Deploy SAP Data Hub on FlexPod with Cisco Container Platform
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>3</td>
</tr>
<tr>
<td>FlexPod overview</td>
<td>6</td>
</tr>
<tr>
<td>Implementation</td>
<td>9</td>
</tr>
<tr>
<td>Create a Kubernetes cluster on the Cisco Container Platform control plane</td>
<td>12</td>
</tr>
<tr>
<td>Install the NetApp Trident CSI plug-in</td>
<td>17</td>
</tr>
<tr>
<td>Configure NetApp iSCSI back end for Trident CSI: Edit and apply back-end JSON template</td>
<td>19</td>
</tr>
<tr>
<td>Configure NetApp Trident CSI persistent volumes as the default storage class</td>
<td>19</td>
</tr>
<tr>
<td>Install SAP Data Hub 2.7</td>
<td>21</td>
</tr>
<tr>
<td>Conclusion</td>
<td>41</td>
</tr>
<tr>
<td>For more information</td>
<td>42</td>
</tr>
</tbody>
</table>
Introduction

Enterprises are inundated with data, but extracting data that is actionable from this vast amount of data can be difficult. Adding to this challenge is the rapid and broad adoption of containerized workloads (Kubernetes) across enterprise, service provider and telco, public sector, healthcare, and many more markets.

A recent survey, summarized in Figure 1, showed that from 2018 to 2019, the use of containers in development, testing, and production environments grew rapidly. Most notably, the use of containers in production environments increased significantly. In 2019, 84 percent of respondents were using containers in production environments: an impressive jump from 73 percent in 2018, and from 23 percent in 2016. This growth is a result of organizations’ increased trust in containers and use of them in user-facing applications. Another 14 percent of survey respondents have future plans to use containers in their production environments.

The proof-of-concept (PoC) environment is the only area in which the use of containers has shown a gradual decline over the past few years: an indication that containers are seen not as just an idea, but instead are being adopted in production deployments in the real world. Only slightly more than 2 percent of respondents reported no plans to use containers in 2019.

In addition, container-based usage is growing rapidly at scale.

As organizations are trusting their production workloads to containers, they also are using more containers. The number of respondents using 249 or fewer containers decreased by 26 percent between 2018 and 2019. Conversely, the number of respondents using 250 or more containers increased by 28 percent, to more than half. The most significant change was in the number of organizations using fewer than 50 containers, which fell by 43 percent (Figure 2).
However, as Kubernetes has matured as a viable platform during this period of rapid adoption, the complexity of installing it and operating it over time at the enterprise level has increased. The container space is crowded with numerous products and services that all aim to provide Kubernetes at various levels of complexity and capability. Such a fragmented environment has led to a number of challenges for both IT operations and cloud administrators as well as for developers, including the following:

- IT operations and cloud administrators lack common tools to manage and deploy Kubernetes in heterogeneous environments.
- Developers lack common development experience with containers (best practices), slowing down development.

This document describes a solution stack—SAP Data Hub on Cisco® Container Platform on NetApp FlexPod—that expressly addresses the problem of achieving rapid deployment and operations for containerized workloads while maintaining data locality to deliver needed outcomes in a cost-effective and timely manner.

SAP Data Hub (a foundational element of SAP Data Intelligence) is a microservices platform built on Kubernetes that runs in public clouds (cloud provider Kubernetes) as well as on an organization’s premises (requires a Kubernetes platform). Using SAP Data Hub, you can manage the data deluge to rapidly deliver enriched, trustworthy, intelligent data, unlocking the value of all data: from the Internet of Things (IoT) to machine learning and beyond.

SAP Data Hub simplifies your end-to-end data orchestration with automated, reliable data processing across the entire data landscape. Innovative data pipelines fluidly and automatically process a wide variety...
of data, in the exact manner in which the data needs to be processed, while eliminating the need for mass data movement.

A guiding goal in the creation and continued rapid evolution of the Cisco Container Platform has been to help customers overcome these challenges and deliver what they want: the capability to install and operate a lightweight containers-as-a-service (CaaS) on-premises platform that is built on pure upstream Kubernetes. With Cisco Container Platform, customers can do the following:

- Quickly (within hours) and seamlessly deploy a production-class Kubernetes environment.
- Rapidly deploy containerized application clusters on-premises and in the public cloud of choice.
- Seamlessly manage applications throughout the lifecycle and dynamically scale them (with Cisco CloudCenter™ Suite, a cloud management platform).
- Quickly set up and accelerate artificial intelligence (AI) and machine-learning (ML) workloads using multiple graphics processing units (GPUs) and Kubeflow (automation framework).
- Eliminate shadow IT and development team silos by delivering a native Kubernetes experience through an easy-to-operate self-service portal, allowing developers to focus on delivering good software.
- Build once, run anywhere. Either build applications on-premises and seamlessly deploy them in the public cloud, or the reverse, through a consistent and secure environment.
- Deploy applications in cloud-native Kubernetes environments, getting the most from cloud investments while maintaining compliance and security through a single-pane deployment model.
- Help ensure that corporate policies (legal, finance, security, and privacy) are enforced by providing a common environment on which IT operations, development, and security teams can operate.
- Give developers access to the best platform and tools while enabling IT operations and security teams to maintain visibility and control over application and Kubernetes resource utilization across the premises and in public clouds.

Cisco Container Platform is a ready-to-use, lightweight, multicluster container management software platform for deploying production-class upstream Kubernetes environments and managing their lifecycle across on-premises and public cloud environments. Cisco Container Platform automates the installation of 100 percent upstream Kubernetes clusters with self-service and centralized automation and management capabilities (Figure 3).
FlexPod overview

FlexPod is a best-practices data center architecture that includes the following components (Figure 4):

- Cisco Unified Computing System™ (Cisco UCS®)
- Cisco Nexus® switches
- Cisco MDS switches
- NetApp All Flash FAS (AFF) systems
These components are connected and configured according to the best practices of both Cisco and NetApp to provide an excellent platform for running a variety of enterprise workloads with confidence. FlexPod can scale up for greater performance and capacity (adding computing, network, or storage resources individually as needed), or it can scale out for environments that require multiple consistent deployments (such as rolling out additional FlexPod stacks). The reference architecture discussed in this document uses Cisco Nexus 9000 Series Switches for the network switching element.

One of the main benefits of FlexPod is its ability to maintain consistency with scale. Each of the component families shown in Figure 4 (Cisco UCS, Cisco Nexus, and NetApp AFF) offers platform and resource options to scale the infrastructure up or down, while supporting the same features and functions that are required under the configuration and connectivity best practices of FlexPod.
FlexPod design principles
FlexPod addresses four main design principles: availability, scalability, flexibility, and manageability. The architecture goals are as follows:

- **Application availability**: Helps ensure that services are accessible and ready to use
- **Scalability**: Addresses increasing demands with appropriate resources
- **Flexibility**: Provides new services and recovers resources without requiring infrastructure modification
- **Manageability**: Facilitates efficient infrastructure operations through open standards and APIs

FlexPod design for SAP Data Hub with Cisco Container Platform
The Cisco Nexus 9000 Series Switches support two modes of operation: NX-OS standalone mode, using Cisco NX-OS Software, and ACI fabric mode, using the Cisco Application Centric Infrastructure (Cisco ACI™) platform. In standalone mode, the switch performs like a typical Cisco Nexus switch, with increased port density, low latency, and 40, 10, and 25 Gigabit Ethernet connectivity. In fabric mode, the administrator can take advantage of the Cisco ACI platform. The design discussed here uses the standalone mode.

