



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.

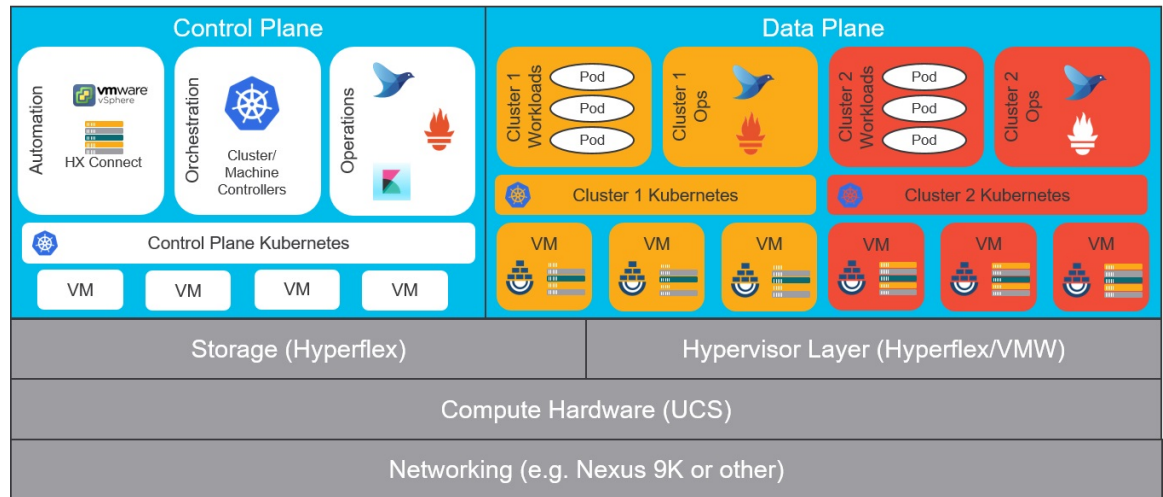
This chapter contains the following topics:

- [Cisco Container Platform Architecture Overview, on page 1](#)
- [Container Network Interface Plugins, on page 2](#)
- [System Requirements, on page 3](#)
- [ACI Integration Requirements, on page 7](#)

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 preceding 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.

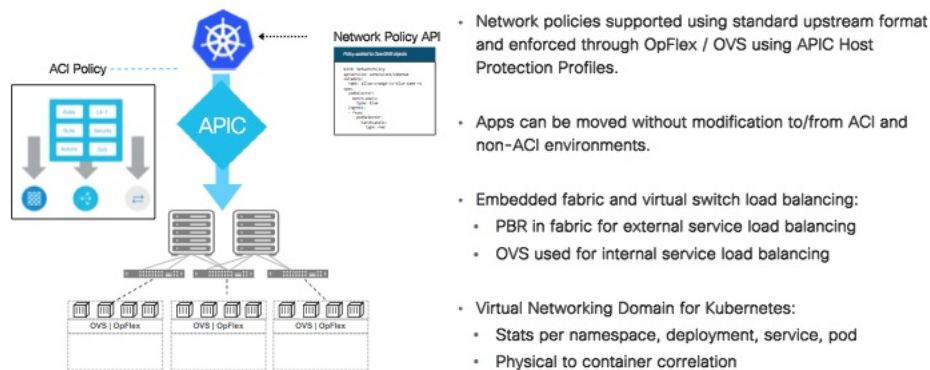
Operationally, all the CNI plugins offer the same experience to the customer. The container network connectivity is seamless and network policies are applied using [Kubernetes NetworkPolicies](#). Under-the-hood, both ACI and Contiv offer advanced feature support. ACI allows you to map CNI NetworkPolicies 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 HyperFlex 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:
- Enter a network Label or the port group, or accept the generated label.
 - 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.

