GTC 2022 Academy Workshop: Hands-on Workbook

Scope

This workbook covers configurations in the NVIDIA Academy GTC Workshop.

Audience

This workbook is intended for Technical Training students registered to the "Configure High Mobility AI-Infrastructure in 5 min" GTC training session.

Objectives

By the end of this workbook, students will be able to:

- Configure switches and servers using Ansible automation tool.
- Configure layer 2 and layer 3 protocols on NVIDIA Cumulus Linux switches.
- Verify configuration and connectivity

Overview

Each student will be using the NVIDIA Air © platform, exercises in this workbook on a group of devices (servers and switches).

Notice

Please follow the instructions below carefully to successfully complete the practice.
If you encounter technical issues, please contact the NVIDIA Academy instructors.

Release Date

Revision 1.0 – November 2021

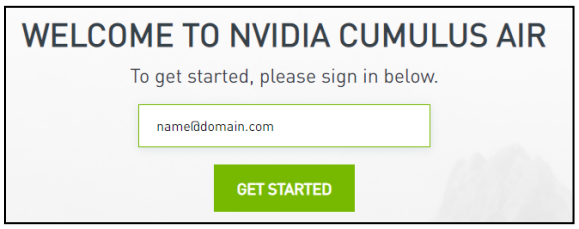
Good Luck,

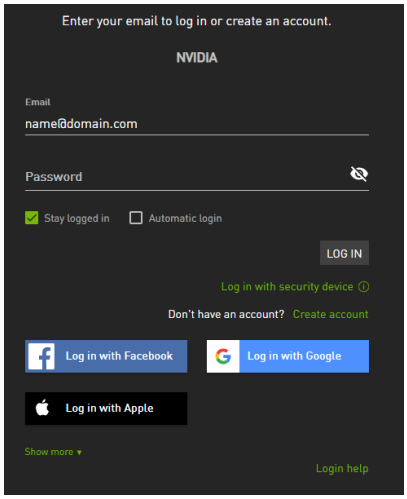
NVIDIA Academy Team

Table of Contents

Hands-on: Prerequisites and Guidelines

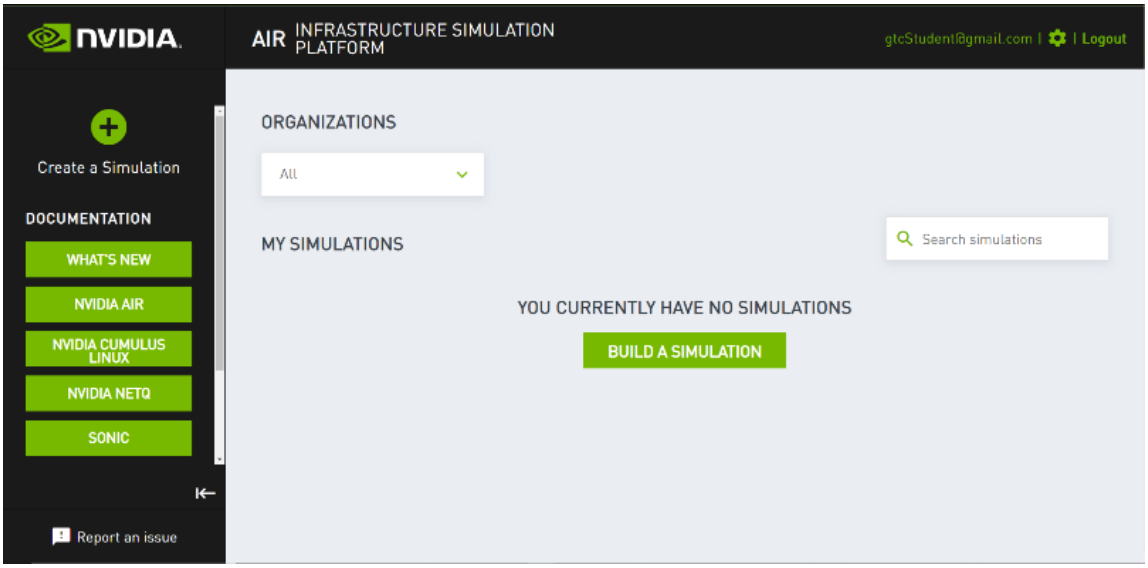
1. Enter the Cumulus Air web page : https://air.nvidia.com/Login
   Click "GET STARTED" button.



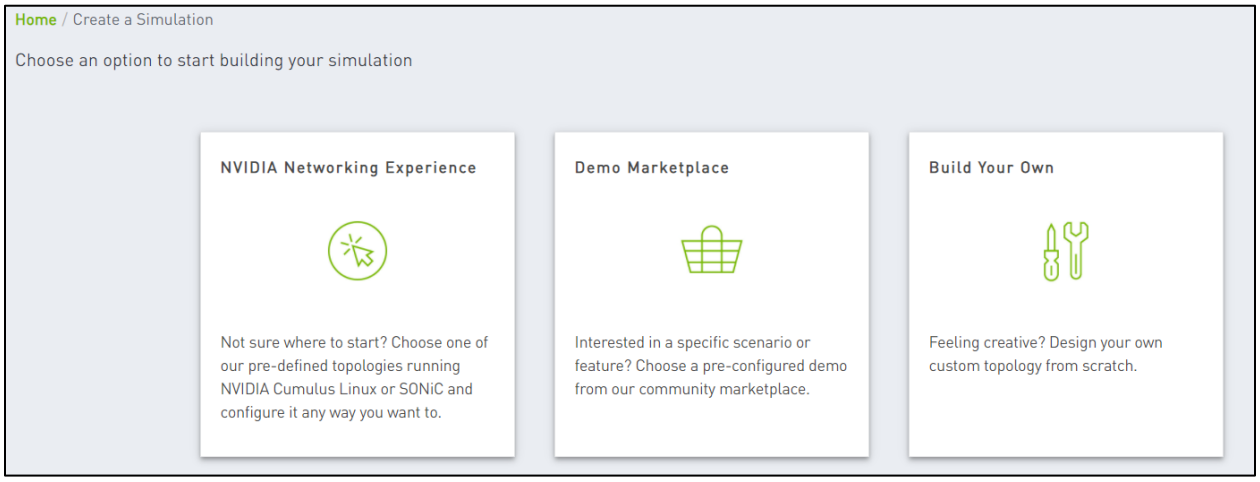


- If you have already created an account, use your credentials to Login.
- To sign up for the first time, click "Create account" and fill in your details.
  Once completed, a confirmation email will be sent, open it to activate your new account.

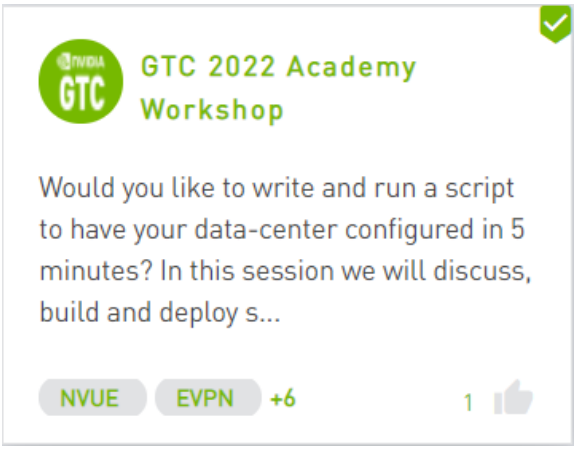
2. Once you are logged in, wait for the page to load and click on the "Create a Simulation" button on the left side of the dashboard..



