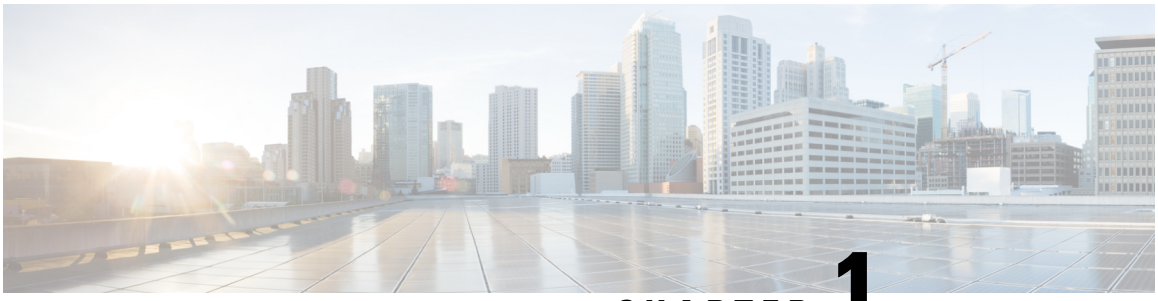
Cisco Container Platform 1.4.0 Installation Guide

First Published: 2018-07-31

Last Modified: 2018-08-02

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

CONTENTS



CHAPTER 1

Cisco Container Platform

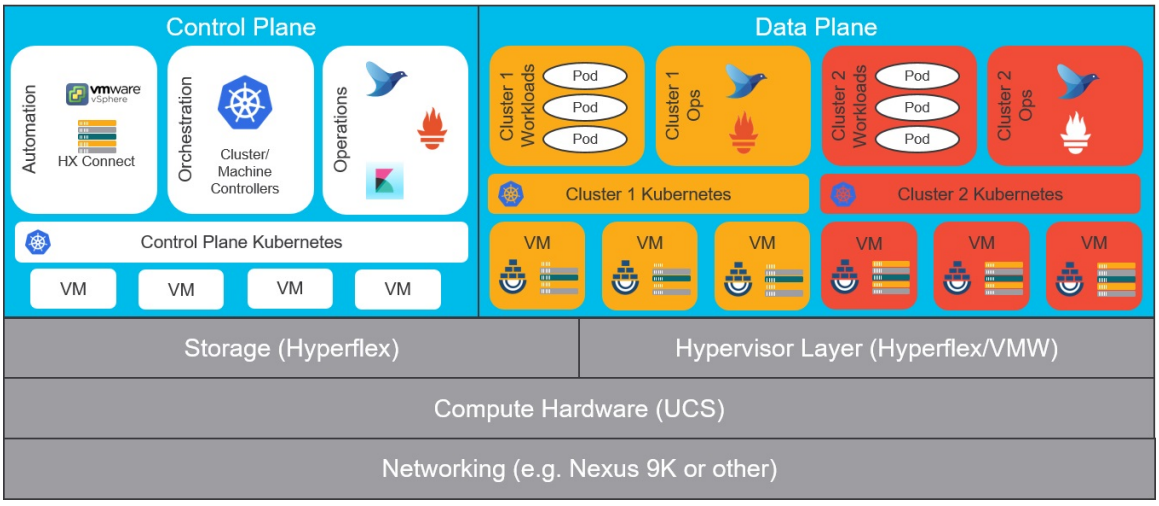
Cisco Container Platform is a turnkey, production grade, extensible platform to deploy and manage multiple Kubernetes clusters. It runs on 100% upstream Kubernetes. Cisco Container Platform offers seamless container networking, enterprise-grade persistent storage, built-in production-grade security, 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 deployment with HyperFlex and ACI integration.

Figure 1: Cisco Container Platform Architecture Overview



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.

At the bottom of the stack, there is an ACI fabric that consists of Nexus switches, Application Policy Infrastructure Controllers (APICs) and Fabric Interconnects (FIs). The next layer up is the UCS servers running the HyperFlex software. HyperFlex provides virtualized compute resources through VMware, and distributed storage resources through the HyperFlex converged data platform.

The next layer up is the Cisco Container Platform Control Plane and Data Plane. In the preceeding figure, Cisco Container Platform Control Plane runs on the four VMs on the left.

Kubernetes tenant clusters are preconfigured to support Persistent Volumes using vSphere Cloud Provider and FlexVolumes using HyperFlex volume plugin. Both implementations use the underlying replicated, highly available HyperFlex data platform for storage.

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
Container Network Interface (CNI)	ACI, Contiv, Calico
SDN	ACI
Container Storage	HyperFlex Flex Driver
Load Balancing	NGINX, Envoy
Service Mesh	Istio, Envoy
Monitoring	Prometheus, Grafana
Logging	Elasticsearch, Fluentd, and Kibana (EFK) stack

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. It can be used for quick evaluation of Cisco Container Platform. Calico is an integrated CNI plugin and is not fully supported under the Cisco commercial support agreement.
- Contiv (Tech Preview) is a user space switch optimized for high performance and scale.

