Deployment Steps

HCI

NetApp
June 26, 2020
Table of Contents

Deployment Steps ........................................................................................................... 1

Workflow Summary: NetApp HCI with Anthos ................................................................. 1
1. Configure Management Switches: NetApp HCI with Anthos ........................................ 1
2. Configure Data Switches: NetApp HCI with Anthos .................................................... 5
3. Deploy NetApp HCI with the NetApp Deployment Engine: NetApp HCI with Anthos ...... 11
4. Configure the vCenter Server: NetApp HCI with Anthos ............................................. 21
5. Deploy and Configure the F5 Big-IP Virtual Edition Appliance: NetApp HCI with Anthos .... 25
6. Complete Anthos Prerequisites: NetApp HCI with Anthos ........................................... 29
7. Deploy the Anthos Admin Workstation: NetApp HCI with Anthos ................................. 35
8. Deploy the Admin and the First User Cluster: NetApp HCI with Anthos ....................... 44
9. Deploy Additional User Clusters: NetApp HCI with Anthos ......................................... 56
10. Enable Access to the Cluster with the GKE Console: NetApp HCI with Anthos ............. 60
11. Install and Configure NetApp Trident Storage Provisioner: NetApp HCI with Anthos .... 63
Deployment Steps

Workflow Summary: NetApp HCI with Anthos

This section provides detailed protocols for implementing the NetApp HCI solution for Anthos.

This deployment is divided into the following high-level tasks:
1. Configure management switches
2. Configure data switches
3. Deploy NetApp HCI with the NetApp Deployment Engine
4. Configure the vCenter Server
5. Deploy and configure the F5 Big-IP Virtual Edition appliance
6. Complete Anthos prerequisites
7. Deploy the Anthos admin workstation
8. Deploy the admin and the first user cluster
9. Deploy additional user clusters
10. Enable access to the cluster with the GKE console
11. Install and configure NetApp Trident storage provisioner

1. Configure Management Switches: NetApp HCI with Anthos

Cisco Nexus 3048 switches are used in this deployment procedure to provide 1Gbps connectivity for in and out-of-band management of the compute and storage nodes. These steps begin after the switches have been racked, powered, and put through the initial setup process. To configure the switches to provide management connectivity to the infrastructure, complete the following steps:

Enable Advanced Features for Cisco Nexus

Run the following commands on each Cisco Nexus 3048 switch to configure advanced features:

1. Enter configuration mode.

   Switch-01# configure terminal

2. Enable VLAN functionality.
3. Enable LACP.

Switch-01(config)# feature lacp

4. Enable virtual port channels (vPCs).

Switch-01(config)# feature vpc

5. Set the global port-channel load-balancing configuration.

Switch-01(config)# port-channel load-balance src-dst ip-l4port

6. Perform the global spanning-tree configuration.

Switch-01(config)# spanning-tree port type network default
Switch-01(config)# spanning-tree port type edge bpdu-guard default

**Configure Ports on the Switch for In-Band Management**

1. Run the following commands to create VLANs for management purposes.

Switch-01(config)# vlan 2
Switch-01(config-vlan)# Name Native_VLAN
Switch-01(config-vlan)# vlan 16
Switch-01(config-vlan)# Name OOB_Network
Switch-01(config-vlan)# vlan 3480
Switch-01(config-vlan)# Name MGMT_Network
Switch-01(config-vlan)# exit

2. Configure the ports ETH1/29-32 as VLAN trunk ports that connect to management interfaces on each HCI storage node.
Configure Ports on the Switch for Out-of-Band Management

1. Run the following commands to configure the ports for cabling the IPMI interfaces on each HCI node.
Switch-01(config)# int eth 1/13
Switch-01(config-if)# description HCI-CMP-01 IPMI
Switch-01(config-if)# switchport mode access
Switch-01(config-if)# switchport access vlan 16
Switch-01(config-if)# spanning-tree port type edge
Switch-01(config-if)# int eth 1/14
Switch-01(config-if)# description HCI-STG-01 IPMI
Switch-01(config-if)# switchport mode access
Switch-01(config-if)# switchport access vlan 16
Switch-01(config-if)# spanning-tree port type edge
Switch-01(config-if)# int eth 1/15
Switch-01(config-if)# description HCI-STG-03 IPMI
Switch-01(config-if)# switchport mode access
Switch-01(config-if)# switchport access vlan 16
Switch-01(config-if)# spanning-tree port type edge
Switch-01(config-if)# exit

In the validated configuration, we cabled odd-node IPMI interfaces to Switch-01, and even-node IPMI interfaces to Switch-02.

Create a vPC Domain to Ensure Fault Tolerance

1. Activate the ports used for the vPC peer-link between the two switches.

Switch-01(config)# int eth 1/1
Switch-01(config-if)# description vPC peer-link Switch-02 1/1
Switch-01(config-if)# int eth 1/2
Switch-01(config-if)# description vPC peer-link Switch-02 1/2
Switch-01(config-if)# exit

2. Perform the vPC global configuration.
**2. Configure Data Switches: NetApp HCI with Anthos**

Mellanox SN2010 switches provide 25Gbps connectivity for the data plane of the compute and storage nodes.

To configure the switches to provide data connectivity to the infrastructure, complete the following steps:

**Create MLAG Cluster to Ensure Fault Tolerance**

1. Run the following commands on each Mellanox SN210 switch for general configuration:
   1. Enter configuration mode.

   ```
   Switch-01 enable
   Switch-01 configure terminal
   ```

   2. Enable the LACP required for the Inter-Peer Link (IPL).

   ```
   Switch-01 (config) # lacp
   ```

   3. Enable the Link Layer Discovery Protocol (LLDP).

   ```
   Switch-01(config-if)# description vPC peer-link
   Switch-01(config-if)# switchport mode trunk
   Switch-01(config-if)# switchport trunk native vlan 2
   Switch-01(config-if)# switchport trunk allowed vlan 16,3480
   Switch-01(config-if)# spanning-tree port type network
   Switch-01(config-if)# vpc peer-link
   Switch-01(config-if)# exit
   ```
Switch-01 (config) # lldp

4. Enable IP routing.

Switch-01 (config) # ip routing

5. Enable the MLAG protocol

Switch-01 (config) # protocol mlag

6. Enable global QoS.

Switch-01 (config) # dcb priority-flow-control enable force

2. For MLAG to function, the switches must be made peers to each other through an IPL. This should consist of two or more physical links for redundancy. The MTU for the IPL is set for jumbo frames (9216), and all VLANs are enabled by default. Run the following commands on each switch in the domain:

1. Create port channel 10 for the IPL.

Switch-01 (config) # interface port-channel 10
Switch-01 (config interface port-channel 10) # description IPL
Switch-01 (config interface port-channel 10) # exit

2. Add interfaces ETH 1/20 and 1/22 to the port channel.

Switch-01 (config) # interface ethernet 1/20 channel-group 10 mode active
Switch-01 (config) # interface ethernet 1/20 description ISL-SWB_01
Switch-01 (config) # interface ethernet 1/22 channel-group 10 mode active
Switch-01 (config) # interface ethernet 1/22 description ISL-SWB_02

3. Create a VLAN outside of the standard range dedicated to IPL traffic.

Switch-01 (config) # vlan 4000
Switch-01 (config vlan 4000) # name IPL VLAN
Switch-01 (config vlan 4000) # exit

4. Define the port channel as the IPL.
5. Set an IP for each IPL member (non-routable; it is not advertised outside of the switch).

```
Switch-01 (config) # interface vlan 4000
Switch-01 (config vlan 4000) # ip address 10.0.0.1 255.255.255.0
Switch-01 (config vlan 4000) # ipl 1 peer-address 10.0.0.2
Switch-01 (config vlan 4000) # exit
```

3. Create a unique MLAG domain name for the two switches and assign a MLAG virtual IP (VIP). This IP is used for keep-alive heartbeat messages between the two switches. Run these commands on each switch in the domain:

1. Create the MLAG domain and set the IP address and subnet.

```
Switch-01 (config) # mlag-vip MLAG-VIP-DOM ip a.b.c.d /24 force
```

2. Create a virtual MAC address for the system MLAG.

```
Switch-01 (config) # mlag system-mac AA:BB:CC:DD:EE:FF
```

3. Configure the MLAG domain so that it is active globally.

```
Switch-01 (config) # no mlag shutdown
```

- The IP used for the MLAG VIP must be in the same subnet as the switch management network (mgmt0).
- The MAC address used can be any unicast MAC address and must be set to the same value on both switches in the MLAG domain.

### Configure Ports to Connect to Storage and Compute Hosts

1. Create each of the VLANs needed to support the services for NetApp HCI. Run these commands on each switch in the domain:

   1. Create VLANs.

2. Create names for each VLAN for easier accounting.

Switch-01 (config) # vlan 1172 name VM_Network
Switch-01 (config) # vlan 3480 name MGMT_Network
Switch-01 (config) # vlan 3481 name Storage_Network
Switch-01 (config) # vlan 3482 name vMotion_Network

2. Create Hybrid VLAN ports on ports ETH1/9-10 so that you can tag the appropriate VLANs for the NetApp HCI compute nodes.