3. Choose "Demo Marketplace"



4. Find and click the "GTC 2022 Academy Workshop" label.

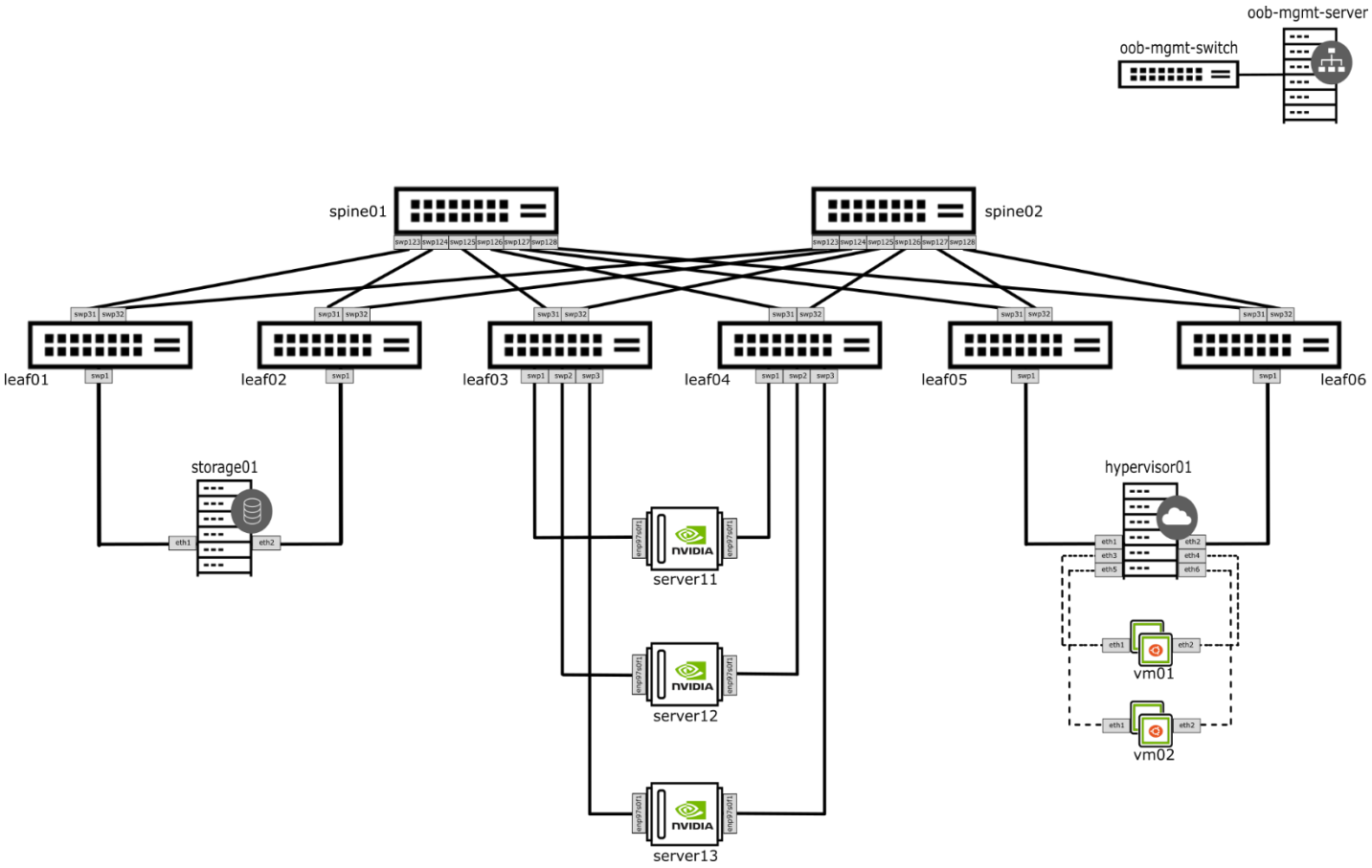


Then click on LAUNCH

5. Wait couple minutes for the simulation to load, and we are ready!

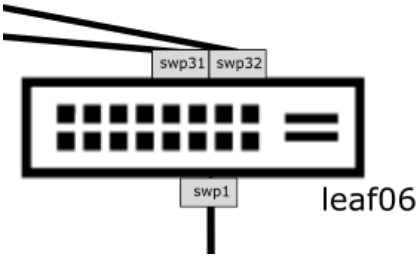
ACADEMY LAB TOPOLOGY

The workshop lab is organized in the following topology:

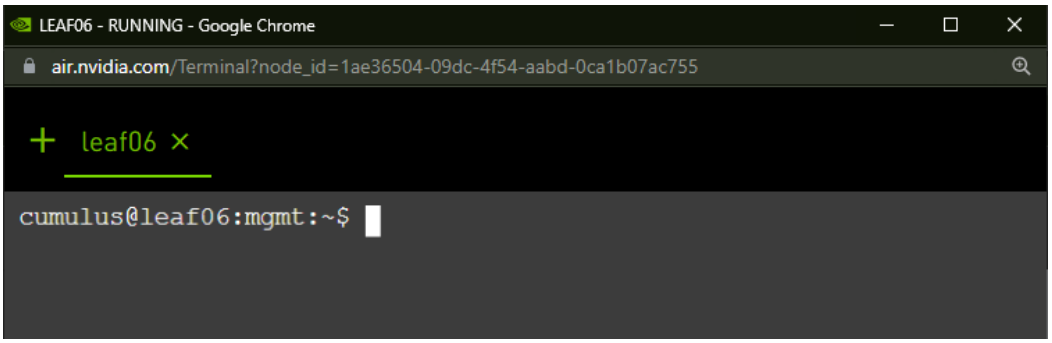


NVIDIA Academy Virtual Lab Access

Click on a NODE to open its console



1. When the login prompt appears, enter the username – "**cumulus**"
2. When the password prompt appears, enter the password – "Academy123" and press Enter.
3. You should now be prompted with the node's name. This indicates that you have successfully accessed the node.



⚠ Please note

The lab can be accessed using SSH rather than the GUI console,
please ask the GTC instructor for more information regarding SSH keys and connecting using your favorite SSH-client.

Workshop Steps

Step 1.1: Getting Started with Ansible - Inventory

Practice objectives:

In this practice session you perform the initial configurations required for Ansible to start working with the group servers and switches.

- You will configure **hosts** and **groups** in an Ansible hosts file.
- You will use **Ansible ping module** to validate the configuration.
- Last, you will use **Ansible Variables** to refine the hosts configuration.

Task 1: Creating an Ansible Inventory (hosts) file

a. Connect to the 'oob-mgmt-server', create a new directory and Use VIM, or another text editor to create a new file named **hosts** :

```
# mkdir practice1
# sudo vi practice1/hosts
```

```
~
~
~
~
~
~
~
~
"practice1/hosts" 0L, 0C
0,0-1         All
```

⚠ to exit VIM:

1. Press ESC
2. Type ':'
3. Type "q!" to exit **without saving** or "wq" to **save and exit**

```
~
~
:wq
```

To edit the file using VIM go to insert mode by typing 'a'
(make sure the word "—INSERT --" appears at the end of the page).

```
~
~
~
~
~
-- INSERT --
```

Task 2: Adding servers to the Inventory (hosts) file

a. While in "INSERT" mode, add the servers host name to the hosts file.

```
# storage
storage01 ansible_user=cumulus ansible_ssh_pass=Academy123

# compute
server11 ansible_user=cumulus ansible_ssh_pass=Academy123
server12 ansible_user=cumulus ansible_ssh_pass=Academy123
server13 ansible_user=cumulus ansible_ssh_pass=Academy123

# virtualization
hypervisor01 ansible_user=cumulus ansible_ssh_pass=Academy123
vm01 ansible_user=cumulus ansible_ssh_pass=Academy123
vm02 ansible_user=cumulus ansible_ssh_pass=Academy123

~
~
~
```

Please note:

- Every line that starts with '#' is considered a comment and can be deleted.
- Instead of configuring each server in a different line, you can use a REGEX expression to capture all compute servers in one line ***# server[11:13]***
- The ssh and user password are required for each server separately, but in the next tasks we will see how all hosts can share them using variables.

Task 3: Testing Ansible connectivity using the "ping" module

a. Save and Exit the hosts file (type ESC, ':', 'wq' and <enter>)

```
# storage
storage01 ansible_user=cumulus ansible_ssh_pass=Academy123

# compute
server11 ansible_user=cumulus ansible_ssh_pass=Academy123
server12 ansible_user=cumulus ansible_ssh_pass=Academy123
server13 ansible_user=cumulus ansible_ssh_pass=Academy123

# virtualization
hypervisor01 ansible_user=cumulus ansible_ssh_pass=Academy123
vm01 ansible_user=cumulus ansible_ssh_pass=Academy123
vm02 ansible_user=cumulus ansible_ssh_pass=Academy123
~
~
~
:wq
```

b. Validate the configuration by using the ping module, make sure to use the inventory file you created (use the -i symbol, followed by the hosts file path)

 # ansible -i practice1/hosts server11 -m ping

```
cumulus@oob-mgmt-server:~$ ansible -i practice1/hosts server11 -m ping
server11 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
cumulus@oob-mgmt-server:~$
```

Please note:

- You might receive a [DEPRECATION WARNING] telling you that there is a later Python version available, please ignore it.

Task 4: Add servers to a "hosts" group

a. Use VIM to edit the hosts file, and enter INSERT mode by typing 'a'

 # sudo vi practice1/hosts

b. Add all servers to a group called "hosts" (use square brackets)

```
[hosts]

# storage
storage01 ansible_user=cumulus ansible_ssh_pass=Academy123

# compute
server11 ansible_user=cumulus ansible_ssh_pass=Academy123
server12 ansible_user=cumulus ansible_ssh_pass=Academy123
server13 ansible_user=cumulus ansible_ssh_pass=Academy123

# virtualization
hypervisor01 ansible_user=cumulus ansible_ssh_pass=Academy123
vm01 ansible_user=cumulus ansible_ssh_pass=Academy123
vm02 ansible_user=cumulus ansible_ssh_pass=Academy123
```

c. Exit VIM and use the Ansible "ping" module to test the new group configuration, make sure to use the inventory file you created (use the -i symbol, followed by the hosts file path)

 # ansible -i practice1/hosts hosts -m ping

```
cumulus@oob-mgmt-server:~$ ansible -i practice1/hosts hosts -m ping

storage01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
server12 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
server13 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
hypervisor01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
server11 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
vm02 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
vm01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
```

Task 5: Add Cumulus Linux switches to the inventory

a. Use VIM to edit the hosts file, and enter INSERT mode by typing 'a'
 # sudo vi practice1/hosts

b. Add the leaf switches ('leaf01' - 'leaf06') to the inventory file, also add the necessary credentials (user and password)
 # leaf01 ansible_user=cumulus ansible_ssh_pass=Academy123

```
[hosts]
# storage
storage01 ansible_user=cumulus ansible_ssh_pass=Academy123

# compute
server[11:13] ansible_user=cumulus ansible_ssh_pass=Academy123

# virtualization
hypervisor01 ansible_user=cumulus ansible_ssh_pass=Academy123
vm01 ansible_user=cumulus ansible_ssh_pass=Academy123
vm02 ansible_user=cumulus ansible_ssh_pass=Academy123

leaf01 ansible_user=cumulus ansible_ssh_pass=Academy123
leaf02 ansible_user=cumulus ansible_ssh_pass=Academy123
leaf03 ansible_user=cumulus ansible_ssh_pass=Academy123
leaf04 ansible_user=cumulus ansible_ssh_pass=Academy123
leaf05 ansible_user=cumulus ansible_ssh_pass=Academy123
leaf06 ansible_user=cumulus ansible_ssh_pass=Academy123


~
~
~
```

c. Exit VIM and use the Ansible "ping" module to test Ansible connectivity to the switches.

```
cumulus@oob-mgmt-server:~$ ansible leaf1 -m ping
leaf01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
```

ansible -i practice1/hosts leaf01 -m ping

Please note:

- If the ansible user or password are incorrect, you will get the following error:

```
Leaf02 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: Permission denied
(publickey,password).",
    "unreachable": true}
```

Task 6: Add Cumulus Linux switches to a "switch" group

a. Use VIM to edit the hosts file, and enter INSERT mode by typing 'a'
 # sudo vi practice1/hosts

d. Add the **leaf** switches to a group called "leaves"

```
.
.
.

[leaves]
Leaf01 ansible_user=cumulus ansible_ssh_pass=Academy123
Leaf02 ansible_user=cumulus ansible_ssh_pass=Academy123
leaf03 ansible_user=cumulus ansible_ssh_pass=Academy123
leaf04 ansible_user=cumulus ansible_ssh_pass=Academy123
leaf05 ansible_user=cumulus ansible_ssh_pass=Academy123
leaf06 ansible_user=cumulus ansible_ssh_pass=Academy123
~
~
~
```

Please note:

- Instead of configuring each leaf in a different line, you can use a REGEX expression to capture all compute servers in one line ***# leaf[01:06]***.

