Configure Network Switches (Automated Deployment)

HCI

NetApp
October 01, 2020

Table of Contents

Configure Network Switches (Automated Deployment)

Prepare Required VLAN IDs

The following table lists the necessary VLANs for deployment, as outlined in this solution validation. You should configure these VLANs on the network switches prior to executing NDE.

Network Segment	Details	VLAN ID
Out-of-band management network	Network for HCI terminal user interface (TUI)	16
In-band management network	Network for accessing management interfaces of nodes, hosts, and guests	3488
VMware vMotion	Network for live migration of VMs	3489
iSCSI SAN storage	Network for iSCSI storage traffic	3490
Application	Network for Application traffic	3487
NFS	Network for NFS storage traffic	3491
IPL*	Interpeer link between Mellanox switches	4000
Native	Native VLAN	2

*Only for Mellanox switches

Switch Configuration

This solution uses Mellanox SN2010 switches running Onyx. The Mellanox switches are configured using an Ansible playbook. Prior to running the Ansible playbook, you should perform the initial configuration of the switches manually:

1. Install and cable the switches to the uplink switch, compute, and storage nodes.
2. Power on the switches and configure them with the following details:
    a. Host name
    b. Management IP and gateway
    c. NTP
3. Log into the Mellanox switches and run the following commands:

```
configuration write to pre-ansible
configuration write to post-ansible
```

The `pre-ansible` configuration file created can be used to restore the switch's configuration to the state before the Ansible playbook execution.

The switch configuration for this solution is stored in the `post-ansible` configuration file.

The configuration playbook for Mellanox switches that follows best practices and requirements for NetApp HCI can be downloaded here.