1. Select the ports you want to work with.

Switch-01 (config) # interface ethernet 1/9-1/10

2. Set the MTU for each port.

Switch-01 (config interface ethernet 1/9-1/10) # mtu 9216 force

3. Modify spanning-tree settings for each port.

Switch-01 (config interface ethernet 1/9-1/10) # spanning-tree bpduguard enable
Switch-01 (config interface ethernet 1/9-1/10) # spanning-tree port type edge
Switch-01 (config interface ethernet 1/9-1/10) # spanning-tree bpduguard enable

4. Set the switchport mode to hybrid.

Switch-01 (config interface ethernet 1/9-1/10) # switchport mode hybrid
Switch-01 (config interface ethernet 1/9-1/10) # exit

5. Create descriptions for each port being modified.

Switch-01 (config) # interface ethernet 1/9 description HCI-CMP-01 PortD
Switch-01 (config) # interface ethernet 1/10 description HCI-CMP-02 PortD
6. Tag the appropriate VLANs for the NetApp HCI environment.

```
Switch-01 (config) # interface ethernet 1/9 switchport hybrid allowed-vlan add 1172
Switch-01 (config) # interface ethernet 1/9 switchport hybrid allowed-vlan add 3480-3482
Switch-01 (config) # interface ethernet 1/10 switchport hybrid allowed-vlan add 1172
Switch-01 (config) # interface ethernet 1/10 switchport hybrid allowed-vlan add 3480-3482
```

3. Create MLAG interfaces and hybrid VLAN ports on ports ETH1/5-8 so that you can distribute connectivity between the switches and tag the appropriate VLANs for the NetApp HCI storage nodes.

1. Select the ports that you want to work with.

```
Switch-01 (config) # interface ethernet 1/5-1/8
```

2. Set the MTU for each port.

```
Switch-01 (config interface ethernet 1/5-1/8) # mtu 9216 force
```

3. Modify spanning tree settings for each port.

```
Switch-01 (config interface ethernet 1/5-1/8) # spanning-tree bpdufilter enable
Switch-01 (config interface ethernet 1/5-1/8) # spanning-tree port type edge
Switch-01 (config interface ethernet 1/5-1/8) # spanning-tree bpduguard enable
```

4. Set the switchport mode to hybrid.

```
Switch-01 (config interface ethernet 1/5-1/8 ) # switchport mode hybrid
Switch-01 (config interface ethernet 1/5-1/8 ) # exit
```

5. Create descriptions for each port being modified.

```
Switch-01 (config) # interface ethernet 1/5 description HCI-STG-01 PortD
Switch-01 (config) # interface ethernet 1/6 description HCI-STG-02 PortD
Switch-01 (config) # interface ethernet 1/7 description HCI-STG-03 PortD
Switch-01 (config) # interface ethernet 1/8 description HCI-STG-04 PortD
```

6. Create and configure the MLAG port channels.
Switch-01 (config) # interface mlag-port-channel 115-118
Switch-01 (config interface mlag-port-channel 115-118) # exit
Switch-01 (config) # interface mlag-port-channel 115-118 no shutdown
Switch-01 (config) # interface mlag-port-channel 115-118 mtu 9216 force
Switch-01 (config) # interface mlag-port-channel 115-118 lacp-individual enable
Switch-01 (config) # interface ethernet 1/5-1/8 lacp port-priority 10
Switch-01 (config) # interface ethernet 1/5-1/8 lacp rate fast
Switch-01 (config) # interface ethernet 1/5 mlag-channel-group 115 mode active
Switch-01 (config) # interface ethernet 1/6 mlag-channel-group 116 mode active
Switch-01 (config) # interface ethernet 1/7 mlag-channel-group 117 mode active
Switch-01 (config) # interface ethernet 1/8 mlag-channel-group 118 mode active

7. Tag the appropriate VLANs for the storage environment.

Switch-01 (config) # interface mlag-port-channel 115-118 switchport mode hybrid
Switch-01 (config) # interface mlag-port-channel 115 switchport hybrid allowed-vlan add 1172
Switch-01 (config) # interface mlag-port-channel 116 switchport hybrid allowed-vlan add 1172
Switch-01 (config) # interface mlag-port-channel 117 switchport hybrid allowed-vlan add 1172
Switch-01 (config) # interface mlag-port-channel 118 switchport hybrid allowed-vlan add 1172
Switch-01 (config) # interface mlag-port-channel 115 switchport hybrid allowed-vlan add 3481
Switch-01 (config) # interface mlag-port-channel 116 switchport hybrid allowed-vlan add 3481
Switch-01 (config) # interface mlag-port-channel 117 switchport hybrid allowed-vlan add 3481
Switch-01 (config) # interface mlag-port-channel 118 switchport hybrid allowed-vlan add 3481

The configurations in this section must also be run on the second switch in the MLAG domain. NetApp recommends that the descriptions for each port are updated to reflect the device ports that are being cabled and configured on the other switch.

Create Uplink Ports for the Switches

1. Create an MLAG interface to provide uplinks to both Mellanox SN2010 switches from the core network.
2. Configure the MLAG members.

3. Set the switchport mode to hybrid and allow all VLANs from the core uplink switches.

4. Verify that the MLAG interface is up.

3. Deploy NetApp HCI with the NetApp Deployment Engine: NetApp HCI with Anthos

NDE delivers a simple and streamlined deployment experience for the NetApp HCI solution.

These steps begin after the nodes have been racked, and cabled, and the IPMI port has been configured on each node using the console.

A detailed guide to using NDE 1.6 to deploy your NetApp HCI system can be found [here](#).

To Deploy the NetApp HCI solution using NDE, complete the following steps:

1. Access the out-of-band management console for one of the storage nodes in the cluster and log in with the default credentials ADMIN/ADMIN.
2. Click the Remote Console Preview image in the center of the screen to download a JNLP file launched by Java Web Start, which launches an interactive console to the system.

3. With the virtual console launched, a user can log in to the HCI Storage node, using the ADMIN/ADMIN username and password combination.

4. The Bond1G interface must have an IP, a netmask, and a gateway set statically; its VLAN set to 3480; and DNS servers defined for the environment.
Select an IP that is within the subnet you intend to use for in-band management, but not an IP you would like to use in production. NDE reconfigures the node with a production IP after initial access.

This task must only be performed on the first storage node. Afterward, the other nodes in the infrastructure are discovered by the Automatic Private IP Address (APIPA) addresses assigned to each storage interface when left unconfigured.

5. The Bond 10G interface must have its MTU setting changed to enable jumbo frames and its bond mode changed to LACP.
Configure each of the four storage nodes in the NetApp HCI solution this way. The NDE process is then able to discover all the nodes in the solution and configure them. You do not need to modify the Bond10g interfaces on the two compute nodes.

6. After completion, open a web browser and visit the IP address you configured for the management port to start the NetApp HCI configuration with NDE.

7. On the Welcome to NetApp HCI page, click the Get Started button.

8. Check each associated box on the Prerequisites page and click Continue.

9. The next page presents End User Licenses for NetApp HCI and VMware vSphere. If you accept the terms, click I Accept at the end of each agreement and then click Continue.

10. Click Configure a New vSphere Deployment, select vSphere 6.5U2, and enter the Fully Qualified Domain Name (FQDN) of your vCenter Server. Then click Continue.
vSphere Configuration

You may elect to configure a new vSphere deployment or to join an existing vSphere deployment.

- Configure a new vSphere deployment
- Configure Using vSphere Version 6.7 Update 1
- Configure Using vSphere Version 6.5 Update 2
- Join and extend an existing vSphere deployment

If you have set up a DNS record for your new vCenter server, then configure your server using its fully qualified domain name and DNS server IP address:

- Configure Using a Fully Qualified Domain Name

**vCenter Server Fully Qualified Domain Name**

anthos-vc.cie.netapp.com

**Note:** The domain name must resolve to an unused IP address.

**DNS Server IP Address**

10.61.184.251

If you have not set up a DNS record for your new vCenter server, you may configure using an IP address that we define:

- Configure Using an IP Address

**Note:** Once defined, the IP address cannot be changed.

11. NDE asks for the credentials to be used in the environment. This is used for VMware vSphere, the NetApp Element storage cluster, and the NetApp Mnode, which provides management functionality for the cluster. When you are finished, click Continue.
12. NDE then prompts for the network topology used to cable the NetApp HCI environment. The validated solution in this document has been deployed using the 2 Cable Option for the compute nodes, and the 4 Cable Option for the storage nodes. Click Continue.
13. The next page presented by NDE is the inventory of the environment as discovered by the APIPA addressed on the storage network. The storage node that is currently running NDE is already selected with a green check mark. Select the corresponding boxes to add additional nodes to the NetApp HCI environment. Click Continue.
If there are any nodes missing from the inventory screen, wait a few minutes and click Refresh Inventory. If the node still fails to appear, additional investigation of environment networking may be required.