Add the **spine** switches, same way the leaves were added.

```
.
.
.
[leaves]
leaf[01:06] ansible_user=cumulus ansible_ssh_pass=Academy123

[spines]
Spine01 ansible_user=cumulus ansible_ssh_pass=Academy123
Spine02 ansible_user=cumulus ansible_ssh_pass=Academy123

~
~
~
```

b. Add the **leaves and spines** to a group called “switches”.

```
.
.
.

[hosts]

# storage
storage01 ansible_user=cumulus ansible_ssh_pass=Academy123

# compute
server[11:13] ansible_user=cumulus ansible_ssh_pass=Academy123

# virtualization
hypervisor01 ansible_user=cumulus ansible_ssh_pass=Academy123
vm01 ansible_user=cumulus ansible_ssh_pass=Academy123
vm02 ansible_user=cumulus ansible_ssh_pass=Academy123

[leaves]
leaf[01:06] ansible_user=cumulus ansible_ssh_pass=Academy123

[spines]
spine[01:02] ansible_user=cumulus ansible_ssh_pass=Academy123


[switches:children]
spines
leaves

~
~
```

c. Exit VIM and use the Ansible "ping" module to test Ansible connectivity to the switches

ansible -i practice1/hosts switches -m ping

```
cumulus@oob-mgmt-server:~$ ansible switches -m ping
leaf01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
leaf02 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
leaf03 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
leaf04 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
leaf05 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
leaf06 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
spine02 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
spine01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
```

Task 7: Add variables to be shared by the groups

a. Add the username and password as variables, to be shared among all devices, then delete the definitions on each device.

```
.
.
.

[hosts]

# storage
storage01 ansible_user=cumulus ansible_ssh_pass=Academy123

# compute
server[11:13] ansible_user=cumulus ansible_ssh_pass=Academy123

# virtualization
hypervisor01 ansible_user=cumulus ansible_ssh_pass=Academy123
vm01 ansible_user=cumulus ansible_ssh_pass=Academy123
vm02 ansible_user=cumulus ansible_ssh_pass=Academy123

[leaves]
leaf[01:06] ansible_user=cumulus ansible_ssh_pass=Academy123

[spines]
spine[01:02] ansible_user=cumulus ansible_ssh_pass=Academy123


[switches:children]
spines
leaves

[all:vars]
ansible_user=cumulus
ansible_ssh_pass=Academy123
```

Please note:

- Different variables can be shared with different groups.
  For example, a different user can be used for the "hosts" group.

Step 1.2: Getting Started with Ansible - Playbooks

Practice objectives:

In this practice session you will create and execute a basic Ansible playbook.

- You will use the **Ansible 'copy'** module to set login messages to switches and hosts.
- You will **execute** the playbook you wrote.

Task 1: Create a new Ansible playbook

a. Access the 'oob-mgmt-server', and create a new yaml file under the 'practice1/' directory

```
# touch /practice1/labPlaybook.yaml
```

b. Use VIM or another text editor to edit the **/practice1/labPlaybook.yaml** file:
 # vi /practice1/labPlaybook.yaml
(the file should be empty).

Task 2: Edit the playbook – add tasks

a. Add a new task to the playbook, the task purpose is to check connectivity to all devices.
   - Set a name for the task
   - Apply task to all devices
   - Use the ping module to check connectivity.

```
- hosts: all
  tasks:
    - name: test connection
      ping:
```

b. Add a new task to the playbook, the task purpose is to set an informative message when login to the lab **switches**.
   - Set a name for the task
   - Apply task to switches only
   - Use the copy module to edit content of '/etc/motd'.

c. Add a new task to the playbook, the task purpose is to set an informative message when login to the lab **servers**.
   - Set a name for the task
   - Apply task to hosts only
   - Use the copy module to edit content of '/etc/motd'.

```
- hosts: all
  tasks:
    - name: test connection
      ping:

- name: change switches message of the day
  hosts: switches
  tasks:
    - name: changing switches motd
      copy:
        content: "Welcome to GTC, this is a virtual switch!"
        dest: '/etc/motd'

- name: change switches message of the day
  hosts: hosts
  tasks:
    - name: changing hosts motd
      copy:
        content: "Welcome to GTC, this is a virtual host!"
        dest: '/etc/motd'
```

Task 4: Execute the playbook

a. Access the 'oob-mgmt-server', and make sure that the host file is configured as follows
 # cat practice1/hosts

```
.
.
.

[hosts]

# storage
storage01

# compute
server[11:13]

# virtualization
hypervisor01
vm01
vm02

[leaves]
leaf[01:06]

[spines]
spine[01:02]

[switches:children]
leaves
spines

[all:vars]
ansible_user=cumulus
ansible_ssh_pass=Academy123

~
~
~
```

b. Do a dry run to check for syntax errors.
It can be done using the '--check' option.

 # sudo ansible-playbook -i practice1/hosts -b practice1/labPlaybook.yaml --check

```
 PLAY RECAP
*********************************************************************************************************************
*********
hypervisor01               : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf01                     : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf02                     : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf03                     : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf04                     : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf05                     : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf06                     : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server11                   : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server12                   : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server13                   : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
spine01                    : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
spine02                    : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
storage01                  : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
vm01                       : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
vm02                       : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

- Use the inventory (hosts) file you wrote in previous exercise; it can be done using the "-i" symbol
- Copying and editing file at the remote hosts requires sudo privileges and Ansible needs to 'become' the root when executing the remote command.
  Use the -b for that, it will make sure Ansible runs the commands with escalated privileges.
  Ansible will run a simulation of the playbook without changing anything on the remote side.

c. Execute the playbook you wrote

 # sudo ansible-playbook -i practice1/hosts -b practice1/labPlaybook.yaml

```
PLAY RECAP
*********************************************************************************************************************
*********
hypervisor01               : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf01                     : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf02                     : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf03                     : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf04                     : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf05                     : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf06                     : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server11                   : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server12                   : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server13                   : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
spine01                    : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
spine02                    : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
storage01                  : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
vm01                       : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
vm02                       : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Step 02: preparation: hypervisor, vm, switches

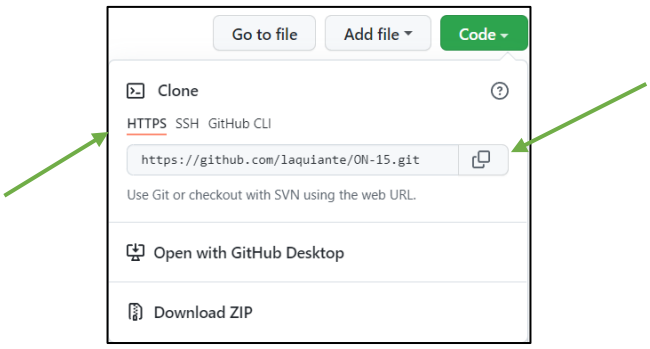
Practice objectives:

In this practice session you will get to know the workshop structure, and do a few preparations for the steps to follow.

- You will **clone** the workshop directory from a git repo.
- You will make the workshop scripts are **executable**.
- You will browse through workshop files and get used to the files structure.

Task 1: Clone the ON-15 git Repository

a. Go to https://github.com/laquiante/ON-15 and copy the web URL.



b. Access the 'oob-mgmt-server', go to the base directory (/home/cumulus) and clone the ON-15 repository.

 # git clone https://github.com/laquiante/ON-15

```
cumulus@oob-mgmt-server:~$ cd ~
cumulus@oob-mgmt-server:~$ git clone https://github.com/laquiante/ON-15
Cloning into 'ON-15'...
remote: Enumerating objects: 3966, done.
remote: Counting objects: 100% (1933/1933), done.
remote: Compressing objects: 100% (1808/1808), done.
remote: Total 3966 (delta 1014), reused 0 (delta 0), pack-reused 2033
Receiving objects: 100% (3966/3966), 9.85 MiB | 7.71 MiB/s, done.
Resolving deltas: 100% (2277/2277), done.

cumulus@oob-mgmt-server:~$ ls
```

