Network Professional Training: 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 Cumulus 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,

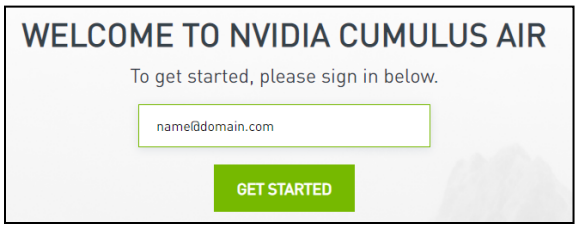
NVIDIA Academy Team

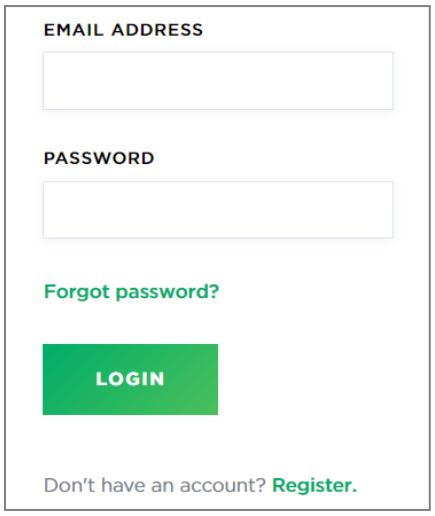
Table of Contents

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 "Register" and fill in your details.
  Once completed, a confirmation email will be sent, open it to activate your new account.

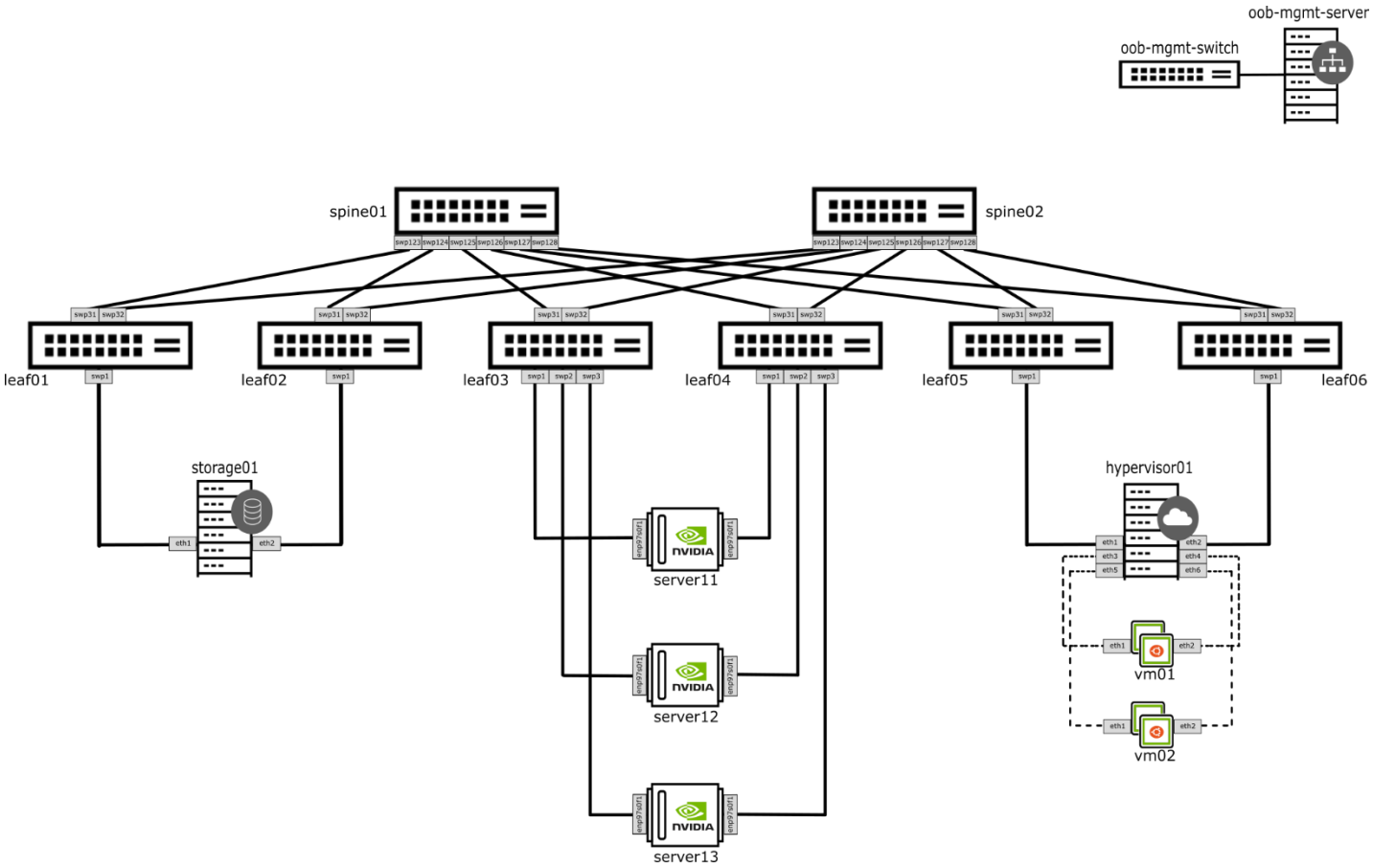
> ⚠ **Please note**
>
> Once you are registered to NVIDIA AIR, please provide your email address to the GTC instructor, the instructor will create the virtual lab which you can access via NVIDIA AIR.

