Cisco Hybrid Cloud Platform with FlashStack Converged Infrastructure Solution

April 2019

Contents

1. Executive Summary **3**

2. Solution Overview **3**

2.1. Solution infrastructure 4
2.1.1. FlashStack with the Cisco Hybrid Cloud Platform 4
2.2. Persistent volume provisioning 4
2.2.1. Static provisioning 4
2.2.2. Dynamic provisioning 5
2.2.3. Storage orchestrators 5
2.3. Scalability and high availability with Cisco Hybrid Cloud Platform and FlashStack 5
2.4. Networking 6
2.4.1. Container network interface plugins 6
2.4.2. Dynamic Host Configuration Protocol (DHCP) 6
2.4.3. Port requirements 6

3. Implementing Networking, Storage, and Compute **7**

3.1. FlashStack 7
3.2. Pure Storage configuration 8
3.2.1. Configuring Pure Storage for VMware 8
3.3. Cisco UCS configuration 9
3.3.1. Network Time Protocol (NTP) server 9
3.4. Installing FlashStack servers 9
3.4.1. VMware ESXi 6.5 U1 9

4. Implementing the container platforms **11**

4.1. Cisco Container Platform 11
4.1.1. Cisco Container Platform requirements 11
4.1.2. Cisco Container Platform control plane 11
4.1.3. Pure Service Orchestrator in a tenant cluster 11

5. Implementing Orchestration, APIs, and Monitoring **18**

5.1. Cisco CloudCenter 18
5.1.1. CloudCenter configuration 18

6. Troubleshooting **18**

6.1. Network issues 18
6.2. Cisco Container Platform (CCP) general troubleshooting 18
6.3. CloudCenter (C3) general troubleshooting 18
6.4. FlashStack 18
6.4.1. NTP server non-reachability 18

7. Reference Details **19**

7.1. Hardware and software versions 19
7.2. Documentation references 19

1. Executive Summary

The Cisco® Hybrid Cloud Platform with Cisco Container Platform, HyperFlex and Application Centric Infrastructure solution guide describes a reference architecture for developing a consistent platform that can span public and private clouds using Cisco and open-source technologies. This solution guide is a deviation from that reference architecture and details the integration of the Cisco Hybrid Cloud Platform into an existing converged infrastructure environment that uses FlashStack™. The use case highlights using Pure Service Orchestrator which integrates seamlessly with the container orchestration framework (Kubernetes) and functions as the control plane virtualization layer that enables containerized environments to move away from consuming storage as a device to consuming storage as a service for persistent storage.

2. Solution Overview

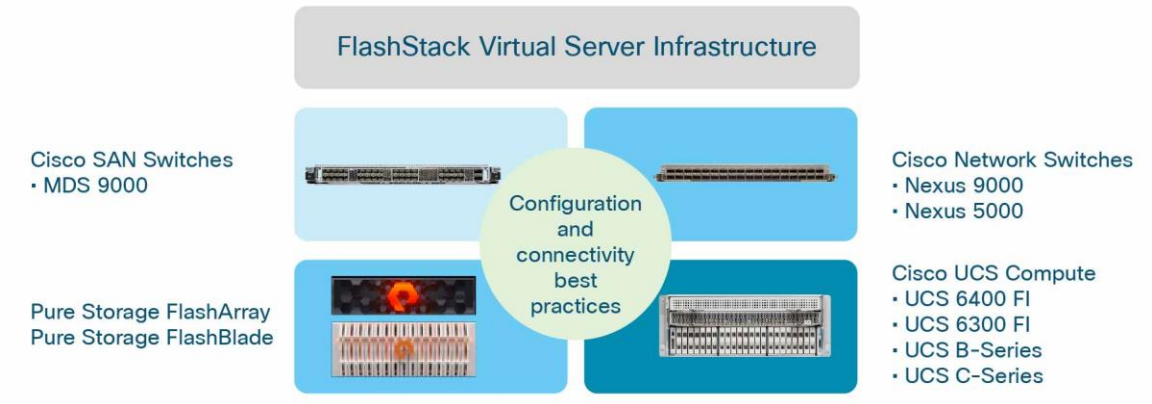
FlashStack is a converged infrastructure solution from Cisco and Pure Storage that brings the power and efficiency of all-flash storage to your entire Cisco UCS®-powered data center. It shortens your time to value, easily scales, and provides all-flash speed and reliability for every workload.

In this use case, FlashStack is a converged solution that can be used as on-premises infrastructure for the Hybrid Cloud Platform, allowing customers to seamlessly integrate compute and storage resources with Kubernetes clusters. It will focus on providing persistent storage for applications deployed in the Cisco Container Platform Kubernetes environment using Pure Service Orchestrator (PSO) to dynamically provision FlexVolume volumes (network file system [NFS] and Small Computer System Interface over IP [iSCSI]) on a Pure Storage FlashArray™.

This solution guide will provide insight on preparing, provisioning, deploying, and managing Cisco Hybrid Cloud Platform based on FlashStack, a Cisco validated, on-premises environment, with additional container-native storage using a Pure Storage FlashArray on Cisco UCS M4 B-Series blade servers and UCS M4 C-Series rack servers. The solution can provide the ability to develop, deploy, and manage containers in on-premises, private, and public cloud environments by bringing automation with a robust platform, such as Cisco CloudCenter™ and Cisco Container Platform.

Use FlashStack Design Guides to ensure your FlashStack platform is designed and configured per Cisco's best practices. Figure 1 shows the FlashStack architecture.