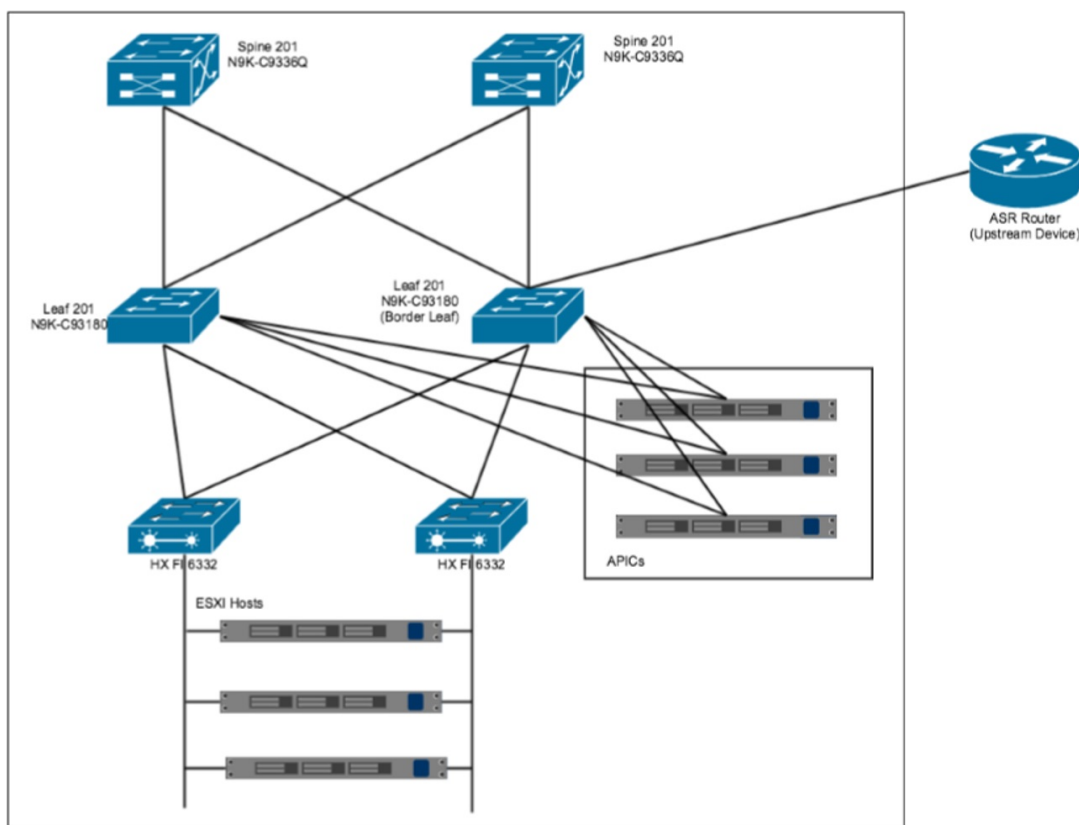
Reserving IP Addresses for Static Allocation

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:

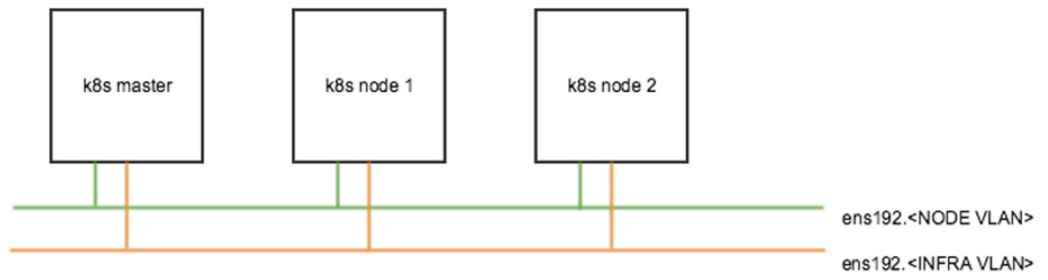
- Configure QOS
- Ensure that the tenant VLAN is allowed

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*.