Task 2: Make the workshop shell script executable

a. Go to the ON-15 directory you cloned, and check the shell scripts permissions.

```
cumulus@oob-mgmt-server:~$ cd ON-15
cumulus@oob-mgmt-server:~/ON-15$ ls -al
total 9076
.
.
.
-rw-rw-r--  1 cumulus cumulus     305 Nov  7 09:21  play-step-02-reference-hypervisor-vms-switches.sh
-rw-rw-r--  1 cumulus cumulus     542 Nov  7 09:21  play-step-02-student-lab.sh
-rw-rw-r--  1 cumulus cumulus    1183 Nov  7 09:21  play-step-03-reference-all-leafs-spines-compute.sh
-rw-rw-r--  1 cumulus cumulus    1436 Nov  7 09:21  play-step-03-student-lab.sh
-rw-rw-r--  1 cumulus cumulus      93 Nov  7 09:21  play-step-04-reference-all-leafs-spines-vms.sh
-rw-rw-r--  1 cumulus cumulus     341 Nov  7 09:21  play-step-04-student-lab.sh
-rw-rw-r--  1 cumulus cumulus      97 Nov  7 09:21  play-step-05a-reference-l2-all-leafs.sh
-rw-rw-r--  1 cumulus cumulus     387 Nov  7 09:21  play-step-05a-student-lab-l2.sh
-rw-rw-r--  1 cumulus cumulus     130 Nov  7 09:21  play-step-05b-reference-multi-homing-all-leafs.sh
-rw-rw-r--  1 cumulus cumulus     387 Nov  7 09:21  play-step-05b-student-lab-multi-homing.sh
-rw-rw-r--  1 cumulus cumulus     132 Nov  7 09:21  play-step-06a-reference-all-leafs.sh
-rw-rw-r--  1 cumulus cumulus     403 Nov  7 09:21  play-step-06a-student-lab-linux-classic.sh
-rw-rw-r--  1 cumulus cumulus     157 Nov  7 09:21  play-step-06b-reference-all-leafs.sh
-rw-rw-r--  1 cumulus cumulus     232 Nov  7 09:21  play-step-06c-reference-all-leafs.sh
-rw-rw-r--  1 cumulus cumulus     407 Nov  7 09:21  play-step-06c-student-lab-configure-leaf01.sh
-rw-rw-r--  1 cumulus cumulus      98 Nov  7 09:21  play-step-07-reference-roles-templates-vars.sh
-rw-rw-r--  1 cumulus cumulus      87 Nov  7 09:21  play-step-08-DC-in-5-min.sh
drwxrwxr-x  5 cumulus cumulus    4096 Nov  7 09:21  step-02
drwxrwxr-x  2 cumulus cumulus    4096 Nov  7 09:21  step-03
drwxrwxr-x  2 cumulus cumulus    4096 Nov  7 09:21  step-04
drwxrwxr-x  2 cumulus cumulus    4096 Nov  7 09:21  step-05
drwxrwxr-x  2 cumulus cumulus    4096 Nov  7 09:21  step-06
drwxrwxr-x  2 cumulus cumulus    4096 Nov  7 09:21  step-06b
drwxrwxr-x  3 cumulus cumulus    4096 Nov  7 09:21  step-06c
drwxrwxr-x  4 cumulus cumulus    4096 Nov  7 09:21  step-07
drwxrwxr-x  2 cumulus cumulus    4096 Nov  7 09:21  step-08.
cumulus@oob-mgmt-server:~/ON-15$
```

Notice that some of the files are not executable…

```
cumulus@oob-mgmt-server:~/ON-15$  ./play-step-02-student-lab.sh
-bash: ./play-step-02-student-lab.sh: Permission denied

cumulus@oob-mgmt-server:~/ON-15$
```

b. Change the permissions so shell scripts can be executed, **do not execute the shell script yet.**

```
cumulus@oob-mgmt-server:~/ON-15$ chmod 777 *.sh

cumulus@oob-mgmt-server:~/ON-15$
```

Please note:

Before the script can perform all necessary tasks, there are a few missing configurations that need to be put into the correct files. Please see next task for more information.

Task 3: Completing the script.

a. Look at the content of the step-2 student lab script.

```
cumulus@oob-mgmt-server:~/ON-15$ cat play-step-02-student-lab.sh
ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-02/prepare_hypervisor/main.yml
ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-02/prepare_vm/main.yml
################################################################################
# the following playbook "student.yml" and maybe dependent files needs work #
################################################################################
sudo ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-02/prepare_switches/student.yml
```

b. One of the files that the script uses is incomplete, find this file and edit it using VIM or other text editor you prefer.
 # vi ~/ON-15/step-02/prepare_switches/student.yml

c. Looking at the playbook, you can see that one of the files that the playbook uses is incomplete.

```
.
.
.
- hosts: switche
  name: copy topology.dot to target devices
  become: yes
  gather_facts: false
  tasks:
    - copy:
# ****************************************************
# ****** use/edit topology-student.dot            *****
# ****** this file is not complete and needs work *****
# ****************************************************
        src: /home/cumulus/ON-15/inventory/files/topology-student.dot
        dest: /etc/ptm.d/topology.dot
        mode: '0644'
        owner: 'root'
        group: 'root'
      notify: restart_ptm
.
.
.
```

PTM (topology.dot file):

The topology.dot file is being used to validate existing topology and network connections. It can also be used for other purposes and to generate a network digital twin.

d. Edit the student topology.dot file and replace the starred lines with the right connections.

```
graph "ALQ" {

# pod 01
****************** -- ***************
"storage01":"eth2" -- "leaf02":"swp1"

# pod 02
"server11":"enp97s0f0" -- "leaf03":"swp1"
"server11":"enp97s0f1" -- "leaf04":"swp1"

"server12":"enp97s0f0" -- "leaf03":"swp2"
"server12":"enp97s0f1" -- "leaf04":"swp2"

"server13":"enp97s0f0" -- "leaf03":"swp3"
"server13":"enp97s0f1" -- "leaf04":"swp3"

# pod 03
"hypervisor01":"eth1" -- "leaf05":"swp1"
"hypervisor01":"eth2" -- "leaf06":"swp1"

"hypervisor01":"eth3" -- "vm01":"eth1"
"hypervisor01":"eth4" -- "vm02":"eth1"
"hypervisor01":"eth5" -- "vm01":"eth2"
"hypervisor01":"eth6" -- "vm02":"eth2"

# leaf-spine
****************** -- ****************
"spine01":"swp124" -- "leaf02":"swp31"
"spine01":"swp125" -- "leaf03":"swp31"
"spine01":"swp126" -- "leaf04":"swp31"
"spine01":"swp127" -- "leaf05":"swp31"
"spine01":"swp128" -- "leaf06":"swp31"

****************** -- ****************
"spine02":"swp124" -- "leaf02":"swp32"
"spine02":"swp125" -- "leaf03":"swp32"
"spine02":"swp126" -- "leaf04":"swp32"
"spine02":"swp127" -- "leaf05":"swp32"
"spine02":"swp128" -- "leaf06":"swp32"

}
```

Please note:

- If you are not sure you made the right fixes, you can look at the complete topology.dot.
  file location: "~/ON-15/inventory/files/topology.dot"

Task 4: Execute the script.

a. Execute play-step-02-student-lab.sh script…

```
cumulus@oob-mgmt-server:~/ON-15$  ./play-step-02-student-lab.sh
.
.
.
PLAY RECAP
*********************************************************************************************************
**********************
leaf01                     : ok=8    changed=8    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
leaf02                     : ok=8    changed=8    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
leaf03                     : ok=8    changed=8    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
leaf04                     : ok=8    changed=8    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
leaf05                     : ok=8    changed=8    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
leaf06                     : ok=8    changed=8    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
spine01                    : ok=10   changed=9    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
spine02                    : ok=8    changed=8    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0

cumulus@oob-mgmt-server:~/ON-15$
```

The script will do some necessary preparations such as:

- Create vSwitches on the hypervisor
- Copy the if-manager config
- Apply changes
- Copy 3rd-party app configurations
- Add security keys
- Install and activate FRR (Free Range Routing) on VMs
- Disable network command line utility (NCLU) and activate the newer version – NVUE on switches.
- Copy the topology.dot file (you fixed)
- Verify out of band connectivity
- Debug and testing

Task 5: Validate the step.

a. access one of the switches in the topology, you can either click on the switch image to open its console or ssh via the oob-mgmt-server

```
cumulus@oob-mgmt-server:~/ON-15$ ssh leaf01
Linux leaf01 4.19.0-cl-1-amd64 #1 SMP Debian 4.19.176-1+cl4.4.0u1 (2021-06-25) x86_64

Welcome to NVIDIA Cumulus VX (TM)

NVIDIA Cumulus VX (TM) is a community supported virtual appliance designed
for experiencing, testing and prototyping NVIDIA Cumulus' latest technology.
For any questions or technical support, visit our community site at:
http://community.cumulusnetworks.com

The registered trademark Linux (R) is used pursuant to a sublicense from LMI,
the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide
basis.
Last login: Mon Nov  1 18:42:21 2021
cumulus@leaf01:mgmt:~$
```

b. make sure *NVUE* was installed correctly by running an *NVUE* command
 # nv show platform hardware --operational

```
cumulus@oob-mgmt-server:~/ON-15$ ssh leaf01
cumulus@leaf01:mgmt:~$ nv show platform hardware --operational

            operational        description
----------  -----------------  ----------------------------------------
model       vx                 The platform's model identifier
system-mac  44:38:39:00:00:4a  The MAC provided by eeprom for system-mac
vendor      cumulus            The platform's vendor
```

c. make sure the BPG daemon is enabled in the FRR daemons file (Free Range Routing) daemons are enabled

#sudo cat /etc/frr/daemons

```
cumulus@leaf01:mgmt:~$ sudo cat /etc/frr/daemons
.
.
.
bgpd=yes
ospfd=no
ospf6d=no
ripd=no
ripngd=no
isisd=no
fabricd=no
pimd=no
ldpd=no
nhrpd=no
eigrpd=no
babeld=no
sharpd=no
pbrd=no
fabricd=no
vrrpd=no
```

Step 03: interfaces layer2, vlans

Task 1: Completing the student script.

a. Take a look at the content of the step-3 student lab script.
cat ~/ON-15/ play-step-03-student-lab.sh

```
cumulus@oob-mgmt-server:~/ON-15$ cat ~/ON-15/play-step-03-student-lab.sh
ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-03/spine01
ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-03/spine02

################################################################################
# the following playbook "leaf01-student" and maybe dependent files needs work #
################################################################################
ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-03/leaf01-student

sudo ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-03/leaf02
sudo ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-03/leaf03
sudo ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-03/leaf04
sudo ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-03/leaf05
sudo ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-03/leaf06
sudo ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-03/storage01
sudo ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-03/server11
sudo ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-03/server12
sudo ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-03/server13
sudo ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-03/vm01
sudo ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-03/vm02
```

b. One of the files that the script uses is incomplete, in this case – leaf01 configuration playbook.
find this playbook and edit it using VIM or other text editor you prefer.
vi ~/ON-15/step-03/leaf01-student

c. Edit the student leaf01 configuration playbook and replace the starred lines with the right connections.

```
---
- hosts: leaf01
  name: create bridge, set loopback and enable switch ports
  become: yes
  gather_facts: no
  tasks:
    - name: nvue set items
      shell: ** *** {{ item }}
      with_items:
      - interface ** ** ******* **************
      - interface **********
      - bridge ****** **** vlan *****
      - interface **** ****** ****** **** ****** **
      - platform ******** value ******

    - name: activate staging buffer
      shell: nv config ***** -y
    - name: iproute2 bridge interface list
      shell: bridge link
      register: br
    - debug: msg={{ br.stdout }}
    - name: iproute2 bridge forwarding database
      shell: bridge fdb
      register: fdb
    - debug: msg={{ fdb.stdout }}
```