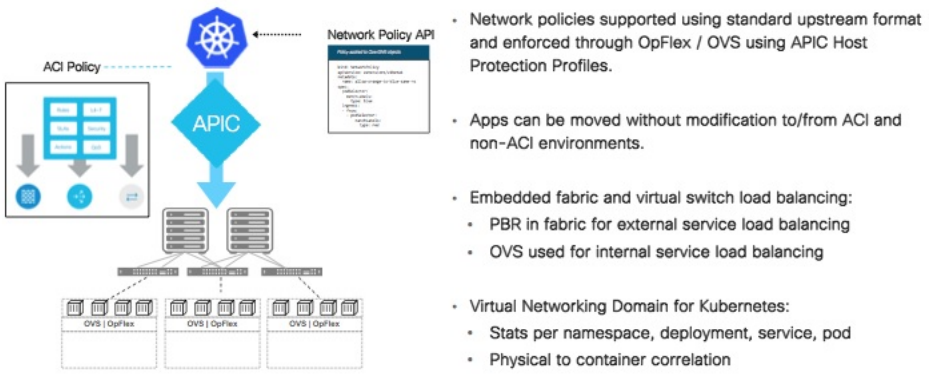
Operationally, all the CNI plugins offer the same experience to the customer. The container network connectivity is seamless and network policies are applied using Kubernetes NetworkPolicies. Under-the-hood, both ACI and Contiv offer advanced feature support. ACI allows you to map CNI NetworkPolicies 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 2: Architecture of ACI



System Requirements

This section describes the software, hardware, storage, and network requirements that are necessary to deploy Cisco Container Platform.

Supported Version Matrix

Cisco Container Platform uses various software and hardware components. The following table provides information on the validated versions of each component.

Component	Validated Version
Kubernetes	1.10

Component	Validated Version
ACI	3.1.2p
HyperFlex software	3.0(1b)+
vSphere	vSphere 6.0 (u2)+ vSphere 6.5

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

Software Requirements

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

- VMware vCenter server 6.5
- VMware client integration plugin
- vSphere Flash client
- HyperFlex 3.0(1b)+

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

Hardware Requirements

- In Cisco Container Platform 1.3.0 or later, the hypervisor hosts need to run CPUs with an Ivy Bridge or newer microarchitecture.

Storage Requirements

Once HyperFlex is installed, you need to configure two shared datastores that are accessible to the hosts in the cluster for the following purposes:

- For persistent volume storage
- For deploying the Cisco Container Platform tenant base VM

Configuring Shared Datastore

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

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

Step 3 Perform these steps to create a datastore for the Kubernetes persistent volume storage:

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**.

Step 4 Perform these steps to create a datastore for deploying the Cisco Container Platform tenant base VM:

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

b) Specify a name, size, and block size for the datastore.

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.

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 get the 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.

Resource Management Requirements

Enabling DRS and HA on Clusters

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 Browse to the 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**.

Network Requirements

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. The user needs 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.
Step 2 On the **Configure** tab, expand **Networking** and select **Virtual switches**.
Step 3 Click **Add host networking**.
Step 4 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 5 Select **New standard switch** and click **Next**.
Step 6 Add physical network adapters to the new standard switch.
Step 7 Under **Assigned adapters**, click **Add adapters**.
Step 8 Select one or more physical network adapters from the list.
Step 9 From the **Failover order group** drop-down list, choose from the Active or Standby failover lists.
Step 10 For higher throughput and to provide redundancy, configure at least two physical network adapters in the Active list.
Step 11 Click **OK**.
Step 12 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 13 On the **Ready to Complete** screen, click **OK**.

Configuring DHCP Server

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

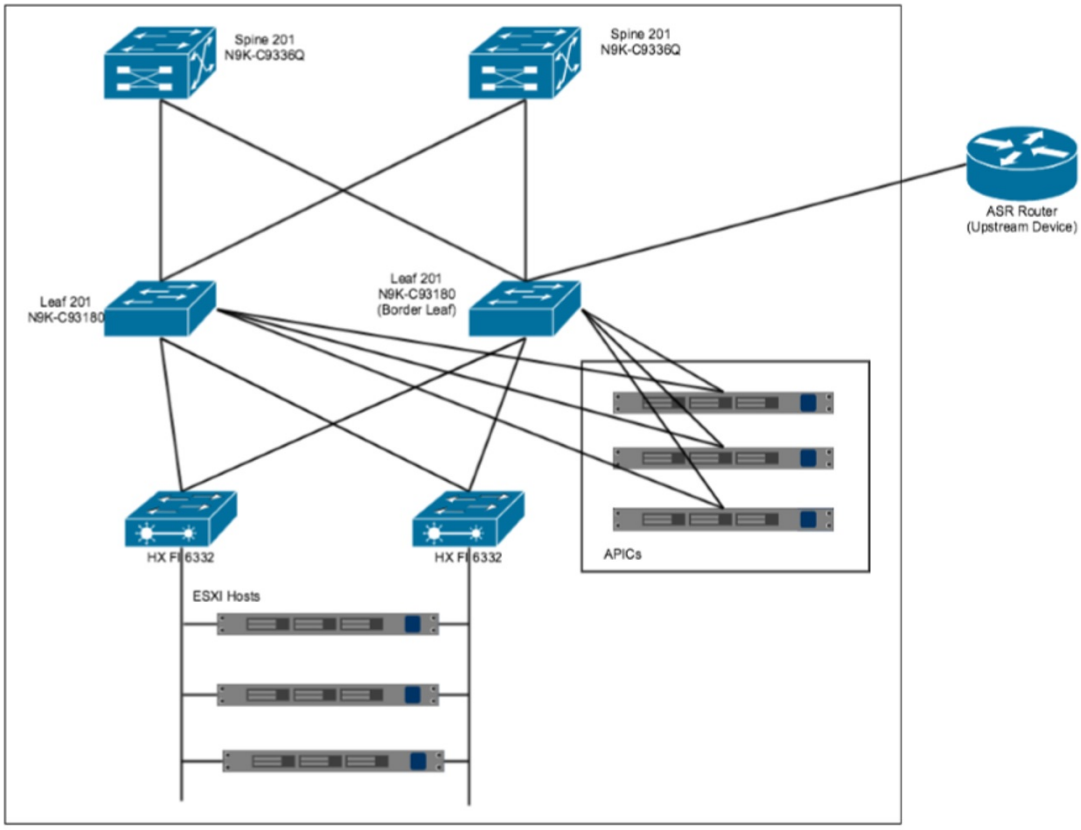
Reserving IP Addresses for Static Allocation