2. Once you are logged in, you will reach the "My Simulations" dashboard.
   Wait for the lab to be Loaded.

3. Click on the "GTCAcademyWorkshop" label.

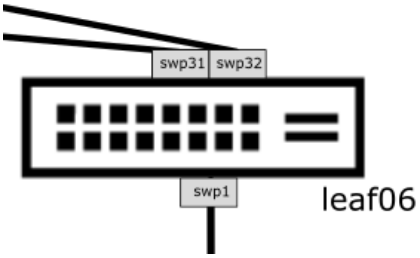
ACADEMY LAB TOPOLOGY

The workshop lab is organized in the following topology:

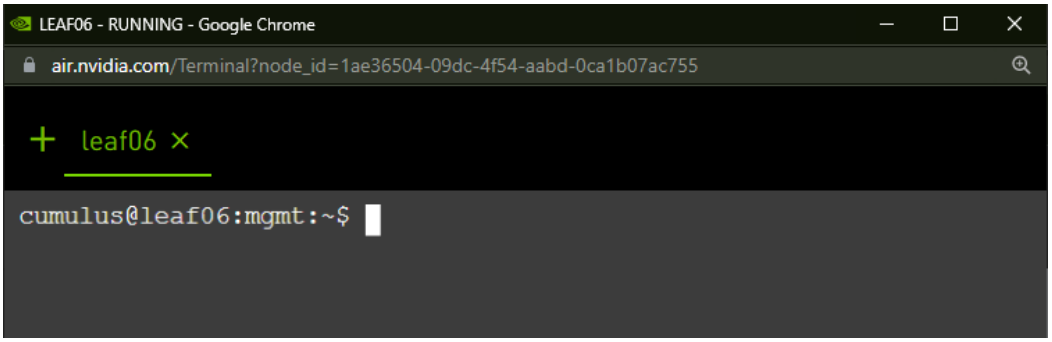


NVIDIA Academy Virtual Lab Access

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.

Practice Units

Practice 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


~
~
~
```

Please note:

- 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]***.

c. Exit VIM and use the Ansible "ping" module to test Ansible connectivity to the switches.
ansible -i practice1/hosts leaf01 -m ping

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

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]***.

b. 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

~
~
~
```

c. 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

~
~
```

d. 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

- 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.
It is very useful to check syntax and authentication errors.