Please note:

- If you are not sure you made the right fixes, you can look at the complete leaf01 file. file location: "***~/ON-15/step-03/leaf01***"

Task 2: Execute the script.

a. Execute play-step-03-student-lab.sh script (after the fixes).

```
cumulus@oob-mgmt-server:~/ON-15$ ./play-step-03-student-lab.sh

PLAY [set loopback and enable switch ports]
*********************************************************************************************************************
*****************************************************************

TASK [set interface loopback IPv4]
*********************************************************************************************************************
*************************************************************************
[WARNING]: Platform linux on host spine01 is using the discovered Python interpreter at /usr/bin/python, but future
installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
changed: [spine01]

TASK [enable switchports]
*********************************************************************************************************************
*************************************************************************
changed: [spine01]

TASK [set hostname]
*********************************************************************************************************************
*************************************************************************
changed: [spine01]

TASK [activate staging buffer]
*********************************************************************************************************************
*************************************************************************
changed: [spine01]

TASK [iproute2 interface list]
*********************************************************************************************************************
*************************************************************************
changed: [spine01]

TASK [debug] ******************************************************************************.
.
.
.
```

The script will perform L2 configurations on the switches using NVUE commands. The servers will be configured using standard Linux shell commands.

Task 3: Validate the script.

a. Access leaf01 and make sure that the interface loopback *lo* uses the correct IPv4 address

 # ip -brief -4 addr

```
cumulus@leaf01:mgmt:~$ ip -br -4 a
lo               UNKNOWN        127.0.0.1/8 192.168.0.1/32
eth0             UP             192.168.200.9/24
```

b. Access leaf01 and make sure that the newly configured bridge *br_A* has learned the MAC address of storage01.

Storage 01 MAC: 44:38:39:00:00:01

 # bridge fdb

```
cumulus@leaf01:mgmt:~$ bridge fdb
44:38:39:00:00:01 dev swp1 vlan 10 master br_A
44:38:39:00:00:02 dev swp1 master br_A permanent
44:38:39:00:00:02 dev swp1 self permanent
```

c. Verify that leaf01 shows lldp connectivity to spine01 and spine02.

 # sudo ptmctl

```
cumulus@leaf01:mgmt:~$ sudo ptmctl
----------------------------------------
port   cbl     BFD     BFD   BFD    BFD
       status  status  peer  local  type
----------------------------------------
swp1   fail    N/A     N/A   N/A    N/A
swp31  pass    N/A     N/A   N/A    N/A
swp32  pass    N/A     N/A   N/A    N/A
```

Please Note:

(Storage01 LLDP uses an unexpected value and will show PTM fail).
The ptm (topology.dot) file contains a fix for reference.
Alternatively, you can run the reference playbook that uses the fix.

```
cumulus@leaf01:mgmt:~$ sudo ptmctl
----------------------------------------
port cbl BFD BFD BFD BFD
status status peer local type
----------------------------------------
swp1 pass N/A N/A N/A N/A
swp31 pass N/A N/A N/A N/A
swp32 pass N/A N/A N/A N/A
```

Step 04: interfaces layer3, bgp

Task 1: Completing the student script.

a. Take a look at the content of the step-4 student lab script.
cat ~/ON-15/ play-step-04-student-lab.sh

```
cumulus@oob-mgmt-server:~/ON-15$ cat play-step-04-student-lab.sh
#############################################################################################
# the following playbook "step-04-student" and maybe dependent files needs work #
#############################################################################################
sudo ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-04/step-04-student
```

b. One of the files that the script uses is incomplete, in this case – a playbook. find this playbook and which files should be edited.
cat ~/ON-15/step-04/ step-04-student
two files that the script uses are incomplete:

- leaf01 interface configuration
  file location: **/home/cumulus/ON-15/step-04/leaf01-student-if**
- leaf01 L3 FRR configuration
  file location: **/home/cumulus/ON-15/step-04/leaf01-student-frr**

```
cumulus@oob-mgmt-server:~/ON-15/step-04$ cat step-04-student
# ************************************************************
# *****             student working area below              *****
# ************************************************************

# Leaf01

- hosts: leaf01
  name: Leaf01 interfaces
  become: yes
  gather_facts: no
  tasks:
    - name: copy eni
      copy:
# ****************************************
# ***** use/edit file leaf01-student-if *****
# ****************************************
        src: /home/cumulus/ON-15/step-04/leaf01-student-if
        dest: /etc/network/interfaces
    - name: activate changes on Leaf1
      shell: /sbin/ifreload -a
    - name: fix daemons to be on the safe side
      copy:
        src: /home/cumulus/ON-15/inventory/files/daemons
        dest: /etc/frr/daemons
    - name: restart frr
      ansible.builtin.shell: systemctl restart frr
    - name: copy frr
      copy:
# ****************************************
# ***** use/edit file leaf01-student-frr *****
# ****************************************
        src: /home/cumulus/ON-15/step-04/leaf01-student-frr
        dest: /etc/frr/frr.conf
    - name: reload frr
      ansible.builtin.shell: systemctl reload frr

# ************************************************************
# *****             Don't change anything below             *****
# ************************************************************
```

c. Edit the student leaf01 interface configuration file and replace the starred lines with the right parameters.
vi ~/ON-15/step-04/leaf01-student-if

```
hostname leaf01
log syslog informational
service integrated-vtysh-config
!
router bgp **********
 bgp router-id 192.168.0.1
 neighbor ***** interface remote-as external
 neighbor ***** interface remote-as external
 !
 address-family **** *******
  network **************
 exit-address-family
!
 address-family l2vpn evpn
  neighbor swp31 activate
  neighbor swp32 activate
  advertise-all-vni
 exit-address-family
!
line vty
!
```

d. Edit the student leaf01 frr configuration file and replace the starred lines with the right parameters.
vi ~/ON-15/step-04/leaf01-student-frr

```
hostname leaf01
log syslog informational
service integrated-vtysh-config
!
router bgp **********
 bgp router-id 192.168.0.1
 neighbor ***** interface remote-as external
 neighbor ***** interface remote-as external
 !
 address-family **** *******
  network **************
 exit-address-family
!
 address-family l2vpn evpn
  neighbor swp31 activate
  neighbor swp32 activate
  advertise-all-vni
 exit-address-family
!
line vty
!
```

Please note:

- If you are not sure you made the right fixes, you can look at the complete files.
  file location: "***~/ON-15/step-04/leaf01-if***
  file location: "***~/ON-15/step-04/leaf01-frr***

Task 2: Execute the script.

a. Execute play-step-04-student-lab.sh script (after the fixes).

```
cumulus@oob-mgmt-server:~/ON-15/step-04$ ./play-step-04-student-lab.sh
.
.
.

PLAY RECAP
*********************************************************************************************************************************
****
leaf01                     : ok=6    changed=6    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf02                     : ok=6    changed=6    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf03                     : ok=6    changed=6    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf04                     : ok=6    changed=6    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf05                     : ok=6    changed=6    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf06                     : ok=6    changed=6    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
spine01                    : ok=6    changed=6    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
spine02                    : ok=6    changed=6    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
vm01                       : ok=6    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
vm02                       : ok=6    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

The script will perform L3 configurations on the switches using NVUE commands. The VMs will be configured using standard Linux shell commands.

Task 3: Validate the script.

a. Access leaf01 and verify the routing table for VRF *Teanant_A*
 # ip route show vrf Tenant_A

```
cumulus@leaf01:mgmt:~$ ip route show vrf Tenant_A
unreachable default metric 4278198272
10.0.10.0/24 dev vlan10 proto kernel scope link src 10.0.10.240
10.0.10.0/24 dev vlan10-v0 proto kernel scope link src 10.0.10.254 metric 1024
```

b. Access storage01 and check that `SVI and VRR addresses of leaf01 can be reached`

```
cumulus@storage01:~$ ping -c 3 10.0.10.240
PING 10.0.10.240 (10.0.10.240) 56(84) bytes of data.
64 bytes from 10.0.10.240: icmp_seq=1 ttl=64 time=0.295 ms
64 bytes from 10.0.10.240: icmp_seq=2 ttl=64 time=0.289 ms
64 bytes from 10.0.10.240: icmp_seq=3 ttl=64 time=0.358 ms
--- 10.0.10.240 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2038ms
rtt min/avg/max/mdev = 0.289/0.314/0.358/0.031 ms
```

c. Verify on leaf01 the "main" or "default" routing table to show the remote loopback addresses.
 # ip route show

```
cumulus@leaf01:mgmt:~$ ip route show
192.168.0.2 nhid 35 proto bgp metric 20
192.168.0.3 nhid 32 proto bgp metric 20
192.168.0.4 nhid 35 proto bgp metric 20
192.168.0.5 nhid 32 proto bgp metric 20
192.168.0.6 nhid 35 proto bgp metric 20
192.168.0.201 nhid 32 proto bgp metric 20
192.168.0.202 nhid 35 proto bgp metric 20
```

Step 05: evpn Layer 2 Layer 2 + Multi-Homing

Task 1: Completing the l2 student script.

a. Take a look at the content of the step-5a student lab script.
cat ~/ON-15/play-step-05a-student-lab.sh