14. You must next configure the permanent network settings for the NetApp HCI deployment. The first page configures infrastructure services (DNS and NTP), vCenter networking, and Mnode networking.
15. The next page allows you to configure each node in the environment. For the compute nodes, it allows you to configure the host name, management network, vMotion network, and storage network. For the storage nodes, name the storage cluster and configure the management and storage networks being used for each node. Click Continue.
16. On the next page, review all the settings that have been defined for the environment by expanding each section, and, if necessary, click Edit to make corrections. There is also a check box on this page that enables or disables the Mnode from sending real-time health and diagnostics information to NetApp Active IQ®. If all the information is correct, click Start Deployment.

If you want to enable Active IQ, verify that your management network can reach the internet. If NDE is unable to reach Active IQ, the deployment can fail.

17. A summary page appears along with a progress bar for each component of the NetApp HCI solution, as well as the overall solution. When complete, you are presented with an option to launch the vSphere client and begin working with your environment.
4. Configure the vCenter Server: NetApp HCI with Anthos

NDE deploys the solution with vCenter server and integrates the solution with the Element cluster by provisioning the Mnode VM and installing the NetApp Element Plug-in for vCenter.

After deployment, you must make a few modifications to the environment, including the creation of additional vDS portgroups, datastores, and resource groups for the deployment of the Anthos on VMware solution.

Complete the following steps to configure your vCenter Server for use:

1. Log into the VMware vCenter server using the Administrator@vsphere.local account and the password chosen for the admin user during NDE configuration.
2. Right-click `NetApp-HCI-Cluster-01` created by NDE and select the option to create a new resource pool. Name this pool `Infrastructure-Resource-Pool` and accept the defaults by clicking OK. This resource pool is used in a later configuration step.
The reservations in this resource pool can be modified based on the resources available in the environment. NetApp HCI is deployed as an all-in-one solution. Therefore, NetApp recommends reserving the resources necessary to provide availability for the infrastructure services by placing them into this resource pool and adjusting the resources appropriately. Infrastructure services include vCenter Server, NetApp Mnode, and F5 Big-IP Load Balancer.
3. Repeat this step to create another resource pool for VMs deployed by Anthos. Name this pool Anthos-Resource-Pool and click the OK button to accept the default values. Adjust the resource availability based on the specific environment in which you are deploying the solution. This resource pool is used in a later deployment step.

4. To configure Element volumes to be used as vSphere datastores, click the dropdown menu and select NetApp Element Management from the list.

5. A Getting Started screen appears with details about your Element cluster.

![NetApp Element Management](image)

6. Click Management, and the vSphere client presents a list of datastores. Click Create Datastore to create one datastore to host VMs and another to host ISOs for future guest installs.

7. Next click the Network menu item in the left panel. This displays a screen with information about the vDS deployed by NDE.

8. Several virtual port groups are defined by the initial configuration. NetApp recommends leaving these alone to support the infrastructure, and additional port groups should be created for user-deployed virtual guests. Right-click the NetApp HCI VDS 01 vDS in the left panel, and then select Distributed Port Group followed by the New Distributed Port Group option from the expanded menu.

9. Create a new distributed port group called `Management_Network`. Then click Next.

10. On the next screen, select the VLAN type as VLAN, and set the VLAN ID to 3480 for management purposes. Click Next, and, after reviewing the options on the summary page, click Next again to complete the creation of the distributed port group.

11. Repeat these steps to create distributed port groups for the `VM_Network` (VLAN 1172), as well as any other networks that might be used in the NetApp HCI environment.
Additional networks can be defined to segment any additional deployed VMs. Examples of this use could be for a dedicated HA network for additional F5 Big-IP appliances if provisioned. Such configurations are in addition to the environment deployed in this validated solution and are considered out of scope for this NVA document.

5. Deploy and Configure the F5 Big-IP Virtual Edition Appliance: NetApp HCI with Anthos

Anthos enables native integration with F5 Big-IP load balancers to expose services from each pod to the world.

This solution makes use of the virtual appliance deployed in VMware vSphere as deployed by NDE. Networking for the F5 Big-IP virtual appliance can be configured in a two-armed or three-armed configuration based on your network environment. The deployment in this document is based on the two-armed configuration. Additional details on configuring the virtual appliance for use with Anthos can be found here.

To deploy the F5 Big-IP Virtual Edition appliance, complete the following steps:

1. Download the virtual application Open Virtual Appliance (OVA) file from F5 here.

   To download the appliance, a user must register with F5. They provide a 30-day demo license for the Big-IP Virtual Edition Load Balancer. NetApp recommends a permanent 10Gbps license for the production deployment of an appliance.

2. Right-click the Infrastructure Resource Pool and select Deploy OVF Template. A wizard launches that allows you to select the OVA file that you just downloaded in Step 1. Click Next.
3. Click Next to continue through each step and accept the default values for each screen presented until you reach the storage selection screen. Select the VM_Datastore that was created earlier, and then click Next.

4. The next screen presented by the wizard allows you to customize the virtual networks for use in the environment. Select VM_Network for the External field and select Management_Network for the Management field. Internal and HA are used for advanced configurations for the F5 Big-IP appliance and are not configured. These parameters can be left alone, or they can be configured to connect to non-infrastructure, distributed port groups. Click Next.

5. Review the summary screen for the appliance, and, if all the information is correct, click Finish to start the deployment.

6. After the virtual appliance is deployed, right-click it and power it up. It should receive a DHCP address on the management network. The appliance is Linux-based, and it has VMware Tools deployed, so you can view the DHCP address it receives in the vSphere client.

7. Open a web browser and connect to the appliance at the IP address from the previous step. The
default login is admin/admin, and, after the first login, the appliance immediately prompts you to change the admin password. It then returns you to a screen where you must log in with the new credentials.

8. The first screen prompts the user to complete the Setup Utility. Begin the utility by clicking Next.

9. The next screen prompts for activation of the license for the appliance. Click Activate to begin. When prompted on the next page, paste either the 30-day evaluation license key you received when you registered for the download or the permanent license you acquired when you purchased the appliance. Click Next.

For the device to perform activation, the network defined on the management interface must be able to reach the internet.

10. On the next screen, the End User License Agreement (EULA) is presented. If the terms in the license are acceptable, click Accept.

11. The next screen counts the elapsed time as it verifies the configuration changes that have been made so far. Click Continue to resume with the initial configuration.

12. The Configuration Change window closes, and the Setup Utility displays the Resource Provisioning
menu. This window lists the features that are currently licensed and the current resource allocations for the virtual appliance and each running service.

13. Clicking the Platform menu option on the left enables additional modification of the platform. Modifications include setting the management IP address configured with DHCP, setting the host name and the time zone the appliance is installed in, and securing the appliance from SSH accessibility.

14. Next click the Network menu, which enables you to configure standard networking features. Click Next to begin the Standard Network Configuration wizard.

15. The first page of the wizard configures redundancy; leave the defaults and click Next. The next page enables you to configure an internal interface on the load balancer. Interface 1.1 maps to the vmnic labeled Internal in the OVF deployment wizard.

**[Big-IP Configuration] | big-IP_config_8.png**

The spaces in this page for Self IP Address, Netmask, and Floating IP address can be filled with a non-routable IP for use as a placeholder. They can also be filled with an internal network that has been configured as a distributed port group for virtual guests if you are deploying the three-armed configuration. They must be completed to continue with the wizard.

16. The next page enables you to configure an external network that is used to map services to the pods deployed in Kubernetes. Select a static IP from the VM_Network range, the appropriate subnet mask, and a floating IP from that same range. Interface 1.2 maps to the vmnic labeled External in the OVF deployment wizard.

**[Big-IP Configuration] | big-IP_config_9.png**

17. On the next page, you can configure an internal-HA network if you are deploying multiple virtual appliances in the environment. To proceed, you must fill the Self-IP Address and the Netmask fields, and you must select interface 1.3 as the VLAN Interface, which maps to the HA network defined by the OVF template wizard.

18. The next page enables you to configure the NTP servers. Then click Next to continue to the DNS setup. The DNS servers and domain search list should already be populated by the DHCP server. Click Next to accept the defaults and continue.

19. For the remainder of the wizard, click Next to continue through the advanced peering setup, the configuration of which is beyond the scope of this document. Then click Finish to exit the wizard.

20. Create individual partitions for the Anthos admin cluster and each user cluster deployed in the environment. Click System in the menu on the left, navigate to Users, and click Partition List.
21. The displayed screen only shows the current common partition. Click Create on the right to create the first additional partition, and name it Anthos-Admin. Then click Repeat, and name the partition Anthos-Cluster1, and click the Repeat button again to name the next partition Anthos-Cluster2. Finally click Finished to complete the wizard. The Partition list screen returns with all the partitions now listed.

6. **Complete Anthos Prerequisites: NetApp HCI with Anthos**

Now that the physical environment is set up, you can begin Anthos deployment. This starts with several prerequisites that you must meet to deploy the solution and access it afterward. Each of these steps are discussed in depth in the Anthos GKE On-Prem Guide.

