Cisco Container Platform

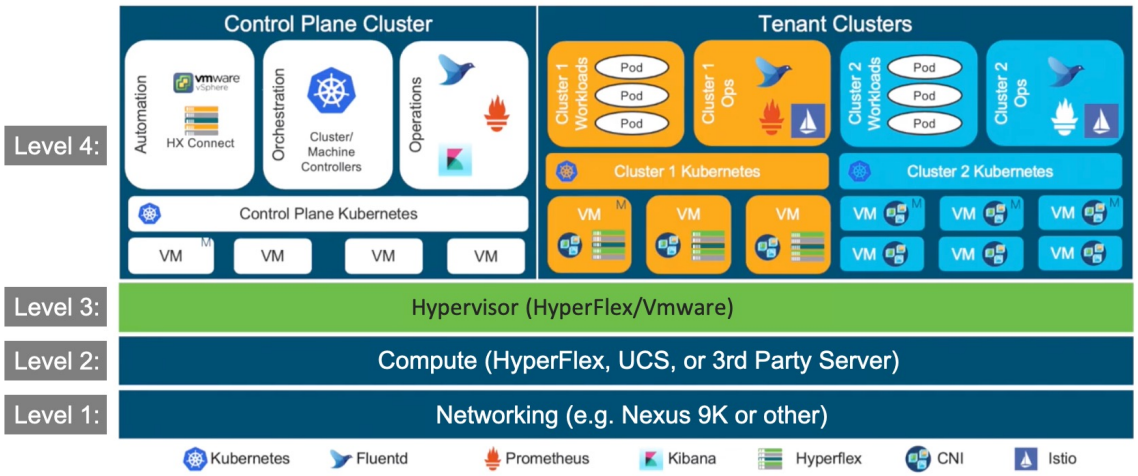
Cisco Container Platform is a turnkey, production grade, extensible platform to deploy and manage multiple Kubernetes clusters. It runs on 100% upstream Kubernetes. Cisco Container Platform offers seamless container networking, enterprise-grade persistent storage, built-in production-grade security,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

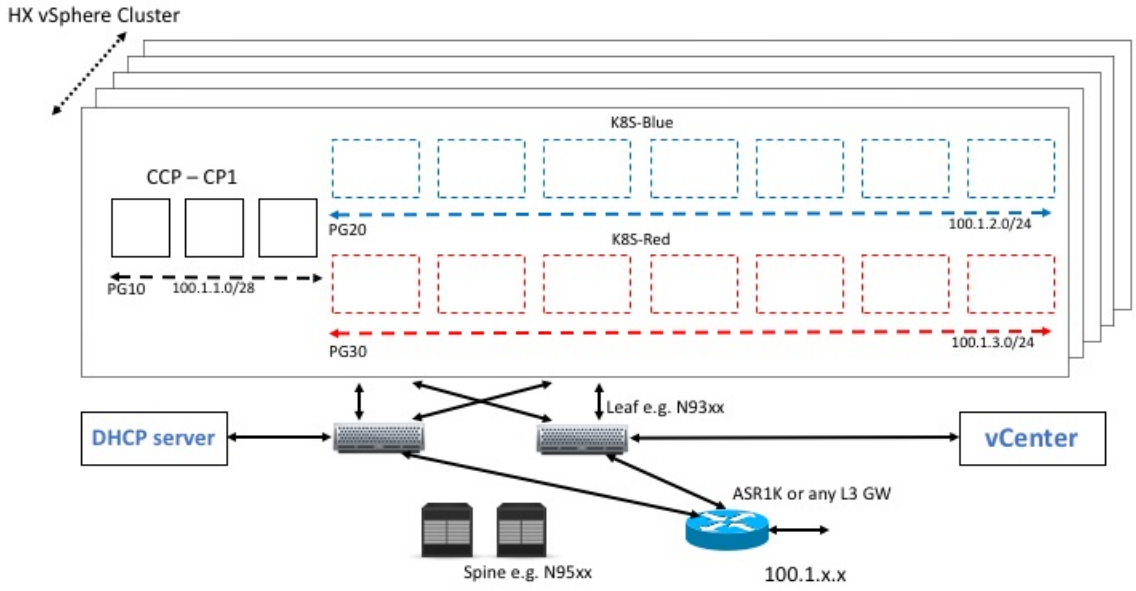
This section describes a sample deployment topology of the Cisco Container Platform and illustrates the network topology requirements at a conceptual level. Future sections of the document such as System Requirements and Installing Cisco Container Platform provide additional configuration details based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