```
cumulus@oob-mgmt-server:~/ON-15$ cat play-step-05a-student-lab-l2.sh
#############################################################################################
# the following playbook "step-05a-student-l2-classic" and maybe dependent files needs work #
#############################################################################################
sudo ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-05/step-05a-student-l2-classic
```

b. One of the files that the script uses is incomplete, in this case – a playbook. find this playbook and which files should be edited.
cat ~/ON-15/step-05/step-05a-student-l2

```
cumulus@oob-mgmt-server:~/ON-15/step-05$ cat step-05a-student-l2
# Leaf01

# ***************************************************************
# *****             student working area below              *****
# ***************************************************************

- hosts: leaf01
  name: Leaf01 interfaces and routing
  become: yes
  gather_facts: no
  tasks:
    - name: copy eni
      copy:
# ****************************************
# ***** use/edit file leaf01-student-if *****
# ****************************************
        src: /home/cumulus/ON-15/step-05/leaf01-student-if
        dest: /etc/network/interfaces
    - name: activate changes on leaf01
      shell: /sbin/ifreload -a

# ***************************************************************
# *****             Don't change anything below             *****
# ***************************************************************

.
.
.
```

c. Edit the student leaf01 interface configuration file and replace the starred lines with the right parameters.
vi ~/ON-15/step-05/leaf01-student-if

```
auto lo
iface lo inet loopback
        address 192.168.0.1/32

auto mgmt
iface mgmt
        address 127.0.0.1/8
        address ::1/128
        vrf-table auto

auto Tenant_A
iface Tenant_A
        vrf-table auto

auto Tenant_B
iface Tenant_B
        vrf-table auto

auto eth0
iface eth0 inet dhcp
        ip-forward off
        ip6-forward off
        vrf mgmt

auto swp1
iface swp1
        bridge-access 10

auto swp31
iface swp31

auto swp32
iface swp32

auto br_A
iface br_A
        bridge-ports swp1 *******
        bridge-vids 10 20
        bridge-vlan-aware yes

auto vlan10
iface vlan10
        address 10.0.10.240/24
        address-virtual 00:00:5e:00:00:01 10.0.10.254/24
        vlan-id 10
        vlan-raw-device br_A
        vrf Tenant_A

auto vxlan10
iface vxlan10
    bridge-access 10
    bridge-arp-nd-suppress on
    bridge-learning off
    mstpctl-bpduguard yes
    mstpctl-portbpdufilter yes
    vxlan-local-tunnelip ***********
    vxlan-id **
```

Please note:
If you are not sure you made the right fixes, you can look at the complete file.
file location: "***~/ON-15/step-05/leaf01-if***

Task 2: Execute the script.

a. Execute play-step-05-student-lab.sh script (after the fixes).

```
cumulus@oob-mgmt-server:~/ON-15$ ./play-step-05a-student-lab-l2.sh
.
.
.

PLAY RECAP
*********************************************************************************************************************************
****
leaf01                     : ok=2    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf02                     : ok=2    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf03                     : ok=2    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf04                     : ok=2    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf05                     : ok=2    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf06                     : ok=2    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

The script will perform L2 configurations on the switches using NVUE commands.

Task 3: Validate the script.

a. Access leaf01, and make sure the remote MAC address of server11 is externally_learned

```
cumulus@leaf01:mgmt:~$ bridge fdb
44:38:39:00:00:02 dev swp1 vlan 10 master br_A permanent
44:38:39:00:00:01 dev swp1 vlan 10 master br_A
44:38:39:00:00:02 dev swp1 master br_A permanent
00:00:5e:00:00:01 dev br_A self permanent
00:00:5e:00:00:01 dev br_A vlan 10 master br_A permanent
00:00:5e:00:00:01 dev vlan10 self permanent
44:38:39:00:00:07 dev vxlan10 vlan 10 extern_learn master br_A
44:38:39:00:00:05 dev vxlan10 vlan 10 extern_learn master br_A
44:38:39:00:00:03 dev vxlan10 vlan 10 extern_learn master br_A
c6:a0:0c:28:ce:02 dev vxlan10 master br_A permanent
00:00:00:00:00:00 dev vxlan10 dst 192.168.0.2 self permanent
00:00:00:00:00:00 dev vxlan10 dst 192.168.0.3 self permanent
00:00:00:00:00:00 dev vxlan10 dst 192.168.0.4 self permanent
00:00:00:00:00:00 dev vxlan10 dst 192.168.0.5 self permanent
00:00:00:00:00:00 dev vxlan10 dst 192.168.0.6 self permanent
44:38:39:00:00:07 dev vxlan10 dst 192.168.0.4 self extern_learn
44:38:39:00:00:05 dev vxlan10 dst 192.168.0.3 self extern_learn
44:38:39:00:00:03 dev vxlan10 dst 192.168.0.2 self extern_learn
```

```
cumulus@server11:~$ ip l
<SNIP>
5: uplink: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 9216 qdisc noqueue state UP mode DEFAULT group default qlen 1000
    link/ether 44:38:39:00:00:05 brd ff:ff:ff:ff:ff:ff
```

```
cumulus@storage01:~$ ip l
<SNIP>
5: uplink: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 9216 qdisc noqueue state UP mode DEFAULT group default qlen 1000
    link/ether 44:38:39:00:00:01 brd ff:ff:ff:ff:ff:ff
```

```
cumulus@leaf01:mgmt:~$ ip l
<SNIP>
3: swp1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9216 qdisc pfifo_fast master br_A state UP mode DEFAULT group default qlen 1000
    link/ether 44:38:39:00:00:02 brd ff:ff:ff:ff:ff:ff
```

b. Access leaf01 and use *vtysh* to check that the "l2vpn evpn send and received number of NLRI" is larger than 0.

 # sudo vtysh

 # leaf01# show bgp l2vpn evpn summary

```
cumulus@leaf01:mgmt:~$ sudo vtysh

Hello, this is FRRouting (version 7.5+cl4.4.0u4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

leaf01#

leaf01# show bgp l2vpn evpn summary
BGP router identifier 192.168.0.1, local AS number 4200000001 vrf-id 0
BGP table version 0
RIB entries 11, using 2200 bytes of memory
Peers 2, using 46 KiB of memory
Neighbor        V         AS   MsgRcvd   MsgSent   TblVer  InQ OutQ  Up/Down State/PfxRcd   PfxSnt
spine01(swp31)  4 4200000201      1308      1305        0    0    0 01:03:23           10       14
spine02(swp32)  4 4200000202      1303      1300        0    0    0 01:03:18           10       14
Total number of neighbors 2
```

> Please note:
> VTYSH is a shell for FRR daemons, for more information
> http://docs.frrouting.org/projects/dev-guide/en/latest/vtysh.html

c. Access storage01 can reach server11.

 # ip addr

```
cumulus@storage01:~$ ip a
<SNIP>
5: uplink: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 9216 /
                            qdisc noqueue state UP group default qlen 1000
    link/ether 44:38:39:00:00:01 brd ff:ff:ff:ff:ff:ff
    inet 10.0.10.1/24 scope global uplink
       valid_lft forever preferred_lft forever
    inet6 fe80::4638:39ff:fe00:1/64 scope link
       valid_lft forever preferred_lft forever

cumulus@storage01:~$ ping 10.0.10.11
PING 10.0.10.11 (10.0.10.11) 56(84) bytes of data.
64 bytes from 10.0.10.11: icmp_seq=1 ttl=64 time=2.38 ms
64 bytes from 10.0.10.11: icmp_seq=2 ttl=64 time=1.06 ms
64 bytes from 10.0.10.11: icmp_seq=3 ttl=64 time=1.19 ms
^C
--- 10.0.10.11 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.066/1.549/2.388/0.595 ms
```

Task 4: Completing the L3 student script.

a. Take a look at the content of the step-5b student lab script.
cat ~/ON-15/play-step-05b-student-lab-multi-homing.sh

```
cumulus@oob-mgmt-server:~/ON-15$ cat play-step-05b-student-lab-multi-homing.sh
#################################################################################################
# the following playbook "step-05b-mh-student-classic" and maybe dependent files needs work #
#################################################################################################
sudo ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-05/step-05b-mh-student-classic
```

b. One of the files that the script uses is incomplete, in this case – a playbook. find this playbook and which files should be edited, complete the starred lines.
vi ~/ON-15/step-05/step-05b-mh-student-linux-classic

```
cumulus@oob-mgmt-server:~/ON-15/step-05$ cat step-05b-mh-student-linux-classic
# Leaf01

# ****************************************************************
# *****              student working area below              *****
# ****************************************************************

# Leaf01

- hosts: ******
  name: Leaf01 interfaces and routing
  become: yes
  gather_facts: no
  tasks:
    - name: copy eni
      copy:
# ****************************************
# *** use/edit file leaf01-student-if-mh ****
# ****************************************
        src: /home/cumulus/ON-15/step-05/leaf01-student-if-mh
        dest: /etc/network/interfaces
    - name: activate changes on leaf01
      shell: /sbin/******** -a

    - name: copy frr
      copy:
# ****************************************
# ** use/edit file leaf01-student-frr-mh ****
# ****************************************
        src: /home/cumulus/ON-15/step-05/leaf01-student-frr-mh
        dest: /etc/frr/frr.conf
    - name: reload frr
      ansible.builtin.shell: ********* reload frr

# ****************************************************************
# *****              Don't change anything below             *****
# ****************************************************************
.
.
.
```