A static IP address is used during Cisco Container Platform installation for the **CCP Control Plane master node virtual IP** to support Cisco Container Platform upgrades. Additionally, Virtual IP address (VIP) is used as an external IP address for each Kubernetes cluster. VIPs are configured using VIP Pools. You can obtain this IP address from the same or a different subnet and you must ensure that it is not part of a DHCP pool.

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.



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 the Availability Zone (AZ) external 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:

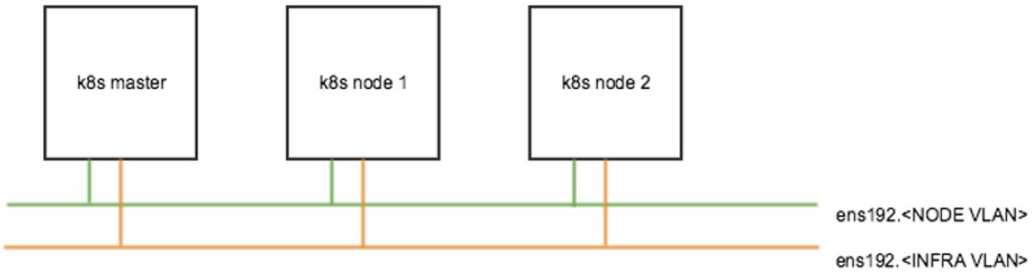
- Configure QOS
- Ensure that the tenant VLAN is allowed

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.

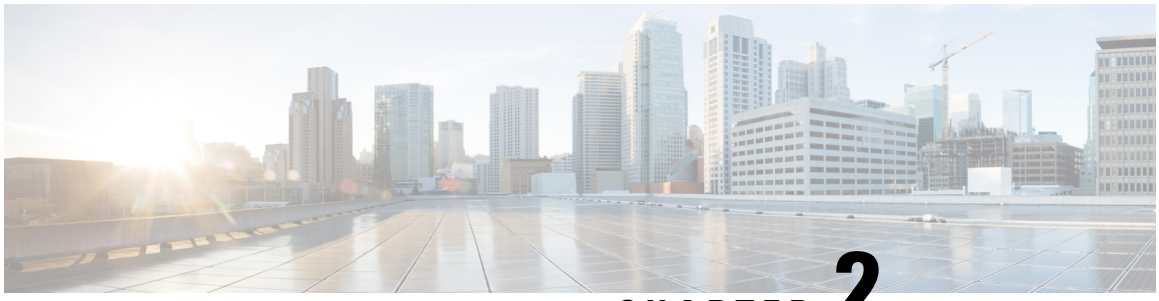
Tenant Cluster with ACI Deployment

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

Unlike other CNI, an ACI tenant cluster requires a couple sub-interface (VLAN interface) for each Kubernetes node. As shown in the following figure, Cisco Container Platform assigns the unique Node VLAN IDs. You need to assign the unique Infra VLAN ID for the clusters during cluster creation.



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



CHAPTER 2

Installing Cisco Container Platform on vSphere Web Client

This chapter contains the following topics:

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 14

  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 17

  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 Storage Requirements, on page 4 and Network Requirements, on page 6.
- Ensure that vSphere has an Enterprise Plus license, which supports DRS and vSphere HA.

Step 1 Download the Cisco Container Platform tenant base VM from Cisco.com.

Step 2 Log in to the **VMware vSphere Web Client** as an administrator.

Step 3 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 4 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.

b) Click **Next**.

Step 5 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 6 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 7 In the **Review details** screen, verify the Cisco Container Platform tenant base VM details, and then click **Next**.