Figure 1. FlashStack architecture



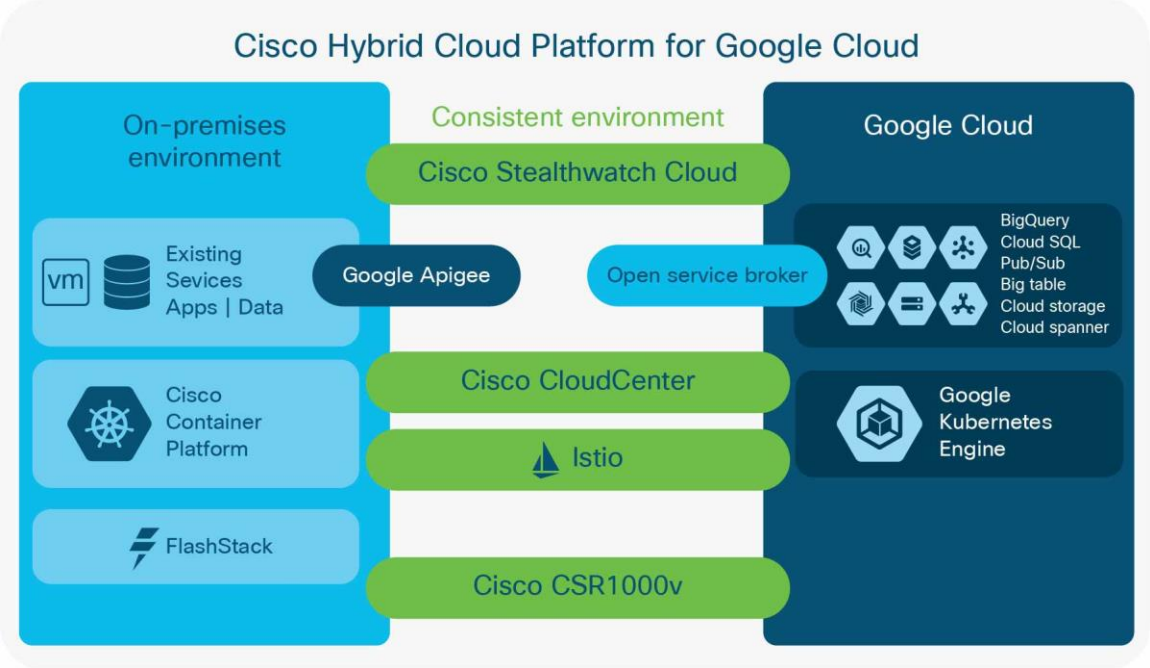
2.1. Solution infrastructure

2.1.1. FlashStack with the Cisco Hybrid Cloud Platform

If your organization is invested in block- or file-based storage products and strategies, Cisco's converged infrastructure solutions can fit right in, with Cisco Hybrid Cloud Platform. These solutions combine Cisco UCS and Cisco Nexus® switches with leading and emerging solutions from storage partners to deliver pre-engineered, tested, and supported systems that are designed to operate as a single unit.

Leveraging a Cisco converged solution can reduce the time spent ensuring everything works together with the Cisco Hybrid Cloud Platform. Validated designs are available and supported for popular infrastructure use cases and applications. Figure 2 depicts an overview of the FlashStack solution.

Figure 2. Overview of FlashStack featuring Pure Storage and Cisco UCS



2.2. Persistent volume provisioning

The Persistent Volume subsystem in Kubernetes uses Persistent Volume Claim to provide persistent storage to applications deployed in a Cisco Container Platform cluster (using iSCSI, NFS, or provider-specific storage system). Interaction between Persistent Volumes (PV) and Persistent Volume Claims (PVC) follow a specific lifecycle. There are three ways that PVs can be provisioned in Kubernetes: static, dynamic, or by leveraging storage orchestrators.

2.2.1. Static provisioning

The cluster administrator creates PVs that are available for use by cluster users. These volumes are available via a Kubernetes API and can be claimed with a PVC if a PV matching the requested criteria is available.

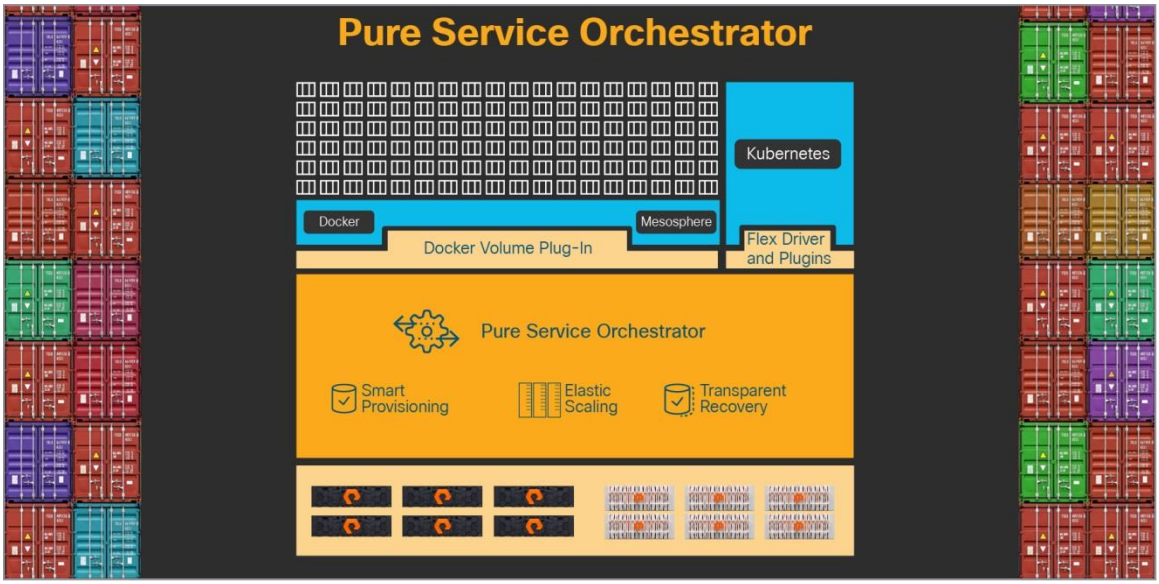
2.2.2. Dynamic provisioning

If no previously created PVs match a PVC, the cluster can attempt to dynamically provision a volume for the PVC based on StorageClasses that must have been previously configured.

