NetApp Solution Automation

NetApp Solutions

Nikhil M Kulkarni
March 02, 2021

Table of Contents

NetApp Solution Automation

Procedure

To configure Mellanox switches using ansible playbooks via CLI, follow the below procedure:

1. Download the Mellanox ONYX Ansible collections

```
ansible-galaxy collection install -r collections/requirements.yml
--force
```

2. Download the Ansible content required for configuring the Mellanox switches

```
git clone https://bitbucket.ngage.netapp.com/scm/ns-
bb/na_with_mellanox.git
```

3. Change the directory to the required solution

```
cd na_with_mellanox/solutions/nar_hci_mellanox_deploy
```

4. Fill the variables below based on the recommendations/suggestions provided. Once all the variables are filled as per requirement, copy the content by clicking 'Copy' button at the top.

```
<style>
div {
position: relative;
}
div button {
position: absolute;
top: 0;
right: 0;
}
button {
  transition-duration: 0.4s;
  background-color: white;
  color: #1563a3;
  border: 2px solid #1563a3;
}
button:hover {
  background-color: #1563a3;
  color: white;
}
#more_vlans {
```

```
  display: block;
}
#more_vlans_button {
  display: none;
}
#more_peerlink_interfaces {
  display: block;
}
#more_peerlink_interfaces_button {
  display: none;
}
#more_storage_interfaces {
  display: block;
}
#more_storage_interfaces_button {
  display: none;
}
#more_storage_mgmt_interfaces {
  display: block;
}
#more_storage_mgmt_interfaces_button {
  display: none;
}
#more_compute_interfaces {
  display: block;
}
#more_compute_interfaces_button {
  display: none;
}
#more_uplink_interfaces {
  display: block;
}
#more_uplink_interfaces_button {
  display: none;
}
</style>
<div class="listingblock"><div class="content"><div><button id="copy-
button" onclick="CopyClassText()">Copy</button></div><pre><code><div
class="CopyMeClass" id="CopyMeID"># vars file for nar_hci_mellanox_deploy

#Hosts group name
#Default hosts group name - 'mellanox'
#Change only if you are changing the hosts group name either in hosts file
or in inventory groups in case of AWX/Tower
hosts: <span <div contenteditable="true"/><i>mellanox</i></span>
```

```
#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
<br>netapp_hci_vlans:
  - {vlan_id: <span <div contenteditable="true"/><i>2</i></span>,
vlan_name: &amp;quot;<span <div
contenteditable="true"/><i>OB_Management</i></span>&amp;quot;}
  - {vlan_id: <span <div contenteditable="true"/><i>3488</i></span>,
vlan_name: &amp;quot;<span <div
contenteditable="true"/><i>IB_Managment</i></span>&amp;quot;}
  - {vlan_id: <span <div contenteditable="true"/><i>3489</i></span>,
vlan_name: &amp;quot;<span <div
contenteditable="true"/><i>vMotion</i></span>&amp;quot;}
  - {vlan_id: <span <div contenteditable="true"/><i>3490</i></span>,
vlan_name: &amp;quot;<span <div
contenteditable="true"/><i>iSCSI_Storage</i></span>&amp;quot;}
  - {vlan_id: <span <div contenteditable="true"/><i>3487</i></span>,
vlan_name: &amp;quot;<span <div
contenteditable="true"/><i>Application</i></span>&amp;quot;}
  - {vlan_id: <span <div contenteditable="true"/><i>3491</i></span>,
vlan_name: &amp;quot;<span <div
contenteditable="true"/><i>NFS</i></span>&amp;quot;}
  - {vlan_id: <span <div contenteditable="true"/><i>4000</i></span>,
vlan_name: &amp;quot;<span <div
contenteditable="true"/><i>IPL</i></span>&amp;quot;}
  - {vlan_id: <span <div contenteditable="true"/><i>2</i></span>,
vlan_name: &amp;quot;<span <div
contenteditable="true"/><i>Native</i></span>&amp;quot;}
<a id="more_vlans" href="javascript:vlandropdown();">More VLANs</a><div
id="select_more_vlans"></div><a id="more_vlans_button"
href="javascript:addvlans();">Enter VLAN details</a><div
id="extra_vlans"></div>
#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
lacp_load_balance: &amp;quot;<span <div contenteditable="true"/><i>source-
destination-ip source-destination-port</i></span>&amp;quot;

#----------------------
# IPL variables
#----------------------

#Inter-Peer Link Portchannel
#ipl_portchannel to be defined in the format - Po100
ipl_portchannel: <span <div contenteditable="true"/><i>Po100</i></span>

#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: <span <div contenteditable="true"/><i>10.0.0.1</i></span>
ipl_ipaddr_b: <span <div contenteditable="true"/><i>10.0.0.2</i></span>

#Define the subnet mask in CIDR number format. Eg: For subnet /22, use
ipl_ip_subnet: 22
ipl_ip_subnet: <span <div contenteditable="true"/><i>24</i></span>

#Inter-Peer Link Interfaces
#members to be defined with Eth in the format. Eg: Eth1/1

peer_link_interfaces:
  description: &amp;quot;<span <div contenteditable="true"/><i>peer link
interfaces</i></span>&amp;quot;
  members:
    - &amp;quot;<span <div contenteditable="true"/><i>Eth1/20</i></span>&amp;quot;
    - &amp;quot;<span <div contenteditable="true"/><i>Eth1/22</i></span>&amp;quot;
<a id="more_peerlink_interfaces" href="javascript:ipldropdown();">More
Peer-Link Interfaces</a><div id="select_more_peerlink_interfaces"></div><a
id="more_peerlink_interfaces_button" href="javascript:addipls();">Enter
peer-link Interface details</a><div id="extra_peerlink_interfaces"></div>
#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 and must be unique for each MLAG domain
```

```
mlag_vip_ip: <span <div
contenteditable="true"/><i>10.195.60.111/24</i></span>