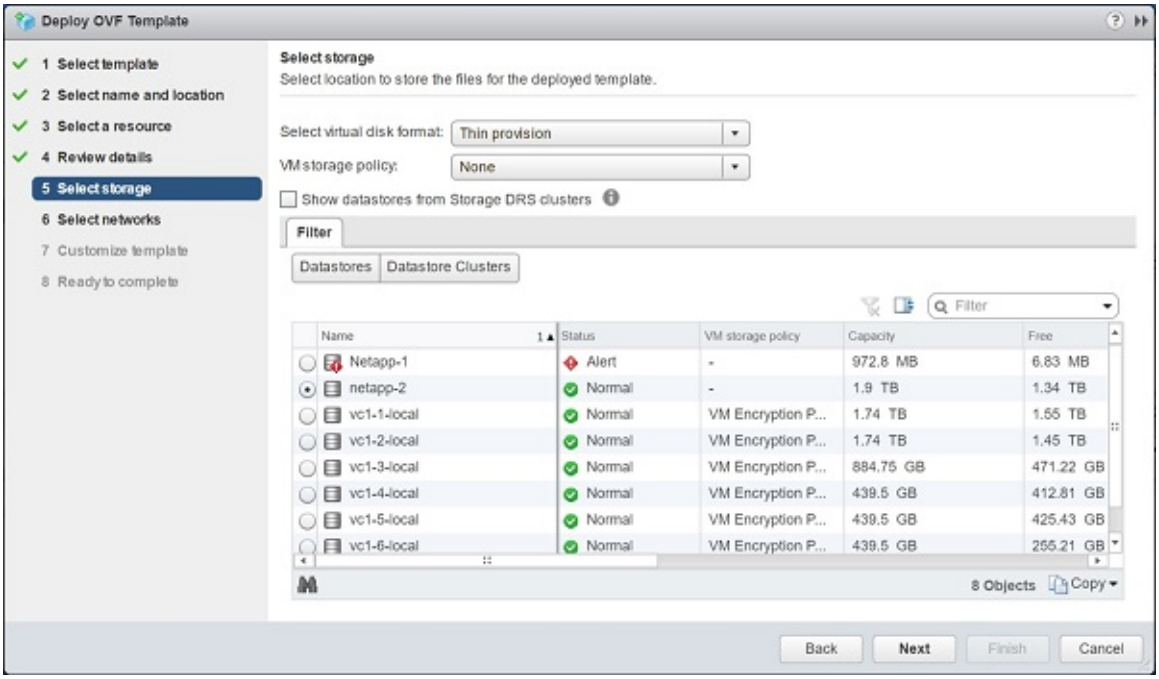
Step 8 In the **Select storage** screen, perform these steps:

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 storage** screen appears.

Figure 3: Select Storage Screen

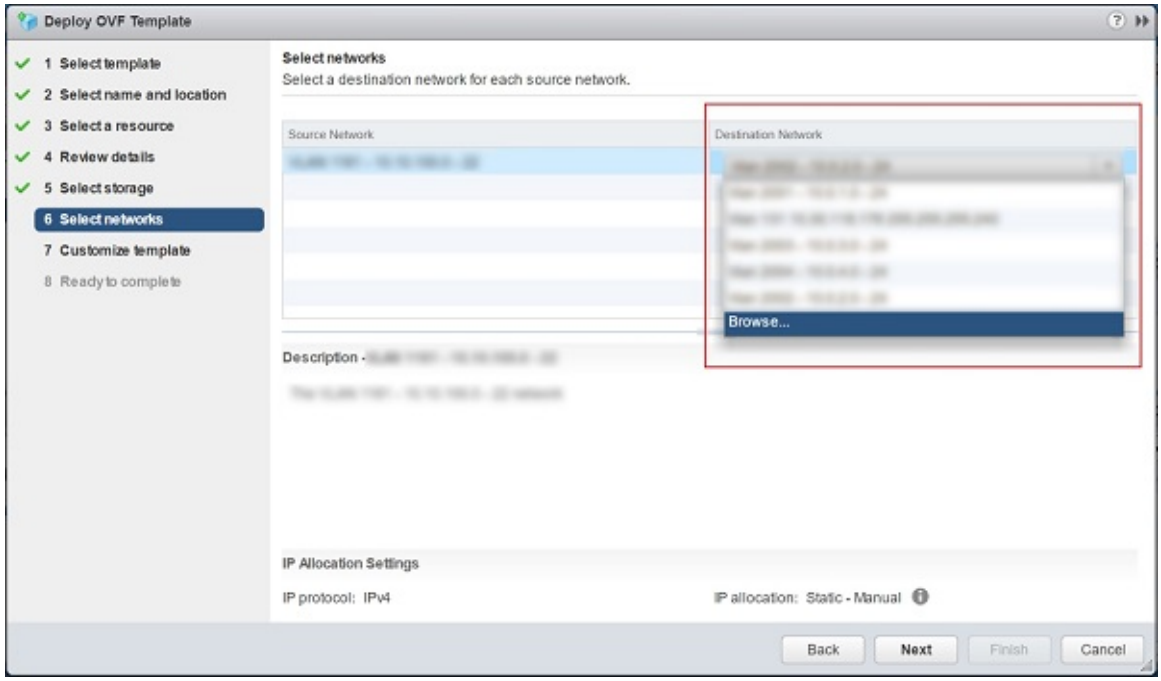


Step 9 In the **Select networks** screen, perform these steps:

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**.
The **Select networks** screen appears.

Figure 4: Select Networks Screen



Step 10 In the **Customize template** screen, click **Next**.

Step 11 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.