2.2.3. Storage orchestrators

Storage orchestrators are vendor-specific plugins that allow Kubernetes to perform dynamic FlexVolume volume provisioning for specific OEM storage solutions. **Pure Service Orchestrator** is maintained by Pure Storage. It delivers persistent storage as a service for containerized applications using Pure Storage all-flash arrays and seamlessly integrates with orchestration tools like Kubernetes. See Figure 3 for high-level architectural overview.

Figure 3. Pure Service Orchestrator architecture



2.3. Scalability and high availability with Cisco Hybrid Cloud Platform and FlashStack

Cisco UCS software supports large-scale, distributed server environments using Cisco UCS B-Series, C-Series, and S-Series servers, and Cisco HyperFlex hyperconverged systems. Generally, when you increase the scale of infrastructure, it also increases the complexity. Cisco UCS makes it easier to manage large-scale, diverse environments.

Microservices and containers offer much greater flexibility and scalability. Containers offer significant advantages over mere virtualization. Containers supercharge today's state-of-the-art converged and hyperconverged platforms and they are cost-effective.

Using Kubernetes to help adopt containers eases deployment pains, but getting Kubernetes up and running can be a major task.

Cisco Container Platform simplifies the task of deploying and managing Kubernetes and takes advantage of UCS's robust hardware platforms and software orchestration capabilities. Cisco Hybrid Cloud Platform uses reliable Cisco equipment that can enable users to deploy Kubernetes, with options for adding scalability, high availability, cloud security, audit tools, and connectivity to hybrid clouds.

2.4. Networking

The solution guide, Cisco Hybrid Cloud Platform with Cisco Container Platform, HyperFlex and Application Centric Infrastructure (ACI) describes the Cisco Hybrid Cloud Platform in a greenfield, hyperconverged deployment, with or without Cisco Application Centric Infrastructure (Cisco ACI®). The FlashStack virtual server infrastructure supports Cisco ACI implemented with Cisco Nexus switches (not covered in this guide). If you intend to use Cisco ACI in your environment, refer to FlashStack Virtual Server Infrastructure with Cisco Application Centric Infrastructure and Pure Storage FlashArray.

2.4.1. Container network interface plugins

Cisco Container Platform supports native Kubernetes NetworkPolicies and provides container-to-container and container-to-non-containerized application layers communication with security policies. The NetworkPolicies are independent of the underlying container network plugin.

Cisco Container Platform supports multiple Kubernetes Container Network Interface (CNI) plugins, such as:

- Cisco ACI, which is the recommended plugin for use with an ACI fabric. ACI is fully supported by Cisco
- Calico, which is recommended when an ACI fabric is not used
- Contiv (Tech Preview), which is a user space switch that is 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.

For more on Cisco Container Platform networking, refer to Chapter: Services and Networking of the Cisco Container Platform 1.4.1.User Guide.

2.4.2. Dynamic Host Configuration Protocol (DHCP)

Cisco Container Platform requires a DHCP server to be present. The Cisco Container Platform installer Virtual Machine (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.

2.4.3. Port requirements

Cisco Container Platform

- Cisco Container Platform control plane VMs must have network access to the vCenter appliance API

Cisco CloudCenter

- For firewall rule requirements for each CloudCenter component (including Load Balancers) refer to the Firewall Rules Overview section of the CloudCenter Installation Guide.

3. Implementing Networking, Storage, and Compute

This section describes the elements that compose the FlashStack architecture and highlights configurations that are needed to support the Cisco Hybrid Cloud Platform for Google Cloud. This section is intended to supplement the Networking, Storage, and Compute section in the document Cisco Hybrid Cloud Platform with Cisco Container Platform, HyperFlex and Application Centric Infrastructure (ACI).

3.1. FlashStack

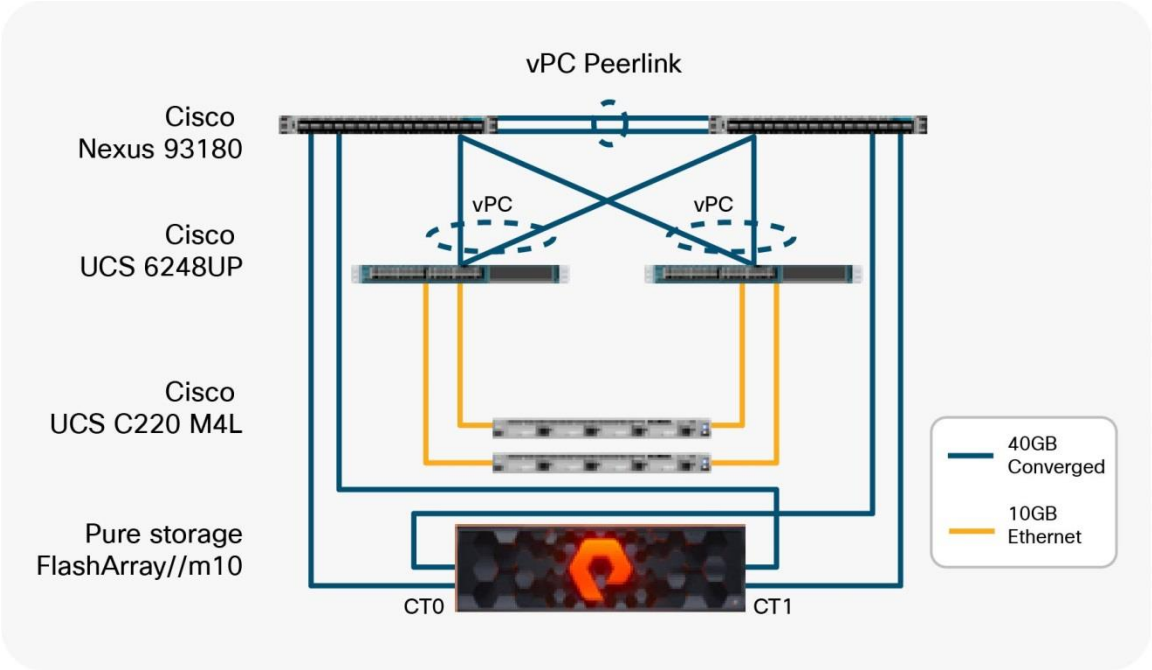
The FlashStack used in this solution guide highlights a 10 Gigabit Ethernet (GE) iSCSI end-to-end deployment utilizing a Cisco Nexus 93180 switch and the Cisco UCS C220 M4 Rack Server. The architecture uses iSCSI volumes for generic VM workloads.