FlexPod with NX-OS mode is designed to be fully redundant in the computing, network, and storage layers. There is no single point of failure from a device or traffic path perspective. Figure 5 shows the connection of the various elements of the FlexPod design for SAP Data Hub with Cisco Container Platform.

Cisco Container Platform automates repetitive tasks and simplifies complex ones so you can make more productive use of containers.

Trident is a fully supported open-source project maintained by NetApp. It has been designed from the foundation to help you meet the sophisticated persistence demands of your containerized applications.

SAP Data Hub provides visibility and access to a broad range of data systems and assets; it allows easy and fast creation of powerful, organization-spanning data pipelines; and it optimizes data-pipeline processing speed with a push-down distributed processing approach at each step.

FlexPod helps you orchestrate and extract value from distributed data for an intelligent enterprise.
This design uses the fourth-generation Cisco UCS 6454 Fabric Interconnects and the Cisco UCS Virtual Interface Card (VIC) 1400 platform in the servers. The Cisco UCS B200 M5 Blade Servers in the Cisco UCS chassis use the Cisco UCS VIC 1440 connected to the Cisco UCS 2408 Fabric Extender I/O module (IOM), and each virtual network interface card (vNIC) has a speed of 20 Gbps. The Cisco UCS BC220 M5 Rack Servers use the Cisco UCS VIC 1457 connected to the Cisco UCS 6454 Fabric Interconnects with 25-Gbps Ethernet, and each vNIC has a speed of 50 Gbps. The fabric interconnects connect through 100-Gbps port channels to virtual port channels (vPCs) across the Cisco Nexus 9336C-FX2 Switches. The connectivity between the Cisco Nexus switches and the NetApp AFF A800 storage cluster is also 100 Gbps, with port channels on the storage controllers and vPCs on the switches. This configuration supports IP-based storage protocols (Network File System [NFS], Common Internet File System [CIFS], and Small Computer System over IP [iSCSI]) over a high-speed network between the storage and the Cisco UCS servers.

**Implementation**

SAP Data Hub is an all-in-one data orchestration solution that discovers, refines, enhances, and manages any type, variety, and volume of data across your entire distributed data landscape. Because this application is delivered as a Kubernetes application, you need a Kubernetes-provisioned cluster. For an on-premises solution, Cisco Container Platform is a ready-to-use lightweight multicluster container management software platform for deploying production-class upstream Kubernetes environments and...
managing their lifecycle across on-premises and public clouds. Cisco Container Platform automates the installation of 100 percent upstream Kubernetes clusters with self-service and centralized automation and management capabilities. SAP Data Hub requires persistent volumes to operate within the Kubernetes pods of the application, and when Cisco Container Platform is installed in a FlexPod environment, deployment of the NetApp Trident Container Storage Interface (CSI) storage plug-in accomplishes this task. The NetApp Trident CSI plug-in allows Kubernetes pods to get the needed storage from either an NFS or an iSCSI back end on the NetApp storage.

This document presents the procedures and references the documents necessary to perform the following tasks:

- Create a Cisco Container Platform Kubernetes tenant cluster specifically to meet the requirements for SAP Data Hub.
- Present the steps and reference extended documentation for installing the NetApp Trident CSI plug-in using iSCSI.
- Present the steps for deploying the SAP Data Hub application on Cisco Container Platform.

**Prerequisites**

The following items need to be preconfigured before you begin the setup and configuration of a Cisco Container Platform tenant cluster on FlexPod:

- A Linux host that meets the following requirements:
  - The kubectl client binary installed (Release v1.14.8)
  - Access to the Internet to download the Trident CSI plug-in and SAP Data Hub registry
  - Network routable to the created Cisco Container Platform Kubernetes tenant cluster
  - Python Release 2.7 and the associated PyYAML package
  - At least 50 GB of free space on the disk for SAP Data Hub images
  - Docker Release 1.12.6 or later
  - The helm client binary installed (Release v2.15.2)

- A Cisco Container Platform control plane installed and configured to deploy Kubernetes tenant clusters

- A FlexPod environment with a storage virtual machine (SVM) configured to accept the iSCSI initiator

- An operational registry service from which to pull and push SAP Data Hub images; Cisco Container Platform can also provide this service from a Kubernetes cluster as documented in the following links:
  - Configuring Add-ons for v3 Clusters
  - Using Harbor Registry in Tenant Clusters

- An operational object storage solution compliant with the Amazon Simple Storage Service (S3) API; the test case presented in this document uses minio for SAP Data Hub checkpoint store validation during the installation process
Validated hardware and software

Table 1 lists the hardware and software versions used during the solution validation process. Note that Cisco and NetApp have interoperability matrices that should be referenced to determine support for any specific implementation of FlexPod. See the following documents for more information:

- [NetApp Interoperability Matrix Tool](#)
- [Cisco UCS Hardware and Software Interoperability Tool](#)

Table 1. Validated hardware and software versions

<table>
<thead>
<tr>
<th>Layer</th>
<th>Device</th>
<th>Image</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computing</td>
<td>Cisco UCS 6454 Fabric Interconnect, Cisco UCS B200 M5 Blade Server with Cisco UCS VIC 1440, and Cisco UCS C220 M5 Rack Server with Cisco UCS VIC 1457</td>
<td>Release 4.1(1b)</td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>Second Generation Intel® Xeon® Scalable processor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td>12 x 32-GB DDR4 memory modules</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network</td>
<td>Cisco Nexus 9336C-FX2 Switch in NX-OS standalone mode</td>
<td>Release 7.0(3)I7(7)</td>
<td></td>
</tr>
<tr>
<td>Storage network</td>
<td>Cisco MDS 9132T 32-Gbps 32-Port Fibre Channel Switch</td>
<td>Release 8.3(2)</td>
<td></td>
</tr>
<tr>
<td>Storage</td>
<td>NetApp AFF A800</td>
<td>NetApp ONTAP Release 9.7</td>
<td></td>
</tr>
<tr>
<td>Operating system</td>
<td>Red Hat Enterprise Linux (RHEL) Release 7.6</td>
<td>Kernel 3.10.0-957</td>
<td></td>
</tr>
<tr>
<td>Software</td>
<td>Cisco Container Platform control plane:</td>
<td>Release 5.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cisco Container Kubernetes tenant cluster</td>
<td>Release 1.14.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NetApp CSI Plug-in</td>
<td>Release 20.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SAP Data Hub</td>
<td>Release 2.7 Update 3</td>
<td></td>
</tr>
</tbody>
</table>
Create a Kubernetes cluster on the Cisco Container Platform control plane

Use the procedure described in this section to create a Kubernetes cluster on the Cisco Container Platform control plane.

Perform initial login

After installing the Cisco Container Platform management control plane (installation procedure), log in to the user interface with the necessary credentials.

Create a cluster for SAP Data Hub

On the main login page, you should already be in the Cisco Container Platform v3 section. If you are not, select Clusters on the menu at the left and make sure that the drop-down is set to Version 3.

Follow the online procedure for creating a VMware vSphere on-premises cluster.