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 19.

Ensure that you have imported the latest Cisco Container Platform tenant base VM to the vCenter instance. For more information, see Importing Cisco Container Platform Tenant Base VM, on page 11.

Step 1 Download the Installer VM from Cisco.com.

Step 2 Log in to the **VMware vSphere Web Client** as an administrator.

Step 3 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 4 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.

b) Click **Next**.

Step 5 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 6 In the **Select a resource** screen, choose the cluster where you want to run the installer VM, and then click **Next**.

Step 7 In the **Review details** screen, verify the template details, and then click **Next**.

Step 8 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 9 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**.

Step 10 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.

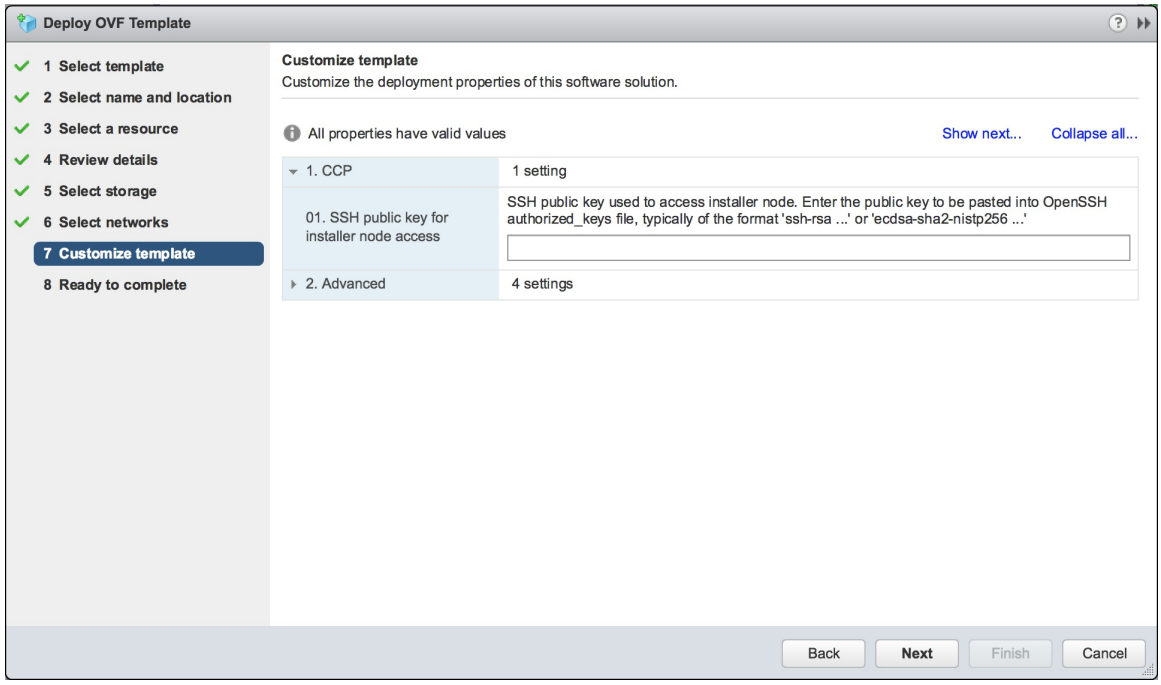
Note
- Ensure that you enter the public key in a single line.
- If you do not have an SSH key pair, you can generate it using the **ssh-keygen** command.

b) Expand **Advanced**, in the **Cluster name** field, enter a unique name for the Control Plane cluster.

Note Deployment of the installer VM fails if another Control Plane cluster with the same name already exists on your HyperFlex instance. You must ensure that you specify a unique name for the Control Plane cluster.

c) In the CIDR for 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.
d) Click **Next**.