Table 1 describes the FlashStack environment used for this solution guide. Figure 4 outlines the FlashStack configuration.

Table 1. FlashStack Component Configuration Reference

Category	Components	Version
Management	Cisco UCS Manager	5.4
	VMware ESXi 6.5 U1	Build number 5310538
	VMware vCenter 6.5	Build number 8024368
Network	Cisco Nexus 93180 Switch	7.0(3)I4(8a)
Compute	Cisco UCS C220 M4L	3.2(3a)
	Cisco UCS 6248UP	Cisco UCS 6248UP
Storage	Pure Storage FlashArray//M	Purity FA 5.1.3

Figure 4. FlashStack configuration



Consult the FlashStack installation instructions within the FlashStack Design and Installation Guides to be sure your infrastructure is in a supported configuration. This guide will provide a high-level overview of a FlashStack installation with tips for integration of Cisco Container Platform.

NOTE: Cisco Container Platform 1.3.0 or later requires the hypervisor hosts to use CPUs with an Ivy Bridge or newer microarchitecture for the use of cryptographical number generation.

3.2. Pure Storage configuration

3.2.1. Configuring Pure Storage for VMware

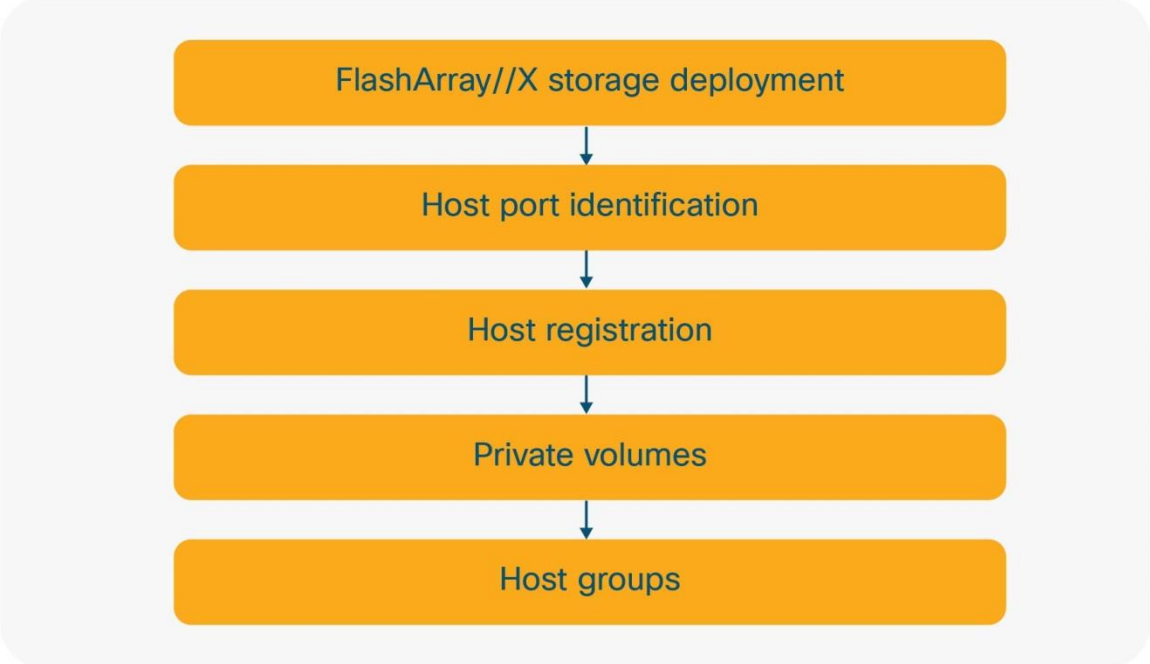
Once your FlashStack is installed and the Pure Storage FlashArray//X is accessible to the FlashStack, the storage to be deployed will include:

- VMware ESXi iSCSI boot LUNs
- Virtual Machine File System (VMFS) datastores

The iSCSI boot LUNs will be set up from the Pure Storage web portal, and the VMFS data stores will have been directly provisioned from the vSphere web client after the Pure Storage vSphere web client plugin has been registered with the VMware vCenter.

Ensure that you have followed FlashStack installation instructions for configuring private volumes and host groups, as shown in Figure 5.