To create a properly sized cluster for SAP Data Hub 2.7 and later, refer to the sizing guide from SAP:

- [SAP Data Hub sizing guide](#)

The cluster created in this document uses the following setup for a Kubernetes 1.14.8 cluster:

- Three-node primary node group (for high availability) with two virtual CPUs (vCPUs) and 16 Gb of RAM each (nodes do not run any SAP Data Hub pods)
- Five-node worker node group with two vCPUs and 32 Gb of RAM for deploying and running the SAP Data Hub application
If you are using an internal container registry (for example, Harbor Registry on a Cisco Container Platform with add-ons), extract the Certificate Authority (CA) certificate from the web service to allow the Kubernetes cluster to pull images with the following command and paste it into the ROOT CA REGISTRIES section when creating the cluster:

**Note:** This command was run from the Linux host mentioned earlier in the Requirements section. Remove the “https://” prefix (for example, change "https:<IP address>:443" to "<IP address>:443").

```
    ubuntu@Ubuntu-jump:~$ H=192.168.92.102:443
    # Generate the CA certificate
    openssl s_client -showcerts -connect ${H} </dev/null 2>/dev/null |
      openssl x509 -outform PEM | tee ${H}_CA.crt
    -----BEGIN CERTIFICATE-----
    MIIE/jCCA2agAwIBAgIQO1GyAb9ggqARzGF7pArTANBgkqhkiG9w0BAQsFADA3
    MR0wDAYDVQQKEwVjaXNjbybwMAwGA1UEBhMCVVRnymQY2NwLWNlcnQt
    bWFuYm9jdWJcAeFw0yMDAxMTgyMDI3NhawFw0yMjEyNzA5MDI3MD4wMTABgNV
    BAoTDGlnYnctbWFuYWdlcjElMCMGA1UEAxMcY2NwLWluZ3Jlc3MuY2NwLmNjcy
    LWhhcmVjcyCCAiIwDQYJKoZIhvcNAQEBQADgggIPADCCAdQgIgYBAMBgkqhkiG9w0BAQsFAD
    V60D9u9yeIT7/y1lWWhHktT5TP+GR4+x+aUMERAK5nBHiOV84ALrWaryRAAnOGio
    fr7m2rFg5zKOA3trvNpN4aGiEC9IPZ41ig+6osGhOKPzPz3PWjE1InDTmuUg
    g3wt8ovpofxy1HR2oBaBpA0eTcy6tILLGmWKwpTL87cbNW/jEDL0cKduRHemynr
    1BSi1b2xZzWISgSlliVZSjFaRvYqoAm1SDBig2KTkXN12Ec68FKyvrVR
    N2rm7FVHx7Y34aNo3q5whKqTOaHD37xl4zAVBrqFMY2qQWUjg8+Ds0uyN2mhS
    6R8WwAqFXThtVhCRZG7/jhGR2yt+q00bdST4LumCPvN16zATuLxV100izlJYvGmL
    mAdSL7Em2HA6NCyRearEjUNwuKQFMiyzAgMAA0GCDIy8MV9MA4GAlUdEB/wQEAwIF
    oDAMbGNRHMBAf8EjaJAMF0GAIUdEQRWMFSCHGnjcC1pbmdyZXNzLmNjcC5jY3A5
    M1o0XJib3KCLm5naW54LWluZ3Jlc3My329udHJvGbxlci5jY3Auc3ZjLmNsdXN0
```
After the cluster is created, it is ready for installation of the NetApp Trident CSI plug-in and SAP Data Hub.
Download the kubeconfig file for access to the cluster using kubectl

To install the NetApp Trident plug-in, a Linux host with the kubectl application must be installed. The kubeconfig file will allow the kubectl application to send commands to the Kubernetes cluster.

You can download the kubeconfig file using either of two methods:

1. From the main cluster menu, open the drop-down menu in the Actions columns of the intended cluster and choose Download Kubeconfig.

2. Selecting the cluster and click the Download Kubeconfig button.
Export the KUBECONFIG environment variable so that it points to the downloaded kubeconfig file and verify that you can communicate with the cluster.

```
ubuntu@Ubuntu-jump:~$ export KUBECONFIG=/Downloads/ccp92-cluster-3.yaml
```

```
ubuntu@Ubuntu-jump:~$ kubectl get nodes
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATUS</th>
<th>ROLES</th>
<th>AGE</th>
<th>VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccp92-cluster-3-0-master-0</td>
<td>Ready</td>
<td>master</td>
<td>5d18h</td>
<td>v1.14.8</td>
</tr>
<tr>
<td>ccp92-cluster-3-0-master-1</td>
<td>Ready</td>
<td>master</td>
<td>5d18h</td>
<td>v1.14.8</td>
</tr>
<tr>
<td>ccp92-cluster-3-0-master-2</td>
<td>Ready</td>
<td>master</td>
<td>5d18h</td>
<td>v1.14.8</td>
</tr>
<tr>
<td>ccp92-cluster-3-1-node-gr-0</td>
<td>Ready</td>
<td>&lt;none&gt;</td>
<td>5d18h</td>
<td>v1.14.8</td>
</tr>
<tr>
<td>ccp92-cluster-3-1-node-gr-1</td>
<td>Ready</td>
<td>&lt;none&gt;</td>
<td>5d18h</td>
<td>v1.14.8</td>
</tr>
<tr>
<td>ccp92-cluster-3-1-node-gr-2</td>
<td>Ready</td>
<td>&lt;none&gt;</td>
<td>5d18h</td>
<td>v1.14.8</td>
</tr>
<tr>
<td>ccp92-cluster-3-1-node-gr-3</td>
<td>Ready</td>
<td>&lt;none&gt;</td>
<td>5d18h</td>
<td>v1.14.8</td>
</tr>
<tr>
<td>ccp92-cluster-3-1-node-gr-4</td>
<td>Ready</td>
<td>&lt;none&gt;</td>
<td>5d18h</td>
<td>v1.14.8</td>
</tr>
</tbody>
</table>

**Collect Kubernetes node IP addresses and iSCSI initiators**

Using the complementary private Secure Shell (SSH) key on the Linux host, run the following script to obtain the IP addresses of the nodes and iSCSI initiator names. This information can be used in NetApp ONTAP System Manager to configure the intended iSCSI access for the dynamically created persistent volume claims created by the Kubernetes cluster.

```
ubuntu@Ubuntu-jump:~$ while read ip; do echo -n "IPaddress=${ip} - " ; ssh $ip cat /etc/iscsi/initiatorname.iscsi </dev/null; done < <(kubectl get no -o jsonpath='{range.items[*].status.addresses[?(@.type=="InternalIP")]}{.address}{"\n"}{end}')
```

```
IPaddress=192.168.92.184 - InitiatorName=iqn.2005-03.org.open-iscsi:e89ed91cfdf
IPaddress=192.168.92.185 - InitiatorName=iqn.2005-03.org.open-iscsi:11868af880
IPaddress=192.168.92.186 - InitiatorName=iqn.2005-03.org.open-iscsi:c78f63f2923
IPaddress=192.168.92.187 - InitiatorName=iqn.2005-03.org.open-iscsi:6191668fc1
IPaddress=192.168.92.188 - InitiatorName=iqn.2005-03.org.open-iscsi:adf51eec7c4d
IPaddress=192.168.92.189 - InitiatorName=iqn.2005-03.org.open-iscsi:1a4b9383b059
IPaddress=192.168.92.191 - InitiatorName=iqn.2005-03.org.open-iscsi:a3e351201f8a
IPaddress=192.168.92.190 - InitiatorName=iqn.2005-03.org.open-iscsi:b67d8c1b46b
```
Install the NetApp Trident CSI plug-in

This section presents the steps for installing the NetApp Trident plug-in as described in the NetApp Trident documentation.

Qualify the Kubernetes cluster

Verify the version, permissions, and network connectivity for the NetApp Trident plug-in.