4. Fill out the credentials to access the switches and variables needed for the environment. The following text is a sample of the variable file for this solution.

```
# vars file for nar_hci_mellanox_deploy
#These set of variables will setup the Mellanox switches for NetApp HCI that uses a 2-
cable compute connectivity option.
#Ansible connection variables for mellanox
ansible_connection: network_cli
ansible_network_os: onyx
#--------------------
# Primary Variables
#--------------------
#Necessary VLANs for Standard NetApp HCI Deployment [native, Management,
iSCSI_Storage, vMotion, VM_Network, IPL]
#Any additional VLANs can be added to this in the prescribed format below
netapp_hci_vlans:
- {vlan_id: 2 , vlan_name: "Native" }
- {vlan_id: 3488 , vlan_name: "IB-Management" }
- {vlan_id: 3490 , vlan_name: "iSCSI_Storage" }
- {vlan_id: 3489 , vlan_name: "vMotion" }
- {vlan_id: 3491 , vlan_name: "NFS " }
- {vlan_id: 3487 , vlan_name: "App_Network" }
- {vlan_id: 4000 , vlan_name: "IPL" }#Modify the VLAN IDs to suit your environment
#Spanning-tree protocol type for uplink connections.
#The valid options are 'network' and 'normal'; selection depends on the uplink switch
model.
uplink_stp_type: network
#----------------------
# IPL variables
#----------------------
#Inter-Peer Link Portchannel
#ipl_portchannel to be defined in the format - Po100
ipl_portchannel: Po100
#Inter-Peer Link Addresses
```

```
#The IPL IP address should not be part of the management network. This is typically a
private network
ipl_ipaddr_a: 10.0.0.1
ipl_ipaddr_b: 10.0.0.2
#Define the subnet mask in CIDR number format. Eg: For subnet /22, use ipl_ip_subnet:
22
ipl_ip_subnet: 24
#Inter-Peer Link Interfaces
#members to be defined with Eth in the format. Eg: Eth1/1
peer_link_interfaces:
  members: ['Eth1/20', 'Eth1/22']
  description: "peer link interfaces"
#MLAG VIP IP address should be in the same subnet as that of the switches' mgmt0
interface subnet
#mlag_vip_ip to be defined in the format - <vip_ip>/<subnet_mask>. Eg: x.x.x.x/y
mlag_vip_ip: <<mlag_vip_ip>>
#MLAG VIP Domain Name
#The mlag domain must be unique name for each mlag domain.
#In case you have more than one pair of MLAG switches on the same network, each domain
(consist of two switches) should be configured with different name.
mlag_domain_name: MLAG-VIP-DOM
#---------------------
# Interface Details
#---------------------
#Storage Bond10G Interface details
#members to be defined with Eth in the format. Eg: Eth1/1
#Only numerical digits between 100 to 1000 allowed for mlag_id
#Operational link speed [variable 'speed' below] to be defined in terms of bytes.
#For 10 Gigabyte operational speed, define 10G. [Possible values - 10G and 25G]
#Interface descriptions append storage node data port numbers assuming all Storage
Nodes' Port C -> Mellanox Switch A and all Storage Nodes' Port D -> Mellanox Switch B
#List the storage Bond10G interfaces, their description, speed and MLAG IDs in list of
dictionaries format
storage_interfaces:
- {members: "Eth1/1", description: "HCI_Storage_Node_01", mlag_id: 101, speed: 25G}
- {members: "Eth1/2", description: "HCI_Storage_Node_02", mlag_id: 102, speed: 25G}
#In case of additional storage nodes, add them here
#Storage Bond1G Interface
#Mention whether or not these Mellanox switches will also be used for Storage Node
Mgmt connections
#Possible inputs for storage_mgmt are 'yes' and 'no'
storage_mgmt: <<yes or no>>
#Storage Bond1G (Mgmt) interface details. Only if 'storage_mgmt' is set to 'yes'
#Members to be defined with Eth in the format. Eg: Eth1/1
#Interface descriptions append storage node management port numbers assuming all
Storage Nodes' Port A -> Mellanox Switch A and all Storage Nodes' Port B -> Mellanox
Switch B
#List the storage Bond1G interfaces and their description in list of dictionaries
```

```
format
storage_mgmt_interfaces:
- {members: "Ethx/y", description: "HCI_Storage_Node_01"}
- {members: "Ethx/y", description: "HCI_Storage_Node_02"}
#In case of additional storage nodes, add them here
#LACP load balancing algorithm for IP hash method
#Possible options are: 'destination-mac', 'destination-ip', 'destination-port',
'source-mac', 'source-ip', 'source-port', 'source-destination-mac', 'source-
destination-ip', 'source-destination-port'
#This variable takes multiple options in a single go
#For eg: if you want to configure load to be distributed in the port-channel based on
the traffic source and destination IP address and port number, use 'source-
destination-ip source-destination-port'
#By default, Mellanox sets it to source-destination-mac. Enter the values below only
if you intend to configure any other load balancing algorithm
#Make sure the load balancing algorithm that is set here is also replicated on the
host side
#Recommended algorithm is source-destination-ip source-destination-port
#Fill the lacp_load_balance variable only if you are using configuring interfaces on
compute nodes in bond or LAG with LACP
lacp_load_balance: "source-destination-ip source-destination-port"
#Compute Interface details
#Members to be defined with Eth in the format. Eg: Eth1/1
#Fill the mlag_id field only if you intend to configure interfaces of compute nodes
into bond or LAG with LACP
#In case you do not intend to configure LACP on interfaces of compute nodes, either
leave the mlag_id field unfilled or comment it or enter NA in the mlag_id field
#In case you have a mixed architecture where some compute nodes require LACP and some
don't,
#1. Fill the mlag_id field with appropriate MLAG ID for interfaces that connect to
compute nodes requiring LACP
#2. Either fill NA or leave the mlag_id field blank or comment it for interfaces
connecting to compute nodes that do not require LACP
#Only numerical digits between 100 to 1000 allowed for mlag_id.
#Operational link speed [variable 'speed' below] to be defined in terms of bytes.
#For 10 Gigabyte operational speed, define 10G. [Possible values - 10G and 25G]
#Interface descriptions append compute node port numbers assuming all Compute Nodes'
Port D -> Mellanox Switch A and all Compute Nodes' Port E -> Mellanox Switch B
#List the compute interfaces, their speed, MLAG IDs and their description in list of
dictionaries format
compute_interfaces:
- members: "Eth1/7"#Compute Node for ESXi, setup by NDE
  description: "HCI_Compute_Node_01"
  mlag_id: #Fill the mlag_id only if you wish to use LACP on interfaces towards
compute nodes
  speed: 25G
- members: "Eth1/8"#Compute Node for ESXi, setup by NDE
  description: "HCI_Compute_Node_02"
```

```
  mlag_id: #Fill the mlag_id only if you wish to use LACP on interfaces towards
compute nodes
  speed: 25G
#In case of additional compute nodes, add them here in the same format as above-
members: "Eth1/9"#Compute Node for Kubernetes Worker node
  description: "HCI_Compute_Node_01"
  mlag_id: 109 #Fill the mlag_id only if you wish to use LACP on interfaces towards
compute nodes
  speed: 10G
- members: "Eth1/10"#Compute Node for Kubernetes Worker node
  description: "HCI_Compute_Node_02"
  mlag_id: 110 #Fill the mlag_id only if you wish to use LACP on interfaces towards
compute nodes
  speed: 10G
#Uplink Switch LACP support
#Possible options are 'yes' and 'no' - Set to 'yes' only if your uplink switch
supports LACP
uplink_switch_lacp: <<yes or no>>
#Uplink Interface details
#Members to be defined with Eth in the format. Eg: Eth1/1
#Only numerical digits between 100 to 1000 allowed for mlag_id.
#Operational link speed [variable 'speed' below] to be defined in terms of bytes.
#For 10 Gigabyte operational speed, define 10G. [Possible values in Mellanox are 1G,
10G and 25G]
#List the uplink interfaces, their description, MLAG IDs and their speed in list of
dictionaries format
uplink_interfaces:
- members: "Eth1/18"
  description_switch_a: "SwitchA:Ethx/y -> Uplink_Switch:Ethx/y"
  description_switch_b: "SwitchB:Ethx/y -> Uplink_Switch:Ethx/y"
  mlag_id: 118  #Fill the mlag_id only if 'uplink_switch_lacp' is set to 'yes'
  speed: 10G
  mtu: 1500
```

