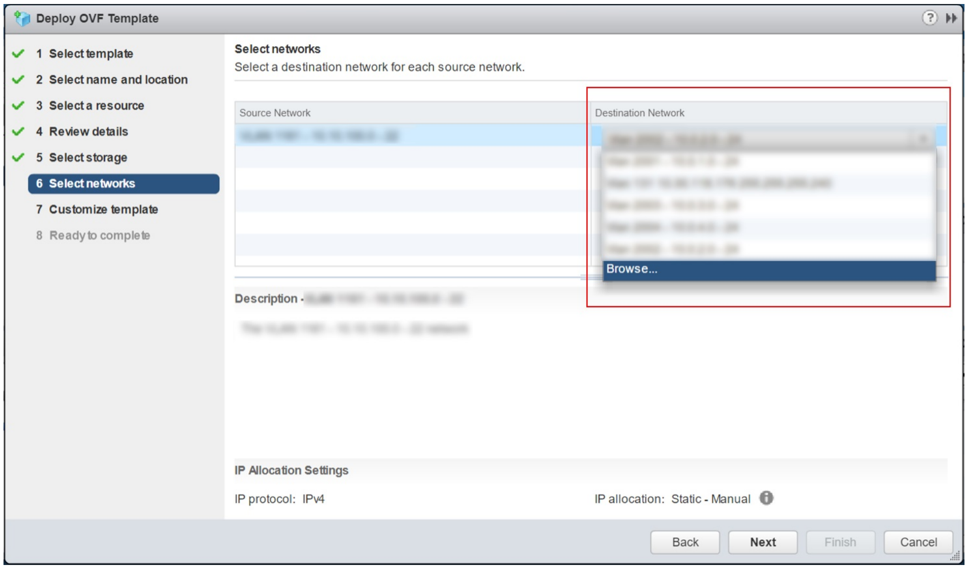
a) From the **Select virtual disk format** drop-down list, choose **Thin Provision** to allocate storage on demand.

b) In the **Filters** tab, choose a destination datastore for the Cisco Container Platform tenant base VM.

c) Click **Next**.

The **Select networks** screen appears.

Figure 8: Select Networks Screen



Select networks

Select a destination network for each source network.

Source Network	Destination Network

Description -

IP Allocation Settings

IP protocol: IPv4

IP allocation: Static - Manual

Step 8 In the **Select networks** screen, perform these steps:

a) From the **Destination Network** column, choose a network for each source network that is available in the Cisco Container Platform tenant base VM.

b) Click **Next**.

Step 9 In the **Customize template** screen, click **Next**.

Step 10 In the **Ready to complete** screen, verify the Cisco Container Platform tenant base VM settings, and then click **Finish**.
The Cisco Container Platform tenant base VM import takes few minutes to complete.

Note You can leave the tenant base VM powered off and continue to Deploying Installer VM.

Deploying Installer VM

Before you begin

Note This deployment is for new installations of Cisco Container Platform. For upgrades, see Upgrading Cisco Container Platform, on page 31.

Ensure that you have imported the Version Mapping Table during the Importing Cisco Container Platform Tenant Base VM.

Step 1 Log in to the **VMware vSphere Web Client** as an administrator.

Step 2 In the **Navigation** pane, right-click the cluster on which you want to deploy Cisco Container Platform, and then choose **Deploy OVF Template**.
The **Deploy OVF Template** wizard appears.

Step 3 In the **Select template** screen, perform these steps:

a) Click the **URL** radio button, and enter the URL of the Installer OVA.

Alternatively, click the **Local file** radio button, and browse to the location where the Installer OVA is saved on your computer.

Note The format of the Installer OVA filename is as follows:

```
kcp-vm-x.y.z.ova
```

Where `x, y, z` corresponds to the major, minor, and patch release of Cisco Container Platform.

b) Click **Next**.

Step 4 In the **Select name and location** screen, perform these steps:

a) In the **Name** field, enter a name for the installer VM.
b) In the **Browse** tab, choose the data center where you want to deploy Cisco Container Platform.
c) Click **Next**.

Step 5 In the **Select a resource** screen, choose the cluster where you want to run the installer VM, and then click **Next**.

Step 6 In the **Review details** screen, verify the template details, and then click **Next**.

Step 7 In the **Select storage** screen, perform these steps:

a) From the **Select virtual disk format** drop-down list, choose **Thin Provision** to allocate storage on demand.
b) In the **Filters** tab, choose a destination datastore to store the installer VM.
c) Click **Next**.

Step 8 In the **Select networks** screen, perform these steps:

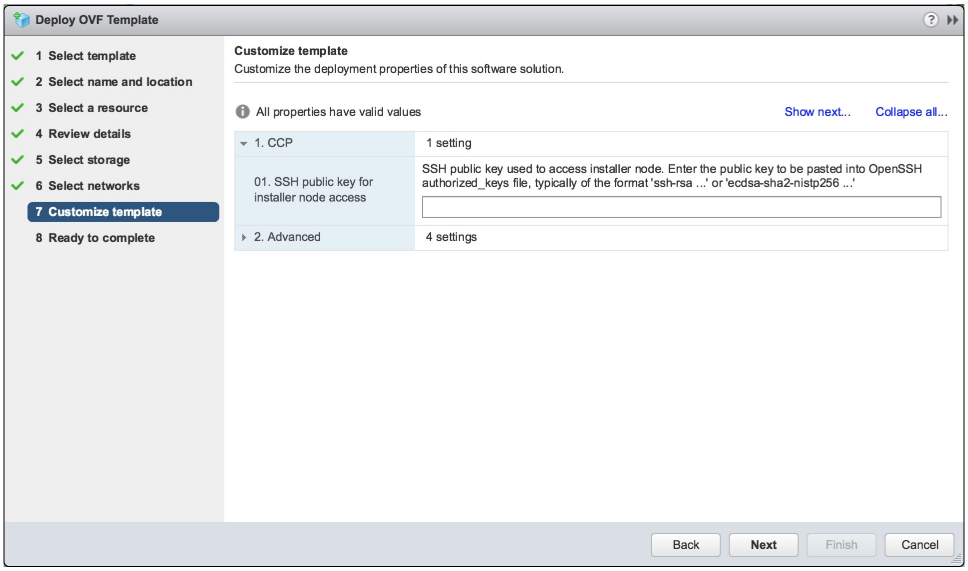
a) From the **Destination Network** column, choose a network for each source network that is available in the installer VM.

Note The selected network must have access to vCenter and the tenant VM networks.

b) Click **Next**.

The **Customize template** screen appears.

Figure 9: Customize Template Screen



Customize template

Customize the deployment properties of this software solution.

All properties have valid values

▾ 1. CCP	1 setting
01. SSH public key for installer node access	SSH public key used to access installer node. Enter the public key to be pasted into OpenSSH authorized_keys file, typically of the format 'ssh-rsa ...' or 'ecdsa-sha2-nistp256 ...'
▸ 2. Advanced	4 settings

Step 9 In the **Customize template** screen, enter the following optional parameters to customize the deployment properties:

a) Expand **CCP**, in the **SSH public key for installer node access** field, enter an ssh public key.

You can use this key to ssh to the installer VM with the username `ccpuser`.

Note
- Ensure that you enter the public key in a single line.
- If you do not have an SSH key pair, you can generate it using the **ssh-keygen** command.
- 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.

b) Expand **Advance** and enter the optional fields as necessary.

In the **CIDR for Kubernetes pod network** field, `192.168.0.0/24` is displayed as the default pod network CIDR of the Kubernetes cluster for the installer. If the CIDR IP addresses conflict with the tenant cluster VM network or the vCenter network, you need to set a different value for the CIDR.

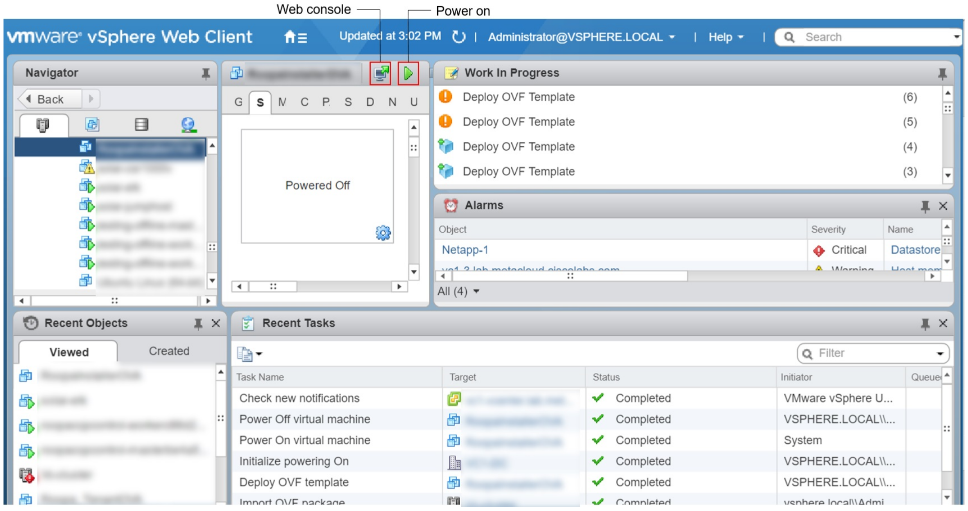
This CIDR is the single large CIDR from which smaller CIDRs are automatically allocated to each node for allocating IP addresses to the pods in the Kubernetes cluster. For more information, refer to https://kubernetes.io/docs/concepts/cluster-administration/networking/.

c) Click **Next**.

Step 10 In the **Ready to complete** screen, verify the installer VM deployment settings, and then click **Finish**.

Step 11 Click the **Power on** button to switch on the VM.

Figure 10: Switching on Installer VM



Work In Progress	
Deploy OVF Template	(6)
Deploy OVF Template	(5)
Deploy OVF Template	(4)
Deploy OVF Template	(3)

Object	Severity	Name
Netapp-1	Critical	Datastore

All (4)

Task Name	Target	Status	Initiator	Queue
Check new notifications	[redacted]	Completed	VMware vSphere U...	
Power Off virtual machine	[redacted]	Completed	VSPHERE.LOCAL\\...	
Power On virtual machine	[redacted]	Completed	System	
Initialize powering On	[redacted]	Completed	VSPHERE.LOCAL\\...	
Deploy OVF template	[redacted]	Completed	VSPHERE.LOCAL\\...	
Import OVF package	[redacted]	Completed	vsphere.local\\Admi	

Once the installer VM is switched on, the installer UI takes a few minutes to become ready. You can view the status of the Installer UI using the **Web console** of vCenter. When the installer UI is ready, you can access it using the URL from the **Web console**.

You can use the ssh private key to access the Installer, control plane VMs, or the tenant cluster VMs. However, logging into these VMs using a username and password is not supported.

Caution After Deploying Cisco Container Platform, do not change the location of the Control Plane VMs by modifying the datacenter or folder location **in vSphere**. Changing these settings will adversely impact the management of clusters.

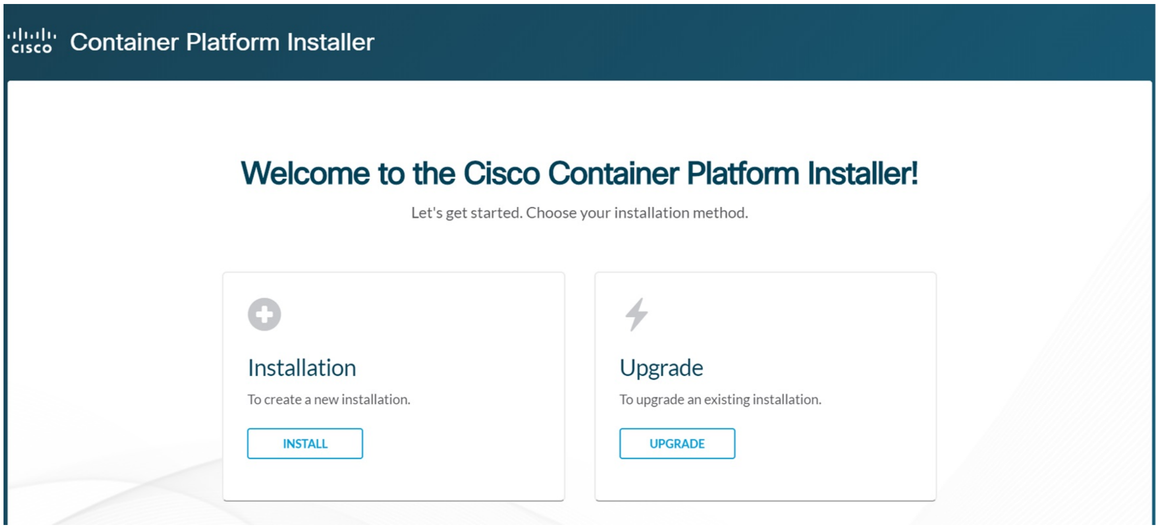
Deploying Cisco Container Platform

The Cisco Container Platform Control Plane is set up using an installer UI. After the installer VM is switched on, the URL of the installer appears on the vCenter **Web console**.