```
ubuntu@Ubuntu-jump:~$ kubectl version
```

```
ubuntu@Ubuntu-jump:~$ # Are you a Kubernetes cluster administrator?

ubuntu@Ubuntu-jump:~$ kubectl auth can-i '*' '*' --all-namespaces
yes
```

```
ubuntu@Ubuntu-jump:~$ # Can you launch a pod that uses an image from Docker Hub and can reach your storage system over the pod network?

ubuntu@Ubuntu-jump:~$ kubectl run -i --tty ping --image=busybox --restart=Never --rm -- ping 192.168.92.10
```

If you don't see a command prompt, try pressing enter.

64 bytes from 192.168.92.10: seq=1 ttl=63 time=0.145 ms
64 bytes from 192.168.92.10: seq=2 ttl=63 time=0.144 ms
64 bytes from 192.168.92.10: seq=3 ttl=63 time=0.185 ms
64 bytes from 192.168.92.10: seq=4 ttl=63 time=0.142 ms

^C

```
--- 192.168.92.10 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.142/0.211/0.441 ms
pod "ping" deleted
```

Download the Trident CSI plug-in

Download the latest version of the Trident installer bundle from the Downloads section and extract the files.

The version used for this document is Release 20.01.0.

```
ubuntu@Ubuntu-jump:~$/netapp$ wget -q https://github.com/NetApp/trident/releases/download/v20.01.0/trident-installer-20.01.0.tar.gz
ubuntu@Ubuntu-jump:~$/netapp$ tar -xf trident-installer-20.01.0.tar.gz
ubuntu@Ubuntu-jump:~$/netapp$ cd trident-installer
/home/ubuntu/trident-installer
```

© 2020 Cisco and/or its affiliates. All rights reserved.
Install Trident CSI on Kubernetes

Run the trident install command and verify that the trident pods are running and that the version is correct.

```
ubuntu@Ubuntu-jump:~/netapp/trident-installer$ ./tridentctl install -n trident
INFO Starting Trident installation. namespace=trident
INFO Created namespace. namespace=trident
INFO Created service account.
INFO Created cluster role.
INFO Created cluster role binding.
INFO Created custom resource definitions. namespace=trident
INFO Added finalizers to custom resource definitions.
INFO Created Trident pod security policy.
INFO Created Trident service.
INFO Created Trident secret.
INFO Created Trident deployment.
INFO Created Trident daemonset.
INFO Waiting for Trident pod to start.
INFO Trident pod started. namespace=trident pod=trident-csi-6bbd889f9f-bszg9
INFO Waiting for Trident REST interface.
INFO Trident REST interface is up. version=20.01.0
INFO Trident installation succeeded.
```

```
ubuntu@Ubuntu-jump:~/netapp/trident-installer$ kubectl get pod -n trident
NAME                           READY   STATUS    RESTARTS   AGE
trident-csi-4bvx9              2/2     Running   0          47s
trident-csi-6bbd889f9f-bszg9   3/3     Running   0          47s
trident-csi-9qph7              2/2     Running   0          47s
trident-csi-f7cjv              2/2     Running   0          47s
trident-csi-hkjdd              2/2     Running   0          47s
trident-csi-nzdr1              2/2     Running   0          47s
trident-csi-vp82h               2/2     Running   0          47s
trident-csi-wwc28              2/2     Running   0          47s
trident-csi-xbngs              2/2     Running   0          47s
```

```
ubuntu@Ubuntu-jump:~/netapp/trident-installer$ ./tridentctl -n trident version
+-----------------+-----------------+
<table>
<thead>
<tr>
<th>SERVER VERSION</th>
<th>CLIENT VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.01.0</td>
<td>20.01.0</td>
</tr>
</tbody>
</table>
+-----------------+-----------------+
```

The Trident CSI plug-in is now running. You can configure either NFS or iSCSI, or both protocols, as back ends for the NetApp storage.
Configure NetApp iSCSI back end for Trident CSI: Edit and apply back-end JSON template

From the sample-input directory found in the Trident installer, copy the backend.json file up one directory, edit it with the credential information for your NetApp storage, and apply it to your Kubernetes cluster.

```json
{
   "version": 1,
   "storageDriverName": "ontap-san",
   "backendName": "aa14-a800iSCSI",
   "managementLIF": "192.168.92.10",
   "dataLIF": "192.168.92.54",
   "svm": "CCP-VMs",
   "username": "xxxxxx",
   "password": "PaxxWoxd"
}
```

Configure NetApp Trident CSI persistent volumes as the default storage class

Use the procedures in this section to configure the NetApp Trident CSI persistent volumes.

Create a Kubernetes storage class

From the sample-input directory found in the trident-installer, copy the storage-class-csi.yaml.templ file up one directory as `storage-class-basic.yaml`, edit the file, and replace `__BACKEND_TYPE__` with the storage driver name `ontap-san`.

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
   name: basic
provisioner: csi.trident.netapp.io
parameters:
   backendType: "ontap-san"
```
The basic Trident storage class is created:

```
kubectl create -f storage-class-basic.yaml
```

The basic storage class is listed:

```
kubectl get sc
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>PROVISIONER</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic</td>
<td>csi.trident.netapp.io</td>
<td>4s</td>
</tr>
<tr>
<td>standard (default)</td>
<td>kubernetes.io/vsphere-volume</td>
<td>40m</td>
</tr>
</tbody>
</table>

The Trident controller is run as:

```
./tridentctl -n trident get sc basic -o json
```

```
{
  "items": [
    {
      "Config": {
        "version": "1",
        "name": "basic",
        "attributes": {
          "backendType": "ontap-san"
        },
        "storagePools": null,
        "additionalStoragePools": null
      },
      "storage": {
        "aa14-a800iSCSI": [
          "aa14_a800_1_NVME_SSD_1",
          "aa14_a800_2_NVME_SSD_1"
        ]
      }
    }
  ]
}
```

**Promote the Trident storage class to the default**

First demote the standard class used by the Kubernetes cluster by applying a Kubernetes **patch** command. Then apply another **patch** command to promote the basic Trident CSI storage class to the default class.

```
kubectl get sc
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>PROVISIONER</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic</td>
<td>csi.trident.netapp.io</td>
<td>1h</td>
</tr>
<tr>
<td>standard (default)</td>
<td>kubernetes.io/vsphere-volume</td>
<td>2h</td>
</tr>
</tbody>
</table>

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

```
storageclass.storage.k8s.io/standard patched
```
Install SAP Data Hub 2.7

This section presents the procedures for installing SAP Data Hub.

Check the configuration of the Linux host for the SAP Data Hub installation

Before running the SAP Data Hub installer, verify that the software and configurations listed in this section are set up.

Install the helm client and verify the installation

Install the same version of helm on the Linux host that is already deployed on the Cisco Container Platform Kubernetes cluster and verify the installation.

```bash
ubuntu@Ubuntu-jump:~/git/sapdhinstall$ curl -s https://get.helm.sh/helm-v2.15.2-linux-amd64.tar.gz | sudo tar zxvf - --strip-components=1 -C /usr/local/bin linux-amd64/helm
```

```bash
ubuntu@Ubuntu-jump:~/git/sapdhinstall$ helm version
Client: &version.Version{SemVer:"v2.15.2", GitCommit:"8dc272473e5f2a7bf58ce79bb5c3691db54c96b", GitTreeState:"clean"}
Server: &version.Version{SemVer:"v2.15.2", GitCommit:"8dc272473e5f2a7bf58ce79bb5c3691db54c96b", GitTreeState:"clean"}
```

Configure Kubernetes cluster roles for SAP Data Hub installation

Run the following commands to configure the cluster roles.

```bash
ubuntu@Ubuntu-jump:~/git/sapdhinstall$ kubectl create clusterrolebinding tiller-cluster-rule --clusterrole=cluster-admin --serviceaccount=kube-system:tiller
clusterrolebinding.rbac.authorization.k8s.io/tiller-cluster-rule created
```

```bash
ubuntu@Ubuntu-jump:~/git/sapdhinstall$ kubectl create clusterrolebinding kubernetes-dashboard --clusterrole=cluster-admin --serviceaccount=kube-system:kubernetes-dashboard
clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
```

Verify Python and the PyYAML module

Run the following command to check that Python and the associated PyYAML module are installed.