#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: <span <div contenteditable="true"/><i>MLAG-VIP-DOM-10-
16</i></span>

#---------------------
# 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: &amp;quot;<span <div
contenteditable="true"/><i>Eth1/13</i></span>&amp;quot;, description:
&amp;quot;<span <div
contenteditable="true"/><i>HCI_Storage_Node_01</i></span>&amp;quot;, mlag_id:
<span <div contenteditable="true"/><i>113</i></span>, speed: <span <div
contenteditable="true"/><i>10G</i></span>}
  - {members: &amp;quot;<span <div
contenteditable="true"/><i>Eth1/14</i></span>&amp;quot;, description:
&amp;quot;<span <div
contenteditable="true"/><i>HCI_Storage_Node_02</i></span>&amp;quot;, mlag_id:
<span <div contenteditable="true"/><i>114</i></span>, speed: <span <div
contenteditable="true"/><i>10G</i></span>}
  - {members: &amp;quot;<span <div
contenteditable="true"/><i>Eth1/15</i></span>&amp;quot;, description:
&amp;quot;<span <div
contenteditable="true"/><i>HCI_Storage_Node_03</i></span>&amp;quot;, mlag_id:
<span <div contenteditable="true"/><i>115</i></span>, speed: <span <div
```

```
contenteditable="true"/><i>10G</i></span>}
  - {members: &amp;quot;<span <div
contenteditable="true"/><i>Eth1/16</i></span>&amp;quot;, description:
&amp;quot;<span <div
contenteditable="true"/><i>HCI_Storage_Node_04</i></span>&amp;quot;, mlag_id:
<span <div contenteditable="true"/><i>116</i></span>, speed: <span <div
contenteditable="true"/><i>10G</i></span>}
<a id="more_storage_interfaces" href="javascript:storagedropdown();">More
Interfaces towards Storage Nodes</a><div
id="select_more_storage_interfaces"></div><a
id="more_storage_interfaces_button"
href="javascript:addstorageinterfaces();">Enter Storage Interface
details</a><div id="extra_storage_interfaces"></div>
#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: <span <div contenteditable="true"/><i>no</i></span>

#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: &amp;quot;<span <div
contenteditable="true"/><i>Eth1/5</i></span>&amp;quot;, description:
&amp;quot;<span <div
contenteditable="true"/><i>HCI_Storage_Node_01</i></span>&amp;quot;}
  - {members: &amp;quot;<span <div
contenteditable="true"/><i>Eth1/6</i></span>&amp;quot;, description:
&amp;quot;<span <div
contenteditable="true"/><i>HCI_Storage_Node_02</i></span>&amp;quot;}
  - {members: &amp;quot;<span <div
contenteditable="true"/><i>Eth1/7</i></span>&amp;quot;, description:
&amp;quot;<span <div
contenteditable="true"/><i>HCI_Storage_Node_03</i></span>&amp;quot;}
  - {members: &amp;quot;<span <div
contenteditable="true"/><i>Eth1/8</i></span>&amp;quot;, description:
&amp;quot;<span <div
contenteditable="true"/><i>HCI_Storage_Node_04</i></span>&amp;quot;}
<a id="more_storage_mgmt_interfaces"
```

```
href="javascript:storagemgmtdropdown();">More management interfaces
towards storage nodes</a><div
id="select_more_storage_mgmt_interfaces"></div><a
id="more_storage_mgmt_interfaces_button"
href="javascript:addstoragemgmtinterfaces();">Enter Storage Management
Interface details</a><div id="extra_storage_mgmt_interfaces"></div>
#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: &amp;quot;<span <div
contenteditable="true"/><i>Eth1/2</i></span>&amp;quot;
    description: &amp;quot;<span <div
contenteditable="true"/><i>HCI_Compute_Node_01</i></span>&amp;quot;
    mlag_id: <span <div contenteditable="true"/><i> </i></span>  #Fill the
mlag_id only if you wish to use LACP on interfaces towards compute nodes
    speed: <span <div contenteditable="true"/><i>10G</i></span>
  - members: &amp;quot;<span <div
contenteditable="true"/><i>Eth1/4</i></span>&amp;quot;
    description: &amp;quot;<span <div
contenteditable="true"/><i>HCI_Compute_Node_02</i></span>&amp;quot;
    mlag_id: <span <div contenteditable="true"/><i>104</i></span>  #Fill
the mlag_id only if you wish to use LACP on interfaces towards compute
nodes
    speed: <span <div contenteditable="true"/><i>10G</i></span>
```

```
<a id="more_compute_interfaces" href="javascript:computedropdown();">More
interfaces towards compute nodes</a><div
id="select_more_compute_interfaces"></div><a
id="more_compute_interfaces_button"
href="javascript:addcomputeinterfaces();">Enter Compute Interface
details</a><div id="extra_compute_interfaces"></div>
#Spanning-tree protocol type for uplink connections.
#The valid options are 'network' and 'normal'; selection depends on the
uplink switch model.
uplink_stp_type: <span <div contenteditable="true"/><i>network</i></span>

#Uplink Switch LACP support
#Possible options are 'yes' and 'no' - Set to 'yes' only if your uplink
switch supports LACP
uplink_switch_lacp: <span <div contenteditable="true"/><i>yes</i></span>

#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: &amp;quot;<span <div
contenteditable="true"/><i>Eth1/18</i></span>&amp;quot;
    description_switch_a: &amp;quot;<span <div
contenteditable="true"/><i>SwitchA:Ethx/y ->
Uplink_Switch:Ethx/y</i></span>&amp;quot;
    description_switch_b: &amp;quot;<span <div
contenteditable="true"/><i>SwitchB:Ethx/y ->
Uplink_Switch:Ethx/y</i></span>&amp;quot;
    mlag_id: <span <div contenteditable="true"/><i>118</i></span>  #Fill
the mlag_id only if 'uplink_switch_lacp' is set to 'yes'
    speed: <span <div contenteditable="true"/><i>10G</i></span>