Step 1 Obtain the URL from the vCenter **Web console** and use a browser to open the installer UI.

The **Welcome** screen appears.

Figure 11: Welcome Screen



Container Platform Installer

Welcome to the Cisco Container Platform Installer!

Let's get started. Choose your installation method.

Installation

To create a new installation.

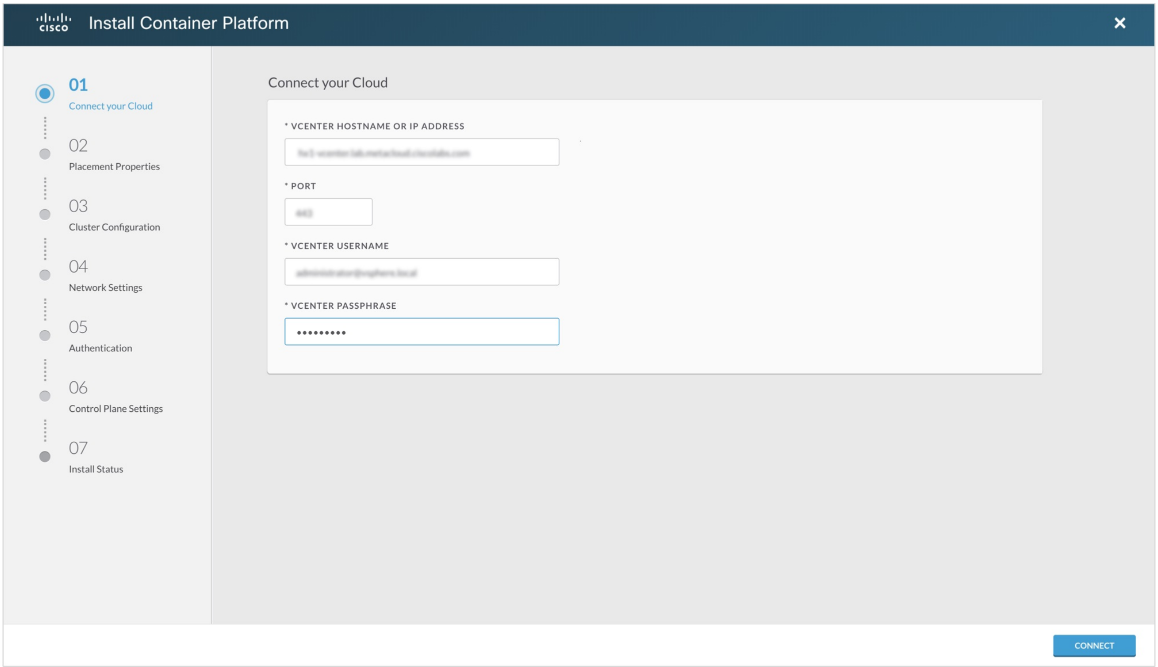
Upgrade

To upgrade an existing installation.

Step 2 Click **Install**.

The **Connect your Cloud** screen appears.

Figure 12: Connect your Cloud Screen



Install Container Platform

Connect your Cloud

* VCENTER HOSTNAME OR IP ADDRESS: [redacted]

* PORT: [redacted]

* VCENTER USERNAME: [redacted]

* VCENTER PASSPHRASE: ••••••••••

Step 3 In the **Connect your Cloud** screen, enter the following information:

a) In the **VCENTER HOSTNAME OR IP ADDRESS** field, enter the IP address of the vCenter instance that you want to use.

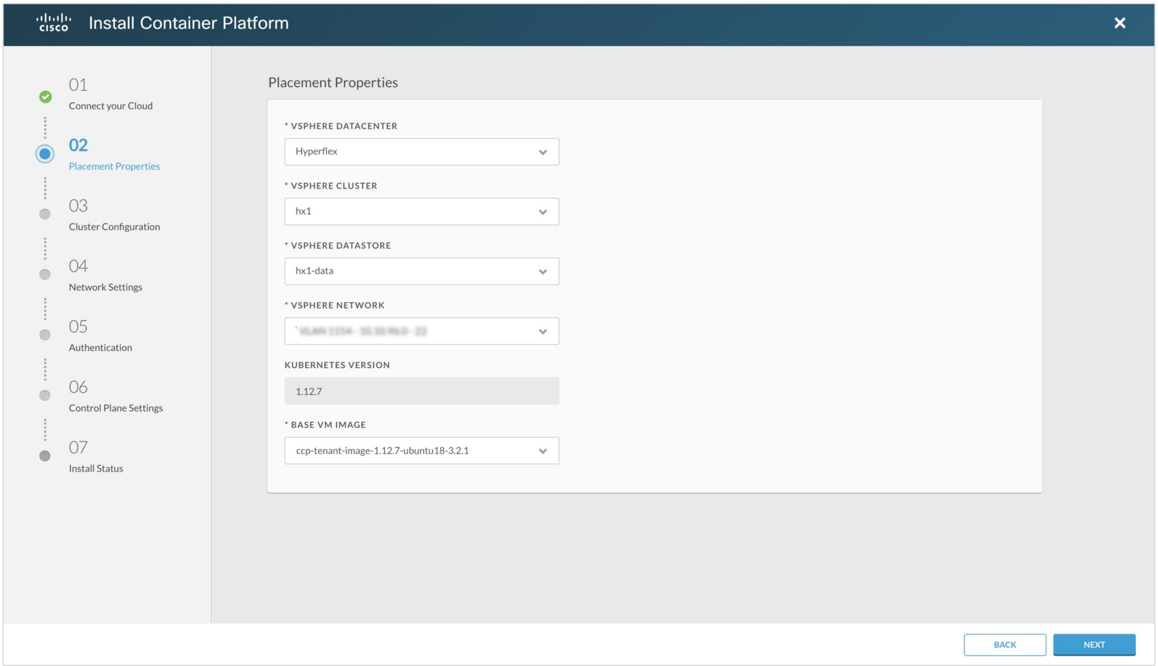
b) In the **PORT** field, enter the port number that your vCenter server uses.

Note The default port for vCenter is `443`.

c) In the **VCENTER USERNAME** field, enter the username of the user with administrator access to the vCenter instance.
d) In the **VCENTER PASSPHRASE** field, enter the passphrase of the vCenter user.
e) Click **CONNECT**.

The **Placement Properties** screen appears.

Figure 13: Placement Properties Screen



Placement Properties

* VSPHERE DATACENTER: Hyperflex

* VSPHERE CLUSTER: hx1

* VSPHERE DATASTORE: hx1-data

* VSPHERE NETWORK: [redacted]

KUBERNETES VERSION: 1.12.7

* BASE VM IMAGE: ccp-tenant-image-1.12.7-ubuntu18-3.2.1

Step 4 In the **Placement Properties** screen, enter the following information:

a) From the **VSPHERE DATACENTER** drop-down list, choose the datacenter.
b) From the **VSPHERE CLUSTER** drop-down list, choose the cluster.
c) From the **VSPHERE DATASTORE** drop-down list, choose the datastore.

Caution Do not use a datastore located in a nested folder or a Storage DRS (SDRS).

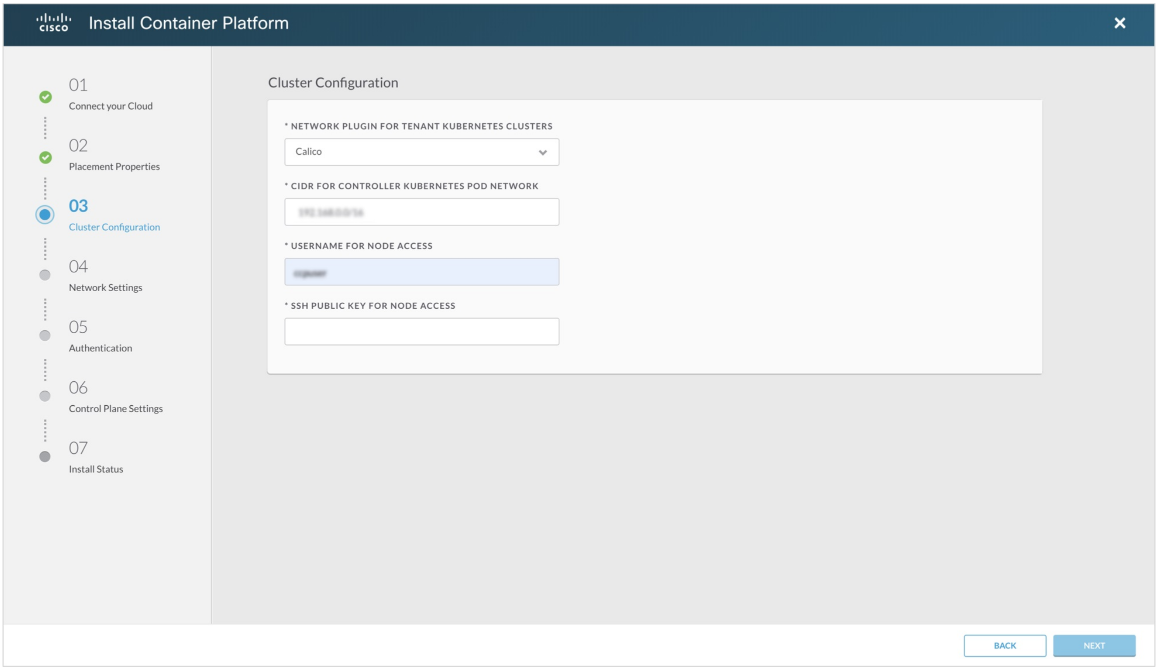
d) From the **VSPHERE NETWORK** drop-down list, choose the network.
e) In the **BASE VM IMAGE** field, enter the Cisco Container Platform tenant base VM name from Step 5 of the Importing Cisco Container Platform Tenant Base VM task.

Caution Do not select a VM name that is located in nested folder.

f) Click **NEXT**.

The **Cluster Configuration** screen appears.

Figure 14: Cluster Configuration Screen



Install Container Platform

01 Connect your Cloud

02 Placement Properties

03 Cluster Configuration

04 Network Settings

05 Authentication

06 Control Plane Settings

07 Install Status

Cluster Configuration

* NETWORK PLUGIN FOR TENANT KUBERNETES CLUSTERS: Calico

* CIDR FOR CONTROLLER KUBERNETES POD NETWORK: [redacted]

* USERNAME FOR NODE ACCESS: [redacted]

* SSH PUBLIC KEY FOR NODE ACCESS:

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

a) From the **NETWORK PLUGIN FOR TENANT KUBERNETES CLUSTERS** drop-down list, choose one of the following options for network connectivity:

- ACI-CNI
- Calico
- Contiv

Note For more information on the network plugins, see Container Network Interface Plugins, on page 4.

b) In the **CIDR FOR CONTROLLER KUBERNETES POD NETWORK** field, **192.168.0.0/16** is displayed as the default pod network CIDR of the Kubernetes cluster for the installer. If the CIDR IP addresses conflict with the tenant cluster VM network or the vCenter network, you need to set a different value for the CIDR.

Note This CIDR is the single large CIDR from which smaller CIDRs are automatically allocated to each node for allocating IP addresses to the pods in the Kubernetes cluster. For more information, refer to https://kubernetes.io/docs/setup/scratch/#network-connectivity.

c) In the **USERNAME FOR NODE ACCESS** field, enter the username of the user who can ssh into the Cisco Container Platform Control Plane nodes.

d) In the **SSH PUBLIC KEY FOR NODE ACCESS** field, enter an ssh public key.

You can use this key to ssh to the Control Plane nodes.

Note:

- Ensure that you enter the public key in a single line.
- If you do not have an SSH key pair, you can generate it using the **ssh-keygen** command.