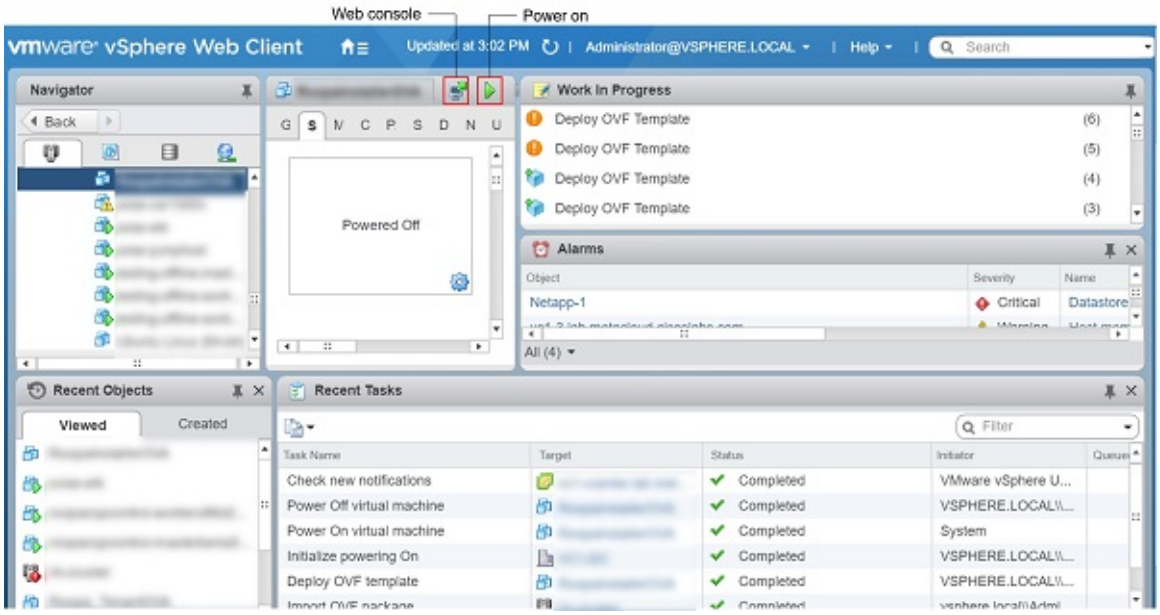
Figure 5: Customize Template Screen



Step 11 In the **Ready to complete** screen, verify the installer VM deployment settings, and then click **Finish**.

Step 12 Click the **Power on** button to switch on the VM.

Figure 6: Switching on Installer VM



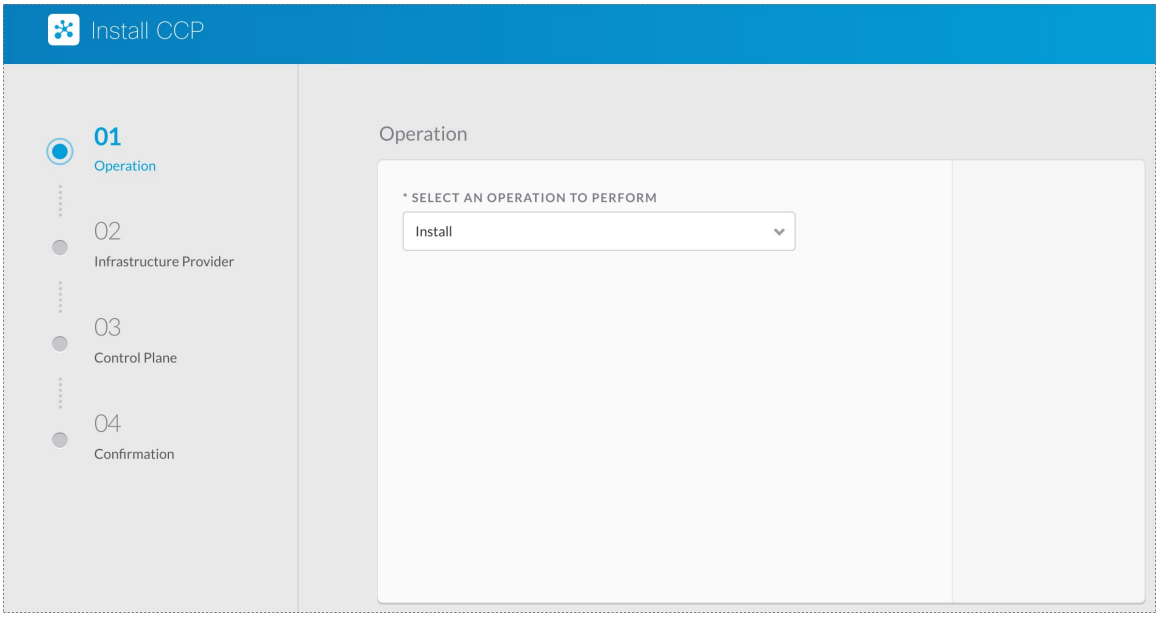
Once the installer VM is switched on, the installer UI takes a few minutes to become ready. You can view the status of the Installer UI using the Web console of vCenter. When the installer UI is ready, you can access it using the URL from the Web console.

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**.

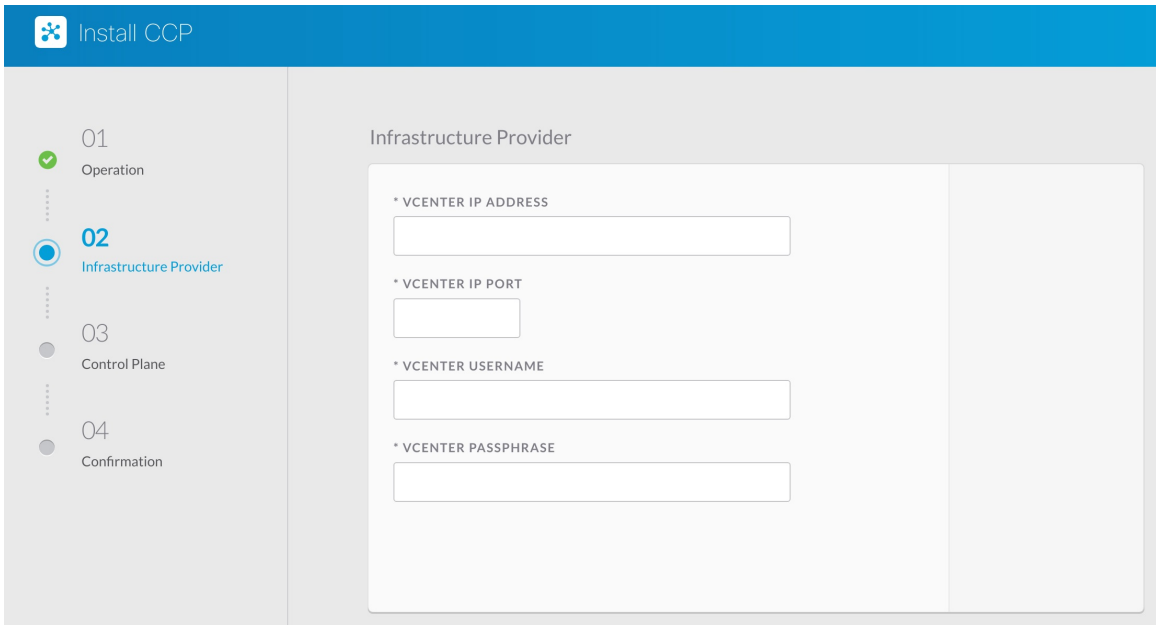
Step 1 Obtain the URL from the vCenter **Web console** and use a browser to open the installer UI.