c. Edit the student leaf01 mh interface configuration file and replace the starred line with the right parameters.
vi ~/ON-15/step-05/leaf01-student-if-mh

```
auto lo
iface lo inet loopback
        address 192.168.0.1/32

auto mgmt
iface mgmt
        address 127.0.0.1/8
        address ::1/128
        vrf-table auto

auto Tenant_A
iface Tenant_A
        vrf-table auto

auto Tenant_B
iface Tenant_B
        vrf-table auto

auto eth0
iface eth0 inet dhcp
        ip-forward off
        ip6-forward off
        vrf mgmt

auto swp1
iface swp1

auto bond1
iface bond1
    bridge-access 10
    bond-slaves swp1
    ********** **********
    bond-lacp-bypass-allow yes
    mstpctl-bpduguard yes
    mstpctl-portadminedge yes

auto swp31
iface swp31

auto swp32
iface swp32

auto br_A
iface br_A
        bridge-ports bond1 vxlan10
        bridge-vids 10 20
        bridge-vlan-aware yes

auto vlan10
iface vlan10
        address 10.0.10.240/24
        address-virtual 00:00:5e:00:00:01 10.0.10.254/24
        vlan-id 10
        vlan-raw-device br_A
        vrf Tenant_A

auto vxlan10
iface vxlan10
    bridge-access 10
    bridge-arp-nd-suppress on
    bridge-learning off
    mstpctl-bpduguard yes
    mstpctl-portbpdufilter yes
    vxlan-local-tunnelip 192.168.0.1
    vxlan-id 10
```

d. Edit the student leaf01 mh frr configuration file and replace the starred lines with the right parameters.
vi ~/ON-15/step-04/leaf01-student-frr-mh

```
frr defaults datacenter
hostname leaf01
log syslog informational
zebra nexthop proto only
service integrated-vtysh-config
!
interface swp31
  **** ** ******
!
interface swp32
  **** ** ******
!
interface bond1
  evpn mh es-df-pref 50000
  evpn mh es-id 1
  evpn mh es-sys-mac 44:38:39:FF:FF:01
!
router bgp 4200000001
 bgp router-id 192.168.0.1
 bgp bestpath as-path multipath-relax
 neighbor swp31 interface remote-as external
 neighbor swp32 interface remote-as external
 !
 address-family ipv4 unicast
  network 192.168.0.1/32
 exit-address-family
!
 address-family l2vpn evpn
  neighbor swp31 activate
  neighbor swp32 activate
  advertise-all-vni
 exit-address-family
!
line vty
!
```

Please note:
- If you are not sure you made the right fixes, you can look at the complete files.
  file location: "***~/ON-15/step-05/leaf01-if-mh***
  file location: "***~/ON-15/step-05/leaf01-frr-mh***

Task 5: Execute the script.

a. Execute play-step-05-student-lab.sh script (after the fixes).

```
cumulus@oob-mgmt-server:~/ON-15$ ./play-step-05b-student-lab-multi-homing.sh
.
.
.

PLAY RECAP *********************************************************************************************************************************
leaf01                     : ok=4    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf02                     : ok=4    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf03                     : ok=4    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf04                     : ok=4    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf05                     : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf06                     : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Task 6: Validate the script.

a. Access leaf01 and make sure the storage and compute nodes are “dual-homed” via Layer 2.

```
cumulus@leaf01:mgmt:~$ bridge fdb
00:00:5e:00:00:01 dev br_A self permanent
44:38:39:00:00:02 dev br_A vlan 10 master br_A permanent
00:00:5e:00:00:01 dev br_A vlan 10 master br_A permanent
00:00:5e:00:00:01 dev vlan10 self permanent
44:38:39:00:00:07 dev vxlan10 vlan 10 extern_learn master br_A
44:38:39:00:00:05 dev vxlan10 vlan 10 extern_learn master br_A
44:38:39:00:00:08 dev vxlan10 vlan 10 extern_learn master br_A
44:38:39:00:00:06 dev vxlan10 vlan 10 extern_learn master br_A
44:38:39:00:00:04 dev vxlan10 vlan 10 extern_learn master br_A
4e:13:2a:8f:21:73 dev vxlan10 vlan 10 master br_A permanent
44:38:39:00:00:16 dev vxlan10 vlan 10 extern_learn master br_A
44:38:39:00:00:1a dev vxlan10 vlan 10 extern_learn master br_A
4e:13:2a:8f:21:73 dev vxlan10 master br_A permanent
00:00:00:00:00:00 dev vxlan10 dst 192.168.0.2 self permanent
00:00:00:00:00:00 dev vxlan10 dst 192.168.0.3 self permanent
00:00:00:00:00:00 dev vxlan10 dst 192.168.0.4 self permanent
00:00:00:00:00:00 dev vxlan10 dst 192.168.0.5 self permanent
00:00:00:00:00:00 dev vxlan10 dst 192.168.0.6 self permanent
44:38:39:00:00:08 dev vxlan10 dst 192.168.0.4 self extern_learn
44:38:39:00:00:07 dev vxlan10 nhid 536870915 self extern_learn
44:38:39:00:00:06 dev vxlan10 dst 192.168.0.3 self extern_learn
44:38:39:00:00:05 dev vxlan10 nhid 536870915 self extern_learn
44:38:39:00:00:04 dev vxlan10 dst 192.168.0.2 self extern_learn
44:38:39:00:00:1a dev vxlan10 dst 192.168.0.6 self extern_learn
44:38:39:00:00:16 dev vxlan10 dst 192.168.0.5 self extern_learn
44:38:39:00:00:01 dev bond1 vlan 10 master br_A static
44:38:39:00:00:03 dev bond1 vlan 10 master br_A static
44:38:39:00:00:02 dev bond1 master br_A permanent
```

Step 06: evpn layer 3 and Ansible features

Task 1: Completing the student script.

a. Take a look at the content of the step-6a student lab script.
cat ~/ON-15/ step-06a-student-lab-linux-classic.sh

```
cumulus@oob-mgmt-server:~/ON-15$ cat play-step-06a-student-lab-linux-classic.sh

#############################################################################################################
# the following playbook "step-06a-student-classic.yaml" and maybe dependent files needs work #
#############################################################################################################
sudo ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-06/step-06a-student-linux-classic.yaml
```

b. One of the files that the script uses is incomplete, in this case – a playbook. find this playbook and which files should be edited, complete the starred lines.
cat ~/ON-15/step-06/ step-06a-student-linux-classic.yaml

```
cumulus@oob-mgmt-server:~/ON-15/step-05$ cat step-05a-student-l2
# Leaf01

# ***************************************************************
# *****              student working area below               *****
# ***************************************************************

# Leaf01

- hosts: leaf01
  name: Leaf01 interfaces and routing
  become: yes
  gather_facts: no
  tasks:
    - name: copy eni
      ****:
# ****************************************
# ***** use/edit file leaf01-student-if *****
# ****************************************
        src: /home/cumulus/ON-15/step-06/leaf01-student-if
        dest: /etc/network/interfaces
    - name: activate changes on leaf01
      shell: /sbin/******** -a

    - name: copy frr
      ****:
# ****************************************
# ***** use/edit file leaf01-student-frr ****
# ****************************************
        src: /home/cumulus/ON-15/step-06/leaf01-student-frr
        dest: /***/***/********
    - name: reload frr
      ansible.builtin.shell: systemctl ****** frr

# ***************************************************************
# *****              Don't change anything below              *****
# ***************************************************************
.
.
.
```

c. Edit the student leaf01 interface configuration file and replace the starred line with the right parameters.
vi ~/ON-15/step-06/leaf01-student-if

```
auto lo
iface lo inet loopback
        address 192.168.0.1/32
        vxlan-local-tunnelip 192.168.0.1

auto mgmt
iface mgmt
        address 127.0.0.1/8
        address ::1/128
        vrf-table auto

auto Tenant_A
iface Tenant_A
        vrf-table auto

auto Tenant_B
iface Tenant_B
        vrf-table auto

auto eth0
iface eth0 inet dhcp
        ip-forward off
        ip6-forward off
        vrf mgmt

auto swp1
iface swp1

auto bond1
iface bond1
    bridge-access 10
    bond-slaves swp1
    es-sys-mac 44:38:39:FF:FF:01
    bond-lacp-bypass-allow yes
    mstpctl-bpduguard yes
    mstpctl-portadminedge yes

auto swp31
iface swp31

auto swp32
iface swp32

auto br_A
iface br_A
        bridge-ports bond1 vxlan10 l3_vni_A
        bridge-vids 10 20
        bridge-vlan-aware yes

auto vlan10
iface vlan10
        address 10.0.10.240/24
        address-virtual 00:00:5e:00:00:01 10.0.10.254/24
        vlan-id 10
        vlan-raw-device br_A
        vrf Tenant_A

auto vxlan10
iface vxlan10
    bridge-access 10
    bridge-arp-nd-suppress off
    bridge-learning off
    mstpctl-bpduguard yes
    mstpctl-portbpdufilter yes
    vxlan-local-tunnelip 192.168.0.1
    vxlan-id 10

auto l3_vni_A
iface l3_vni_A
    bridge-access 1001
    bridge-arp-nd-suppress off
    bridge-learning off
    mstpctl-bpduguard yes
    mstpctl-portbpdufilter yes
    vxlan-id *****

auto l3_svi
iface l3_svi
    hwaddress 44:38:39:01:01:01
    ******* ****
    *************** ****
    *** ********
```

d. Edit the student leaf01 frr configuration file and replace the starred lines with the right parameters.
vi ~/ON-15/step-06/leaf01-student-frr

```
frr defaults datacenter
hostname leaf01
log syslog informational
zebra nexthop proto only
service integrated-vtysh-config
!
vrf Tenant_A
 vni *****
 exit-vrf
!
interface swp31
  evpn mh uplink
!
interface swp32
  evpn mh uplink
!
interface bond1
  evpn mh es-df-pref 50000
  evpn mh es-id 1
  evpn mh es-sys-mac 44:38:39:FF:FF:01
!
router bgp 4200000001
 bgp router-id 192.168.0.1
 bgp bestpath as-path multipath-relax
 neighbor swp31 ********* remote-as external
 neighbor swp32 ********* remote-as external
 !
 address-family ipv4 unicast
  network 192.168.0.1/32
 exit-address-family
!
 address-family l2vpn evpn
  ******** ***** ********
  ******** ***** ********
  advertise-all-vni
 exit-address-family
!
line vty
!
```

Please note:

- If you are not sure you made the right fixes, you can look at the complete files.
  file location: "***~/ON-15/step-06/leaf01-if-mh***
  file location: "***~/ON-15/step-06/leaf01-frr-mh***

Task 2: Execute the script.

e. Execute play-step-06-student-lab.sh script (after the fixes).