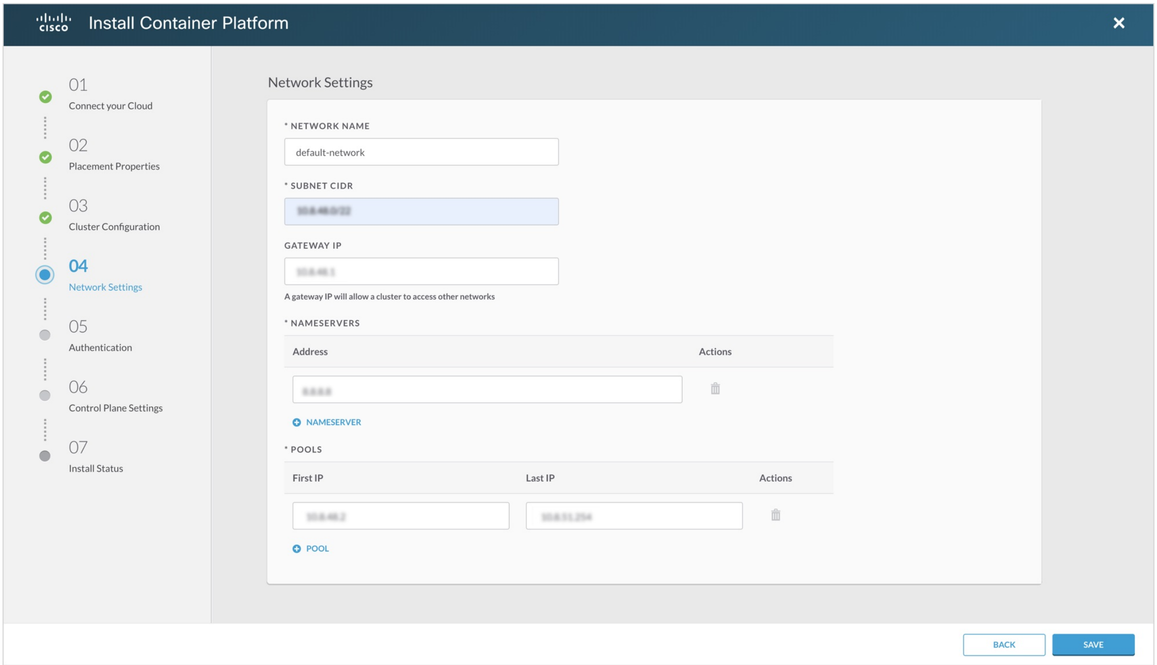
- Ensure that you use the Ed25519 or ECDSA format for the public key.

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

e) Click **NEXT**.

The **Network Settings** screen appears.

Figure 15: Network Settings Screen



Install Container Platform

01 Connect your Cloud
02 Placement Properties
03 Cluster Configuration
04 Network Settings
05 Authentication
06 Control Plane Settings
07 Install Status

Network Settings

* NETWORK NAME: default-network

* SUBNET CIDR: [redacted]

GATEWAY IP: [redacted]

A gateway IP will allow a cluster to access other networks

* NAMESERVERS

Address	Actions
[redacted]	🗑

NAMESERVER

* POOLS

First IP	Last IP	Actions
[redacted]	[redacted]	🗑

POOL

Step 6 In the **Network Settings** screen, enter the following information:

Note These network settings will be used to configure the Cisco Container Platform web interface.

a) In the **NETWORK NAME** field, enter the name of the network that you want to use.

b) In the **SUBNET CIDR** field, enter a CIDR for your subnet.

c) In the **GATEWAY IP** field, enter the gateway IP address that you want to use.

d) Under **NAMESERVER**, enter the IP address of the necessary DNS nameserver.

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

e) Under **POOLS**, enter a range for the VIP network pool by specifying the **First IP** and **Last IP** that are within the Subnet CIDR specified above. The VIP network pool range enables us to prevent provisioning of tenant clusters with IP address ranges from overlapping subnets.

   The IP address for the Control Plane is also allocated from this network pool range.

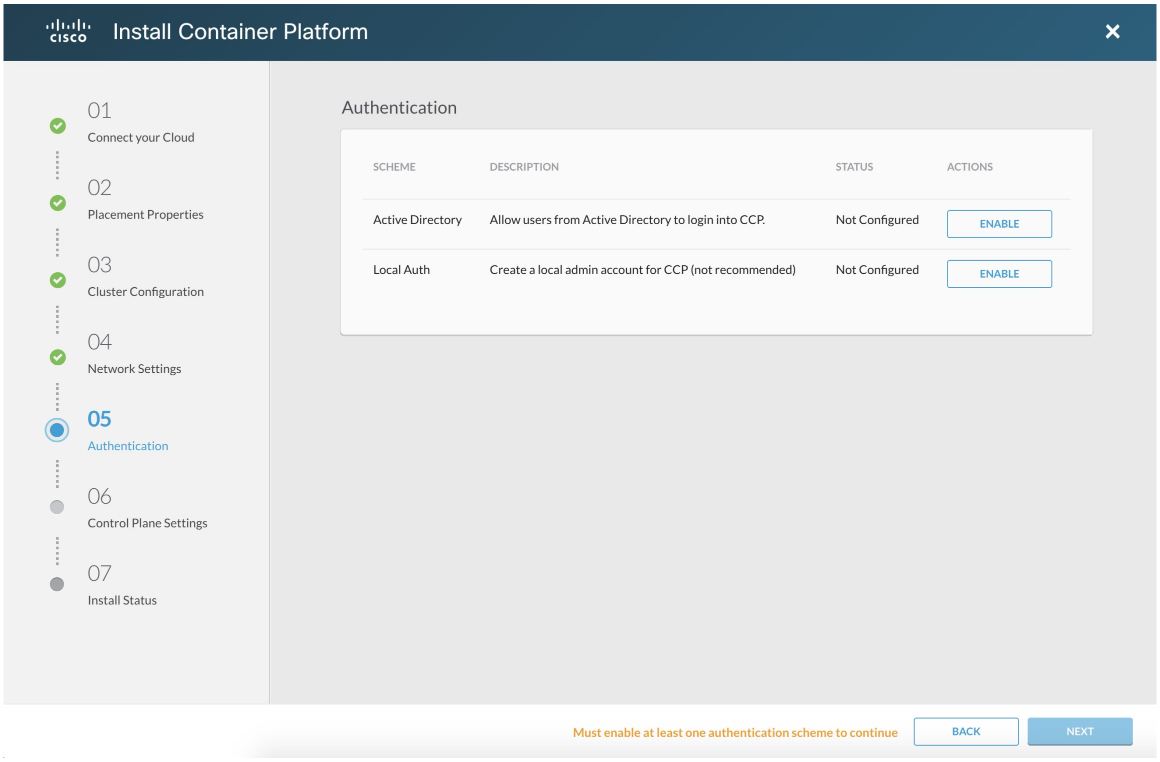
   You can click **+POOL** to enter multiple pools in the subnet.

   Note You must ensure that these IP addresses are not part of a DHCP pool.

f) Click **SAVE**.

The **Authentication** screen appears.

Figure 16: Authentication Screen



Install Container Platform

01 Connect your Cloud
02 Placement Properties
03 Cluster Configuration
04 Network Settings
05 Authentication
06 Control Plane Settings
07 Install Status

Authentication

SCHEME	DESCRIPTION	STATUS	ACTIONS
Active Directory	Allow users from Active Directory to login into CCP.	Not Configured	ENABLE
Local Auth	Create a local admin account for CCP (not recommended)	Not Configured	ENABLE

Must enable at least one authentication scheme to continue

Step 7 In the **Authentication** screen, click the **Enable** button next to the type of authentication that you want to configure.

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

a) If you have enabled **Active Directory**, specify the following information in the **Active Directory** screen:

1. Use the toggle button to enable or disable validation of Active Directory settings.

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

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

4. To establish a secure connection using SSL/TLS, enable **STARTTLS**.

5. To ensure security of your data, disable **SKIP CERTIFICATE VERIFICATION**.

   Caution If you enable **SKIP CERTIFICATE VERIFICATION**, TLS will accept any certificate presented by the AD server. In this mode, TLS is susceptible to data loss.

6. In the **BASE DN** field, enter the LDAP query to select the AD group that contains the users who must be granted the **User** role.

   For example:

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

   Note Base DN is the Distinguished Name for the base entity. All searches for users and groups will be scoped to this distinguished name.

7. In the **ADMIN GROUP QUERY** field, enter the LDAP query to select the AD group that contains the users who must be granted the **Administrator** role.

For example:

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

8. In the **SERVICE ACCOUNT DN** field, enter the service account domain name that is used for accessing the LDAP server.

9. In the **SERVICE ACCOUNT PASSPHRASE** field, enter the passphrase of the AD account.

10. Click **SAVE**.

b) If you have enabled **Local** (not recommended), specify the following information in the **LOCAL AUTHENTICATION** screen:

1. In the **LOCAL ADMIN USERNAME** field, enter the admin username.

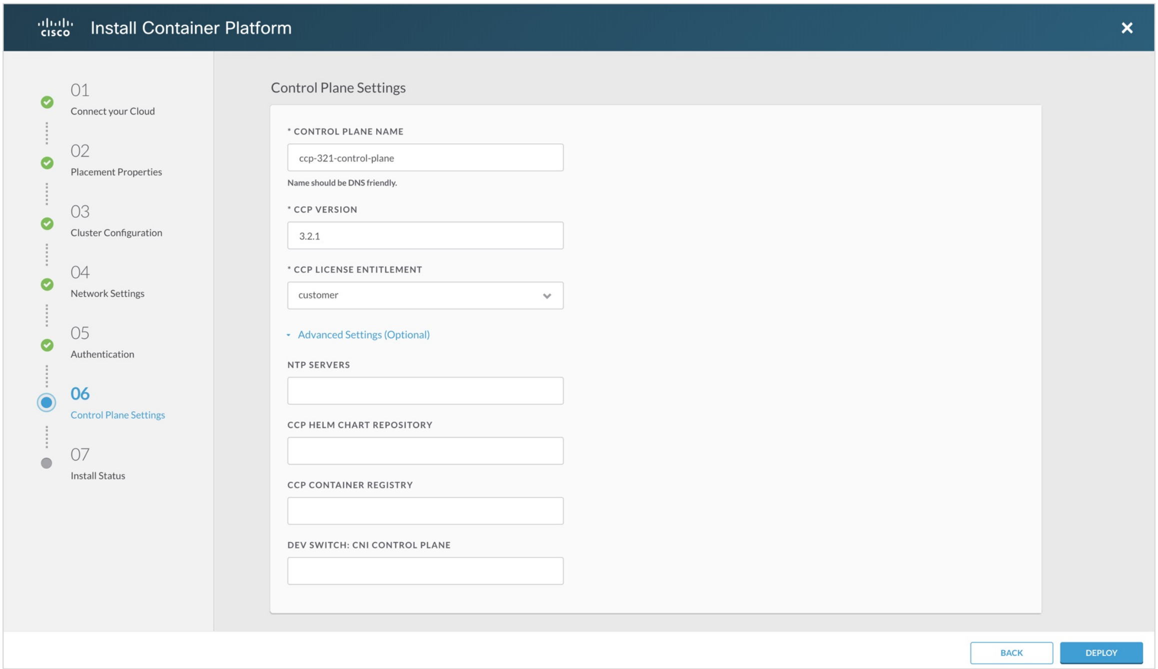
2. In the **LOCAL ADMIN PASSPHRASE** field, enter a passphrase.

3. In the **CONFIRM LOCAL ADMIN PASSPHRASE** re-enter the admin passphrase.

4. Click **SAVE**.

The **Control Plane Settings** screen appears.

Figure 17: Control Plane Settings Screen



Control Plane Settings

* CONTROL PLANE NAME

ccp-321-control-plane

Name should be DNS friendly.

* CCP VERSION

3.2.1

* CCP LICENSE ENTITLEMENT

customer

Advanced Settings (Optional)

NTP SERVERS

CCP HELM CHART REPOSITORY

CCP CONTAINER REGISTRY

DEV SWITCH: CNI CONTROL PLANE

Step 8 In the **Control Plane Settings** screen, enter the following information:

a) In the **CONTROL PLANE NAME** field, enter the name of the Cisco Container Platform cluster.

Note

- The cluster name must start with an alphanumeric character (a-z, A-Z, 0-9). It can contain a combination of hyphen (-) symbols and alphanumeric characters (a-z, A-Z, 0-9). The maximum length of the cluster name is 46 characters.
- Deployment of the installer VM fails if another Control Plane cluster with the same name already exists on the same datastore. You must ensure that you specify a unique name for the Control Plane cluster.

b) In the **CCP VERSION** field, enter the version of the Cisco Container Platform cluster.

c) From the **CCP LICENSE ENTITLEMENT** drop-down list, choose an entitlement option that indicates the type of Smart Licensing that you want to use.

Note The **Partner** option will only be used in conjunction with a **Not for Retail (NFR)** or **Trial** license.

d) Expand **Advanced Settings**, in the **NTP SERVERS** field, enter the list of any NTP servers in your environment.

This field is optional.

e) Click **DEPLOY** and then monitor the installation progress through the vCenter **Web console**.

Caution After deploying Cisco Container Platform, do not change the location of the Control Plane VMs by modifying the datacenter or folder location **in vSphere**. Changing these settings will adversely impact the management of clusters.

CHAPTER 5

Upgrading Cisco Container Platform

Upgrading Cisco Container Platform and upgrading tenant clusters are independent operations. You must upgrade the Cisco Container Platform to allow tenant clusters to upgrade. Specifically, tenant clusters cannot be upgraded to a higher version than the Control Plane. For example, if the Control Plane is at version 1.10, the tenant cluster cannot be upgraded to the 1.11 version.

Upgrading Cisco Container Platform is a three-step process:

Note Taking a snapshot of the VMs managed by Cisco Container Platform is currently unsupported and results in failures during upgrades.

You can update the size of a single IP address pool during an upgrade. However, we recommend that you plan ahead for the free IP address requirement by ensuring that the free IP addresses are available in the Control Plane cluster prior to the upgrade.