Step 2 From the **SELECT AN OPERATION TO PERFORM** drop-down list, choose **Install**.



Step 3 In the **Infrastructure Provider** screen, enter the following information and then click **Next**:

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



Step 4 In the **Control Plane** screen, enter the following information:

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

b) From the **NETWORK PLUGIN FOR TENANT K8S CLUSTERS** drop-down list, choose one of the following options for network connectivity:

- Calico
- ACI
- Contiv (Tech Preview)

c) From the **VSPHERE DATACENTER** drop-down list, choose the datacenter.

d) From the **VSPHERE CLUSTER** drop-down list, choose the cluster.

e) From the **VSPHERE DATASTORE** drop-down list, choose the datastore.

f) From the **VSPHERE NETWORK** drop-down list, choose the network.

g) 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.

h) In the **CIDR FOR 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.

i) In the **CCP CONTROLLER MASTER NODE VIRTUAL IP** field, enter the IP address that is used to support a Cisco Container Platform upgrade.

j) In the **USERNAME FOR NODE ACCESS** field, enter the username of the user who can ssh into the CCP Control Plane nodes.

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

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

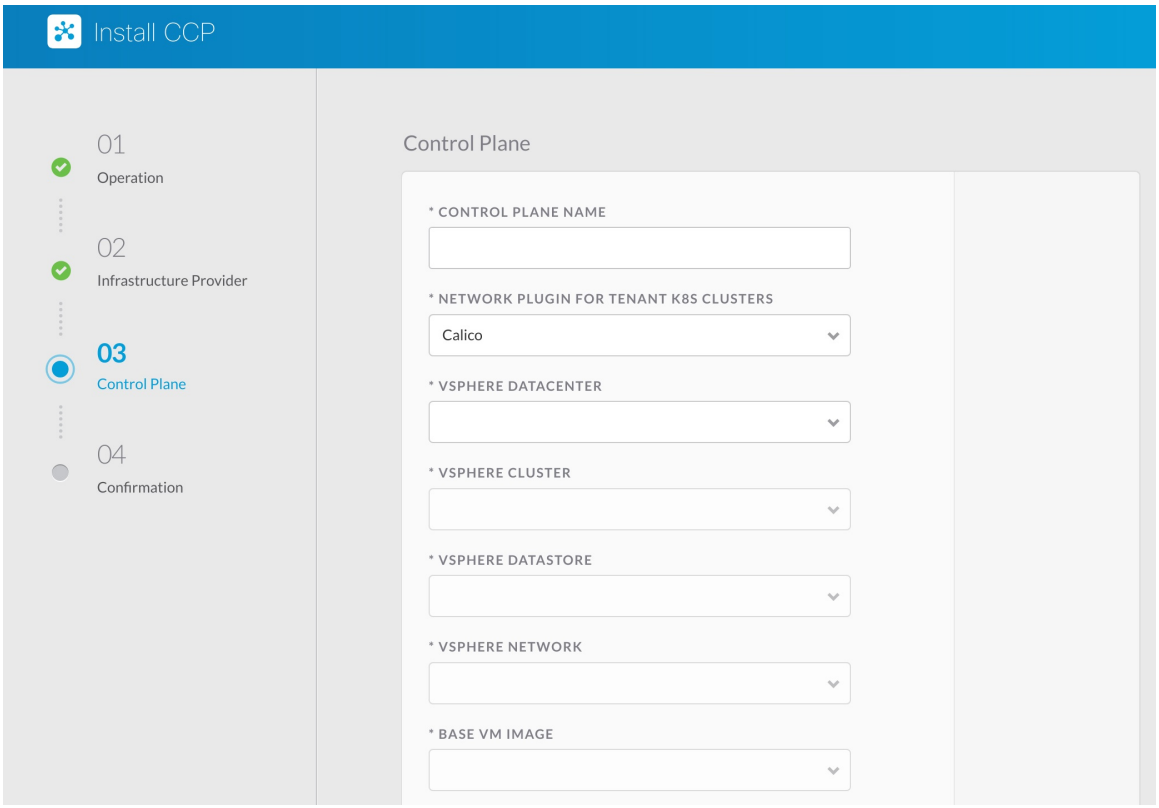
Note
- Ensure that you enter the public key in a single line.
- If you do not have an SSH key pair, you can generate it using the **ssh-keygen** command.

l) In the **CCP LICENSE ENTITLEMENT** field, enter the appropriate entitlement.
m) In the **CCP CONTROL PLANE ADMIN PASSPHRASE** field, enter the password of the Cisco Container Platform Control Plane.
n) In the **NTP SERVERS** field, enter the list of any NTP servers in your environment.