Install the same version of helm on the Linux host that is already deployed on the Cisco Container Platform Kubernetes cluster and verify the installation.
ubuntu@Ubuntu-jump:$ python2.7 -c 'import yaml;' & echo success!
success!

**Check the Docker version and log in to push images to the container repository**

Verify that Docker Release 1.12 or later is installed on the Linux install host.

ubuntu@Ubuntu-jump:$ docker --version
Docker version 19.03.5, build 633a0ea838

ubuntu@Ubuntu-jump:$ docker version
Client: Docker Engine - Community
  Version:          19.03.5
  API version:      1.40
  Go version:       go1.12.12
  Git commit:       633a0ea838
  Built:            Wed Nov 13 07:29:52 2019
  OS/Arch:          linux/amd64
  Experimental:     false

Server: Docker Engine - Community
  Engine:
    Version:          19.03.5
    API version:      1.40 (minimum version 1.12)
    Go version:       go1.12.12
    Git commit:       633a0ea838
    Built:            Wed Nov 13 07:28:22 2019
    OS/Arch:          linux/amd64
    Experimental:     false
  containerd:
    Version:          1.2.10
    GitCommit:        b34a5c8af56e510852c35414db4c1f4fa6172339
  runc:
    Version:          1.0.0-rc8+dev
    GitCommit:        3e425f80a8c931f88e6d94a8c831b9d5aa481657
  docker-init:
    Version:          0.18.0
    GitCommit:        fec3683
If you are publishing the downloaded SAP Data Hub container images to a local repository with the self-signed certificate, then you need to obtain the self-signed CA certificate and log in with the appropriate credentials.

```bash
ubuntu@Ubuntu-jump:$ H=192.168.92.102; if [ "${H}" = "${H##*:}" ]; then CON="${H}:443"; else CON=${H}; fi; # The if conditional appends the port 443 if a port isn’t given.

ubuntu@Ubuntu-jump:$ sudo mkdir -p /etc/docker/certs.d/${H}

ubuntu@Ubuntu-jump:$ openssl s_client -showcerts -connect ${CON} </dev/null 2>/dev/null | openssl x509 -outform PEM | sudo tee /etc/docker/certs.d/${H}/ca.crt

-----BEGIN CERTIFICATE-----
MIIE/jCCA2agAwIBAgIQO1GyxAb9gdqArzGF7pArTANBgkqhkiG9w0BAQsFADA3
MRo0DAYVQEQKxVjaXNzbAKBgvNAoTA2NjcDEZMBcGA1UEAxMQY2NwLWNlcnc9t
bWFuYWlscjEJungaMDAxMTgyMDI3NDhafW0yMjAxMtcyMDI3NDhafMD4xFTATBgNV
BAdOTGN1cnQtbfWFuYWldcjE1MCMGA1UEAxMcY2NwLWNlcnc9t3Jlci3Mu2Y2Nw
LWNlcnc9tDkY
LWxhcmlvcmVuY29tCmVuY29tPi9OaGVQc29sdnNwMVQdBWluZ3Jlci5jY3A5
LWhhcmVjcmVuY29tPi9OaGVQc29sdnNwMVQdBWluZ3Jlci5jY3A5
-----END CERTIFICATE-----

ubuntu@Ubuntu-jump:$ docker login ${H} # Enter in authentication information to login

Username: admin
Password: ********
```

WARNING! Your password will be stored unencrypted in /home/ubuntu/.docker/config.json. Configure a credential helper to remove this warning. See https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded

**Install SAP Data Hub Foundation using Software Lifecycle Container Bridge without Maintenance Planner and Host Agent**

This section describes the installation process for SAP Data Hub using the Software Lifecycle (SL) plug-in. Other methods can be used as well; the deployment of SAP Data Hub is the same.

Before beginning, verify that you have downloaded SAP Data Hub 2.7 Foundation zip file and extracted it on the Linux host machine.

Change the directory to the SAP Data Hub installation source.

Change the directory to slplugin/workdir, run the ./setup.sh file, and begin the SAP Data Hub installation

Here is an example of the installation process:

```
ubuntu@Ubuntu-jump:~/SAPDataHub-2.7.155-Foundation/slplugin/workdir$ ./setup.sh
2020-03-04T09:37:41.051-0800 INFO  cmd/cmd.go:244 1> admin@ccp92-cluster-3
2020-03-04T09:37:41.054-0800 INFO  k8s/context.go:95
----------------------------
Current kubernetes context: admin@ccp92-cluster-3
----------------------------
```