If you are upgrading from a Cisco Container Platform version:

- 3.1.x or earlier, you must ensure that at least five IP addresses are available.
- 3.2 or later, you must ensure that at least three IP addresses are available.

Upgrading Cisco Container Platform Tenant Base VM

You can follow the instructions in the Installing Cisco Container Platform, on page 17 > Importing Cisco Container Platform Tenant Base VM section.

Note The older tenant images are no longer required, you can delete them from your vCenter instance.

Deploying Upgrade VM

Follow the instructions in the Installing Cisco Container Platform, on page 17 > Deploying Installer VM section to deploy the latest VM.

It may take a few minutes for the deployment of the VM to complete. You can view the status of the upgrade task using the Web console of vCenter.

Note Depending on CNI usage, the port used to access Cisco Container Platform may change as part of the upgrade.

Upgrading Cisco Container Platform Control Plane

The Cisco Container Platform Control Plane is upgraded using an installer UI. After the installer VM is switched on, the URL of the installer appears on the vCenter **Web console**.

Step 1 Obtain the URL from the vCenter **Web console** and use a browser to open the installer UI.

Step 2 Click **Upgrade**.

Step 3 In the **Connect your Cloud** screen, enter the following information:

a) In the **VCENTER HOSTNAME OR IP ADDRESS** field, enter the IP address of the vCenter instance that you want to use.

b) In the **PORT** field, enter the port of the vCenter instance that you want to use.

c) In the **VCENTER USERNAME** field, enter the username of the user with administrator access to the vCenter instance.

d) In the **VCENTER PASSPHRASE** field, enter the passphrase of the vCenter user.

e) Click **CONNECT**.

Step 4 In the **Authenticate CCP** screen, enter the following information:

a) In the **EXISTING CISCO CONTAINER PLATFORM (CCP) URL** field, for accessing Cisco Container Platform in the following format:

https://*<CCP_IP_Address>:<Port>*

b) To establish a secure connection, enable **VERIFY SSL**.

c) In the **ADMIN USERNAME** field, enter the username for the **Administrator** user of the Cisco Container Platform Control Plane.

d) In the **ADMIN PASSPHRASE** field, enter the current passphrase for an **Administrator** user of the Cisco Container Platform Control Plane.

e) Click **CONNECT**.

Step 5 In the **Verify Network** screen, enter the following information:

a) In the **SUBNET CIDR** field, enter the actual CIDR of the VM network.

Note

- This network will be used for VM network configuration. You must ensure that the CIDR matches VM network configured on the vsphere.
- When the **SUBNET CIDR** is updated, the **GATEWAY IP** and **IP ADDRESS RANGE** are also updated accordingly.

b) In the **GATEWAY IP** field, enter the gateway IP address of the VM network.

Note Ensure that you enter the correct gateway IP address for the VM network. An incorrect gateway IP address causes failures during Control Plane upgrading.

c) Under **Nameservers** enter at atleast on DNS server addresss.

Note This nameserver(s) will be used in the DNS configuration of the Control Plane. You must ensure that Cisco Container Platform has access to this DNS server.

d) Under **POOLS**, enter the available IP address ranges that can be used for the Control Panel..

Note Do not adjust the address range if there are enough free IP addresses across the pools in the Control Plane's subnet to support the Control Plane upgrade.

You can extend the pool range as long as it does not overlap with any other pools in the subnet.

e) Click **NEXT**.

Step 6 In the **Control Plane Settings** screen, enter the following information:

a) In the **CONTROL PLANE NAME** field, enter the name of the Cisco Container Platform cluster.

Note You need to enter the same cluster name that you used during installation.

b) From the **VSPHERE DATACENTER** drop-down list, choose the datacenter.
c) From the **BASE VM IMAGE** drop-down list, choose the Cisco Container Platform tenant base VM name.
d) In the **CCP VERSION** field, enter the version of the Cisco Container Platform cluster.
e) From the **CCP LICENSE ENTITLEMENT** drop-down list, choose an entitlement option that indicates the type of Smart Licensing that you want to use.

Note The **Partner** option will only be used in conjunction with a **Not for Retail (NFR)** or **Trial** license.

f) Click **UPGRADE**.

The **Upgrade Status** screen appears.

After the upgrade is complete, click **LAUNCH** to access the upgraded Cisco Container Platform web interface.

CHAPTER 6

Uninstalling Cisco Container Platform

This chapter contains the following sections:

Uninstalling Cisco Container Platform

Uninstalling Cisco Container Platform removes all containers and services associated with it. You will no longer be able to create or manage tenant clusters on this Cisco Container Platform instance.

Step 1 Open the Cisco Container Platform web interface, log in to the Control Plane cluster using its VIP address, and then delete all the Kubernetes tenant clusters that belong to the Cisco Container Platform instance.

For more information on deleting Kubernetes clusters, refer to the *Cisco Container Platform User Guide*.

Step 2 Follow these steps to delete the Control Plane and installer node VMs:

a) In the vSphere web client, right-click the VM, choose **Power** > **Power off**, and then click **Yes** in the confirmation dialog box.

b) Right-click each VM and choose **Delete from Disk**.

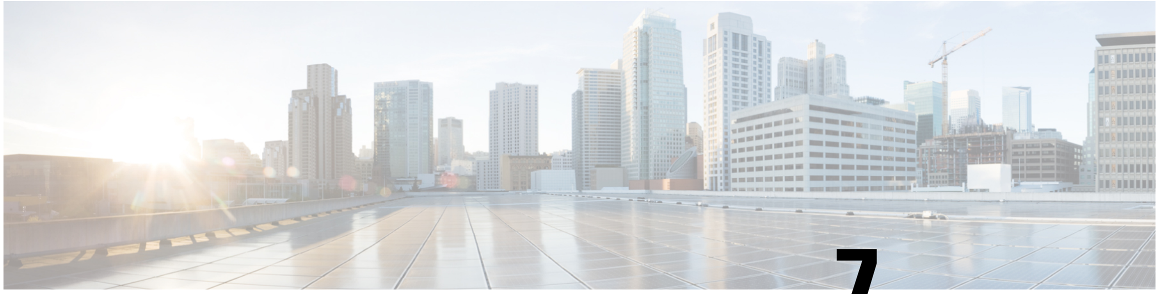
Step 3 Follow these steps to delete the Control Plane cluster data disks:

a) In the vSphere web client, choose **Home** > **Storage**.

b) From the left pane, choose the datastore that is used to install the Control Plane VMs. This is the same as the datastore to which the installer VM is imported to unless you have changed it in the installer UI.

c) If you have installed the Control Plane using the default name, right-click the folder name with the prefix **ccpcontrol** or if you have provided a different name to the Control Plane in the installer UI, right-click the folder with that name.

d) Choose **Delete File**.



CHAPTER 7

Back up and Restore Cisco Container Platform

This chapter contains the following topics:

Back up Cisco Container Platform

You can back up the Cisco Container Platform application data that pertains to the following components:

- Application users
- Virtualization providers
- Tenant clusters

The logging or monitoring data from Prometheus, Grafana, and the EFK stack is not included in the backup archive.

Ensure that the backup is up-to-date before restoring. Tenant cluster tasks such as creating, deleting, upgrading, and scaling tenant clusters, or altering the number of Load Balancer Virtual IP addresses create changes in the data that will not be present in the backup. If you perform such tasks, make a fresh backup of the control plane.

Note Restoring a Cisco Container Platform environment from an outdated backup may lead to unexpected IP address conflicts, unmanageable tenant clusters, or an unsuccessful restore.

Before you begin

Ensure that at least 6 consecutive IP addresses are available in the same pool where the Cisco Container Platform Control Plane is deployed.

When the target for a restore is a new cluster, you must ensure that additional free IP addresses are available to avoid conflicts with the IP addresses that are currently in use.

For more information on the requirement for additional free IP addresses, refer to the *Managing Networks* section of the *Cisco Container Platform User Guide*.

Step 1 Log in to the console of the master node of the Cisco Container Platform Control Plane.

Note

- Note down the IP addresses assigned to the VMs of the control plane.
- Note down the IP address of the ingress-nginx-controller service.

Step 2 Run the **backup-k8s-artifacts.sh** script to create a backup of the Kubernetes artifacts.

backup-k8s-aritfacts.sh

```
./ccp_related_files/backup-k8s-artifacts.sh
    backing up provider secrets
    backing up kubeconfig secrets
    backing up network resources
    backing up cluster resources
    backing up your-cluster
    Cleaning up artifacts.
    Backup artifacts available at k8s-artifacts-backup-2021-08-03.tar
```

Run the following command:

```
./ccp_related_files/backup-k8s-artifacts.sh
```

Step 3 Copy the tar file generated in Step 2 to a secure location outside of the current master node of the control plane.

Step 4 Run the **percona-backup.sh** script to create a backup of the percona database that contains data related to ccp-api, ccp-networks, and ccp-appdata. This data is used when restoring the control plane data.

percona-backup.sh

```
/ccp_related_files/percona-backup.sh ccp-percona-db-backup.tar
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
tar: Removing leading `/' from member names
tar: Removing leading `/' from member names

Retrieving allocated CCP Control Cluster IPs
Retrieving IP pools for control plane subnets
Retrieving all allocated IPs within the control subnets
Determining IPs in pools
Determining available IP ranges

--------------------------------------------------
IP Range 10.10.96.126 to 10.10.96.140 has 15 free IPs.     # <==

Backup is valid only if control plane operations are not performed
. No cluster create, upgrade, delete, scale out or in, change in
number of LoadBalancer VIPs, etc. Restoring a backup after changes
can result in IP collisions.

Percona DB backup saved to percona-db-backup/percona-db.tar
CCP Backup saved to ccp-percona-db-backup.tar


ATTENTION

Part of the backup includes a decryption key needed to restore the CPP Control Cluster.
Store this key securely and separately from the backup data.
```

```
The key is below:

8b2de348aca874342a1288c77fe821d6567fb619ab60a72e13af5f8f9dcbe3d21d669335d3651ec54dd6dc5dae8729b8a27f0cdcbfd9b302a23f2bb35a939be5
```

Run the following command:

```
./ccp_related_files/percona-backup.sh ccp-percona-db-backup.tar
```

Step 5 After running the backup script, note down the following information from the console:

- The valid IP pool ranges with enough free IPs to create a replacement Cisco Container Platform Control Plane.

  Save the IP address ranges. When you need to restore, you will start by installing a new control plane instance. The installer will ask for this information.

- The encryption key is needed to decrypt the backup data encryption key that is stored on your disk.

  Save the encryption key. You will need it whenever you restore the database to a new Cisco Container Platform Control Plane. For more information, see Restore Cisco Container Platform, on page 39. Without the encryption key, you will not be able to restore the Cisco Container Platform Control Plane cluster.

Step 6 Use the **scp** utility to copy the **ccp-percona-db-backup.tar** file from Step 4 to a secure location outside of the master node control plane.

Note Keep the backup archive under secure conditions. Anyone with access to it has administrative capabilities on all tenant clusters.

Restore Cisco Container Platform

You can restore a valid backup to a new Cisco Container Platform Control Plane instance of the same version that has control over all the existing Cisco Container Platform settings and tenant clusters. Do not restore the backup onto the same Cisco Container Platform Control Plane instance.

Restoring a backup archive to a new Control Plane will not restore the Ingress, kube-apiserver, and node IP addresses. They will remain the same as when you created the new Control Plane.

Before you begin

You will need these items which you noted during the Back up Cisco Container Platform:

- The encryption key.
- The IP Address Range setting from your old cluster.

Step 1 Power off the VMs that belong to the previous Control Plane instance.

Step 2 Install a new Cisco Container Platform control plane instance. Follow one of these approaches:

- **Approach 1:** Restore the Cisco Container Platform control plane to the same IP address.

  When installing the Cisco Container Platform control plane, you can assign the IP address of the original control plane to the new control plane.

During the installation of Cisco Container Platform, in the **Network Settings** step, enter the same Pool IP range as that used in the original instance of the Cisco Container Platform control plane.

For example, if the original control plane was deployed using the IP address range 10.96.96.6 to 10.96.96.10, you can use the same IP pool range, so that the new control plane will be deployed with the same IP addresses.

- **Approach 2:** Restore to the Cisco Container Platform control plane within the same Pool IP range as that used in the original instance, but using a different IP address than the original control plane.

  You must note down the IP address range suggested during the execution of the **percona-backup.sh script** in Back up Cisco Container Platform, on page 37.

  During the installation of Cisco Container Platform, in the **Network Settings** step, enter a Pool IP address that falls within the suggested IP range.

  For example, if the original control plane was deployed using the IP address range 10.96.96.6 to 10.96.96.10. The suggested IP range from the output of **percona-backup.sh** is 10.10.96.126 to 10.10.96.140, then you can provide any IP address range from the above range such as:

  - Entire range: 10.10.96.126 - 10.10.96.140
  - Partial IP address range: 10.10.96.135 - 10.10.96.140

Note You must ensure that the IP address range must have a minimum of six IP addresses available.

Step 3 After the new Cisco Container Platform control plane is up and running, copy the backup artifacts (.tar) files from your secure location to the master node of the control plane.