To prepare your environment for the deployment of Anthos on VMware, complete the following steps:

1. Create a Google Cloud project following the instructions available here.

   Your organization might already have a project in place intended for this purpose. Check with your cloud administration team to see if a project exists and is already configured for access to Anthos on VMware. All projects intended for use with Anthos must be whitelisted by Google. This includes the primary user account, additional team members, and the access service account created in a later step.

2. Create a deployment workstation from which to manage the installation of Anthos on VMware. The deployment workstation can be Linux, MacOS, or Windows. For the purposes of this validated deployment, Red Hat Enterprise Linux 7 was used.

   This workstation can be hosted either internal or external to the NetApp HCI deployment. The only requirement is that it must be able to successfully communicate with the deployed VMware vCenter Server and the internet to function correctly.

3. Install Google Cloud SDK for interactions with Google Cloud. It can be downloaded as an archive of binaries for manual install or installed by either the apt-get (Ubuntu/Debian) or yum (RHEL) package managers.
[user@rhel7 ~]$ sudo yum install google-cloud-sdk
Failed to set locale, defaulting to C
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
Resolving Dependencies
--> Running transaction check
--> Package google-cloud-sdk.noarch 0:270.0.0-1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=============================================================================================================

Package       Arch Version             Repository
Size
=============================================================================================================

Installing:
google-cloud-sdk noarch 270.0.0-1 google-cloud-sdk
36 M

Transaction Summary

=============================================================================================================

Install 1 Package

Total download size: 36 M
Installed size: 174 M
Is this ok [y/d/N]: y
Downloading packages:
6d81c821884ae40244c746f6044fc1bcd801143a0d9c8da06767036b8d090a24-google-cloud-sdk-270.0.0-1.noarch | 36 MB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing: google-cloud-sdk-270.0.0-1.noarch
1/1
  Verifying: google-cloud-sdk-270.0.0-1.noarch
1/1

Installed:
google-cloud-sdk.noarch 0:270.0.0-1

Complete!
The gcloud binary must be at least version 265.0.0. You can update a manual install with a `gcloud components update`. However, if SDK was installed by a package manager, future updates must also be performed using that same package manager.

4. Install `govc`, the CLI for VMware vSphere. Installing `govc` allows you to interact directly with the management of VMware vCenter. Govc is available as a pre-packaged binary available in a gzip format for download. For installation, a user must download the gzip archive, unzip, and copy the resulting binary to a local path directory such as `/usr/local/bin`.

```
[user@rhel7 ~]$ wget https://github.com/vmware/govmomi/releases/download/v0.20.0/govc_linux_amd64.gz
--2019-11-06 09:06:10--
https://github.com/vmware/govmomi/releases/download/v0.20.0/govc_linux_amd64.gz
Resolving github.com (github.com)... 13.250.177.223
Connecting to github.com (github.com)|13.250.177.223|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7650188 (7.3M) [application/octet-stream]
Saving to: 'govc_linux_amd64.gz.1'

100%[================================================================================================================================] 7,650,188  --.K/s in 0.1s
2019-11-06 09:06:12 (76.4 MB/s) - 'govc_linux_amd64.gz' saved [7650188/7650188]
```

```
[user@rhel7 ~]$ gunzip govc_linux_amd64.gz
[user@rhel7 ~]$ sudo cp govc_linux_amd64 /usr/local/bin/govc
[user@rhel7 ~]$ which govc
/usr/local/bin/govc
```

5. Install **Hashicorp Terraform**. Terraform enables the automated deployment of VMs in a VMware vSphere environment and is used to deploy the Anthos admin workstation in a later step. Like the `govc` install, Terraform can be downloaded as a zip archive and unzipped, and the resulting binary can be copied to a directory in the local user’s path.
6. With the workstation configured, log in to Google Cloud with your credentials. To do so, enter the login command from the deployment workstation and retrieve a link that can be copied and pasted into a browser to allow interactive sign-in to Google services. After you have logged in, the web page presents a code that you can copy and paste back into the deployment workstation when prompted.
7. Before you can install Anthos on VMware, you must create four service accounts, each with a specific purpose in interacting with Google Cloud. The following table lists the accounts and their purposes.

<table>
<thead>
<tr>
<th>Account Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>access-service-account</td>
<td>Used to download the Anthos binaries from Cloud Storage.</td>
</tr>
<tr>
<td>register-service-account</td>
<td>Used to register Anthos clusters to the Google Cloud console.</td>
</tr>
<tr>
<td>connect-service-account</td>
<td>Used to maintain the connection between user clusters and the Google Cloud.</td>
</tr>
<tr>
<td>stackdriver-service-account</td>
<td>Used to write logging and monitoring data to Stackdriver.</td>
</tr>
</tbody>
</table>

Each account is assigned an email address that references your approved Google Cloud project name. The examples below all list the project Anthos-Dev which was used during the NetApp validation. Make sure to substitute your appropriate project name in syntax examples where necessary.
8. Enable several APIs so that your environment can communicate with Google Cloud. The pods deployed in your clusters must be able to access https://www.googleapis.com and https://gkeconnect.googleapis.com to function as expected. Therefore, the VM_Network that the worker nodes are attached to must have internet access. To enable the necessary APIs, run the following command from the deployment workstation:

```
[user@rhel7 ~]$ gcloud services enable \
    cloudresourcemanager.googleapis.com \
    container.googleapis.com \
    gkeconnect.googleapis.com \
    gkehub.googleapis.com \
    serviceusage.googleapis.com \
    stackdriver.googleapis.com \
    monitoring.googleapis.com \
    logging.googleapis.com
```

9. The final step needed to prepare your environment to deploy Anthos is to limit certain privileges to your service accounts. You need the associated email address for each service account listed in Step 7.

1. Using the register service account, assign the roles for `gkehub.admin` and `serviceusage.serviceUsageViewer`.
2. Using the connect service account, assign the roles for `gkehub.connect`.

3. With the stackdriver service account, assign the roles for `stackdriver.resourceMetadata.writer`, `logging.logWriter`, and `monitoring.metricWriter`.

7. Deploy the Anthos Admin Workstation: NetApp HCI with Anthos

The admin workstation is a vSphere VM deployed within your NetApp HCI environment that is preinstalled with all the tools necessary to administer the Anthos on VMware solution. Follow the instructions in this section to download,
deploy, and configure the Anthos admin workstation.

The admin workstation image is packaged as an OVA file and is only available for download to those users with whitelisted access service accounts. If you are unable to download the OVA from cloud storage, contact your project administrator or open a support ticket with Google Cloud support.

To deploy the Anthos admin workstation, complete the following steps:

1. Download the latest version of the virtual appliance here. You must first set your account to the whitelisted access service account that has permission to download the OVA.

   ```
   [user@rhel7 ~]$ gcloud config set account 'access-service-account@anthos-dev.iam.gserviceaccount.com'
   Updated property [core/account].
   [user@rhel7 ~]$ gsutil cp gs://gke-on-prem-release/admin-appliance/1.1.1-gke.2/gke-on-prem-admin-appliance-vsphere-1.1.1-gke.2.{ova,ova.sig} ~/
   Copying gs://gke-on-prem-release/admin-appliance/1.1.1-gke.2/gke-on-prem-admin-appliance-vsphere-1.1.1-gke.2.ova...
   ==> NOTE: You are downloading one or more large file(s), which would run significantly faster if you enabled sliced object downloads. This feature is enabled by default but requires that compiled crcmod be installed (see "gsutil help crcmod").
   - [0 files][ 40.2 MiB/  5.1 GiB]
   ```

2. When the OVA file is downloaded, use govc to load it into your vCenter. You need to set several environment variables for your current user that provide specific information about your VMware vSphere environment deployed on NetApp HCI.