    mtu: <span <div contenteditable="true"/><i>1500</i></span>
<a id="more_uplink_interfaces" href="javascript:uplinkdropdown();">More
interfaces towards uplink switches</a><div
id="select_more_uplink_interfaces"></div><a
id="more_uplink_interfaces_button"
href="javascript:adduplinkinterfaces();">Enter Uplink Interface
details</a><div
id="extra_uplink_interfaces"></div></div></code></pre></div></div>
```

```
<script>
function CopyClassText(){
    var textToCopy = document.getElementById("CopyMeID");
    var currentRange;
    if(document.getSelection().rangeCount > 0)
    {
        currentRange = document.getSelection().getRangeAt(0);
        window.getSelection().removeRange(currentRange);
    }
    else
    {
        currentRange = false;
    }
    var CopyRange = document.createRange();
    CopyRange.selectNode(textToCopy);
    window.getSelection().addRange(CopyRange);
    document.getElementById("more_vlans").style.display = "none";
    document.getElementById("more_peerlink_interfaces").style.display =
"none";
    document.getElementById("more_storage_interfaces").style.display =
"none";
    document.getElementById("more_storage_mgmt_interfaces").style.display
= "none";
    document.getElementById("more_compute_interfaces").style.display =
"none";
    document.getElementById("more_uplink_interfaces").style.display =
"none";
    var command = document.execCommand("copy");
      if (command)
      {
          document.getElementById("copy-button").innerHTML = "Copied!";
          setTimeout(revert_copy, 3000);
      }
    window.getSelection().removeRange(CopyRange);
    if(currentRange)
    {
        window.getSelection().addRange(currentRange);
    }
}
function revert_copy() {
      document.getElementById("copy-button").innerHTML = "Copy";
      document.getElementById("more_vlans").style.display = "block";
      document.getElementById("more_peerlink_interfaces").style.display =
"block";
      document.getElementById("more_storage_interfaces").style.display =
"block";
```

```
document.getElementById("more_storage_mgmt_interfaces").style.display =
"block";
      document.getElementById("more_compute_interfaces").style.display =
"block";
      document.getElementById("more_uplink_interfaces").style.display =
"block";
}
function vlandropdown() {
    document.getElementById("more_vlans").style.display = "none";
    document.getElementById("more_vlans_button").style.display = "block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper = document.getElementById("select_more_vlans");
    while (x < 100) {
      buildup += '<option value="' + x + '">' + x + '</option>';
      x++;
    }
    myHTML += '<a id="more_vlans_info">How many extra VLANs do you wish to
add?</a><select name="number_of_extra_vlans" id="number_of_extra_vlans">'
+ buildup + '</select>';
    wrapper.innerHTML = myHTML;
}

function addvlans() {
    var y = document.getElementById("number_of_extra_vlans").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_vlans");
    while (j < y) {
        j++;
        myHTML += '  - {vlan_id: <span <div contenteditable="true"/><i>
</i></span>, vlan_name: &amp;quot;<span <div contenteditable="true"/><i>
</i></span>&amp;quot;}<br>';
    }
    wrapper.innerHTML = myHTML;
    document.getElementById("select_more_vlans").style.display = "none";
    document.getElementById("more_vlans_button").style.display = "none";
}
function ipldropdown() {
    document.getElementById("more_peerlink_interfaces").style.display =
"none";

document.getElementById("more_peerlink_interfaces_button").style.display =
"block";
```

```
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper =
document.getElementById("select_more_peerlink_interfaces");
    while (x < 10) {
        buildup += '<option value="' + x + '">' + x + '</option>';
        x++;
    }
    myHTML += '<a id="more_peerlink_interfaces_info">How many more Peer-
Link Interfaces do you wish to add?</a><select
name="number_of_extra_peerlink_interfaces"
id="number_of_extra_peerlink_interfaces">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function addipls() {
    var y =
document.getElementById("number_of_extra_peerlink_interfaces").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_peerlink_interfaces");
    while (j < y) {
        j++;
        myHTML += '    - &quot;<span <div contenteditable="true"/><i>
</i></span>&quot;<br>';
    }
    wrapper.innerHTML = myHTML;

document.getElementById("select_more_peerlink_interfaces").style.display =
"none";

document.getElementById("more_peerlink_interfaces_button").style.display =
"none";
}
function storagedropdown() {
    document.getElementById("more_storage_interfaces").style.display =
"none";

document.getElementById("more_storage_interfaces_button").style.display =
"block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper =
document.getElementById("select_more_storage_interfaces");
    while (x < 10) {
```

```
        buildup += '<option value="' + x + '">' + x + '</option>';
        x++;
    }
    myHTML += '<a id="more_storage_interfaces_info">How many more Storage
Interfaces do you wish to add?</a><select
name="number_of_extra_storage_interfaces"
id="number_of_extra_storage_interfaces">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function addstorageinterfaces() {
    var y =