Step 4 Restore the Kubernetes artificates such as cluster providers, cluster kubeconfig, cluster objects, and network objects using the **restore-k8s-artifacts.sh** script.

restore-k8s-artifacts.sh

```
$ ./ccp_related_files/restore-k8s-artifacts.sh backup-1620255959.tar
 + echo 'extracting tar file'
 extracting tar file
 + tar -xf backup-1620255959.tar -C restore-1620258443/
 + echo 'restoring artifacts'
 restoring artifacts
 + echo 'restoring providers'
 restoring providers
 + kubectl apply -f restore-1620258443/k8s-provider.yaml
 secret/vsphere-provider-ec8eb851-6091-4812-9e4e-1668134b4001 created
 + echo 'restoring kubeconfig-secrets'
 restoring kubeconfig-secrets
 + kubectl apply -f restore-1620258443/kubeconfig-secrets.yaml
 secret/vsc-006-kubeconfig created
 + echo 'restoring network resources'
 restoring network resources
 + for i in {1..7}
 + kubectl apply -f restore-1620258443/001-k8s-network.yaml
 ipallocator.net.ccp.cisco.com/vsc-006-ipallocator created
 + sleep 2
 + for i in {1..7}
 + kubectl apply -f restore-1620258443/002-k8s-network.yaml
 clusternetwork.net.ccp.cisco.com/vsc-006 created
 + sleep 2
 + for i in {1..7}
 + kubectl apply -f restore-1620258443/003-k8s-network.yaml
 metallb.net.ccp.cisco.com/vsc-006-lb created
 + sleep 2
 + for i in {1..7}
+ kubectl apply -f restore-1620258443/004-k8s-network.yaml
ipaddress.net.ccp.cisco.com/vsc-006-lb-qc47p created
ipaddress.net.ccp.cisco.com/vsc-006-master-gro-aed3ea78b7-ens192 created
ipaddress.net.ccp.cisco.com/vsc-006-node-group-5e6bde4243-ens192 created
ipaddress.net.ccp.cisco.com/vsc-006-node-group-e295e46618-ens192 created
ipaddress.net.ccp.cisco.com/vsc-006-vip created
+ sleep 2
+ for i in {1..7}
+ kubectl apply -f restore-1620258443/005-k8s-network.yaml
cni.net.ccp.cisco.com/vsc-006-cni created
+ sleep 2
+ for i in {1..7}
+ kubectl apply -f restore-1620258443/006-k8s-network.yaml
netconfig.net.ccp.cisco.com/vsc-006-master-gro-aed3ea78b7 created
netconfig.net.ccp.cisco.com/vsc-006-node-group-5e6bde4243 created
netconfig.net.ccp.cisco.com/vsc-006-node-group-e295e46618 created
+ sleep 2
+ for i in {1..7}
+ kubectl apply -f restore-1620258443/007-k8s-network.yaml
nginxingress.net.ccp.cisco.com/vsc-006-ingress created
+ sleep 2
+ echo 'restoring cluster resources'
restoring cluster resources
+ kubectl apply -f restore-1620258443/k8s-clusters-resources.yaml
cluster.tlc.ccp.cisco.com/vsc-006 created
vspherecluster.vsphere.ccp.cisco.com/vsc-006 created
+ echo 'restoring complete!'
restoring complete!
+ echo 'SUCCESS!'
SUCCESS!
```

Run the following command:

```
./ccp_related_files/restore-k8s-artifacts.sh k8s-artifacts-backup-1621282525.tar
```

Step 5 Restore the percona-db database in one of the following ways:

- **Approach 1:** To install the Cisco Container Platform control plane to the **same IP address**, use the following **perona-restore.sh** script:

  perona-restore.sh

  ```
  $ ./ccp_related_files_new/percona-restore.sh ccp-percona-db-backup.tar
  Determining if restoring to same or new cluster
  Obtaining appdata from secret
  Restoring to the same CCP Control Cluster, no network data extraction required
  Restoring database, ignore 'command terminated with exit code 137' messages
  restart_mysql_pod
  Restarting mysql (1/2)
  pod "mysql-0" deleted
  Waiting for mysql (1/12)
  Restarting mysql (2/2)
  pod "mysql-0" deleted
  Waiting for mysql (1/12)
  rotate_user_credentials
  Rotating user credentials
  mysql: [Warning] Using a password on the command line interface can be insecure.
  mysql: [Warning] Using a password on the command line interface can be insecure.
  mysql: [Warning] Using a password on the command line interface can be insecure.
  mysql: [Warning] Using a password on the command line interface can be insecure.
  mysql: [Warning] Using a password on the command line interface can be insecure.
  restart_pods
  Restarting pods (by deleting)
  pod "kaas-api-74d784fffc-h9zjf" deleted
  pod "kaas-appdata-547d6ff889-kfwf5" deleted
  ```

```
pod "kaas-ccp-aks-operator-8559549855-l2sx8" deleted
pod "kaas-ccp-cluster-operator-bc4f85c9c-cqrbl" deleted
pod "kaas-ccp-eks-operator-5ff44786cd-tfcwk" deleted
pod "kaas-ccp-gke-operator-549888d77c-q6xzh" deleted
pod "kaas-ccp-vsphere-operator-757f9ff84c-xbzdv" deleted
pod "kaas-corc-5b8c797589-q8pfj" deleted
pod "kaas-cx-aes-key-job-3nnek-cgjzq" deleted
pod "kaas-dashboard-f5556f997-gj5jf" deleted
pod "kaas-network-77cd6df999-tvl68" deleted
pod "kaas-network-77cd6df999-zf8j2" deleted
pod "kaas-network-initdb-vuhu3-4tft9" deleted
pod "kaas-sddc-bfb47bd4-jk7cc" deleted
pod "kaas-slagent-6f5bc4dd8b-4f4xs" deleted
pod "kaas-slagent-q5lha-5wd2z" deleted
wait_for_pods
Wait for all kaas pods, especially CORC, appdata, and slagent
pod/kaas-api-74d784fffc-h57ss condition met
pod/kaas-appdata-547d6ff889-cdlhk condition met
pod/kaas-ccp-aks-operator-8559549855-q94sn condition met
pod/kaas-ccp-cluster-operator-bc4f85c9c-zhwvw condition met
pod/kaas-ccp-eks-operator-5ff44786cd-gmncj condition met
pod/kaas-ccp-gke-operator-549888d77c-f9x5j condition met
pod/kaas-ccp-vsphere-operator-757f9ff84c-n5q5h condition met
pod/kaas-corc-5b8c797589-dbx6j condition met
pod/kaas-dashboard-f5556f997-bhkrc condition met
pod/kaas-network-77cd6df999-frsk7 condition met
pod/kaas-network-77cd6df999-wkf9w condition met
pod/kaas-sddc-bfb47bd4-6wfkm condition met
pod/kaas-slagent-6f5bc4dd8b-fbqjq condition met
resore_appdata
Restoring existing appdata - in place of backup
Wrote output to /dev/null
Restoring to the same CCP Control Cluster, no network data alignment required
align_uuid
Aligning the network uuids of restored cluster
Retrieving current ccp-appdata contents
Creating a copy of current ccp-appdata contents
Retrieving current ccp-appdata contents
updating uuid of ipaddress associated with node ccpres-same-worker0182107096
updating uuid of ipaddress associated with node ccpres-same-workerdfb4de8f98
updating uuid of ipaddress associated with node ccpres-same-masterf00682a62b
updating uuid of ipaddress associated with node ccpres-same-workerfeac01c85e
updating uuid of loadbalancer ip 10.10.96.115
updating uuid of node_pool default-pool
updating uuid for ip 10.10.96.119
updating uuid for ip 10.10.96.118
updating uuid for ip 10.10.96.116
updating uuid for ip 10.10.96.117
Testing retrievel of CCP Control Cluster data
Wrote output to /dev/null
Testing retrievel of CCP Tenant data
Wrote output to /dev/null
secret "temp-appdata" deleted

SUCCESS!
```

Run the following command:

```
./ccp_related_files_new/percona-restore.sh ccp-percona-db-backup.tar
```

- **Approach 2:** To install the Cisco Container Platform control plane within the same IP address range configured for the original instance, but **using different IP addresses than the original control plane**, use the following **perona-restore.sh** script:

perona-restore.sh

```
$ ./ccp_related_files_new/percona-restore.sh ccp-percona-db-backup.tar
Determining if restoring to same or new cluster
Obtaining appdata from secret

===================================
Enter Decryption Key presented during backup (key will not echo to screen):
Starting data extraction utility pod kube-system:ccp-backed-up-db
pod/ccp-backed-up-db created
Waiting for data extraction utility pod to be Ready
Waiting for pod
Waiting for pod
Configure data extraction utility pod
Copying backup data from local into the utility pod
Extract data from the backup database
Unpacking data
Loading data
Configuring mysql
Waiting for mysqld
waiting for mysql...
waiting for mysql...
mysqld is alive
Retrieve subnet data from backup
Running subnet data query
Copy network data extracted from backup down to local
Delete utility pod
pod "ccp-backed-up-db" deleted
Retrieving new CCP Control Plane data
Retrieving new IP allocations for CCP Control Plane
Network checks pass, can continue with data restore
Dump new CCP Control Cluster network data
mysqldump: [Warning] Using a password on the command line interface can be insecure.
Copy restore network data locally
tar: Removing leading `/' from member names
Retrieve new CPP Control Cluster subnet data
Running subnet data query
mysql: [Warning] Using a password on the command line interface can be insecure.
Copy new subnet data to local
tar: Removing leading `/' from member names

Retrieving encrypted aes key from backup
Decrypting aes key
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
Updating Control Cluster aes key
Flag --export has been deprecated, This flag is deprecated and will be removed in future.
Warning: kubectl apply should be used on resource created by either kubectl create
--save-config or kubectl apply
secret/cx-aes-key configured
Restoring database, ignore 'command terminated with exit code 137' messages
Restarting mysql (1/2)
pod "mysql-0" deleted
Waiting for mysql (1/12)
Restarting mysql (2/2)
pod "mysql-0" deleted
Rotating user credentials
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
Restarting pods (by deleting)
pod "kaas-api-6c766d99c-thzlw" deleted
```

```
     pod "kaas-appdata-64b4656dd9-9mr62" deleted
     pod "kaas-ccp-aks-operator-5859d8f779-lb6qd" deleted
     pod "kaas-ccp-cluster-operator-b6d9cf9f7-zmnmv" deleted
     pod "kaas-ccp-eks-operator-64d46dcf5d-hr9ft" deleted
     pod "kaas-ccp-gke-operator-758ff99d65-b7vmp" deleted
     pod "kaas-ccp-vsphere-operator-7fb9b6768f-gr9q8" deleted
     pod "kaas-corc-64cd569468-wrbwn" deleted
     pod "kaas-cx-aes-key-job-ozb8s-9nnn6" deleted
     pod "kaas-dashboard-7f8fbbd7db-42bmc" deleted
     pod "kaas-network-7948cf457f-4dwrk" deleted
     pod "kaas-network-7948cf457f-krxkx" deleted
     pod "kaas-network-initdb-jaz9p-5gtcv" deleted
     pod "kaas-sddc-74978bfd56-zcdj5" deleted
     pod "kaas-slagent-5957db454d-bdz57" deleted
     pod "kaas-slagent-7vmip-lkh6l" deleted
     Wait for all kaas pods, especially CORC, appdata, and slagent
     pod/kaas-api-6c766d99c-r9dd7 condition met
     pod/kaas-appdata-64b4656dd9-vhchp condition met
     pod/kaas-ccp-aks-operator-5859d8f779-mjx58 condition met
     pod/kaas-ccp-cluster-operator-b6d9cf9f7-hkvdt condition met
     pod/kaas-ccp-eks-operator-64d46dcf5d-9g9pn condition met
     pod/kaas-ccp-gke-operator-758ff99d65-j5jqn condition met
     pod/kaas-ccp-vsphere-operator-7fb9b6768f-mm59p condition met
     pod/kaas-corc-64cd569468-kmwrt condition met
     pod/kaas-dashboard-7f8fbbd7db-crtsp condition met
     pod/kaas-network-7948cf457f-qzwp7 condition met
     pod/kaas-network-7948cf457f-tlz8d condition met
     pod/kaas-sddc-74978bfd56-g64lk condition met
     pod/kaas-slagent-5957db454d-6lgxp condition met
     Restoring existing appdata - in place of backup
     Wrote output to /dev/null
     Waiting for new CCP Control Cluster data to become available
     Wrote output to /dev/null
     Aligning subnet information in new CCP Control Cluster data to be compatable with tenant
data
     Retrieving new CCP Control Plane data
     Retrieving restored subnet configuration
     Updating appdata vip subnet identifier from 9a15ef9a-e2ff-482a-8a55-06a8bb0f224d to
b5c36f49-d449-42d2-8c95-6a1756a175c3
     Updating appdata vip subnet identifier from 9a15ef9a-e2ff-482a-8a55-06a8bb0f224d to
b5c36f49-d449-42d2-8c95-6a1756a175c3
     Done updating subnet data
     Copy the new CCP Control Cluster's network DB to the db pod
     Update the CCP Control cluster IPs to the new IPs
     Load copy of current ccp-network db
     mysql: [Warning] Using a password on the command line interface can be insecure.
     mysql: [Warning] Using a password on the command line interface can be insecure.
     mysqldump: [Warning] Using a password on the command line interface can be insecure.
     Create copied db of restored ccp-network db
     Create new work database to combine data
     mysql: [Warning] Using a password on the command line interface can be insecure.
     mysql: [Warning] Using a password on the command line interface can be insecure.
     See if we can add the created_at column
     mysql: [Warning] Using a password on the command line interface can be insecure.
     ERROR 1067 (42000) at line 2: Invalid default value for 'created_at'
     Already had created_at column
     See if we can add the owner column
     mysql: [Warning] Using a password on the command line interface can be insecure.
     ERROR 1060 (42S21) at line 2: Duplicate column name 'owner'
     Already had owner column
     Copy system IPs into work db
     mysql: [Warning] Using a password on the command line interface can be insecure.
     Checking that the correct number of IPs are present in work db
     mysql: [Warning] Using a password on the command line interface can be insecure.
```

```
Copy reconciled data back to current ccp-network db
mysql: [Warning] Using a password on the command line interface can be insecure.
Restore new owner data if available
mysql: [Warning] Using a password on the command line interface can be insecure.
Restore new created_at data if available
mysql: [Warning] Using a password on the command line interface can be insecure.
Drop temporary databases
mysql: [Warning] Using a password on the command line interface can be insecure.
Updated ccp-network system IP's for control cluster to match restored subnets and pools
Aligning the network uuids of restored cluster
Retrieving current ccp-appdata contents
Creating a copy of current ccp-appdata contents
Retrieving current ccp-appdata contents
updating uuid of ipaddress associated with node ccp610res-same-worker1a8e8814b1
updating uuid of ipaddress associated with node ccp610res-same-masterbb3eefc4d0
updating uuid of ipaddress associated with node ccp610res-same-workerd258545f91
updating uuid of ipaddress associated with node ccp610res-same-workereee85e4c64
updating uuid of loadbalancer ip 10.10.96.115
updating uuid of node_pool default-pool
updating uuid for ip 10.10.96.117
updating uuid for ip 10.10.96.116
updating uuid for ip 10.10.96.119
updating uuid for ip 10.10.96.118
Testing retrievel of CCP Control Cluster data
Wrote output to /dev/null
Testing retrievel of CCP Tenant data
Wrote output to /dev/null
secret "temp-appdata" deleted

SUCCESS!
```