The fingerprint for the switch's key must match with that present in the host machine from where the playbook is being executed. To ensure this, add the key to `/root/. ssh/known_host` or any other appropriate location.

Rollback the Switch Configuration

1. In case of any timeout failures or partial configuration, run the following command to roll back the switch to the initial state.

```
configuration switch-to pre-ansible
```

This operation requires a reboot of the switch.

2. Switch the configuration to the state before running the Ansible playbook.

```
configuration delete post-ansible
```

3. Delete the post-ansible file that had the configuration from the Ansible playbook.

```
configuration write to post-ansible
```

4. Create a new file with the same name post-ansible, write the pre-ansible configuration to it, and switch to the new configuration to restart configuration.

IP Address Requirements

The deployment of the NetApp HCI inferencing platform with VMware and Kubernetes requires multiple IP addresses to be allocated. The following table lists the number of IP addresses required. Unless otherwise indicated, addresses are assigned automatically by NDE.

IP Address Quantity	Details	VLAN ID	IP Address
One per storage and compute node*	HCI terminal user interface (TUI) addresses	16	
One per vCenter Server (VM)	vCenter Server management address	3488	
One per management node (VM)	Management node IP address		
One per ESXi host	ESXi compute management addresses		
One per storage/witness node	NetApp HCI storage node management addresses		
One per storage cluster	Storage cluster management address		
One per ESXi host	VMware vMotion address	3489	
Two per ESXi host	ESXi host initiator address for iSCSI storage traffic	3490	

IP Address Quantity	Details	VLAN ID	IP Address
Two per storage node	Storage node target address for iSCSI storage traffic		
Two per storage cluster	Storage cluster target address for iSCSI storage traffic		
Two for mNode	mNode iSCSI storage access		

The following IPs are assigned manually when the respective components are configured.

IP Address Quantity	Details	VLAN ID	IP Address
One for Deployment Jump Management network	Deployment Jump VM to execute Ansible playbooks and configure other parts of the system – management connectivity	3488	
One per Kubernetes master node – management network	Kubernetes master node VMs (three nodes)	3488	
One per Kubernetes worker node – management network	Kubernetes worker nodes (two nodes)	3488	
One per Kubernetes worker node – NFS network	Kubernetes worker nodes (two nodes)	3491	
One per Kubernetes worker node – application network	Kubernetes worker nodes (two nodes)	3487	
Three for ONTAP Select – management network	ONTAP Select VM	3488	
One for ONTAP Select – NFS network	ONTAP Select VM – NFS data traffic	3491	
At least two for Triton Inference Server Load Balancer – application network	Load balancer IP range for Kubernetes load balancer service	3487	

*This validation requires the initial setup of the first storage node TUI address. NDE automatically assigns the TUI address for subsequent nodes.

DNS and Timekeeping Requirement

Depending on your deployment, you might need to prepare DNS records for your NetApp HCI system. NetApp HCI requires a valid NTP server for timekeeping; you can use a publicly available time server if you do not have one in your environment.

This validation involves deploying NetApp HCI with a new VMware vCenter Server instance using a fully qualified domain name (FQDN). Before deployment, you must have one Pointer (PTR) record and one Address (A) record created on the DNS server.

Next: Virtual Infrastructure with Automated Deployment