document.getElementById("number_of_extra_storage_interfaces").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_storage_interfaces");
    while (j < y) {
        j++;
        myHTML += '  - {members: &amp;quot;<span <div
contenteditable="true"/><i>Ethx/y</i></span>&amp;quot;, description:
&amp;quot;<span <div
contenteditable="true"/><i>HCI_Storage_Node_x</i></span>&amp;quot;, mlag_id:
<span <div contenteditable="true"/><i>xxx</i></span>, speed: <span <div
contenteditable="true"/><i>10G</i></span>}<br>';
    }
    wrapper.innerHTML = myHTML;

document.getElementById("select_more_storage_interfaces").style.display =
"none";

document.getElementById("more_storage_interfaces_button").style.display =
"none";
}
function storagemgmtdropdown() {
    document.getElementById("more_storage_mgmt_interfaces").style.display
= "none";

document.getElementById("more_storage_mgmt_interfaces_button").style.displ
ay = "block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper =
document.getElementById("select_more_storage_mgmt_interfaces");
    while (x < 10) {
        buildup += '<option value="' + x + '">' + x + '</option>';
        x++;
```

```
    }
    myHTML += '<a id="more_storage_mgmt_interfaces_info">How many more
Storage Management Interfaces do you wish to add?</a><select
name="number_of_extra_storage_mgmt_interfaces"
id="number_of_extra_storage_mgmt_interfaces">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function addstoragemgmtinterfaces() {
    var y =
document.getElementById("number_of_extra_storage_mgmt_interfaces").value;
    var j=0;
    var myHTML = '';
    var wrapper =
document.getElementById("extra_storage_mgmt_interfaces");
    while (j < y) {
        j++;
        myHTML += '  - {members: &quot;<span <div
contenteditable="true"/><i>Ethx/y</i></span>&quot;, description:
&quot;<span <div
contenteditable="true"/><i>HCI_Storage_Node_x</i></span>&quot;}<br>';
    }
    wrapper.innerHTML = myHTML;

document.getElementById("select_more_storage_mgmt_interfaces").style.displ
ay = "none";

document.getElementById("more_storage_mgmt_interfaces_button").style.displ
ay = "none";
}
function computedropdown() {
    document.getElementById("more_compute_interfaces").style.display =
"none";

document.getElementById("more_compute_interfaces_button").style.display =
"block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper =
document.getElementById("select_more_compute_interfaces");
    while (x < 10) {
        buildup += '<option value="' + x + '">' + x + '</option>';
        x++;
    }
    myHTML += '<a id="more_compute_interfaces_info">How many more Compute
Interfaces do you wish to add?</a><select
```

```
name="number_of_extra_compute_interfaces"
id="number_of_extra_compute_interfaces">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function addcomputeinterfaces() {
    var y =
document.getElementById("number_of_extra_compute_interfaces").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_compute_interfaces");
    while (j < y) {
        j++;
        myHTML += '  - members: &quot;<span <div
contenteditable="true"/><i>Ethx/y</i></span>&quot;<br>    description:
&quot;<span <div
contenteditable="true"/><i>HCI_Compute_Node_x</i></span>&quot;<br>
mlag_id: <span <div contenteditable="true"/><i> </i></span>  #Fill the
mlag_id only if you wish to use LACP on interfaces towards compute
nodes<br>    speed: <span <div
contenteditable="true"/><i>10G</i></span><br>';
    }
    wrapper.innerHTML = myHTML;

document.getElementById("select_more_compute_interfaces").style.display =
"none";

document.getElementById("more_compute_interfaces_button").style.display =
"none";
}
function uplinkdropdown() {
    document.getElementById("more_uplink_interfaces").style.display =
"none";
    document.getElementById("more_uplink_interfaces_button").style.display
= "block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper =
document.getElementById("select_more_uplink_interfaces");
    while (x < 10) {
        buildup += '<option value="' + x + '">' + x + '</option>';
        x++;
    }
    myHTML += '<a id="more_uplink_interfaces_info">How many more Uplink
Interfaces do you wish to add?</a><select
name="number_of_extra_uplink_interfaces"
```

```
id="number_of_extra_uplink_interfaces">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function adduplinkinterfaces() {
    var y =
document.getElementById("number_of_extra_uplink_interfaces").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_uplink_interfaces");
    while (j < y) {
        j++;
        myHTML += '  - members: &quot;<span <div
contenteditable="true"/><i>Ethx/y</i></span>&quot;<br>
description_switch_a: &quot;<span <div
contenteditable="true"/><i>SwitchA:Ethx/y ->
Uplink_Switch:Ethx/y</i></span>&quot;<br>    description_switch_b:
&quot;<span <div contenteditable="true"/><i>SwitchB:Ethx/y ->
Uplink_Switch:Ethx/y</i></span>&quot;<br>    mlag_id: <span <div
contenteditable="true"/><i>xxx</i></span>  #Fill the mlag_id only if
uplink_switch_lacp is set to yes<br>    speed: <span <div
contenteditable="true"/><i>10G</i></span><br>    mtu: <span <div
contenteditable="true"/><i>1500</i></span><br>';
    }
    wrapper.innerHTML = myHTML;
    document.getElementById("select_more_uplink_interfaces").style.display
= "none";
    document.getElementById("more_uplink_interfaces_button").style.display
= "none";
}
</script>
```