Run the following command:

```
./ccp_related_files_new/percona-restore.sh ccp-percona-db-backup.tar
```

When prompted, enter the AES key that was generated during the creation of the percona-backup. For more information, see Back up Cisco Container Platform, on page 37.

Back Up Harbor Database

The database on Harbor tenant contains information such as user data and audit logs. This information can be backed up as a safety precaution before attempting a tenant upgrade on a Harbor tenant as the upgrade process may perform a database migration.

Note This backup process does not include docker images hosted on the Harbor registry.

Step 1 Log in to the console of the master node of Harbor tenant.

Step 2 Run the following command.

```
/opt/ccp/charts/harbor-db-backup.sh ./harbor_db_backup.sql default ccp-harbor
```

Step 3 Copy the `harbor_db_backup.sql` backup file to a secure location.

Restore Harbor Database

You can restore a valid Harbor database on a new or an existing Harbor tenant.

Step 1 Copy the backup from the secure location to Harbor tenant master.

```
scp ./harbor_db_backup.sql <harbor_tenant_master>:/tmp/harbor_db_backup.sql
```

Step 2 Log in to the console of the master node of Harbor tenant.

Step 3 Run the following command.

```
/opt/ccp/charts/harbor-db-restore.sh /tmp/harbor_db_backup.sql default ccp-harbor
```

APPENDIX A

Troubleshooting Cisco Container Platform

This appendix describes the problems that may occur during the installation and operation of Cisco Container Platform and the possible ways of resolving these problems.

It contains the following topics:

Multi-master vSphere or Openstack Tenant Cluster Fails to Upgrade

During the upgrade of a multi-master vSphere or Openstack tenant cluster, the upgrade fails because the etcd leader is reported missing.

The following error is displayed in the cloud-init logs of one of the master nodes:

```
Error: etcdserver: leader changed
```

Recommended Solution

Before upgrading a multi-master tenant cluster, run the following scripts to ensure that the etcd leader is available during the upgrade process:

Step 1 Run the get-etcd-pod-name.sh script on the control plane to get the name of the etcd pod.

```
get-etcd-pod-name.sh
```

```
 #! /bin/bash

 if [ -z "$1" ]; then
 echo "usage ./get-etcd-pod-name.sh <tenant-cluster-name>"
  exit 1
 fi

 CLUSTER=$1

 etcdNodes=$(kubectl get vsc $CLUSTER -o go-template='{{$array:=""}}{{range $key, $value :=
.status.masterGroupStatus.nodes}}{{ $output := printf "%s%s" "etcd-" $key }}{{ $array = printf "%s
%s" $array $output }}{{ end }}{{$array}}')

 echo "nodes in cluster $CLUSTER"
 IFS=' ' read -ra PODS <<<"$etcdNodes"
 for i in "${PODS[@]}"; do
     echo $i
 done

 if [ ${#PODS[@]} -eq 1 ]; then
     echo "cluster $CLUSTER has only 1 master node, no need of etcd migration to perform."
     exit 0
 fi

 echo "########################################################"
 # the master node at index 2 is the last master node to upgrade
 lastMasterNodeUID=$(kubectl get vsc $CLUSTER -o jsonpath={.status.masterGroupIndexUIDMap."2"})
 lastMasterNodeName="etcd-""$CLUSTER""-master-gro-""$lastMasterNodeUID"
 echo "migrate etcd-leader to etcd pod: $lastMasterNodeName"
 echo "use the above etcd pod name as input for script to execute on tenant cluster"
```

Sample output of this script for a multi-master vSphere cluster:

control-plane-output

```
 ccpuser@ccp800-master80bcc3ccdc:~/move-etcd-leader-20210402/control-plane$ ./get-etcd-pod-name.sh
vsc-multimaster-001
 nodes in cluster vsc-multimaster-001
 etcd-vsc-multimaster-001-master-gro-9a160a491e
 etcd-vsc-multimaster-001-master-gro-9e3ee81be6
 etcd-vsc-multimaster-001-master-gro-dd55508cf3
 ########################################################
 migrate etcd-leader to etcd pod: etcd-vsc-multimaster-001-master-gro-9a160a491e
 use the above etcd pod name as input for script to execute on tenant cluster
```

Step 2 Copy the following script **move-etcd-leader.sh** and the job template **move-leader-job-ccp.yaml** to the same directory on the master node.

move-etcd-leader.sh

```
 #!/bin/bash

 set -eo pipefail

 ETCD_POD=$1

 if [ -z "$ETCD_POD" ]; then
     echo "usage ./move-etcd-leader.sh <ETCD_POD_NAME>"
     exit 1
 fi

 set -u

 # info of new leader
 ETCD_POD_IP=$(kubectl get po $ETCD_POD -n kube-system -o jsonpath={.status.podIP})
 export NEWLEADERIP=$ETCD_POD_IP

 # info of current/old leader
 ETCD_PODS=$(kubectl get pods -n kube-system -l component=etcd,tier=control-plane -o
jsonpath="{.items[*]['.metadata.name']}")
 IFS=' ' read -ra PODS <<<"$ETCD_PODS"
 ETCD_LEADER_POD=""
 for i in "${PODS[@]}"; do
     STATUS=$(kubectl exec $i -n kube-system -- /bin/sh -c "etcdctl
--cacert=/etc/kubernetes/pki/etcd/ca.crt --cert=/etc/kubernetes/pki/etcd/peer.crt
--key=/etc/kubernetes/pki/etcd/peer.key --endpoints="https://127.0.0.1:2379" endpoint status")
     IFS="," read -r -a statusArray <<<"$STATUS"
     if [ "${statusArray[4]}" == " true" ]; then
         ETCD_LEADER_POD=$i
         break
     fi
 done

 if [ -z "$ETCD_LEADER_POD" ]; then
     echo "etcd leader pod not found, exiting"
     exit 1
 fi

 if [ "$ETCD_POD" == "$ETCD_LEADER_POD" ]; then
     echo "the chosen pod is already the leader, nothing to do"
     exit 0
 fi

 KUBENODE=$(kubectl get po $ETCD_LEADER_POD -n kube-system -o jsonpath={.spec.nodeName})
 ETCDIMAGE=$(kubectl get pod $ETCD_POD -n kube-system -o jsonpath={.status.containerStatuses[0].image})


 echo "future leader's IP address:" $NEWLEADERIP
 echo "node chosen to run the job on:" $KUBENODE

 ESCAPED_ETCDIMAGE=$(echo $ETCDIMAGE | sed 's/\//\\\//g')

 DATE=$(date +"%s")

 cat move-leader-job-ccp.yaml |
     sed 's/\$DATE/'$DATE'/g' |
     sed 's/\$NEWLEADERIP/'$NEWLEADERIP'/g' |
     sed 's/\$KUBENODE/'$KUBENODE'/g' |
     sed 's/\$ETCDIMAGE/'$ESCAPED_ETCDIMAGE'/g' | kubectl apply -f -

 JOB_POD=$(kubectl get pods -n kube-system -l job-name=moveetcdleader$DATE -o
jsonpath={.items[0].metadata.name})
 echo "the job's pod is kube-system/$JOB_POD"
 echo "sleeping for 5 seconds to allow the pod to start"
 for i in {1..5}; do
     sleep 1
     echo waited $i seconds
 done
 JOB_NAME=moveetcdleader$DATE

 kubectl logs -f -n kube-system --pod-running-timeout=60s -l job-name=$JOB_NAME

 EXIT_CODE=$(kubectl get pod -n kube-system -l job-name=$JOB_NAME -o
jsonpath="{.items[0].status.containerStatuses[0].state.terminated.exitCode}")
 if [ "$EXIT_CODE" == "0" ]; then
     echo "job finished running without errors"
     echo "deleting job"
     kubectl delete job -n kube-system $JOB_NAME
 fi
```

move-leader-job-ccp.yaml

```
 apiVersion: batch/v1
 kind: Job
 metadata:
 name: moveetcdleader$DATE
 namespace: kube-system
 spec:
 activeDeadlineSeconds: 90
 template:
     metadata:
     labels:
         jobtype: "forced-etcd-leader-migration"
     spec:
     containers:
         - name: etcdctl
         # replace with the etcd image present on the node
         image: $ETCDIMAGE
         command: ["/bin/sh", "-x", "-e", "-c"]
         args:
             - |
             ETCDCTLOPTS="--cacert=/etc/kubernetes/pki/etcd/ca.crt
--cert=/etc/kubernetes/pki/etcd/peer.crt --key=/etc/kubernetes/pki/etcd/peer.key
--endpoints=\"https://127.0.0.1:2379\""
             # Using files instead of pipes because /bin/sh doesnt support -o pipefile
             # Get this nodes endpoint status
             nodeStatus=$(etcdctl $ETCDCTLOPTS endpoint status)
             # etcdctl $ETCDCTLOPTS endpoint status > nodeStatus 2>&1
             # Check if the IsLeader boolean is false
             # Array output includes leading spaces; do not remove
             nodeIsLeader=$(echo "$nodeStatus" | cut -d ',' -f5)
             if [ "$nodeIsLeader" = " false" ]; then
                 # Not the leader, exit
                 echo "Not the leader"
                 exit 0
             fi
             # Get every node in the clusters endpoint status
             etcdctl $ETCDCTLOPTS endpoint status --cluster > clusterStatus 2>&1
             # Print the clusterStatus
             echo "==== start etcd nodes list ===="
             while read -r line; do
                 echo "$line"
             done < clusterStatus
             echo "==== end etcd nodes list ======="
             # Filter the list of status to find the first non-leader node
             while read -r line; do
             #IFS="," read -r -a statusArray <<< "$line"
                 leaderIp=$(echo "$line" | cut -d ',' -f1)
                 if [ "${leaderIp}" = "https://$NEWLEADERIP:2379" ]; then
                     # pull the node ID from the status for the node with this IP
                     # Read is used to remove leading spaces
                     #IFS=" " read -r transferID <<< "${statusArray[1]}"
                     transferID=$(echo "$line" | cut -d ',' -f2)
                     break
                 fi
             done < clusterStatus
             if [ -z "${transferID-}" ]; then
                     # No other etcd nodes to transfer leadership too
                     echo "No other members"
                     exit 1
             fi
             # Transfer the leadership from this node to the node we found above
             etcdctl $ETCDCTLOPTS move-leader $transferID
         volumeMounts:
            - mountPath: /etc/kubernetes/pki/etcd
            name: etcd-certs
    hostNetwork: true
    restartPolicy: Never
    nodeSelector:
        kubernetes.io/hostname: "$KUBENODE"
    tolerations:
        - effect: NoSchedule
        key: node.kubernetes.io/not-ready
        operator: Exists
        - effect: NoSchedule
        key: node-role.kubernetes.io/master
        operator: Exists
        - effect: NoSchedule
        key: node.kubernetes.io/unschedulable
        operator: Exists
        - effect: NoSchedule
        key: node.cloudprovider.kubernetes.io/uninitialized
        operator: Equal
        value: "true"
    volumes:
        - hostPath:
            path: /etc/kubernetes/pki/etcd
            type: Directory
        name: etcd-certs
```

Step 3 Run the **move-etcd-leader.sh** script on one of the master nodes of the tenant cluster.