```
 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
```

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, switche

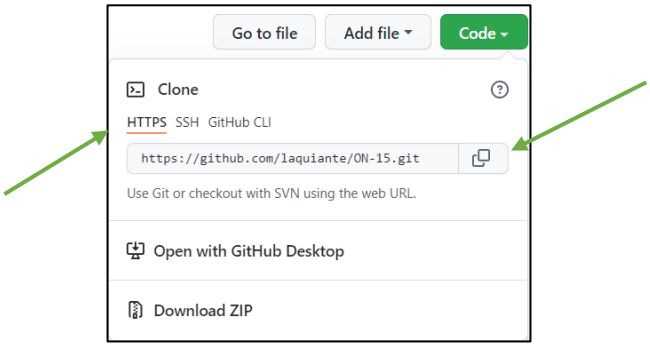
Practice objectives:

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', and clone the ON-15 repository.

 # 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: 3854, done.
remote: Counting objects: 100% (1821/1821), done.
remote: Compressing objects: 100% (1700/1700), done.
remote: Total 3854 (delta 948), reused 0 (delta 0), pack-reused 2033
Receiving objects: 100% (3854/3854), 1.53 MiB | 4.18 MiB/s, done.
Resolving deltas: 100% (2211/2211), done.

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

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 908
.
.
.
-rw-rw-r--  1 cumulus cumulus    290 Nov  3 09:50 play-step-02-reference-hypervisor-vms-switches.sh
-rw-rw-r--  1 cumulus cumulus    527 Nov  3 09:50 play-step-02-student-lab.sh
-rw-rw-r--  1 cumulus cumulus   1113 Nov  3 09:50 play-step-03-reference-all-leafs-spines-compute.sh
-rw-rw-r--  1 cumulus cumulus   1366 Nov  3 09:50 play-step-03-student-lab.sh
-rw-rw-r--  1 cumulus cumulus    329 Nov  3 09:50 play-step-04-reference-all-leafs-spines-vms.sh
-rw-rw-r--  1 cumulus cumulus    336 Nov  3 09:50 play-step-04-student-lab.sh
-rw-rw-r--  1 cumulus cumulus     92 Nov  3 09:50 play-step-05a-reference-l2-all-leafs.sh
-rw-rw-r--  1 cumulus cumulus    382 Nov  3 09:50 play-step-05a-student-lab-l2.sh
-rw-rw-r--  1 cumulus cumulus    125 Nov  3 09:50 play-step-05b-reference-multi-homing-all-leafs.sh
-rw-rw-r--  1 cumulus cumulus    382 Nov  3 09:50 play-step-05b-student-lab-multi-homing.sh
-rw-rw-r--  1 cumulus cumulus    127 Nov  3 09:50 play-step-06a-reference-all-leafs.sh
-rw-rw-r--  1 cumulus cumulus    398 Nov  3 09:50 play-step-06a-student-lab-linux-classic.sh
-rw-rw-r--  1 cumulus cumulus    152 Nov  3 09:50 play-step-06b-reference-all-leafs.sh
-rw-rw-r--  1 cumulus cumulus    227 Nov  3 09:50 play-step-06c-reference-all-leafs.sh
-rw-rw-r--  1 cumulus cumulus    402 Nov  3 09:50 play-step-06c-student-lab-configure-leaf01.sh
-rw-rw-r--  1 cumulus cumulus    176 Nov  3 09:50 play-step-07-reference.sh
drwxrwxr-x  5 cumulus cumulus   4096 Nov  3 09:50 step-02
drwxrwxr-x  2 cumulus cumulus   4096 Nov  3 09:50 step-03
drwxrwxr-x  2 cumulus cumulus   4096 Nov  3 09:50 step-04
drwxrwxr-x  2 cumulus cumulus   4096 Nov  3 09:50 step-05
drwxrwxr-x  2 cumulus cumulus   4096 Nov  3 09:50 step-06
drwxrwxr-x  2 cumulus cumulus   4096 Nov  3 09:50 step-06b
drwxrwxr-x  3 cumulus cumulus   4096 Nov  3 09:50 step-06c
drwxrwxr-x  4 cumulus cumulus   4096 Nov  3 09:50 step-07.
.
.
.
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 it can be executed,
do not execute the shell script yet.

```
cumulus@oob-mgmt-server:~/ON-15$ chmod 777 play-step-02-student-lab.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:~$ 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 #
#############################################################################
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
.
.
.
```

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

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

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

b. 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.

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 #
#############################################################################################
ansible-playbook -i /home/cumulus/ON-15/inventory/files/hosts ./step-04/step-04-student
```

b. 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.