   ```
   [user@rhel7 ~]$ export GOVC_URL=https://anthos-vc.cie.netapp.com/sdk
   [user@rhel7 ~]$ export GOVC_USERNAME=administrator@vsphere.local
   [user@rhel7 ~]$ export GOVC_PASSWORD=vcpassword
   [user@rhel7 ~]$ export GOVC_DATASTORE=VM_Datastore
   [user@rhel7 ~]$ export GOVC_DATACENTER=NetApp-HCI-Datacenter-01
   [user@rhel7 ~]$ export GOVC_RESOURCE_POOL= Anthos-Resource-Pool
   [user@rhel7 ~]$ export GOVC_INSECURE=true
   ```

3. Because the NetApp HCI deployment uses a vDS, you need to create a custom options file that you pass as an additional argument to govc. Create a file called `options_specs`, and commit the following information to the file:
4. To import the OVA file as template in VMware vSphere, enter the following command referencing the downloaded virtual appliance file, as well as the options file that you just created. The console reports upload progress, and you can also verify the upload by browsing to the Recent Tasks window in VMware vSphere.

```
[user@rhel7 ~]$ govc import.ova -options options_spec .
gke-on-prem-admin-appliance-vsphere-1.1.1-gke.2.ova
[06-11-19 10:56:39] Uploading gke-on-prem-admin-appliance-vsphere-1.1.1-gke.2.vmdk... (23%, 26.8MiB/s)
```

5. With the template uploaded, we can now use Terraform to deploy the Anthos admin workstation VM. Google provides a Terraform template and variable files for environments using DHCP as well those using static IP addresses. Create a directory for these Terraform files, and copy and paste the text for both the TF and TFVARS files available here into files created with a text editor.

```
[user@rhel7 ~]$ mkdir terraform
[user@rhel7 ~]$ cd terraform
[user@rhel7 ~]$ vi terraform.tfvars
```

6. See the following `terraform.tfvars` file contents.
# vCenter Server username
vcenter_user = "administrator@vsphere.local"

# vCenter Server password
vcenter_password = "vcpassword"

# vCenter Server IP or hostname
vcenter_server = "https://anthos-vc.cie.netapp.com"

# Path in which the admin workstation's VM's public key should be saved
ssh_public_key_path = "~/.ssh/vsphere_workstation.pub"

# Hostname for the VM
vm_name = "admin-workstation"

# vSphere datastore to use for storage
datastore = "VM_Datastore"

# vSphere datacenter in which to create the VM
datacenter = "NetApp-HCI-Datacenter-01"

# vSphere cluster in which to create the VM
cluster = "NetApp-HCI-Cluster-01"

# vSphere resource pool in which to create VM, if you are using a non-default resource pool
# If you are using the default resource pool, provide a value like "CLUSTER-NAME/Resources"
resource_pool = "Anthos-Resource-Pool"

# vSphere network to use for the VM
network = "VM_Network"

# Number of CPUs for this VM. Recommended minimum 4.
num_cpus = 4

# Memory in MB for this VM. Recommended minimum 8192.
memory = 8192

# The VM template (OVA) to clone. Change the version if you imported a different version of the OVA.
vm_template = "gke-on-prem-admin-appliance-vsphere-1.1.1-gke.2"

[user@rhel7 ~]$ vi terraform.tf
7. See the following `terraform.tf` file contents.

```
#########################
####### VARIABLES #######
#########################

# The following variables are declared in the accompanying TFVARS file

# vCenter Server username
variable "vcenter_user" {} 

# vCenter Server password
variable "vcenter_password" {} 

# vCenter Server address
variable "vcenter_server" {} 

# Path in which the VM's public key should be saved
variable "ssh_public_key_path" { default = "~/.ssh/vsphere_workstation.pub" } 

# vSphere network to use for the VM
variable "network" { default = "VM Network"}

# Hostname for the VM
variable "vm_name" { default = "vsphere-workstation" } 

# vSphere datacenter in which to create the admin workstation VM
variable "datacenter" {} 

# vSphere datastore to use for storage
variable "datastore" {} 

# vSphere cluster in which to create the VM
variable "cluster" {} 

# vSphere resource pool in which to create the VM
variable "resource_pool" {} 

# Number of CPUs for this VM. Recommended minimum 4.
variable "num_cpus" { default = 4 }

# Memory in MB for this VM. Recommended minimum 8192.
variable "memory" { default = 8192 }

# The VM template (OVA) to clone
variable "vm_template" {} 
```
### vsphere Data ###

```hcl
provider "vsphere" {
  version = "~> 1.5"
  user     = "${var.vcenter_user}"
  password = "${var.vcenter_password}"
  vcenter_server = "${var.vcenter_server}"

  # if you have a self-signed cert
  allow_unverified_ssl = true
}
```

```hcl
data "vsphere_datastore" "datastore" {
  name     = "${var.datastore}"
  datacenter_id = "${data.vsphere_datacenter.dc.id}"
}
```

```hcl
data "vsphere_datacenter" "dc" {
  name = "${var.datacenter}"
}
```

```hcl
data "vsphere_compute_cluster" "cluster" {
  name     = "${var.cluster}"
  datacenter_id = "${data.vsphere_datacenter.dc.id}"
}
```

```hcl
data "vsphere_resource_pool" "pool" {
  name     = "${var.resource_pool}"
  datacenter_id = "${data.vsphere_datacenter.dc.id}"
}
```

```hcl
data "vsphere_network" "network" {
  name     = "${var.network}"
  datacenter_id = "${data.vsphere_datacenter.dc.id}"
}
```

```hcl
data "vsphere_virtual_machine" "template_from_ovf" {
  name     = "${var.vm_template}"
  datacenter_id = "${data.vsphere_datacenter.dc.id}"
}
```

```hcl
data "template_file" "dhcp_ip_config" {
  template = "EOF
  network:
EOF"
```
version: 2
ethernets:
  ens192:
    dhcp4: true
EOF

data "template_file" "user_data" {
  template = <<EOF
#cloud-config
apt:
  primary:
    - arches: [default]
      uri: http://us-west1.gce.archive.ubuntu.com/ubuntu/
write_files:
  - path: /etc/netplan/99-dhcp.yaml
    permissions: '0644'
    encoding: base64
    content: |
      $$
      ${dhcp_ip_config}
runcmd:
  - netplan apply
  - /var/lib/gke/guest-startup.sh
EOF
  vars = {
    dhcp_ip_config = "${base64encode(data.template_file.dhcp_ip_config.rendered)}"
  }
}

### vSphere Resources ###

resource "vsphere_virtual_machine" "vm" {
  name = "${var.vm_name}"
  resource_pool_id = "${data.vsphere_resource_pool.pool.id}"
  datastore_id = "${data.vsphere_datastore.datastore.id}"
  num_cpus = "${var.num_cpus}"
  memory = "${var.memory}"
  guest_id = "${data.vsphere_virtual_machine.template_from_ovf.guest_id}"
  enable_disk_uuid = "true"
  scsi_type = "${data.vsphere_virtual_machine.template_from_ovf.scsi_type}"
  network_interface {
    network_id = "${data.vsphere_network.network.id}"
    adapter_type = "${data.vsphere_virtual_machine.template_from_ovf.network_interface_types[0]}"
  }
  wait_for_guest_net_timeout = 15
nested_hv_enabled = false
cpu_performance_counters_enabled = false

disk {
  label     = "disk0"
  size      = "${max(50, data.vsphere_virtual_machine.template_from_ovf.disks.0.size)}"
  eagerly_scrub = "${data.vsphere_virtual_machine.template_from_ovf.disks.0.eagerly_scrub}"
  thin_provisioned = "${data.vsphere_virtual_machine.template_from_ovf.disks.0.thin_provisioned}"
}

cdrom {
  client_device = true
}

vapp {
  properties = {
    hostname   = "${var.vm_name}"
    public-keys = "${file(var.ssh_public_key_path)}"
    user-data  = "${base64encode(data.template_file.user_data.rendered)}"
  }
}

clone {
  template_uuid = "${data.vsphere_virtual_machine.template_from_ovf.id}"
}

output "ip_address" {
  value = "${vsphere_virtual_machine.vm.default_ip_address}"
}

Values specific to the deployed environment have been added to the terraform.tfvars file. However, you should not modify the terraform.tf file in any manner.

8. Create an SSH public/private keypair used to log in to the admin workstation after it is deployed. Name the public key so that it matches the variable that was assigned in the terraform.tfvars file.
[user@rhel7 ~]$ ssh-keygen -t rsa -f ~/.ssh/vsphere_workstation -N ""
Generating public/private rsa key pair.
Your identification has been saved in /home/user/.ssh/vsphere_workstation2.
Your public key has been saved in /home/user/.ssh/vsphere_workstation2.pub.
The key fingerprint is:
SHA256:qEk8G13LhwBKqf85ekHHzkIzuX2MkZUXGNEHvFT2vw user@rhel7
The key’s randomart image is:

9. Navigate to the directory created to host the TF and TFVARS files. Within this directory, initialize Terraform and use it to launch the deployment of the admin workstation VM.
8. Deploy the Admin and the First User Cluster: NetApp HCI with Anthos

All Kubernetes clusters deployed as a part of the Anthos solution are deployed from the Anthos admin workstation that you just created. A user logs into the
admin workstation using SSH, the public key created in a previous step, and the IP address provided at the end of the VM deployment. They can then begin creating their first clusters.

There are specific procedures for deploying clusters that use static IP addresses [here](#), and procedures for environments with DHCP can be found [here](#). In this guide, we use the second set of instructions.

To deploy the initial admin and first user cluster, complete the following steps:

1. Login to the admin workstation using the SSH public key and the IP address output at the end of Terraform deployment.

```bash
[user@rhel7 ~]$ ssh -i ~/.ssh/vsphere_workstation ubuntu@10.63.172.21
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-62-generic x86_64)
* Documentation:  https://help.ubuntu.com
* Management:     https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

36 packages can be updated.
23 updates are security updates.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@admin-workstation:$
```

2. Log in with the `gcloud auth` command as you did before from your deployment workstation by copying the URL into a web browser, signing into your Google account, and pasting the verification code back into the workstation.
3. Set the project that you intend to deploy your clusters in for Anthos on VMware, because the project is not set by default when you log in.

```
ubuntu@admin-workstation:~$ gcloud config set project anthos-dev
Updated property [core/project].
```

4. Register gcloud as a Docker credential helper, which enables it to manage the credentials for Docker registries used for deployment. This way, the default credential store is not used for operations involving the credentials of the specified registries.
By default, Anthos on VMware uses a pre-existing, Google-owned container image registry that requires no additional setup. If you choose to use a private Docker registry for deployment, then you must configure that registry separately based on instructions found here. This step is beyond the scope of this deployment guide.