Figure 5. FlashArray storage deployment workflow



3.3. Cisco UCS configuration

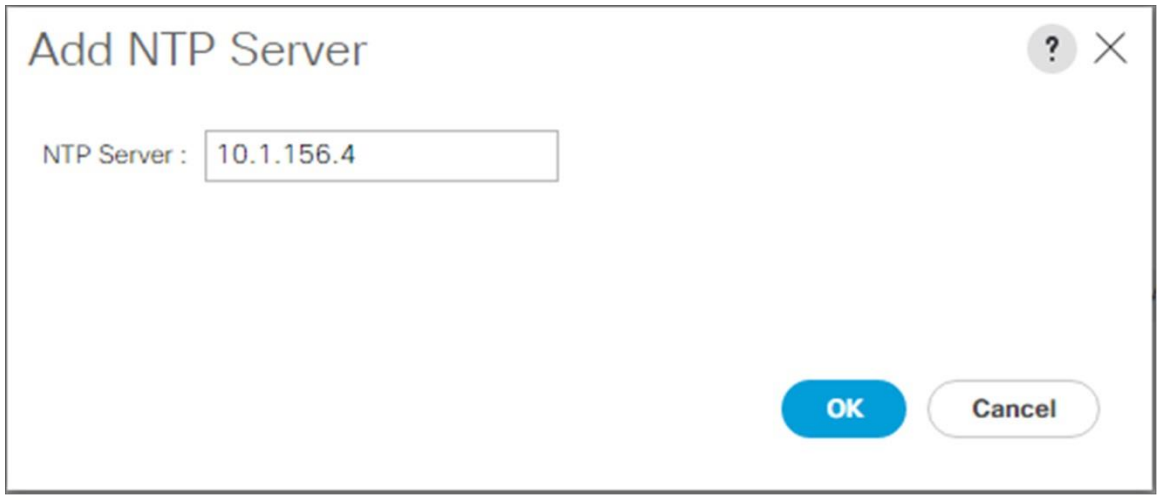
3.3.1. Network Time Protocol (NTP) server

To avoid unpredictable NTP offset issues in your Kubernetes cluster environment, ensure that Cisco UCS, Cisco Nexus switches, and all VMs are synchronized to common NTP servers.

To synchronize the Cisco UCS environment to the common NTP servers, complete the following steps:

1. In Cisco UCS Manager, click Admin on the left.
2. Expand All > Time Zone Management.
3. Select Timezone.
4. In the Properties pane, select the appropriate time zone in the Timezone menu.
5. Click Save Changes, and then click OK.
6. Click Add NTP Server.
7. Enter and click OK.

Figure 6. NTP server details for UCS Manager



3.4. Installing FlashStack servers

3.4.1. VMware ESXi 6.5 U1

Several methods exist for installing ESXi in a VMware environment. These procedures focus on how to configure VMware and Pure Storage best practices for data stores.

For detailed instruction on installing VMware ESXi 6.5 U1 for FlashStack, refer to the document, FlashStack Virtual Server Infrastructure with iSCSI Storage for VMware vSphere 6.5 U1.

More about VMWare configuration

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

Cisco Container Platform requires a VMware vCenter cluster with high availability and Distributed Resource Scheduler (DRS) enabled.

3.4.1.1. 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). You will need to create either VSS or DVS switches for managing the VM traffic.

Follow these steps to configure a vSphere standard switch:

1. In the vSphere web client, navigate to the host.
2. On the Configure tab, expand Networking, and then select Virtual switches.
3. Click Add host networking.
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 then click Next.
5. Select New standard switch and click Next.
6. Add physical network adapters to the new standard switch.
7. Under Assigned adapters, click Add adapters.
8. Select one or more physical network adapters from the list.
9. From the Failover order group drop-down list, choose from the Active or Standby failover lists.
10. For higher throughput and to provide redundancy, configure at least two physical network adapters in the Active list.
11. Click OK.
12. Enter connection settings for the adapter or the port group as follows:
    1. Enter a network label or the port group or accept the generated label.
    2. Set the VLAN ID to configure VLAN handling in the port group.
    3. On the Ready to Complete screen, click OK.

4. Implementing the container platforms

4.1. Cisco Container Platform

This section describes how to provide persistent storage for applications deployed in the Cisco Container Platform Kubernetes environment using Pure Service Orchestrator to dynamically provision Network File System (NFS) and Small Computer System Interface over IP (iSCSI) volumes on a Pure Storage array.

Base installation and cloud-specific steps needed to install and pre-configure the Cisco Container Platform environment for Cisco Hybrid Cloud Platform can be found in the solution guide Cisco Hybrid Cloud Platform with Cisco Container Platform, HyperFlex and Application Centric Infrastructure (ACI).

4.1.1. Cisco Container Platform requirements

4.1.1.1. 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.

4.1.1.2. Static IP addresses

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

Table 2 summarizes the static and DHCP IP address requirements for the Cisco Container Platform components.

Table 2. IP requirements

Component	Static IP	DHCP IP
Cisco Container Platform web interface	1	4
	-	1
Tenant clusters	2 + the number of load balancer VIPs desired for applications	1 + the number of workers

4.1.2. Cisco Container Platform control plane

For detailed installation instructions on installing the Cisco Container Platform control plane and deploying a tenant cluster, follow the install guide for the current version certified for the Cisco Hybrid Cloud Platform for Google Cloud: Cisco Container Platform 1.4.0 Installation Guide.

4.1.3. Pure Service Orchestrator in a tenant cluster

Pure Storage developed a way to provide a persistent data layer that can meet the expectations for ease of use and scalability.

The Pure Service Orchestrator (PSO) allows smart provisioning and scales across many arrays, both block and file. On-demand, persistent storage for developers is placed on the best array or adheres to your policies based on labels.

To install Pure Service Orchestrator, you can use the traditional shell script as described in the readme file.

The second way, which may fit into your own software deployment strategy, is to use Helm. Since using Helm provides a very quick and simple way to install and may be new to you, the rest of this document will focus on how to get started with PSO using Helm.

4.1.3.1. Installing Pure Service Orchestrator via Helm

Be sure to install Helm using the correct Role-Based Access Control (RBAC) instructions described in the blog, "Getting Started with Helm for K8s."

You can also read the Helm documentation posted on its website.