```
cumulus@oob-mgmt-server:~/ON-15$ ./play-step-05b-student-lab-multi-homing.sh
.
.
.

PLAY RECAP *********************************************************************************************************************************************
leaf01                     : ok=4    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf02                     : ok=4    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf03                     : ok=4    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf04                     : ok=4    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf05                     : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
leaf06                     : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Task 3: Validate the script.

d. Access leaf01 and make sure there are two VNIs – one for L2 and one for L3.
 # sh bgp l2vpn evpn vni

```
leaf01# sh bgp l2vpn evpn vni
Advertise Gateway Macip: Disabled
Advertise SVI Macip: Disabled
Advertise All VNI flag: Enabled
BUM flooding: Head-end replication
Number of L2 VNIs: 1
Number of L3 VNIs: 1
Flags: * - Kernel
  VNI        Type RD                    Import RT                Export RT                Tenant VRF
* 10         L2   192.168.0.1:2         59905:10                 59905:10                 Tenant_A
* 10010      L3   10.0.10.254:4         59905:10010              59905:10010              Tenant_A
```

e. Access storage01 and make sure it can reach server13

Task 4: Optimizing the playbook

a. Take a look at the content of the step-6c student lab script.
cat ~/ON-15/play-step-06c-student-lab.sh

```
cumulus@oob-mgmt-server:~/ON-15$ cat play-step-06c-student-lab-configure-leaf01.sh
##############################################################################################
# the following playbook "step-06c-student-linux-classic" and maybe dependent files needs work #
##############################################################################################
sudo ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-06c/step-06c-student-linux-classic.yaml
```

b. One of the files that the script uses is incomplete, in this case – a playbook. find this playbook and which files should be edited, complete the starred lines.
cat ~/ON-15/step-06/step-06c-student-linux-classic.yaml

```
# ****************************************************************
# *****             student working area below              *****
# ****************************************************************

- hosts: leaf01
  name: task-step
  become: yes
  gather_facts: no

  tasks:
    - name: render file via template
      template:

# ********************************************************************
# ***** your working area file: leaf01-if.j2 and following files *****
# ********************************************************************
        src: /home/cumulus/ON-15/step-06c/templates/leaf01-if.j2
        dest: /etc/network/interfaces
    - name: kick it
      ansible.builtin.shell: /sbin/ifreload -a

    - name: copy frr
      copy:
        src: /home/cumulus/ON-15/step-06c/leaf01-frr
        dest: /etc/frr/frr.conf
    - name: reload frr
      ansible.builtin.shell: systemctl reload frr

# ****************************************************************
# *****             Don't change anything below             *****
# ****************************************************************

- name: Include the play for all other switches than leaf01
  import_playbook: /home/cumulus/ON-15/step-06c/step-06c-all-but-leaf-
01.yaml
```

c. Edit the student leaf01-if.j2 file and replace the starred lines with the right parameters.
vi ~/ON-15/step-06c/templates/leaf01-if.j2

```
{% set ip_lo = "************" %}
{% set *******vlan = "**" %}
{% *** ****** = "*********" %}

auto lo
iface lo inet loopback
    address {{ ip_lo }}

auto mgmt
iface mgmt
        address 127.0.0.1/8
        address ::1/128
        vrf-table auto

auto {{ ****** }}
iface {{ ****** }}
        vrf-table auto

auto Tenant_B
iface Tenant_B
        vrf-table auto

auto eth0
iface eth0 inet dhcp
        ip-forward off
        ip6-forward off
        vrf mgmt

auto swp1
iface swp1

auto bond1
iface bond1
    bridge-access {{ access_vlan }}
    bond-slaves swp1
    es-sys-mac 44:38:39:FF:FF:01
    bond-lacp-bypass-allow yes
    mstpctl-bpduguard yes
    mstpctl-portadminedge yes

auto swp31
iface swp31

auto swp32
iface swp32

auto br_A
iface br_A
        bridge-ports bond1 vxlan10 l3_vni_A
        bridge-vids {{ access_vlan }} 20
        bridge-vlan-aware yes

auto vlan10
iface vlan10
        address 10.0.10.240/24
        address-virtual 00:00:5e:00:00:01 10.0.10.254/24
        vlan-id {{ ********** }}
        vlan-raw-device br_A
        *** ************

auto vxlan10
iface vxlan10
    bridge-access {{ access_vlan }}
    bridge-arp-nd-suppress off
    bridge-learning off
    mstpctl-bpduguard yes
    mstpctl-portbpdufilter yes
    vxlan-local-tunnelip 192.168.0.1
    vxlan-id 10

auto l3_vni_A
iface l3_vni_A
    bridge-access 1001
    bridge-arp-nd-suppress off
    bridge-learning off
    mstpctl-bpduguard yes
    mstpctl-portbpdufilter yes
    vxlan-id 10010

auto l3_svi
iface l3_svi
    hwaddress 44:38:39:01:01:01
    vlan-id 1001
    vlan-raw-device br_A
    vrf {{ ****** ** 
```

If you are not sure you made the right fixes, you can look at the complete file. file location: "***~/ON-15/step-06c/templates/ leaf01-if.reference***

Step 07: roles, playbook optimization (Optional)

So far, we used scripts and playbooks to apply the configurations we needed. However, the way the scripts are organized is not best practice.

Step 07 shows a way to organize our assets which resembles a "production ready automation" structure.

Task 1: Explore the general structure

Take a look at the content of *step-07* directory.

ls ~/ON-15/step07

```
cumulus@oob-mgmt-server:~$ ls ON-15/step-07/
playbooks  roles
```

The assets are stored in two sub-directories

- playbooks
- roles

each directory stores the resources needed for it to operate.

The "*roles*" directory will store necessary configurations, organized into categories.

ls ~/ON-15/step07

```
cumulus@oob-mgmt-server:~$ ls ON-15/step-07/roles/
frr  hostname  interfaces
```

Each category contains templates that it uses, and the sub-roles required for the complete process:

- handlers – preparing and applying configurations necessary to the role.
- tasks – doing the necessary steps for the roles.

```
cumulus@oob-mgmt-server:~$ ls ON-15/step-07/roles/frr/
handlers  tasks  templates
```

- handlers and tasks directories contains a *main.yaml* file that is used to fulfill the sub role.
- Templates folder will contain j2 files (jinja) to be used in the sub-roles *main.yaml*

The "playbooks" directory contains a main.yaml file that uses the roles defined in the roles directory. In addition,this is where the host and group variables are stored.

```
cumulus@oob-mgmt-server:~$ ls ON-15/step-07/playbooks/
group_vars  host_vars  main.yaml

cumulus@oob-mgmt-server:~$ cat ON-15/step-07/playbooks/main.yaml
---
- hosts: spine
  become: true
  user: cumulus
  gather_facts: false

  roles:
    - /home/cumulus/ON-15/step-07/roles/interfaces
    - /home/cumulus/ON-15/step-07/roles/frr
    - /home/cumulus/ON-15/step-07/roles/hostname
```

Task 2: Apply the structure to the leaves

a. At this point, step-07 structure is applied on the spine switches.

```
cumulus@oob-mgmt-server:~$ ls ON-15/step-07/playbooks/host_vars/
spine01.yaml  spine02.yaml
```

b. As a challenge exercise, you can try and apply the same structure to the leaf switches, and get one step closer to a "production ready automation" best practice.

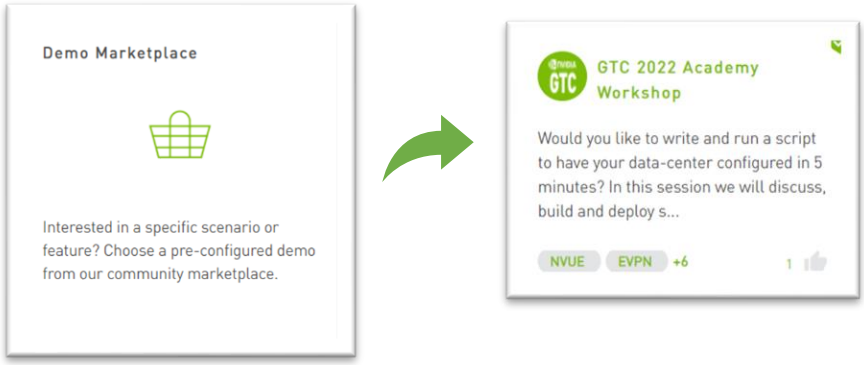
Step 08: DC Infra in 5 minutes

Finally, see how everything is coming up together to one script that configures the entire DC in less than 5 minutes.

Task 1: Create a new DC simulation

Go to the NVIDIA AIR and create a new DC simulation, you can follow the same steps shown in the beginning of the lab guide (page 3-4).

Create new Simulation



Task 2: Clone the ON-15 repo

Access the oob-management-server of the new simulation and clone the repo.

```
# git clone https://github.com/laquiante/ON-15
```

```
cumulus@oob-mgmt-server:~$ git clone https://github.com/laquiante/ON-15
Cloning into 'ON-15'...
remote: Enumerating objects: 3966, done.
remote: Counting objects: 100% (1933/1933), done.
remote: Compressing objects: 100% (1808/1808), done.
remote: Total 3966 (delta 1014), reused 0 (delta 0), pack-reused 2033
Receiving objects: 100% (3966/3966), 9.85 MiB | 7.71 MiB/s, done.
Resolving deltas: 100% (2277/2277), done.

cumulus@oob-mgmt-server:~$ ls
ON-15
```

Task 3: Execute the last script

Change the permissions and execute the last shell script

```
# chmod 777 ~/ON-15/play-step-08-DC-in-5-min.sh
# ./play-step-08-DC-in-5-min.sh
```

Wait 5 minutes (a little bit less) and there it is… 😎