5. In the next step to deploy an admin cluster, create a private key file in the JSON format for each of the service accounts created in the prerequisites section.
<table>
<thead>
<tr>
<th>NAME</th>
<th>EMAIL</th>
<th>DISABLED</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:stackdriver-service-account@anthos-dev.iam.gserviceaccount.com">stackdriver-service-account@anthos-dev.iam.gserviceaccount.com</a></td>
<td>False</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:register-service-account@anthos-dev.iam.gserviceaccount.com">register-service-account@anthos-dev.iam.gserviceaccount.com</a></td>
<td>False</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:access-service-account@anthos-dev.iam.gserviceaccount.com">access-service-account@anthos-dev.iam.gserviceaccount.com</a></td>
<td>False</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:connect-service-account@anthos-dev.iam.gserviceaccount.com">connect-service-account@anthos-dev.iam.gserviceaccount.com</a></td>
<td>False</td>
<td></td>
</tr>
</tbody>
</table>

6. For the next step, you must use the access service key created in the previous step to activate the associated service account.

```
ubuntu@admin-workstation:~$ gcloud auth activate-service-account --key-file=access-key.json
Activated service account credentials for: [access-service-account@anthos-dev.iam.gserviceaccount.com]
```

7. The deployment of the first clusters is performed using inputs from a config file generated by GKE. A generic config file can be created with no additional input, or an existing config file can be referenced to create additional clusters.

```
ubuntu@admin-workstation:~$ gcloud iam service-accounts list
NAME                                      EMAIL
DISABLED
stackdriver-service-account@anthos-dev.iam.gserviceaccount.com
False
register-service-account@anthos-dev.iam.gserviceaccount.com
False
access-service-account@anthos-dev.iam.gserviceaccount.com
False
connect-service-account@anthos-dev.iam.gserviceaccount.com
False
```

```
ubuntu@admin-workstation:~$ gcloud iam service-accounts keys create access-key.json
--iam-account access-service-account@anthos-dev.iam.gserviceaccount.com
created key [8d5f8ce039dd98766e18a3c5ee6794912fb8d095] of type [json] as [access-key.json] for [access-service-account@anthos-dev.iam.gserviceaccount.com]
```

```
ubuntu@admin-workstation:~$ gcloud iam service-accounts keys create register-key.json
--iam-account register-service-account@anthos-dev.iam.gserviceaccount.com
created key [f08b494c665321f83bcb8c8526ba21185b456a11] of type [json] as [register-key.json] for [register-service-account@anthos-dev.iam.gserviceaccount.com]
```

```
ubuntu@admin-workstation:~$ gcloud iam service-accounts keys create connect-key.json
--iam-account connect-service-account@anthos-dev.iam.gserviceaccount.com
created key [c964021ff6157d3df2a15db49f5c85b7b1495c2] of type [json] as [connect-key.json] for [connect-service-account@anthos-dev.iam.gserviceaccount.com]
```

```
ubuntu@admin-workstation:~$ gcloud iam service-accounts keys create stackdriver-key.json
--iam-account stackdriver-service-account@anthos-dev.iam.gserviceaccount.com
created key [3c9427dfdef161d139ff998be896565c1df0b122] of type [json] as [stackdriver-key.json] for [stackdriver-service-account@anthos-dev.iam.gserviceaccount.com]
```
8. The `config.yaml` file created by running the previous command has several variables that must be customized for the current environment.

   1. First, you must determine the full path location and the name of the current GKE bundle that is deployed into the environment. The file exists in the `/var/lib/gke/bundles` directory on the admin workstation.

```
ubuntu@admin-workstation:~$ ls /var/lib/gke/bundles
  gke-onprem-vsphere-1.1.1-gke.2-full.tgz  gke-onprem-vsphere-1.1.1-gke.2.tgz
```

   2. Next, you must get the fully recognized host name or IP address of our vCenter Server as displayed in its default SSL certificate. Connect to vSphere and dump the certificate contents into a file called `vcenter.pem`. Examining this file gives you the information that you need for the value of `Subject: CN` (common name).

```
ubuntu@admin-workstation:~$ true | openssl s_client -connect anthos-vc.cie.netapp.com:443 -showcerts 2>/dev/null | sed -ne '/-BEGIN/,/-END/p' > vcenter.pem
ubuntu@admin-workstation:~$ openssl x509 -in vcenter.pem -text -noout | grep Subject:
   Subject: CN = anthos-vc.cie.netapp.com, C = US
```

   If the value added to the `config.yaml` file does not match that of the CN found in the certificate, communication with the vCenter server fails.

9. With the information from those two commands and the generated `vcenter.pem` file, we can now edit the `config.yaml` file to prepare for deployment. Editing this file is very similar to the edits that you performed to the `terraform.tfvars` file to provide specifics regarding the VMware vCenter instance deployed in NetApp HCI.

   When deploying the cluster, determine which IP addresses to use for the control plane and ingress VIPs for both the admin and user cluster. Also determine the compute and memory resources that must be reserved for each node deployed, because it is not possible to edit a cluster after it has been deployed.

```
# Absolute path to a GKE bundle on disk
bundlepath: "'/var/lib/gke/bundles/gke-onprem-vsphere-1.1.1-gke.2-full.tgz"
```
# Specify which vCenter resources to use for deployment
vcenter:
  # The credentials and address GKE should use to connect to vCenter
credentials:
    address: "anthos-vc.cie.netapp.com"
    username: "administrator@vsphere.local"
    password: "vcpass"
datacenter: "NetApp-HCI-Datacenter-01"
datastore: "VM_Datastore"
cluster: "NetApp-HCI-Cluster-01"
network: "VM_Network"
resourcepool: "Anthos Resource Pool"
  # Provide the name for the persistent disk to be used by the deployment (ending
  # in .vmdk). Any directory in the supplied path must be created before deployment.
  # Not required when adding additional user clusters
datadisk: "anthos-admin-data-disk.vmdk"
  # Provide the path to vCenter CA certificate pub key for SSL verification
cacertpath: "/home/ubuntu/vcenter.pem"
# Specify the proxy configuration.
proxy:
  # The URL of the proxy
  url: ""
  # The domains and IP addresses excluded from proxying
  noproxy: ""
# Specify admin cluster settings for a fresh GKE On-Prem deployment. Omit this section
# and use the --adminconfig flag when adding a new user cluster to an existing deployment
admincluster:
  # In-Cluster vCenter configuration
  vcenter:
    # If specified it overwrites the network field in global vcenter configuration
    network: ""
    # The absolute or relative path to the yaml file to use for static IP allocation.
    ipblockfilepath: ""
    # Do not include if using DHCP
    # manuallbspec:
    #   ingresshttpnodeport: 32527
    #   ingresshttpsnodeport: 30139
    #   controlplanenodeport: 30968
    #   addonsnodeport: 31405
    # Specify pre-defined nodeports if using "manual" load balancer mode
    # manuallbspec:
    #   ingresshttpnodeport: 32527
    #   ingresshttpsnodeport: 30139
    #   controlplanenodeport: 30968
    #   addonsnodeport: 31405
    # Specify the already-existing partition and credentials to use with F5
  bigip:
    # To re-use credentials across clusters we recommend using YAML node anchors.
    # See https://yaml.org/spec/1.2/spec.html#id2785586
    credentials:
      address: "172.21.224.22"
      username: "admin"
password: "lbpass"

partition: "Anthos-Admin-Part"

# # Optionally specify a pool name if using SNAT
# snatpoolname: ""

# The VIPs to use for load balancing
vips:
  # Used to connect to the Kubernetes API
  controlplanevip: "10.63.172.98"
  # Shared by all services for ingress traffic
  ingressvip: "10.63.172.99"
  # # Used for admin cluster addons (needed for multi cluster features). Must be the same
  # # across clusters
  # addonsvip: ""
  # The Kubernetes service CIDR range for the cluster. Must not overlap with the pod CIDR range
  serviceiprange: 10.96.232.0/24
  # The Kubernetes pod CIDR range for the cluster. Must not overlap with the service CIDR range
  podiprange: 192.168.0.0/16