4. Create a new variable file (yml file) in the current directory (directory same as that of the playbook) and paste the copied variables into the file and save it.
5. Fill the IP addresses/hostnames of Mellanox switches below and copy the content. Open the hosts file, clear all the information present and paste the copied content to it.

```
<style>
div {
position: relative;
}
div button {
position: absolute;
top: 0;
right: 0;
}
```

```
button {
  transition-duration: 0.4s;
  background-color: white;
  color: #1563a3;
  border: 2px solid #1563a3;
}
button:hover {
  background-color: #1563a3;
  color: white;
}
</style>
<div class="listingblock"><div class="content"><div><button id="copy-host-
button" onclick="CopyHostClass()">Copy</button></div><pre><code><div
class="CopyMeHostClass" id="CopyMeHostID">[mellanox]
<span <div contenteditable="true"/><i>x.x.x.x</i></span> id="A" #IP
Address of the Mellanox Switch A
<span <div contenteditable="true"/><i>y.y.y.y</i></span> id="B" #IP
Address of the Mellanox Switch B</div></code></pre></div></div>
<script>
function CopyHostClass(){
    var textToCopy = document.getElementById("CopyMeHostID");
    var currentRange;
    if(document.getSelection().rangeCount > 0)
    {
        currentRange = document.getSelection().getRangeAt(0);
        window.getSelection().removeRange(currentRange);
    }
    else
    {
        currentRange = false;
    }
    var CopyRange = document.createRange();
    CopyRange.selectNode(textToCopy);
    window.getSelection().addRange(CopyRange);
    var command = document.execCommand("copy");
      if (command)
      {
          document.getElementById("copy-host-button").innerHTML =
"Copied!";
          setTimeout(revert_host_copy, 3000);
      }
    window.getSelection().removeRange(CopyRange);
    if(currentRange)
    {
        window.getSelection().addRange(currentRange);
    }
```

```
}
function revert_host_copy() {
      document.getElementById("copy-host-button").innerHTML = "Copy";
}
</script>
```