Other methods can be used as well; the deployment of SAP Data Hub is the same.

```
2020-03-04T09:37:41.129-0800 INFO  cmd/cmd.go:244 1> Kubernetes master is running at https://192.168.92.183:6443
2020-03-04T09:37:41.130-0800 INFO  cmd/cmd.go:244 1> KubeDNS is running at https://192.168.92.183:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
2020-03-04T09:37:41.130-0800 INFO  cmd/cmd.go:244 1> To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
2020-03-04T09:37:41.131-0800 INFO  k8s/context.go:120
----------------------------
Current kubernetes cluster: Kubernetes master is running at https://192.168.92.183:6443
KubeDNS is running at https://192.168.92.183:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

--------

**SLC Bridge executable information**

Executable:  /home/ubuntu/SAPDataHub-2.7.155-Foundation/slplugin/bin/slplugin
Build date:  2019-09-30 14:28:13 UTC
Git branch:  fa/rel-1.0
Git revision:  f66f2654ec64185b328ea73e51860a47a16a3af0
Platform:  linux
Architecture:  amd64
Target Software Level
You are about to install or update the following product from directory /home/ubuntu/SAPDataHub-2.7.155-Foundation:

Product:                     SAP DATA HUB - FOUNDATION 2
Software Component Version:  SAP DATA HUB - FOUNDATION 2
Technical Product Name:      DH_FOUNDATION
Technical Release:           2.0
Support Package:             SP007
Patch Level:                 3
PPMS ID:                     73554900100200008830
Support Package PPMS ID:     73555000101100041085
Support Component:           EIM-DH
Product Version PPMS ID:     73554900100900002861

Choose action Next [n/<F1>]: n

Checking the prerequisites for product SAP DATA HUB - FOUNDATION 2 succeeded.

Kubernetes cluster context:

Cluster name:    ccp92-cluster-3
API server URL:   https://192.168.92.183:6443

Editable Prerequisites
Enter the path to the 'kubectl' configuration file. The configuration information contained in this file will specify the cluster on which you are about to perform the deployment.

Path to the KUBECONFIG file [<F1>]: /home/ubuntu/.kube/ccp92-cluster-3.KUBECONFIG

Prerequisite Check Result

<table>
<thead>
<tr>
<th>Name</th>
<th>Current Value</th>
<th>Result</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>LINUX_X64</td>
<td>+ (passed)</td>
<td></td>
</tr>
<tr>
<td>Shell</td>
<td>/bin/bash</td>
<td>+ (passed)</td>
<td></td>
</tr>
<tr>
<td>KUBECONFIG</td>
<td>/home/ubuntu/.kube/ccp92-cluster-3.KUBECONFIG</td>
<td>+ (passed)</td>
<td></td>
</tr>
<tr>
<td>Helm Version</td>
<td>Client: v2.15.2, Server: v2.15.2</td>
<td>+ (passed)</td>
<td></td>
</tr>
<tr>
<td>Kubernetes Client Version</td>
<td>v1.14.8</td>
<td>+ (passed)</td>
<td></td>
</tr>
<tr>
<td>Python Version</td>
<td>/usr/bin/python2.7</td>
<td>+ (passed)</td>
<td></td>
</tr>
<tr>
<td>PyYAML Check</td>
<td></td>
<td>+ (passed)</td>
<td></td>
</tr>
</tbody>
</table>

Choose Retry to retry the Prerequisite Check.

Choose Back to go back to Product Information Dialog.

Choose Next to continue.

Choose action Retry/Back/Next [r/b/n/<F1>]: n

************************
* Kubernetes Namespace *
************************

Specify the Kubernetes namespace in which the actions will be taken.

- The namespace cannot be formed by only digits.
- The namespace must consist of one or more hyphen-separated groups. Each group must contain only lower case letters or only numbers. In the case of only one group, it may contain a mix of lower case letters and digits.
- It must follow the regular expression: '^[([a-z])|--([0-9])+--([a-z])+([0-9])*)+$\|^([a-z])|--([0-9])+--([a-z])+([0-9])*)$' 
- Examples of a valid namespace: valid-namespace, valid-2-namespace, 2-3-117
- Examples of an invalid namespace: 00, 0le-example, invalid-name2, invalid-namespace2-example

Kubernetes Namespace [<F1>]: sapdh

**************************
* License Agreement *
************************

By running the Software Lifecycle Container Bridge for deploying SAP Data Hub and by using built-in operators of SAP Data Hub, Docker images will be built by automatically downloading and installing: (a) Docker images from SAP Docker Registry, (b) Docker images from third party registries, and (c) open source prerequisites from third party open source repositories.

The Docker images from the SAP Docker Registry are part of the SAP Data Hub product. Use of these images is governed by the terms of your commercial agreement with SAP for the SAP Data Hub.

The Docker images from third party registries and open source prerequisites from third party open source repositories (collectively, the “Third Party Prerequisites”) are prerequisites of SAP Data Hub that usually would have to be downloaded and installed by customers from third party repositories before deploying the SAP Data Hub. For the customers' convenience, the Software Lifecycle Container Bridge and built-in operators automatically download and install the Third Party Prerequisites on behalf of the customer. The Third Party Prerequisites are NOT part of the SAP Data Hub and SAP does not accept any responsibility for the Third Party Prerequisites, including providing support. Use of the Third Party Prerequisites is solely at customers’ risk and subject to any third party licenses applicable to the use of such prerequisites. Customers are responsible for keeping the Third Party Prerequisites up-to-date, and are asked to make use of the respective community support and / or to consider commercial support offerings.

The Third Party Prerequisites and associated license information are listed in the Release Note for SAP Data Hub that is published at the download site for SAP Data Hub.

By clicking "I authorize", you authorize the download and installation of Docker images from the SAP Docker Registry and Third Party Prerequisites from third party repositories, and acknowledge the foregoing disclaimer.

I authorize: n

possible values [y/n] [<F1>]: y

************************
* Installation Type *
************************

Choose one of the installation types.

- Basic Installation: You are only prompted for a small selection of installation parameters. For the other installation parameters, default values are used.
- Advanced Installation (recommended): You are prompted for all parameters. In case of specific installation requirements, this installation option is recommended.

1. Basic Installation
   > 2. Advanced Installation

possible values [1,2] [<F1>]: 2
* Use Container Images *

Choose if you want to use saved container images.

- In case of installation without internet connection, you need to save the container images in pre-installation host which has internet access and you need to transfer the saved container images into the installation host. This option allows you to specify the path of the folder which contains saved container images.

> 1. Do not use
   2. Use

possible values [1,2] [F1]: 

* Enter Logon Information *

You require S-User credentials to log on to repositories.sap.ondemand.com

User Name [F1]: S0019247791
Password [F1]:

* Choose a Technical User *

Choose an existing Technical User or create a new Technical User to access repositories.sap.ondemand.com.

> 1. 0000394598-yydpbdh
   2. Create new Technical User

possible values [1,2] [F1]: 

* Container Registry *

Specify the container registry to push the SAP Data Hub images. This container registry will be used by Kubernetes and by SAP Data Hub Modeler. The container registry must be accessible from the installation host including the necessary authentication.
- Examples: 012345678910.dkr.ecr.us-east-1.amazonaws.com, eu.gcr.io/my-project-name, myregistry.azurecr.io, myhost:5000

Container Registry [<F1>]: 192.168.92.102/sapdh

* Image Pull Secret *

Choose if you want to use an image pull secret for "192.168.92.102/sapdh".

- It is necessary when the container registry needs authentication and there is no authentication mechanism in the cluster to access the container registry.
- In some cloud environments, authentication is managed with IAM roles or cloud specific service accounts. In these cases, you don't need to use image pull secrets.

  > 1. Do not use an image pull secret
  2. Use an image pull secret
  possible values [1,2] [<F1>]: 1

* Certificate Domain *

Specify the SAN (Subject Alternative Name) for the certificate, which must match the fully qualified domain name (FQDN) of the Kubernetes node to be accessed externally. By using this certificate domain, SAP Data Hub generates a self-signed certificate for TLS and JWT.

- The length of the certificate domain must be less than 64.
- The certificate domain must consist of lower case letters, upper case letters, digits or the following special characters * . -
- The certificate domain may start with a * followed by a dot or an alphanumerical character.
- * character can only be found at the beginning.
- The certificate domain cannot end with a dot.
- Examples of a valid certificate domain: my-domain5465.com, *.my-domain.com
- Examples of an invalid certificate domain: my-domain.*.com, *4.my.domain.com, *.*.my-domain.com, my_domain.com

Certificate Domain [<F1>]: sapdh.example.com
Specify a password for the "system" user of "system" tenant.

- It must contain between 8 and 255 characters.
- It must contain at least one lower case, one upper case, one numerical and one special character.
- The allowed special characters are . @ # $ % * + _ ? !.
- It cannot contain spaces.

Password [<F1>]: xxxxxxxx
Confirm: xxxxxxxx

**********************************************
* SAP Data Hub Initial Tenant Name *
**********************************************

Specify a name for the SAP Data Hub initial tenant that is going to be created automatically.

- It must contain between 4 and 60 characters.
- It must consist of lower case letters, digits or hyphens.
- It must not begin or end with hyphens and must not contain multiple consecutive hyphens.
- It must follow the regular expression: '^[a-z0-9]+(-[a-z0-9]+)*$'

Tenant Name [<F1>]: default

**********************************************
* SAP Data Hub Initial Tenant Administrator Username *
**********************************************

Specify a name for administrator user of "default" tenant.

- It must contain between 4 and 60 characters.
- It must consist of lower case letters, digits or hyphens.
- It must not begin or end with hyphens and must not contain multiple consecutive hyphens.
- It must follow the regular expression: '^[a-z0-9]+(-[a-z0-9]+)*$'

Username [<F1>]: admin

********************************************************************
* SAP Data Hub Initial Tenant Administrator Password Configuration *
********************************************************************
Specify if you want to use the same "system" user password for "admin" user of "default" tenant.

1. Use the same password
   > 2. Do not use the same password
possible values [1,2] [<F1>]: 1