# Specify settings when deploying a new user cluster. Used both with a fresh deployment
# or when adding a new cluster to an existing deployment.
usercluster:
  antiaffinitygroups:
    enabled: false
  # In-Cluster vCenter configuration
  vcenter:
    # If specified it overwrites the network field in global vcenter configuration
    network: ""
    # # The absolute or relative path to the yaml file to use for static IP allocation.
    # # Do not include if using DHCP
    # ipblockfilepath: ""
    # # Specify pre-defined nodeports if using "manual" load balancer mode
    # manuallbspec:
    #   ingresshttpnodeport: 30243
    #   ingresshttpsnodport: 30879
    #   controlplanenodeport: 30562
    #   addonsnodeport: 0
    # Specify the already-existing partition and credentials to use with F5
  bigip:
    # To re-use credentials across clusters we recommend using YAML node anchors.
    # See https://yaml.org/spec/1.2/spec.html#id2785586
    credentials:
      address: "172.21.224.22"
      username: "admin"
      password: "lbpass"
      partition: "Anthos-Cluster01-Part"
## Optionally specify a pool name if using SNAT
# snatpoolname: ""

## The VIPs to use for load balancing
vips:
  # Used to connect to the Kubernetes API
  controlplanevip: "10.63.172.105"
  # Shared by all services for ingress traffic
  ingressvip: "10.63.172.106"
  # # Used for admin cluster addons (needed for multi cluster features). Must be the same
  # across clusters
  # addonsvip: ""

## A unique name for this cluster
clustername: "anthos-cluster01"

## User cluster master nodes must have either 1 or 3 replicas
masternode:
  cpus: 4
  memorymb: 8192
  # How many machines of this type to deploy
  replicas: 1

## The number of worker nodes to deploy and their size. Min. 2 replicas
workernode:
  cpus: 4
  memorymb: 8192
  # How many machines of this type to deploy
  replicas: 3

## The Kubernetes service CIDR range for the cluster
serviceiprange: 10.96.0.0/12

## The Kubernetes pod CIDR range for the cluster
podiprange: 192.168.0.0/16

## Uncomment this section to use OIDC authentication
# oidc:
  # issuerurl: ""
  # kubectlredirecturl: ""
  # clientid: ""
  # clientsecret: ""
  # username: ""
  # usernameprefix: ""
  # group: ""
  # groupprefix: ""
  # scopes: ""
  # extraparams: ""
  # # Set value to string "true" or "false"
  # usehttpproxy: ""
  # # # The absolute or relative path to the CA file (optional)
  # capath: ""
  # # Optionally provide an additional serving certificate for the API server
  # sni:
# certpath: ""
# keypath: ""
# Which load balancer mode to use "Manual" or "Integrated"
lbmode: Integrated
# Specify which GCP project to connect your GKE clusters to
gkeconnect:
  projectid: "anthos-dev"
# The absolute or relative path to the key file for a GCP service account used to
# register the cluster
  registerserviceaccountkeypath: "/home/ubuntu/register-key.json"
# The absolute or relative path to the key file for a GCP service account used by
# the GKE connect agent
  agentserviceaccountkeypath: "/home/ubuntu/connect-key.json"
# Specify which GCP project to connect your logs and metrics to
stackdriver:
  projectid: "anthos-dev"
# A GCP region where you would like to store logs and metrics for this cluster.
  clusterlocation: "us-east1"
  enablevpc: false
# The absolute or relative path to the key file for a GCP service account used to
# send logs and metrics from the cluster
  serviceaccountkeypath: "/home/ubuntu/stackdriver-key.json"
# # Optionally use a private Docker registry to host GKE images
# privateregistryconfig:
# # Do not include the scheme with your registry address
# credentials:
#   address: ""
#   username: ""
#   password: ""
# # The absolute or relative path to the CA certificate for this registry
#   cacertpath: ""
# The absolute or relative path to the GCP service account key that will be used to
# pull GKE images
  gcrkeypath: "/home/ubuntu/access-key.json"
# Configure kubernetes apiserver audit logging
cloudauditlogging:
  projectid: ""
# A GCP region where you would like to store audit logs for this cluster.
  clusterlocation: ""
# The absolute or relative path to the key file for a GCP service account used to
# send audit logs from the cluster
  serviceaccountkeypath: ""

10. Because spacing in YAML files can be very important, you can check the syntax of the config file by running the following command. If the command outputs any failures, be sure to examine the file and make any needed corrections.
Using DHCP skips the step to validate node IP availability. This is an expected behavior and deployment can continue.

11. Preparing the cluster for deployment and deploying the cluster are performed with two commands:

1. The `gkectl prepare` command initializes the vSphere environment by uploading the node OS image, marking it as a template, and validating the build attestations for all container images.
2. The `gkectl create cluster` command deploys the cluster as depicted in the `config.yaml` file.

```bash
ubuntu@admin-workstation:~$ gkectl create cluster --config config.yaml
```

3. The process runs for several minutes and can be monitored on screen and in vCenter by watching the resource pool as the VMs populate. When complete, you should be able to see the `gke-admin` cluster (three nodes) and the first user cluster (four nodes).

During the deployment process, the standard out might display several messages about the current node not being available or not being ready. This is normal and happens when the control plane checks for machines that have not yet completed deployment or received DHCP addresses.

When using DHCP, if a deployment fails because nodes cannot be reached, there might not be enough available addresses in the pool. Leases for previously failed deployments might need to be cleared manually to allow for additional deployment attempts.
4. You can access and execute commands against the user cluster that has been deployed using the kubectl command line tool and the kubeconfig file generated by the process (stored in the working directory).

```
ubuntu@Anthos-Admin-Workstation:~$ kubectl get nodes --kubeconfig anthos-cluster01-kubeconfig
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATUS</th>
<th>ROLES</th>
<th>AGE</th>
<th>VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>anthos-cluster01-75c6cbdbdc-8wk6l</td>
<td>Ready</td>
<td>&lt;none&gt;</td>
<td>149m</td>
<td>v1.13.7-gke.20</td>
</tr>
<tr>
<td>anthos-cluster01-75c6cbdbdc-qhndd</td>
<td>Ready</td>
<td>&lt;none&gt;</td>
<td>149m</td>
<td>v1.13.7-gke.20</td>
</tr>
<tr>
<td>anthos-cluster01-75c6cbdbdc-tthgd</td>
<td>Ready</td>
<td>&lt;none&gt;</td>
<td>149m</td>
<td>v1.13.7-gke.20</td>
</tr>
</tbody>
</table>

9. Deploy Additional User Clusters: NetApp HCI with Anthos

With Anthos, organizations can scale their environments to incorporate multiple user clusters and segregate workloads between teams. A single admin cluster can
support up to five user clusters, and each user cluster can support up to twenty-five nodes.

To add additional user clusters to your deployment, complete the following steps:

1. Copy the config.yaml file to a new file named anthos-cluster02-config.yaml.

   ```
   ubuntu@Anthos-Admin-Workstation:~$ cp config.yaml anthos-cluster02-config.yaml
   ```

2. Make the following edits to the newly created file:
   1. Comment out the sections that refer to the existing admin cluster with (#).
   2. When you get to the usercluster section, update the following fields:
      1. Update the partition name under the bigip section.
      2. Update the controlplanvip and ingressvip values under the vip section.
      3. Update the clustername value.

   ```yaml
   usercluster:
      # In-Cluster vCenter configuration
      vcenter:
         # If specified it overwrites the network field in global vcenter configuration
         network: ""
         # The absolute or relative path to the yaml file to use for static IP allocation.
         # Do not include if using DHCP
         # ipblockfilepath: ""
         # Specify pre-defined nodeports if using "manual" load balancer mode
         # manuallbspec:
         #   ingresshttpnodeport: 30243
         #   ingresshttpsnodeport: 30879
         #   controlplanenodeport: 30562
         #   addonsnodeport: 0
         # Specify the already-existing partition and credentials to use with F5
         bigip:
            # To re-use credentials across clusters we recommend using YAML node anchors.
            # See https://yaml.org/spec/1.2/spec.html#id2785586
            credentials:
               address: "172.21.224.22"
               username: "admin"
               password: "NetApp!23"
               partition: "Anthos-Cluster02-Part"
            # Optionally specify a pool name if using SNAT
   ```
# snatpoolname: ""
# The VIPs to use for load balancing
vips:
  # Used to connect to the Kubernetes API
controlplanevip: "10.63.172.108"
  # Shared by all services for ingress traffic
  ingressvip: "10.63.172.109"
  # # Used for admin cluster addons (needed for multi cluster features). Must be the same
  ## # across clusters
  # addonsvip: ""
  # A unique name for this cluster
clustername: "anthos-cluster02"
  # User cluster master nodes must have either 1 or 3 replicas
masternode:
  cpus: 4
  memorymb: 8192
  # How many machines of this type to deploy
  replicas: 1
  # The number of worker nodes to deploy and their size. Min. 2 replicas
workernode:
  cpus: 4
  memorymb: 8192
  # How many machines of this type to deploy
  replicas: 3
  # The Kubernetes service CIDR range for the cluster
serviceiprange: 10.96.0.0/12
  # The Kubernetes pod CIDR range for the cluster
podiprange: 192.168.0.0/16

3. Run the following command to check the config file again to verify that there are no syntax errors. Because you have removed the admin section, you must reference the kubeconfig file for the admin cluster named kubeconfig (found in the working directory).
4. If all the checks succeed as expected, you can deploy this new user cluster in a manner very similar to the first cluster creation, referencing the `kubeconfig` file from the admin cluster.

5. As with the previous deployment, the process runs for several minutes and can be monitored on screen and in vCenter by watching the resource pool as the VMs populate. When complete, you should be able to see the new user cluster (four nodes).
6. You can access and execute commands against the deployed user cluster using the kubectl command line tool and the `kubeconfig` file generated by the process (stored in the working directory).