Note The argument provided to this script is the output from the script executed in Step1.

Sample output of the script executed on the chosen master node of the tenant cluster:

tenant-cluster-master-node

```
 ccpuser@vsc-multimaster-001-master-gro-9a160a491e:~/move-etcd-leader-20210402/tenant-cluster$
./move-etcd-leader.sh etcd-vsc-multimaster-001-master-gro-9a160a491e
 future leader's IP address: 10.10.96.45
 node chosen to run the job on: vsc-multimaster-001-master-gro-9e3ee81be6
 job.batch/moveetcdleader1620435247 created
 the job's pod is kube-system/moveetcdleader1620435247-bppqs
 sleeping for 5 seconds to allow the pod to start
 waited 1 seconds
 waited 2 seconds
 waited 3 seconds
 waited 4 seconds
 waited 5 seconds
 + echo https://10.10.96.45:2379, 19a72824bde0e51, 3.4.10, 14 MB, false, false, 9, 15195, 15195,
 + leaderIp=https://10.10.96.45:2379
 + [ https://10.10.96.45:2379 = https://10.10.96.45:2379 ]
 + cut -d , -f2
 + echo https://10.10.96.45:2379, 19a72824bde0e51, 3.4.10, 14 MB, false, false, 9, 15195, 15195,
 + transferID= 19a72824bde0e51
 + break
 + [ -z  19a72824bde0e51 ]
 + etcdctl --cacert=/etc/kubernetes/pki/etcd/ca.crt --cert=/etc/kubernetes/pki/etcd/peer.crt
--key=/etc/kubernetes/pki/etcd/peer.key --endpoints="https://127.0.0.1:2379" move-leader 19a72824bde0e51

 Leadership transferred from 4a4da6f515bfe1ee to 19a72824bde0e51
 job finished running without errors
 deleting job
 job.batch "moveetcdleader1620435247" deleted
```

After both these scripts are executed successfully, you can perform the upgrade of the tenant cluster as usual.

Installation of Cisco Container Platform Fails

If installation of Cisco Container Platform fails, you can reattempt the installation.

Recommended Solution

Reboot the installer VM and then access the installer UI again.

In case you want to update an OVA parameter on the installer node, for example, update the **CIDR for Kubernetes pod network** parameter, you can follow these steps:

1. From the right pane of the vSphere Web Client, navigate to the installer VM.
2. Right-click the installer VM and choose **Power off**.

   The installer VM is turned off.
3. Right-click the installer VM and choose **Edit Settings**.

   The **Edit Settings** dialog box appears.
4. Click the **vApp Options** tab, and then open and update the required property value.
5. Click **OK**.
6. From the right pane, right-click the installer VM and choose **Power on**.

   The installer VM is turned on. After the installer VM is turned on, the URL of the installer appears on the vCenter **Web console**.
7. Obtain the URL from the vCenter **Web console** and use a browser to access the installer UI to continue with the installation.

Unable to Upgrade Cisco Container Platform due to Network Misconfiguration

When you enter a wrong IP address range for the Control Plane in the **Verify Network** screen of the **Upgrade** wizard, the following error message is appears:

```
Cannot patch address pool <uuid> with data: <some-data>
```

Recommended Solution

You must go back to the **Verify Network** screen of the **Upgrade** wizard and configure the IP address range for the Control Plane again.

For more information, see Upgrading Cisco Container Platform Control Plane, on page 32.

Unable to Deploy NGINX Ingress Controller Using Helm

When deploying the NGINX Ingress controller using Helm fails as RBAC is not configured in Helm, the following error message appears:

```
It seems the cluster it is running with Authorization enabled (like RBAC) and there is no
permissions for the ingress controller. Please check the configuration
```

Recommended Solution

As Cisco Container Platform uses RBAC for authentication, Helm also needs to be configured to use RBAC.

Enable the RBAC parameter in Helm using the following command:

```
--set rbac.create=true
```

Unable to Start NGINX Ingress Controller Pod

When kube-proxy is used, setting both the `controller.service.externalIPs` and `controller.hostNetwork` variables to `true` for the NGINX-Ingress chart results in an invalid configuration.

Both kube-proxy and NGINX uses port 80 for communication, causing a port conflict, and the NGINX Ingress controller pod is set to the `CrashLoopBackOff` state.

The following error message appears:

```
Port 80 is already in use. Please check the flag --http-port
```

Recommended Solution

Ensure that both the `controller.service.externalIPs` and `controller.hostNetwork` variables are not set to `true` at the same time.

Unable to Power on Worker VMs after a Shutdown

Worker VMs may fail to power on after a shutdown and the following error message appears:

```
File system specific implementation of LookupAndOpen[file] failed.
```

Recommended Solution

Follow these steps to resolve the problem:

1. From the left pane, click the VM that you want to power on.
2. From the right pane, from the **Actions** drop-down list, choose **Edit Settings**.

   The **Edit Settings** window displays the multiple hard disks of the VM.
3. Except for the primary hard disk (Hard disk 1), click each hard disk, and then click the **Remove** icon.

   Ensure that the **Delete files from datastore** check box is not checked.
4. Click **OK**.

Application Pods Crash When Using Contiv CNI in Tenant Clusters

When you use Contiv as the CNI for a tenant cluster, you need to ensure that the application pods that need HugePages must have the following section in the pod manifest. Otherwise, the pods may crash.

```
resources:
   limits:
     hugepages-2Mi: 512Mi
     memory: 512Mi
```

The preceeding section in the pod manifest limits 512 MB in memory for HugePages for the pod. It allocates 256 HugePages, with each HugePage having 2MB size.

HugePages are allocated to the pods only if you have enabled HugePages on the host. Otherwise, the HugePage allocation in the pod manifest is ignored by Kubernetes. The following table shows the Cisco Container Platform CNIs that use HugePages.

Cisco Container Platform CNI	Use HugePages
Contiv	Yes
ACI	No
Calico	No

Example of Allocating HugePages for Applications

Step 1 Check the total and free HugePages on the worker nodes. Each HugePage is 2048 KB in size.

```
$ grep -i huge /proc/meminfo
AnonHugePages: 0 kB
ShmemHugePages: 0 kB
HugePages_Total: 1024
HugePages_Free: 972
HugePages_Rsvd: 0
HugePages_Surp: 0
Hugepagesize: 2048 kB

$ sudo sysctl -a | grep -i huge
vm.hugepages_treat_as_movable = 0
vm.hugetlb_shm_group = 0
vm.nr_hugepages = 1024
vm.nr_hugepages_mempolicy = 1024
vm.nr_overcommit_hugepages = 0
```

Step 2 If the host has less HugePages, increase the HugePages allocation.

```
sudo su
echo 2048 > /proc/sys/vm/nr_hugepages

# Check the increased number of HugePages
cat /proc/sys/vm/nr_hugepages
grep -i huge /proc/meminfo
sudo sysctl -a | grep -i huge
```

Note You need to perform these steps on all the hosts.

Step 3 Create the `bookinfo.yaml` file that allocates HugePages to the `reviews-v1` pod.

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
name: reviews-v1
spec:
```

```
template:
    metadata:
    labels:
        app: reviews
        version: v1
    spec:
    containers:
    - name: reviews
        image: istio/examples-bookinfo-reviews-v1:1.5.0
        imagePullPolicy: IfNotPresent
        resources:
        limits:
            hugepages-2Mi: 512Mi
            memory: 512Mi
        ports:
        - containerPort: 9080
```

Step 4 Deploy `bookinfo.yaml` and check usage of HugePages.

```
$ kubectl create -f istio-$ISTIO_VERSION/samples/bookinfo/kube/bookinfo.yaml
deployment.extensions "reviews-v1" created

$ kubectl get pods | grep reviews
reviews-v1-6f56455f68-t6phs                                 1/1       Running   0          3m

# Check usage of HugePages by the pods
$ kubectl describe pod reviews-v1-6f56455f68-t6phs | grep -i '^Name:\|Image:\|huge\|mem'
Name:           reviews-v1-6f56455f68-t6phs
    Image:          istio/examples-bookinfo-reviews-v1:1.5.0
    hugepages-2Mi:  512Mi
    memory:         512Mi
    hugepages-2Mi:  512Mi
    memory:         512Mi

# Check usage of HugePages on each host
$ grep -i huge /proc/meminfo
AnonHugePages:         0 kB
ShmemHugePages:        0 kB
HugePages_Total:    1024
HugePages_Free:      972
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB

$ sudo sysctl -a | grep -i huge
vm.hugepages_treat_as_movable = 0
vm.hugetlb_shm_group = 0
vm.nr_hugepages = 1024
vm.nr_hugepages_mempolicy = 1024
vm.nr_overcommit_hugepages = 0
```

Step 5 Check the decrease of the `HugePages_Free` field in the output when the `reviews-v1` pod is using HugePages.

```
grep  -i huge /proc/meminfo
```

How to Create Sosreports

Sosreports are used by support engineers for troubleshooting customer support issues. They contain system log files, configuration details, and system information from your Cisco Container Platform environment.

Note

- For Control Plane issues, you need to run the sosreport from the Control Plane master VM, if available.
- For tenant cluster issues, you need to run the sosreport from the Control Plane master VM and the tenant plane master VM.
- For network issues impacting pods on a particular worker, you need to run the sosreport from the impacted tenant worker node.

Follow these steps to create an sosreport:

Step 1 ssh to the VM.

Step 2 Run sosreport on the node of your choice.

```
sudo sosreport
```

The sosreport is created and saved in the following location:

```
/tmp/sosreport-xxxxxx.tar.xz
```

Step 3 Validate the sosreport file using the following checksum:

```
xxxxxxxxx
```

Step 4 Securely transfer the sosreport file to your customer representative.
The file transfer method can vary depending on your deployment environment. For example, you can use Secure Copy (SCP) for Portable Operating System Interface systems (POSIX) and Windows Secure Copy (WinSCP) for windows clients. For more information, refer to Uploading Files to Cisco Technical Assistance Center (TAC).

Troubleshoot vSphere Operator in Cisco Container Platform

Follow these steps to troubleshoot the vSphere operator in Cisco Container Platform:

Step 1 Check the status of the pod, logs, and CRD of the vSphere operator.

```
$ kubectl get pods --all-namespaces | grep 'NAME\|vsphere-operator'
 NAMESPACE     NAME                                        READY   STATUS      RESTARTS   AGE
 default       kaas-ccp-vsphere-operator-788487bc68-h47dh  1/1     Running     2          156m

 kubectl logs kaas-ccp-vsphere-operator-788487bc68-h47dh --all-containers=true

 $ kubectl get crds | grep vsphereclusters
 vsphereclusters.vsphere.ccp.cisco.com      2019-07-15T18:16:45Z
```

```
$ kubectl get vsphereclusters
NAME           AGE
vhosakot-vs    2h
```

Step 2 Generate sosreport on the master node of the cluster running vSphere operator.

See also, How to Create Sosreports, on page 56.

```
cd <sosreport directory>/sos_commands/kubernetes/
cat vsphereclusters.vsphere.ccp.cisco.com
```

Step 3 Check the `status` of `VsphereCluster` CR in the sosreport.

Troubleshoot Net Tinker in Cisco Container Platform

Follow these steps to troubleshoot the Net Tinker operator in Cisco Container Platform:

Step 1 Check the status of the pod, logs, and CRDs of the net tinker.

```
$ kubectl get pods --all-namespaces | grep 'NAME\|tinker'
 NAMESPACE     NAME                                     READY    STATUS      RESTARTS    AGE
 default       ccp-tinker-manager-85cf7fffd5-mnc24      2/2      Running     0           158m

 kubectl logs ccp-tinker-manager-85cf7fffd5-mnc24 --all-containers=true

 $ kubectl get crds | grep net.ccp.cisco.com
 clusternetworks.net.ccp.cisco.com            2019-07-15T18:17:15Z
 cnis.net.ccp.cisco.com                       2019-07-15T18:17:15Z
 ipaddresses.net.ccp.cisco.com                2019-07-15T18:17:15Z
 ipallocators.net.ccp.cisco.com               2019-07-15T18:17:15Z
 metallbs.net.ccp.cisco.com                   2019-07-15T18:17:15Z
 netconfigs.net.ccp.cisco.com                 2019-07-15T18:17:15Z
 nginxingresses.net.ccp.cisco.com             2019-07-15T18:17:15Z

 kubectl get clusternetworks,cnis,ipaddresses,ipallocators,metallbs,netconfigs,nginxingresses
```

Step 2 Generate sosreport on the master node of the cluster running net tinker.

See also, How to Create Sosreports, on page 56.