******************************
* Cluster Proxy Settings *
******************************

Choose if you want to configure proxy settings. It is necessary when the Kubernetes cluster is running behind a proxy.

1. Configure
   > 2. Do not configure
possible values [1,2] [<F1>]: 2

******************************
* Checkpoint Store Configuration *
******************************

Choose if you want to use SAP Data Hub streaming tables and enable the checkpoint store.

> 1. Do not enable checkpoint store
   2. Enable checkpoint store
possible values [1,2] [<F1>]: 2

******************************
* Checkpoint Store Type *
******************************

Specify the checkpoint store type.

> 1. Amazon S3
   2. Windows Azure Storage Blob (WASB)
   3. Google Cloud Storage (GCS)
   4. WebHDFS
   5. Alibaba OSS
possible values [1,2...5] [<F1>]: 1
************************
* Amazon S3 Access Key *
************************

Specify the Amazon S3 Access Key.
Amazon S3 Access Key [<F1>]: xxxxxxxx

******************************
* Amazon S3 Secret Access Key *
******************************

Specify the Amazon S3 Secret Access Key.
Amazon S3 Secret Access Key [<F1>]: xxxxxxxx

************************
* Amazon S3 Host *
************************

Specify the Amazon S3 Host.
Amazon S3 Host (optional) [<F1>]:

************************
* Amazon S3 Region *
************************

Specify the Amazon S3 Region to be connected.
Amazon S3 Region (optional) [<F1>]:

************************
* Amazon S3 Host *
************************

Specify the Amazon S3 Host.
Amazon S3 Host (optional) [<F1>]: http://192.168.92.197:9000

************************
* Amazon S3 Region *
************************

Specify the Amazon S3 Region to be connected.
Amazon S3 Region (optional) [<F1>]:

************************
* Amazon S3 Path    *
************************

Specify Amazon S3 bucket and directory (in the form my-bucket/directory).
Amazon S3 bucket and directory [<F1>]: sapdh

************************
*       Timeout        *
************************

Specify the timeout in seconds for checkpoint store.
Timeout [<F1>]: 180

******************************
* Checkpoint Store Validation *
******************************

Choose if you want to validate Checkpoint Store.

> 1. Do not validate Checkpoint Store
   2. Validate Checkpoint Store
possible values [1,2] [<F1>]: 2

******************************
* Storage Class Configuration *
******************************

Choose if you want to configure StorageClasses for ReadWriteOnce PersistentVolumes.

- SAP Data Hub needs some ReadWriteOnce PersistentVolumes. During installation and runtime, some PersistentVolumeClaims are created. SAP Data Hub assumes there is at least one dynamic volume provisioner on the cluster and the dynamic volume provisioners are going to provision PersistentVolumes.
- A StorageClass provides a way for administrators to describe the “classes” of storage they offer. Different classes might map to quality-of-service levels, or to backup policies, or to arbitrary policies determined by the cluster administrators.
- SAP Data Hub doesn't set StorageClasses of PersistentVolumeClaims by default. This settings enable you to set the StorageClasses.

> 1. Do not configure storage classes
2. Configure storage classes
possible values [1,2] [<F1>]: 1

*******************************************
* Docker Container Log Path Configuration *
*******************************************

Choose whether the configuration of your kubernetes cluster requires a custom docker container log path configuration. This option is only required if the directory /var/lib/docker/containers resides on different mount volumes of the physical cluster nodes than the root directory (e.g. /mnt/docker/containers) which may be the case for on-premise installations. In this case the installation of SAP Data Hub Diagnostics with the default docker log path setting will fail. You do not need to modify the docker container log path on standard cloud environments (including SAP Cloud Platform, Amazon Web Services, Google Cloud Platform, and Microsoft Azure).

> 1. Do not configure container log path
2. Configure container log path
possible values [1,2] [<F1>]: 1

**************************************
******************
* Container Registry Settings for SAP Data Hub Modeler *
********************************************************

Choose if you want to use a different container registry from "192.168.92.102/sapdh" for SAP Data Hub Modeler.

1. Use a different registry
> 2. Use default registry
possible values [1,2] [<F1>]: 1

*******************************************************
* Container Registry for SAP Data Hub Modeler *
*******************************************************

Container registry for SAP Data Hub Modeler.
Container Registry [<F1>]: 192.168.92.102/sapdhm

****************************************************************
* Image Pull Secret Settings for SAP Data Hub Modeler Registry *
****************************************************************

Choose if you want to use an image pull secret for "192.168.92.102/sapdhm".
- It is necessary when the container registry needs authentication and there is no authentication mechanism in the cluster to access the container registry.
- In some cloud environments, authentication is managed with IAM roles or cloud specific service accounts. In these cases, you don't need to use image pull secrets.

> 1. Do not use an image pull secret
   2. Use an image pull secret

possible values [1,2] [<F1>]: 1

*****************************************************************************
* Loading NFS Modules          *
*****************************************************************************

Choose if you want to enable loading the kernel modules (nfsd and nfsv4) on all Kubernetes nodes. These modules are necessary for System Management. You can disable if you are certain that these modules (nfsd and nfsv4) are already loaded on all Kubernetes nodes.

> 1. Enable loading NFS modules
   2. Disable loading NFS modules

possible values [1,2] [<F1>]: 1

*****************************************************************************
* Enable Network Policies      *
*****************************************************************************

Choose if you want to enable Network Policies.

1. Enable network policies
   > 2. Disable network policies

possible values [1,2] [<F1>]: 2

*****************************************************************************
*   Helm Timeout               *
*****************************************************************************

Specify the timeout in seconds for helm deployments. The default duration is enough for most of the environments. Increasing the value is necessary when the network or volume provisioner is so slow that deployments of components fail because of timeouts without any other issues.

Timeout in seconds [<F1>]: 1200
***************
* Pod Wait Timeout *
***************

Specify the timeout in seconds for waiting a pod to be ready. The default duration is enough for most of the environments. Increasing the value is necessary when the network or volume provisioner is so slow that a pod may not be ready in this amount of time.

Timeout in seconds [F1]: 300

***************
* Additional Installation Parameters *
***************

You can specify additional installation parameters. The parameters are documented in the section "Configuration Parameters for Kubernetes Deployment" in the official SAP Data Hub documentation. Use -e flag for each additional parameter that you give and use spaces between them.

- Example: -e vora-dqp.components.disk.replicas=3 -e vora-dqp.components.dlog.storageSize=100Gi

Additional Installation Parameters [F1]: e vora-dqp.components.disk.replicas=3 -e vora-dqp.components.dlog.storageSize=100Gi

***************
* Parameter Summary *
***************

Choose 'Next' to start the deployment with the displayed parameter values or choose 'Back' to revise the parameters.

KUBECONFIG
  Path to the KUBECONFIG file: /home/ubuntu/.kube/ccp92-cluster-3.KUBECONFIG

Kubernetes Namespace
  Kubernetes Namespace: sapdh

License Agreement
  I authorize: y

Installation Type
  1. Basic Installation
> 2. Advanced Installation

Use Container Images
> 1. Do not use
2. Use

Container Repository Username
Username: 0000394598-yydphbdh

Container Registry
Container Registry: 192.168.92.102/sapdh

Image Pull Secret
> 1. Do not use an image pull secret
2. Use an image pull secret

Certificate Domain
Certificate Domain: sapdh.example.com

SAP Data Hub System Tenant Administrator Password

SAP Data Hub Initial Tenant Name
Tenant Name: default

SAP Data Hub Initial Tenant Administrator Username
Username: admin

SAP Data Hub Initial Tenant Administrator Password Configuration
> 1. Use the same password
2. Do not use the same password

Cluster Proxy Settings
1. Configure
> 2. Do not configure

Checkpoint Store Configuration
1. Do not enable checkpoint store
> 2. Enable checkpoint store

Checkpoint Store Type
> 1. Amazon S3
2. Windows Azure Storage Blob (WASB)
3. Google Cloud Storage (GCS)
4. WebHDFS
5. Alibaba OSS

Amazon S3 Access Key

Amazon S3 Secret Access Key

Amazon S3 Host
    Amazon S3 Host (optional): http://192.168.92.197:9000

Amazon S3 Region
    Amazon S3 Region (optional): 

Amazon S3 Path
    Amazon S3 bucket and directory: sapdh

Timeout
    Timeout: 180

Checkpoint Store Validation
    1. Do not validate Checkpoint Store
    > 2. Validate Checkpoint Store

Storage Class Configuration
    > 1. Do not configure storage classes
    > 2. Configure storage classes

Docker Container Log Path Configuration
    > 1. Do not configure container log path
    > 2. Configure container log path

Container Registry Settings for SAP Data Hub Modeler
    > 1. Use a different registry
    > 2. Use default registry

Container Registry for SAP Data Hub Modeler
    Container Registry: 192.168.92.102/sapdhm

Image Pull Secret Settings for SAP Data Hub Modeler Registry
    > 1. Do not use an image pull secret
    > 2. Use an image pull secret
Loading NFS Modules
   > 1. Enable loading NFS modules
      2. Disable loading NFS modules

Enable Network Policies
   1. Enable network policies
   > 2. Disable network policies

Helm Timeout
   Timeout in seconds: 1200

Pod Wait Timeout
   Timeout in seconds: 300

Additional Installation Parameters
   Additional Installation Parameters: -e vora-dqp.components.disk.replicas=3 -e vora-dqp.components.dlog.storageSize=100Gi

Choose 'Next' to start the deployment with the displayed parameter values or choose 'Back' to revise the parameters.

Choose action Back/Next [b/n/<F1>]: n
2020-03-04T09:45:33-0800 [INFO] Running in SLPlugin mode

SAP Data Hub will be installed on the cluster.

**Expose SAP Data Hub services using Kubernetes service LoadBalancer**

Run the following commands to expose the SAP Data Hub services to an IP address for access.