Note The **CCP HELM CHART REPO**, **CCP CONTAINER REGISTRY**, and **DEV SWITCH: CNI-CONTROL-PLANE** fields are used for internal purpose. You may skip these fields.

Step 5 In the **Infrastructure Provider** screen, enter the following information and then click **Next**:

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



Step 6 Click **Deploy** and then monitor the installation progress through the vCenter **Web console**.

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:

- Upgrading Cisco Container Platform Tenant Base VM, on page 20
- Deploying Upgrade VM, on page 20
- Upgrading Cisco Container Platform, on page 20

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

Upgrading Cisco Container Platform Tenant Base VM

You can follow the instructions in the Installing Cisco Container Platform > 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

Download the latest installer VM and follow the instructions in the Installing Cisco Container Platform > 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 CCP may change as part of the upgrade.

Upgrading Cisco Container Platform

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 From the **SELECT AN OPERATION TO PERFORM** drop-down list, choose **Upgrade**.

Step 3 In the **Infrastructure Provider** screen, enter the following information, and then click **NEXT**:

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

Step 4 In the **Existing Setup** screen, enter the following information, and then click **UPGRADE**:

a) In the **CCP URL** field, enter the URL for accessing Cisco Container Platform in the following format:
https://<*CCP_IP_Address*>:<*Port*>

Step 5 In the **Control Plane** screen, enter the following information:

a) In the **CONTROL PLANE NAME** field, enter the name of the Cisco Container Platform cluster.
b) From the **VSPHERE DATACENTER** drop-down list, choose the datacenter.
c) 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.
d) In the **CIDR FOR 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.
e) In the **USERNAME FOR NODE ACCESS** field, enter the username of the user who can ssh into the Cisco Container Platform Control Plane nodes.
f) In the **SSH PUBLIC KEY FOR INSTALLER 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.
- You can use the private key to securely connect to the Cisco Container Platform Control Plane VMs through SSH, after installation.
- If you do not have an SSH key pair, you can generate it using the **ssh-keygen** command.

Step 6 In the **CCP LICENSE ENTITLEMENT** field, enter the appropriate entitlement.

Step 7 In the **CCP CONTROL PLANE ADMIN PASSPHRASE** field, enter the password of the Cisco Container Platform Control Plane.
The **CCP HELM CHART REPO**, **CCP CONTAINER REGISTRY**, and **DEV SWITCH: CNI-CONTROL-PLANE** fields are used for internal purpose. You may skip these fields

Step 8 Click **UPGRADE**.

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 and 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*.

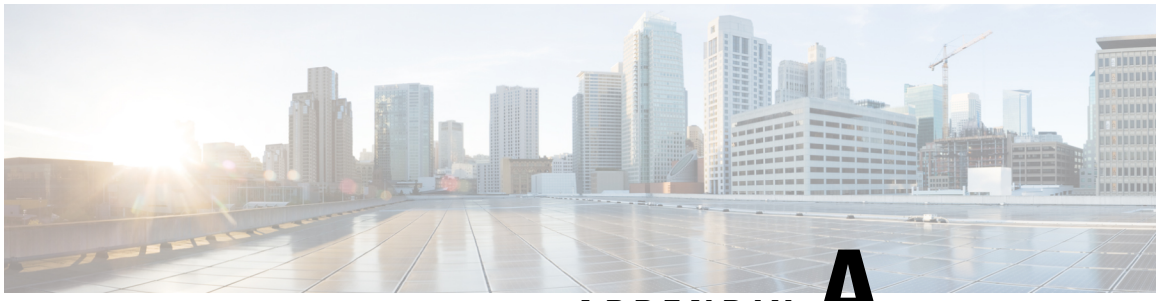
Step 2 Follow these steps to delete the Control Plane and installer node VMs:

a) In 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 **OVF properties** window.
c) If you have installed the Control Plane using the default name, right-click the folder named **ccpcontrol** or if you have provided a different name to the Control Plane in the installer node **OVF properties** window, right-click the folder with that name.
d) Choose **Delete File**.



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:

Unable to Deploy NGINX Ingress Controller Using Helm

Description	Error Message	Recommended Solution
Deploying the NGINX Ingress controller using Helm fails as RBAC is not configured in Helm.	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	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

Description	Error Message	Recommended Solution
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.	Port 80 is already in use. Please check the flag --http-port	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

Description	Error Message	Recommended Solution
Worker VMs may fail to power on after a shutdown.	File system specific implementation of LookupAndOpen[file] failed.	Follow these steps to resolve the problem: **1.** In the left pane, click on the VM that you want to powere on. **2.** In 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. **Note** 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
```

Troubleshooting Cisco Container Platform

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
```

Troubleshooting Cisco Container Platform

Troubleshooting Cisco Container Platform