Once Helm is fully functioning with your Cisco Container Platform cluster, run the following commands to set up and Pure Storage Helm repo:

```
helm repo add pure https://purestorage.github.io/helm-charts
helm repo update
helm search pure-k8s-plugin
```

Next, create a YAML file with contents maintaining the format:

```
arrays:
  FlashArrays:
    - MgmtEndPoint: "10.2.1.00"
      APIToken: "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
      Labels:
        rack: "ATCDC1-B01 RU20-21"
        env: "prod"
  FlashBlades:
    - MgmtEndPoint: "10.2.1.200"
      APIToken: " xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx "
      NfsEndPoint: "10.2.1.201"
      Labels:
        rack: "ATCDC1-B01 RU30"
        env: "dev"
```

Execute a trial run of the installation using the YAML file you created. The command will show the output but will not change anything on your cluster.

```
helm install --name pure-storage-driver pure/pure-k8s-plugin -f pure-m10.yaml --
dry-run --debug
```

If you are satisfied with the output of the trial run, you can run the installation.

```
helm install --name pure-storage-driver pure/pure-k8s-plugin -f pure-m10.yaml
```

Check the GitHub page hosting the Pure Storage repo for more detail.

4.1.3.2. Setting the default StorageClass

Since you may have more than Pure Storage in your environment we will not force ‘pure’ as the default StorageClass in Kubernetes.

If you already installed the plugin via Helm and need to set the default class to Pure Storage, then run this command:

```
kubectl patch storageclass pure-block -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

If you have another storage class set to default and you wish to change it to Pure Storage, you must first remove the default tag from the other StorageClass and then run the command above.

Having two defaults will produce undesired results. To remove the default tag, run this command:

```
kubectl patch storageclass you-storage-class -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"false"}}}'
```

Read more about these commands from the K8s documentation.

4.1.3.3. Updating your array information

If you need to add a new FlashArray or FlashBlade™ simply add the information to your YAML file and update via Helm. You may edit the config map within Kubernetes and there are good reasons to do it that way. But for simplicity, we will stick to using Helm for changes to the array information YAML file. Once your file contains the new array or label run the following command:

```
helm upgrade pure-storage-driver pure/pure-k8s-plugin -f
your_own_dir/yourvalues.yaml
```

4.1.3.4. Upgrading using Helm

With the same general process, you can use the following command and update the version of Pure Service Orchestrator:

```
helm upgrade pure-storage-driver pure/pure-k8s-plugin -f
your_own_dir/yourvalues.yaml --version target version
```

4.1.3.5. Persistent storage in a tenant cluster

The Kubernetes PersistentVolume API provides several plugins for integrating storage into Kubernetes for containers to consume. Below, we’ll show how Pure Service Orchestrator can dynamically provision volumes for an application.

4.1.3.5.1 Application deployment

This example application will install a simple database called CockroachDB using the Helm installation method. Helm will download a pre-configured chart from a defined repository. The persistent volume uses the pure-block driver to dynamically create the required volumes on the FlashStack's Pure Storage array based on our default volume class of "pure".

```
ccpuser@coe-cluster1-master21c841cc05:~$ kubectl get storageclass
NAME                 PROVISIONER                    AGE
pure                 pure-provisioner               1h
pure-block (default) pure-provisioner               1h
pure-file            pure-provisioner               1h
standard             kubernetes.io/vsphere-volume   82d
```

Note that for a production cluster, you will likely to want to modify the storage and StorageClass parameters for the Helm chart. This chart defaults to 100 GB of disk space per pod. However, you may want a different size depending on your use case, and the default persistent volume StorageClass in your environment may not be set to "pure".

4.1.3.5.2. Install the application

To install the application via Helm use the following command:

```
helm install --name crdb stable/cockroachdb
```

The following text shows the output of the Helm installation, which shows the cluster IP for the database and corresponding port for accessing the database.

```
ccpuser@coe-cluster1-master21c841cc05:~$ helm install --name crdb stable/cockroachdb
NAME:   crdb
LAST DEPLOYED: Mon Dec 10 05:41:40 2018
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/Pod(related)
NAME                        READY  STATUS             RESTARTS  AGE
crdb-cockroachdb-init-pxfxc  0/1    ContainerCreating  0         0s

==> v1/Service
NAME                     TYPE       CLUSTER-IP      EXTERNAL-IP  PORT(S)              AGE
crdb-cockroachdb-public  ClusterIP  10.107.189.158  <none>       26257/TCP,8080/TCP  0s
crdb-cockroachdb         ClusterIP  None            <none>       26257/TCP,8080/TCP  0s

==> v1beta1/StatefulSet
NAME             DESIRED  CURRENT  AGE
crdb-cockroachdb  3        0        0s

==> v1/Job
NAME                  DESIRED  SUCCESSFUL  AGE
crdb-cockroachdb-init  1        0           0s

==> v1beta1/PodDisruptionBudget
NAME                    MIN AVAILABLE  MAX UNAVAILABLE  ALLOWED DISRUPTIONS  AGE
crdb-cockroachdb-budget  N/A            1                0                    0s


NOTES:
CockroachDB can be accessed via port 26257 at the
following DNS name from within your cluster:

crdb-public.default.svc.cluster.local

Because CockroachDB supports the PostgreSQL wire protocol, you can connect to
the cluster using any available PostgreSQL client.

For example, you can open up a SQL shell to the cluster by running:

    kubectl run -it --rm cockroach-client \
        --image=cockroachdb/cockroach \
        --restart=Never \
        --command -- ./cockroach sql --insecure --host crdb-cockroachdb-public.default

From there, you can interact with the SQL shell as you would any other SQL shell,
confident that any data you write will be safe and available even if parts of
your cluster fail.

Finally, to open up the CockroachDB admin UI, you can port-forward from your
local machine into one of the instances in the cluster:

    kubectl port-forward crdb-cockroachdb-0 8080

Then you can access the admin UI at http://localhost:8080/ in your web browser.