```
ubuntu@Ubuntu~:~/git/sapdhinstall$ kubectl expose service -n sapdh vsystem --type=LoadBalancer --name=my-vsyste-loadbalancer
service/my-ysystem-loadbalancer exposed

ubuntu@Ubuntu~:~/git/sapdhinstall$ kubectl expose service -n sapdh vora-tx-coordinator-ext --type=LoadBalancer --name=my-vora-tx-coordinator-ext
service/my-vora-tx-coordinator-ext exposed

ubuntu@Ubuntu~:~/git/sapdhinstall$ kubectl expose service -n sapdh vora-textanalysis --type=LoadBalancer --name=my-vora-textanalysis
service/my-vora-textanalysis exposed

ubuntu@Ubuntu~:~/git/sapdhinstall$ kubectl -n sapdh get svc/my-vora-textanalysis svc/my-vsyste-loadbalancer svc/my-vora-tx-coordinator-ext
```
Log in to the SAP Data Hub vsystem web interface

From the IP address assigned as the my-vsyste-system-loadbalancer address, open a browser and connect using the HTTPS version of that address. Log in with the provided tenant, admin user name, and password. The SAP Data Hub Launchpad Applications page will be displayed.

Reconcile the Kubernetes persistent volume claims in the NetApp storage

Run the following command to list the PersistentVolumeClaim (PVC) claims created by the NetApp CSI storage plug-in with the values in NetApp ONTAP System Manager.

```
kubectl get pvc -n sapdh -l app=vora
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLUSTER-IP</th>
<th>EXTERNAL-IP</th>
<th>PORT(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>my-vora-textanalysis</td>
<td>LoadBalancer</td>
<td>10.106.80.104</td>
<td>192.168.92.195</td>
<td>10002:32721/TCP</td>
</tr>
<tr>
<td>my-vsyste-system-loadbalancer</td>
<td>LoadBalancer</td>
<td>10.107.242.90</td>
<td>192.168.92.192</td>
<td></td>
</tr>
<tr>
<td>8797:32152/TCP,8125:30438/TCP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>my-vora-tx-coordinator-ext</td>
<td>LoadBalancer</td>
<td>10.99.75.240</td>
<td>192.168.92.194</td>
<td></td>
</tr>
<tr>
<td>10004:31599/TCP,30115:32009/TCP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
data-vora-disk-2 Bound pvc-64ca0788-5e41-11ea-b399-005056927098 50Gi RWO basic
data-vora-dlog-0 Bound pvc-f8ee184b-5e40-11ea-b399-005056927098 100Gi RWO basic
data-dir-vora-consul-0 Bound pvc-3ad25138-5e40-11ea-b399-005056927098 2Gi RWO basic
data-dir-vora-consul-1 Bound pvc-3ad67db8-5e40-11ea-b399-005056927098 2Gi RWO basic
data-dir-vora-consul-2 Bound pvc-3ad98b76-5e40-11ea-b399-005056927098 2Gi RWO basic
trace-hana-0 Bound pvc-3a88a8db-5e40-11ea-b399-005056927098 10Gi RWO basic

```bash
table
<table>
<thead>
<tr>
<th>NAME</th>
<th>STATUS</th>
<th>VOLUME</th>
<th>CAPACITY</th>
<th>ACCESS MODES</th>
<th>STORAGECLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>storage-diagnostics-elasticsearch-0</td>
<td>Bound</td>
<td>pvc-198fdd3f-5e42-11ea-b399-005056927098</td>
<td>40Gi</td>
<td>RWO</td>
<td>basic</td>
</tr>
<tr>
<td>storage-diagnostics-prometheus-server-0</td>
<td>Bound</td>
<td>pvc-19b36c76-5e42-11ea-b399-005056927098</td>
<td>10Gi</td>
<td>RWO</td>
<td>basic</td>
</tr>
</tbody>
</table>
```

```
Conclusion

FlexPod infrastructure is an excellent platform for deploying SAP Data Hub as an all-in-one data orchestration solution with Cisco Container Platform and the NetApp Trident CSI plug-in to discover, refine, enhance, and manage any type, variety, and volume of data across your entire distributed data landscape.

FlexPod Datacenter is the optimal shared infrastructure foundation for deploying SAP Data Hub to allow high-performance access to applications that need it. FlexPod is well suited as the platform of choice, providing the scalability and reliability needed to support these capabilities.

With FlexPod, Cisco and NetApp have created a platform that is both flexible and scalable for multiple use cases and applications. FlexPod adds yet another feature to help organizations efficiently and effectively support business-critical applications running simultaneously from the same shared infrastructure. The
flexibility and scalability of FlexPod also enables customers to start with a right-sized infrastructure that can grow with and adapt to their evolving business requirements.

This validation effort confirms SAP Data Hub as an all-in-one data orchestration solution well suited to run on FlexPod.

For more information

Consult the following references for additional information about the topics discussed in this document.

**Products and solutions**


**Interoperability matrixes**

Configuration guides