```
ubuntu@Anthos-Admin-Workstation:~$ kubectl get nodes --kubeconfig anthos-cluster02-kubeconfig
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATUS</th>
<th>ROLES</th>
<th>AGE</th>
<th>VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>anthos-cluster02-84744f5bd8-8rqk6</td>
<td>Ready</td>
<td>&lt;none&gt;</td>
<td>9m16s</td>
<td>v1.13.7-gke.20</td>
</tr>
<tr>
<td>anthos-cluster02-84744f5bd8-fl786</td>
<td>Ready</td>
<td>&lt;none&gt;</td>
<td>9m28s</td>
<td>v1.13.7-gke.20</td>
</tr>
<tr>
<td>anthos-cluster02-84744f5bd8-fnsmp</td>
<td>Ready</td>
<td>&lt;none&gt;</td>
<td>9m21s</td>
<td>v1.13.7-gke.20</td>
</tr>
</tbody>
</table>

10. Enable Access to the Cluster with the GKE Console: NetApp HCI with Anthos

After clusters are deployed and registered with Google Cloud, they must be logged into with the Google Cloud console to be managed and to receive additional cluster details. The official procedure to gain access to Anthos user clusters after they are deployed is detailed here.
The project and the specific user must be whitelisted to access on-premises clusters in the Google Cloud console and use Anthos on VMware services. If you are unable to see the clusters after they are deployed, you might need to open a support ticket with Google.

Figure 1. Non-whitelisted view.

Figure 2. View of clusters.

To enable access to your user clusters using the GKE console, complete the following steps:

1. Create a `node-reader.yaml` file that gives you the ability to access the cluster.

```yaml
kind: clusterrole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: node-reader
rules:
- apiGroups: [""
  resources: ["nodes"]
  verbs: ["get", "list", "watch"]
```

2. Apply this file to the cluster that you want to log into with the `kubectl` command.

```bash
ubuntu@Anthos-Admin-Workstation:~$ kubectl apply -f node-reader.yaml --kubeconfig anthos-cluster01-kubeconfig

clusterrole.rbac.authorization.k8s.io/node-reader created
```

3. Create a Kubernetes service account (KSA) that you can use to login. Name this account after the user that uses this account to log into the cluster.

```bash
ubuntu@Anthos-Admin-Workstation:~$ kubectl create serviceaccount netapp-user --kubeconfig anthos-cluster01-kubeconfig

serviceaccount/netapp-user created
```

4. Create cluster role-binding resources to bind both the view and newly created node-reader roles to the newly created KSA.
5. If you need to extend permissions further, you can grant the KSA user a role with cluster admin permissions in a similar manner.

```
kubectl create clusterrolebinding netapp-user-admin
--clusterrole cluster-admin --serviceaccount default:netapp-user --kubeconfig anthos-cluster01-kubeconfig
```

clusterrolebinding.rbac.authorization.k8s.io/netapp-user-admin created

6. With the KSA account created and assigned with correct permissions, you can create a bearer token to allow access with the GKE Console. To do so, set a system variable for the secret name, and pass that variable through a `kubectl` command to generate the token.

```
SECRET_NAME=$(kubectl get serviceaccount netapp-user --kubeconfig anthos-cluster01-kubeconfig -o jsonpath='{$.secrets[0].name}')
kubectl get secret $SECRET_NAME --kubeconfig anthos-cluster01-kubeconfig -o jsonpath='{$.data.token}' | base64 -d
```

7. With this token, you can visit the Google Cloud Console and log in to the cluster by clicking the login button and pasting in the token.
1. After login is complete, you see a green check mark next to the cluster name, and information is displayed about the physical environment. Clicking the cluster name displays more verbose information.

11. Install and Configure NetApp Trident Storage Provisioner: NetApp HCI with Anthos

Trident is a storage orchestrator for containers. With Trident, microservices and containerized applications can take advantage of enterprise-class storage services provided by the full NetApp portfolio of storage systems for persistent storage mounts. Depending on an application’s requirements, Trident dynamically provisions storage for ONTAP-based products such as NetApp AFF and FAS systems and Element storage systems like NetApp SolidFire® and NetApp HCI.

To install Trident on the deployed user cluster and provision a persistent volume, complete the following steps:

1. Download the installation archive to the admin workstation and extract the contents. The current version of Trident is 19.10, which can be downloaded here.
2. First set the location of the user cluster’s **kubeconfig** file as an environment variable so that you don’t have to reference it, because Trident has no option to pass this file.

```bash
ubuntu@Anthos-Admin-Workstation:~$ export KUBECONFIG=/anthos-cluster01-kubeconfig
```

3. Navigate to the Trident directory and execute the tridentctl tool to install trident to your cluster. NetApp recommends installing Trident into its own namespace within the cluster. You can then verify that the install finished correctly.
4. The next step in enabling Trident integration with the NetApp HCI solution and Anthos is to create a backend that enables communication with the storage system. There are sample backend files available in the downloaded installation archive in the `sample-input` folder. Copy the `backend-solidfire.json` to your working directory and edit it to provide information detailing the storage system environment.

```bash
ubuntu@Anthos-Admin-Workstation:~/trident-installer$ cp sample-input/backend-solidfire.json ./
ubuntu@Anthos-Admin-Workstation:~/trident-installer$ vi backend-solidfire.json
```

5. Edit the user, password, and MVIP value on the EndPoint line.

6. Edit the SVIP value.

7. With this back-end file in place, run the following command to create your first backend.

```
ubuntu@Anthos-Admin-Workstation:~/trident-installer$ ./tridentctl -n trident create backend -f backend.json
```

```
+-------------------+----------------+--------------------------------------+--------
<table>
<thead>
<tr>
<th>NAME</th>
<th>STORAGE DRIVER</th>
<th>UUID</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>solidfire-backend</td>
<td>solidfire-san</td>
<td>a5f9e159-c8f4-4340-a13a-c615fef0f433</td>
<td>online</td>
</tr>
<tr>
<td>solidfire-backend</td>
<td>solidfire-san</td>
<td>0</td>
<td>online</td>
</tr>
</tbody>
</table>
```

8. With the backend created, you must next create a storage class. Just as with the backend, there is a sample storage class file that can be edited for the environment available in the sample-inputs folder. Copy it to the working directory and make necessary edits to reflect the backend created.

```
ubuntu@Anthos-Admin-Workstation:~/trident-installer$ cp sample-input/storage-class-csi.yaml.templ ./storage-class-basic.yaml
ubuntu@Anthos-Admin-Workstation:~/trident-installer$ vi storage-class-basic.yaml
```

9. The only edit that must be made to this file is to define the `backendType` value to the name of the storage driver from the newly created backend. Also note the name-field value, which must be referenced in a later step.
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
    name: basic-csi
provisioner: csi.trident.netapp.io
parameters:
    backendType: "solidfire-san"

10. Run the **kubectl** command to create the storage class.

```bash
ubuntu@Anthos-Admin-Workstation:~/trident-installer$ kubectl create -f sample-input/storage-class-basic.yaml
```

11. With the storage class created, you must then create the first persistent volume claim (PVC). There is a sample **pvc-basic.yaml** file that can be used to perform this action located in sample-inputs as well. The only edit that must be made to this file is ensuring that the **storageClassName** field matches the one just created.

```yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
    name: basic
spec:
    accessModes:
        - ReadWriteOnce
    resources:
        requests:
            storage: 1Gi
    storageClassName: basic-csi
```

12. Create the PVC by issuing the **kubectl** command, Creation can take some time depending on the size of the backing volume being created, so you can watch the process as it completes.
```bash
ubuntu@Anthos-Admin-Workstation:~/trident-installer$ kubectl create -f sample-input/pvc-basic.yaml

ubuntu@Anthos-Admin-Workstation:~/trident-installer$ kubectl get pvc --watch

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATUS</th>
<th>VOLUME</th>
<th>CAPACITY</th>
<th>ACCESS MODES</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic</td>
<td>Pending</td>
<td>pvc-2azg0d2c-b13e-12e6-8d5f-5342040d22bf</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>basic</td>
<td>5s</td>
<td>pvc-2azg0d2c-b13e-12e6-8d5f-5342040d22bf</td>
<td></td>
<td>RWO</td>
</tr>
<tr>
<td>basic</td>
<td>Bound</td>
<td>pvc-2azg0d2c-b13e-12e6-8d5f-5342040d22bf</td>
<td>1Gi</td>
<td>RWO</td>
</tr>
<tr>
<td>basic</td>
<td>7s</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```