```
cd <sosreport directory>/sos_commands/kubernetes/

 # find . | grep
'clusternetworks\|cnis\|ipaddresses\|ipallocators\|metallbs\|netconfigs\|nginxingresses'
 ./cnis.net.ccp.cisco.com
 ./clusternetworks.net.ccp.cisco.com
 ./netconfigs.net.ccp.cisco.com
 ./ipaddresses.net.ccp.cisco.com
 ./ipallocators.net.ccp.cisco.com
 ./nginxingresses.net.ccp.cisco.com
 ./metallbs.net.ccp.cisco.com

 find . | grep 'clusternetworks\|cnis\|ipaddresses\|ipallocators\|metallbs\|netconfigs\|nginxingresses'
 | xargs cat
```

Step 3 Check `status` of CRs of the net tinker in the sosreport.

Unable to Delete EKS Clusters Properly

Orphaned AWS resources in your environment can cause errors when an EKS cluster does not get cleaned up properly. You can use the `ccpeksctl` utility library to delete the orphaned AWS resources.

The binary `ccpeksctl` is located in the root directory of the `ccp-eks-operator` pod.

Recommended Solution

Ensure that you have access to the Cisco Container Platform control plane nodes.

Follow these steps to execute the binary:

1. Identify the EKS operator pod deployed in the Kubernetes cluster.

   ```
   $ kubectl get pods
   ```

2. Access and execute the binary from the pod.

   ```
   $ kubectl exec ccp-eks-operator-7fd7cf9646-gjw27 -- ./ccpeksctl -help
   ```

Examples

- Display the documentation of the arguments used in the `ccpeksctl` utility.

```
    $ ./ccpeksctl -help
     usage ccpeksctl ARGS
           -help          Print this help
           -dryrun        Optional. Default value to false.
           -operation     Default value "list". Allowed values ["list","delete"].
           -uuid          Required. UUID value from the eks CR.
                                  This value can be obtained from "Tags" section for
any of the AWS resources provisioned by Cisco Container Platform. Check the value for
key "ccp-cluster-id".
           -provider      Required. Kubernetes secret name stored containing aws
credentials.
           -region        Optional. Default value to "us-west-2" AWS region where the
resources are located.
```

- List the AWS resources with a given UUID.

```
    $ ./ccpeksctl -operation list -provider eks-provider-key -region us-west-2 -uuid
<uuid-id-from-eks-cr>

      Resource Type    Resource Status         Resource Name
      eks              ACTIVE                  eks-cluster-name
      ec2              running                 i-0ff7cc1e5404f9385
      ec2              running                 i-0ebe882d8e2b37bcc
      subnet                                   subnet-023f5744de56fb436
      subnet                                   subnet-05d5d3ab8f77f5084
      subnet                                   subnet-0a50312f228b576ec
      vpc              available               vpc-0e9c996a97f1ca69a
      rtb                                      rtb-0b654dc7e1a86073a
      rtb                                      rtb-09346eaf83b16094c
      rtb                                      rtb-09de12ff95ec94db1
      nat              available               nat-08e5c13eb71a5c6e4

      nat              available               nat-03e3e7d563844f571

      nat              available               nat-0b85fdb963b27494f

      igw                                      igw-05cd989ac1b06f05a
```

```
      sg                                    sg-0dfb973f19c8b7bbd

      eip            eipalloc-0797eab6930c175b5      100.21.234.225

      eip            eipalloc-0077504a99fafd655      35.155.169.195

      eip            eipalloc-07218a8786b742551      35.161.22.10

      cf             CREATE_COMPLETE
eks-cluster-name-group1-networkconfig-iam-bb8d3aeb-dd41-4968-8620-50ab7865fd49
```

- Test your delete command to verify the AWS resources that will be deleted.

  ```
    $ ./ccpeksctl -operation delete -dryrun -provider eks-provider-key -region us-west-2
  -uuid <uuid-id-from-eks-cr>
  ```

- Delete the AWS resources tagged with UUID.

  ```
     $ ./ccpeksctl -operation delete -provider eks-provider-key -region us-west-2 -uuid
  <uuid-id-from-eks-cr>
  ```

Unable to Manage Tenant Clusters due to a vCenter Password Update

If the password of the vCenter account used to install the control plane is changed, Cisco Container Platform cannot connect to the vCenter server. Cluster creation and cluster reboot failures may occur for existing control plane and tenant clusters.

Recommended Solution

For v3 tenant clusters, you need to change the vCenter password in the Cisco Container Platform control plane, and the changes are updated in the `cloud-config` secret. However, for v2 tenant clusters, in addition to updating the vCenter password in the Cisco Container Platform control plane, you need to update the `cloud-config` secret that stores the vCenter password.

Follow these steps to resolve the problem:

Step 1 Log in to Cisco Container Platform and go to **Infrastructure Providers**.

Step 2 Click **Edit** under actions and change the vCenter password to reflect the password of the vCenter account.

Step 3 Update the `cloud-config` secret that stores the vCenter password:

Note This step applies only for v2 tenant clusters.

a) SSH to the master node of the control plane.

b) Encode the new vCenter password.

```
$ echo -n "<vcenter-password>" | base64 -w 0
```

c) Open the `cloud-config` secret in a text editor.

```
$ kubectl -n kube-system edit secret <secret-name>
```

Where, the secret-name is `/etc/kubernetes/pki/cloud-config`

d) In the `cloud-config` secret, replace the vCenter password with the newly encoded password.

e) Restart the kubelet to use the changed password.

```
$ sudo systemctl restart kubelet.service
```

APPENDIX B

Version Mapping Table

This chapter contains the following topic:

Version Mapping Table

Cisco Container Platform Version	Kubernetes Version	Image Names
1.0.0	1.10	Control Plane Installer – kcp-vm-1.0.0.ova Tenant Image – ccp-tenant-image-1.10.1-1.0.0.ova
1.0.1	1.10	Control Plane Installer – kcp-vm-1.0.1.ova Tenant Image – ccp-tenant-image-1.10.1-1.0.1.ova
1.1.0	1.10	Control Plane Installer – kcp-vm-1.1.0.ova Tenant Image – ccp-tenant-image-1.10.1-1.1.0.ova
1.4.0	1.10	Control Plane Installer – kcp-vm-1.4.0.ova Tenant Image – ccp-tenant-image-1.10.1-1.4.0.ova
1.5.0	1.10	Control Plane Installer – kcp-vm-1.5.0.ova Tenant Image – ccp-tenant-image-1.10.1-ubuntu16-1.5.0.ova
2.0.1	1.10 1.11	Control Plane Installer – kcp-vm-2.0.1.ova Tenant Image (Kubernetes 1.10) – ccp-tenant-image-1.10.1-ubuntu16-2.0.0.ova Tenant Image (Kubernetes 1.11) – ccp-tenant-image-1.11.3-ubuntu18-2.0.0.ova

Cisco Container Platform Version	Kubernetes Version	Image Names
2.1.0	1.10 1.11	Control Plane Installer – kcp-vm-2.1.0.ova Tenant Image (Kubernetes 1.10) – ccp-tenant-image-1.10.1-ubuntu16-2.1.0.ova Tenant Image (Kubernetes 1.11) – ccp-tenant-image-1.11.3-ubuntu18-2.1.0.ova
2.2.2	1.10 1.11	Control Plane Installer – kcp-vm-2.2.2.ova Tenant Image (Kubernetes 1.10) – ccp-tenant-image-1.10.11-ubuntu16-2.2.2.ova Tenant Image (Kubernetes 1.11) – ccp-tenant-image-1.11.5-ubuntu18-2.2.2.ova
3.0.0	1.11 1.12	Control Plane Installer – kcp-vm-3.0.0.ova Tenant Image (Kubernetes 1.11) – ccp-tenant-image-1.11.5-ubuntu18-3.0.0.ova Tenant Image (Kubernetes 1.12) – ccp-tenant-image-1.12.3-ubuntu18-3.0.0.ova
3.1.0	1.11 1.12	Control Plane Installer – kcp-vm-3.1.0.ova Tenant Image (Kubernetes 1.11) – ccp-tenant-image-1.11.5-ubuntu18-3.1.0.ova Tenant Image (Kubernetes 1.12) – ccp-tenant-image-1.12.3-ubuntu18-3.1.0.ova
3.2.0	1.11 1.12	Control Plane Installer – kcp-vm-3.2.0.ova Tenant Image (Kubernetes 1.11) – ccp-tenant-image-1.11.5-ubuntu18-3.2.0.ova Tenant Image (Kubernetes 1.12) – ccp-tenant-image-1.12.3-ubuntu18-3.2.0.ova
4.0.0	1.12 1.13	Control Plane Installer – kcp-vm-4.0.0.ova Tenant Image (Kubernetes 1.12) – ccp-tenant-image-1.12.7-ubuntu18-4.0.0.ova Tenant Image (Kubernetes 1.13) – ccp-tenant-image-1.13.5-ubuntu18-4.0.0.ova
4.1.0	1.12 1.13	Control Plane Installer – kcp-vm-4.1.0.ova Tenant Image (Kubernetes 1.12) – ccp-tenant-image-1.12.7-ubuntu18-4.1.0.ova Tenant Image (Kubernetes 1.13) – ccp-tenant-image-1.13.5-ubuntu18-4.1.0.ova

Cisco Container Platform Version	Kubernetes Version	Image Names
4.2.0	1.12 1.13	Control Plane Installer – kcp-vm-4.2.0.ova Tenant Image (Kubernetes 1.12) – ccp-tenant-image-1.12.7-ubuntu18-4.2.0.ova Tenant Image (Kubernetes 1.13) – ccp-tenant-image-1.13.5-ubuntu18-4.2.0.ova
5.0.0	1.13 1.14	Control Plane Installer – kcp-vm-5.0.0.ova Tenant Image (Kubernetes 1.13) – ccp-tenant-image-1.13.10-ubuntu18-5.0.0.ova Tenant Image (Kubernetes 1.14) – ccp-tenant-image-1.14.6-ubuntu18-5.0.0.ova
5.1.0	1.13 1.14	Control Plane Installer – kcp-vm-5.1.0.ova Tenant Image (Kubernetes 1.13) – ccp-tenant-image-1.13.12-ubuntu18-5.1.0.ova Tenant Image (Kubernetes 1.14) – ccp-tenant-image-1.14.8-ubuntu18-5.1.0.ova
6.0.0	1.15 1.16	Control Plane Installer – kcp-vm-6.0.0.ova Tenant Image (Kubernetes 1.15) – ccp-tenant-image-1.15.6-ubuntu18-6.0.0.ova Tenant Image (Kubernetes 1.16) – ccp-tenant-image-1.16.3-ubuntu18-6.0.0.ova
6.1.0	1.15 1.16	Control Plane Installer – kcp-vm-6.1.0.ova Tenant Image (Kubernetes 1.15) – ccp-tenant-image-1.15.6-ubuntu18-6.1.0.ova Tenant Image (Kubernetes 1.16) – ccp-tenant-image-1.16.3-ubuntu18-6.1.0.ova
7.0.0	1.16 1.17	Control Plane Installer – kcp-vm-7.0.0.ova Tenant Image (Kubernetes 1.16) – ccp-tenant-image-1.16.12-ubuntu18-7.0.0.ova Tenant Image (Kubernetes 1.17) – ccp-tenant-image-1.17.6-ubuntu18-7.0.0.ova
8.0.0	1.17 1.18	Control Plane Installer – kcp-vm-8.0.0.ova Tenant Image (Kubernetes 1.17) – ccp-tenant-image-1.17.14-ubuntu18-8.0.0.ova Tenant Image (Kubernetes 1.18) – ccp-tenant-image-1.18.14-ubuntu18-8.0.0.ova

Cisco Container Platform Version	Kubernetes Version	Image Names
9.0.0	1.18 1.19	Control Plane Installer – kcp-vm-9.0.0.ova Tenant Image (Kubernetes 1.18) – ccp-tenant-image-1.18.20-ubuntu18-9.0.0.ova Tenant Image (Kubernetes 1.19) – ccp-tenant-image-1.19.13-ubuntu18-9.0.0.ova
9.0.1	1.18 1.19	Control Plane Installer – kcp-vm-9.0.1.ova Tenant Image (Kubernetes 1.18) – ccp-tenant-image-1.18.20-ubuntu18-9.0.1.ova Tenant Image (Kubernetes 1.19) – ccp-tenant-image-1.19.15-ubuntu18-9.0.1.ova

Note It is required that you use the latest Kubernetes version OVA for the Installing Cisco Container Platform.