For more information on using CockroachDB, please see the project's docs at
https://www.cockroachlabs.com/docs/
```

4.1.3.5.3. Validation

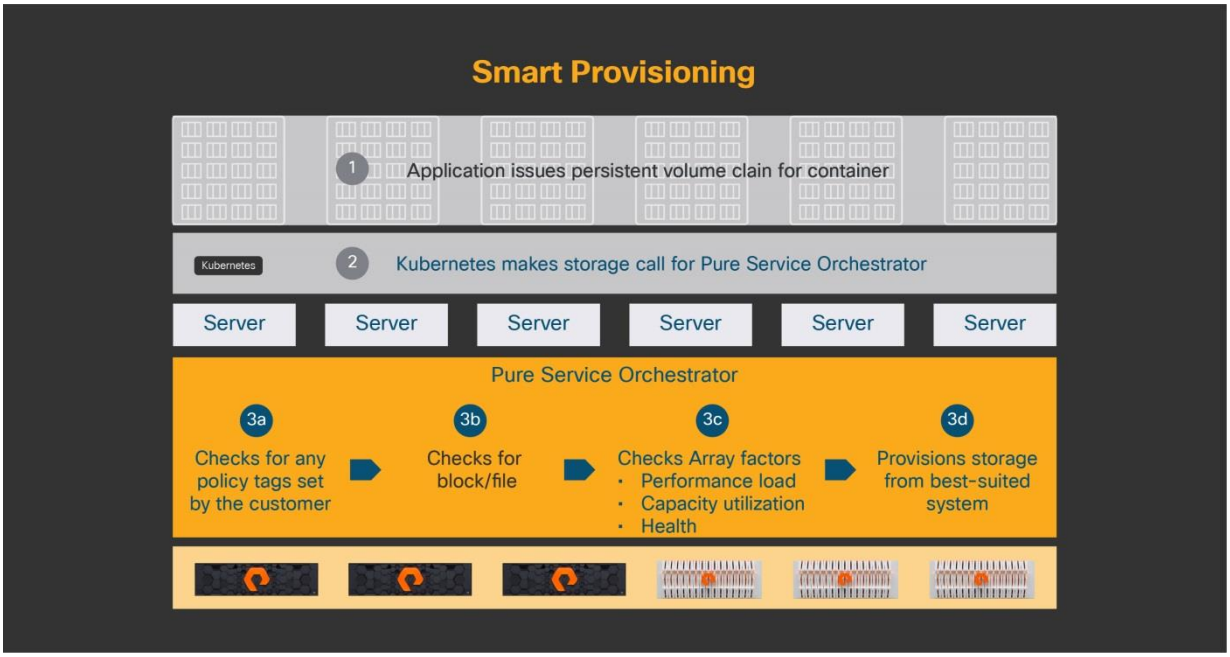
Validate that the application is deployed and that the volume claims are created, with 100-G capacity:

```
kubectl get pvc
```

```
c841cc05:~$ kubectl get pvc
NAME                            STATUS    VOLUME
CAPACITY   ACCESS MODES   STORAGECLASS   AGE
datadir-crdb-cockroachdb-0   Bound     pvc-d4f10142-fc3b-11e8-b417-005056a0e37d
100Gi      RWO            pure           6m
datadir-crdb-cockroachdb-1   Bound     pvc-d4f4c846-fc3b-11e8-b417-005056a0e37d
100Gi      RWO            pure           6m
datadir-crdb-cockroachdb-2   Bound     pvc-d4f85ca1-fc3b-11e8-b417-005056a0e37d
100Gi      RWO            pure           6m
```

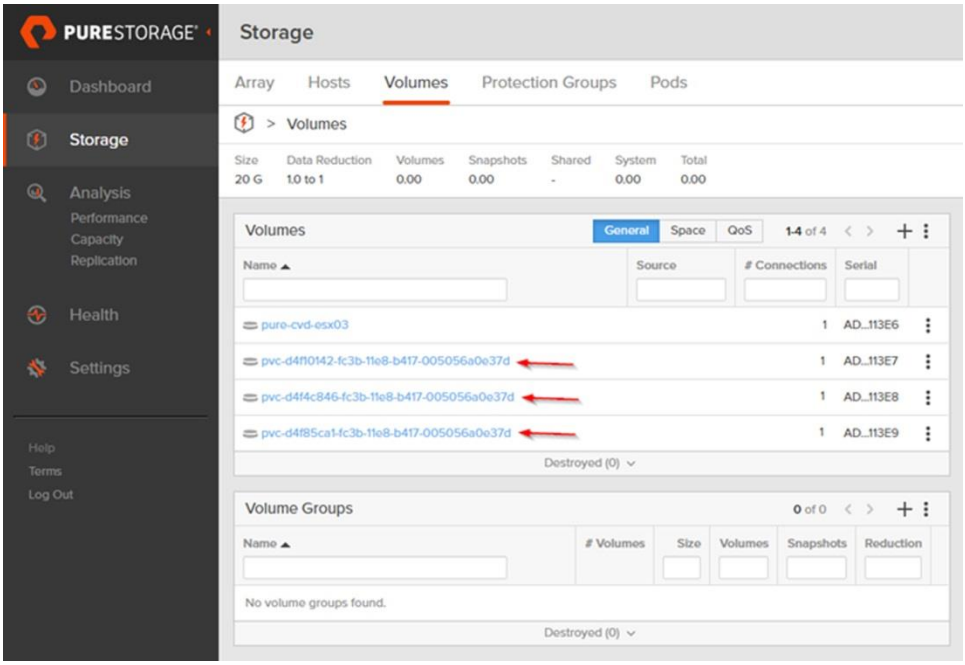
Pure Service Orchestrator used the Pure Storage API to dynamically create the volumes on the Pure Storage array that best matches the storage criteria for the application, as shown in Figure 7.

Figure 7. Volumes of the Pure Storage array



In this example, Pure Service Orchestrator dynamically created the following volumes on the Pure M10 array previously configured:

- pvc-d4f10142-fc3b-11e8-b417-005056a0e37d
- pvc-d4f4c846-fc3b-11e8-b417-005056a0e37d
- pvc-d4f85ca1-fc3b-11e8-b417-005056a0e37d

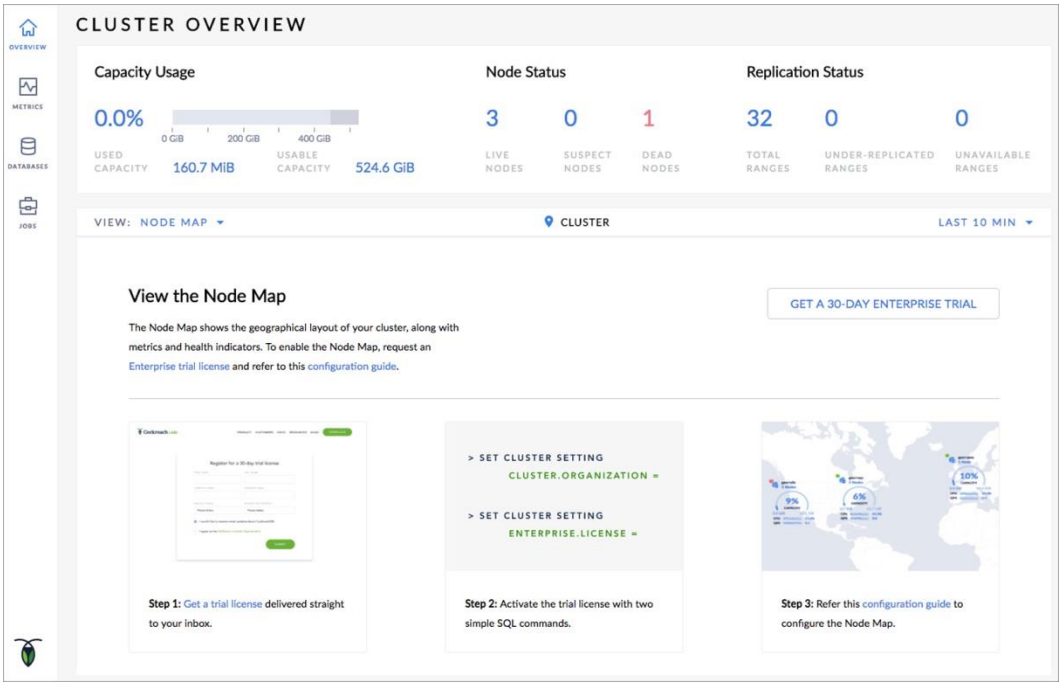


To access the UI admin for CockroachDB use the following command to port forward on your Kubernetes cluster:

```
kubectl port-forward crdb-cockroachdb-0 8080
```

Access the admin UI by visiting http://your-ip:8080/ in your web browser. Figure 8 is an image of what it looks like.

Figure 8. CockroachDB admin UI



5. Implementing Orchestration, APIs, and Monitoring

5.1. Cisco CloudCenter

Basic installation and cloud-specific steps needed to install and pre-configure the Cisco CloudCenter environment for Cisco Hybrid Cloud Platform can be found in the solution guide Cisco Hybrid Cloud Platform with Cisco Container Platform, HyperFlex and Application Centric Infrastructure (ACI).

5.1.1. CloudCenter configuration

5.1.1.1. Creating and connecting Orchestrator for on-premises Kubernetes clusters (Cisco Container Platform)

5.1.1.2. Create Cisco Container Platform Advanced Messaging Queuing Protocol (AMQP) and CCO hosts

If you are using an existing CloudCenter installation you will need to deploy another AMQP and CCO instance in your FlashStack VM environment to manage your Kubernetes deployments. Then, perform the steps in the following document to: Convert to a Kubernetes-Specific CCO

5.1.1.3. Configure a Cloud and Cloud Account

Instructions for adding a Kubernetes cloud in CloudCenter can be found in the solution guide Cisco Hybrid Cloud Platform with Cisco Container Platform, HyperFlex and Application Centric Infrastructure (ACI).

6. Troubleshooting

Following is a quick-reference troubleshooting guide for the overall solution.

6.1. Network issues

6.2. Cisco Container Platform (CCP) general troubleshooting

Troubleshooting Cisco Container Platform

Troubleshooting Pure Service Orchestrator

6.3. CloudCenter (C3) general troubleshooting

The published CloudCenter documentation is very thorough.

C3 Troubleshooting

C3 Log Files

6.4. FlashStack

6.4.1. NTP server non-reachability

Confirm that all Cisco UCS servers are using the same NTP server as the rest of your infrastructure.

7. Reference Details

7.1. Hardware and software versions

Componenets	Version
Cisco Container Platform	1.4.1
Cisco CloudCenter	4.9.1
Cisco UCS Manager	3.2(3c)
VMware ESXi	6.5 U1

7.2. Documentation references

Document	Description
Cisco Container Platform Installation Guide	Describes installation of Cisco Container Platform in your deployment environment
Cisco Container Platform User Guide	Describes administration and management of Kubernetes clusters, and how to deploy applications on them
Cisco Container Platform API Guide	Describes the Cisco Container Platform APIs
CloudCenter Installation Guide	Describes installation of Cisco CloudCenter in your environment
CVD Infrastructure Guide	Describes base infrastructure installation for Cisco Hybrid Cloud Platform for Google Cloud
FlashStack Design and Installation Guides	Describes reference solutions for Cisco FlashStack
FlashStack Virtual Server Infrastructure with iSCSI Storage for VMware vSphere 6.5 U1	Describes installation of the FlashStack solution used in this solution guide
Pure Service Orchestrator	Describes deployment of Pure Service Orchestrator

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **https://www.cisco.com/go/offices**.

Printed in USA C07-741916-00 04/19