6. Run the playbook by passing the variable file and mellanox switch username. Fill the password for Mellanox switches when prompted.

```
<style>
div {
position: relative;
}
div button {
position: absolute;
top: 0;
right: 0;
}
button {
  transition-duration: 0.4s;
  background-color: white;
  color: #1563a3;
  border: 2px solid #1563a3;
}
button:hover {
  background-color: #1563a3;
  color: white;
}
</style>
<div class="listingblock"><div class="content"><div><button id="copy-
playbook-button"
onclick="CopyPlaybookClass()">Copy</button></div><pre><code><div
class="CopyMePlaybookClass" id="CopyMePlaybookCommand">ansible-playbook -i
hosts nar_hci_mellanox_deploy_playbook.yml -u <span <div
contenteditable="true"/><i>mellanox_switch_username</i></span> -k -e
@mellanox_vars.yml</div></code></pre></div></div>
<script>
function CopyPlaybookClass(){
    var textToCopy = document.getElementById("CopyMePlaybookCommand");
    var currentRange;
    if(document.getSelection().rangeCount > 0)
    {
        currentRange = document.getSelection().getRangeAt(0);
        window.getSelection().removeRange(currentRange);
    }
```

```
    else
    {
        currentRange = false;
    }
    var CopyRange = document.createRange();
    CopyRange.selectNode(textToCopy);
    window.getSelection().addRange(CopyRange);
    var command = document.execCommand("copy");
      if (command)
      {
          document.getElementById("copy-playbook-button").innerHTML =
"Copied!";
          setTimeout(revert_playbook_copy, 3000);
      }
    window.getSelection().removeRange(CopyRange);
    if(currentRange)
    {
        window.getSelection().addRange(currentRange);
    }
}
function revert_playbook_copy() {
      document.getElementById("copy-playbook-button").innerHTML = "Copy";
}
</script>
```

Replace mellanox_switch_username with the appropriate username for logging in the mellanox switches before running the playbook with above command.