NVIDIA® Cumulus Linux Virtual Workshop: Lab Guide

NVIDIA Cumulus Linux Virtual Workshop

Built for NVIDIA Cumulus Linux v5.14.0

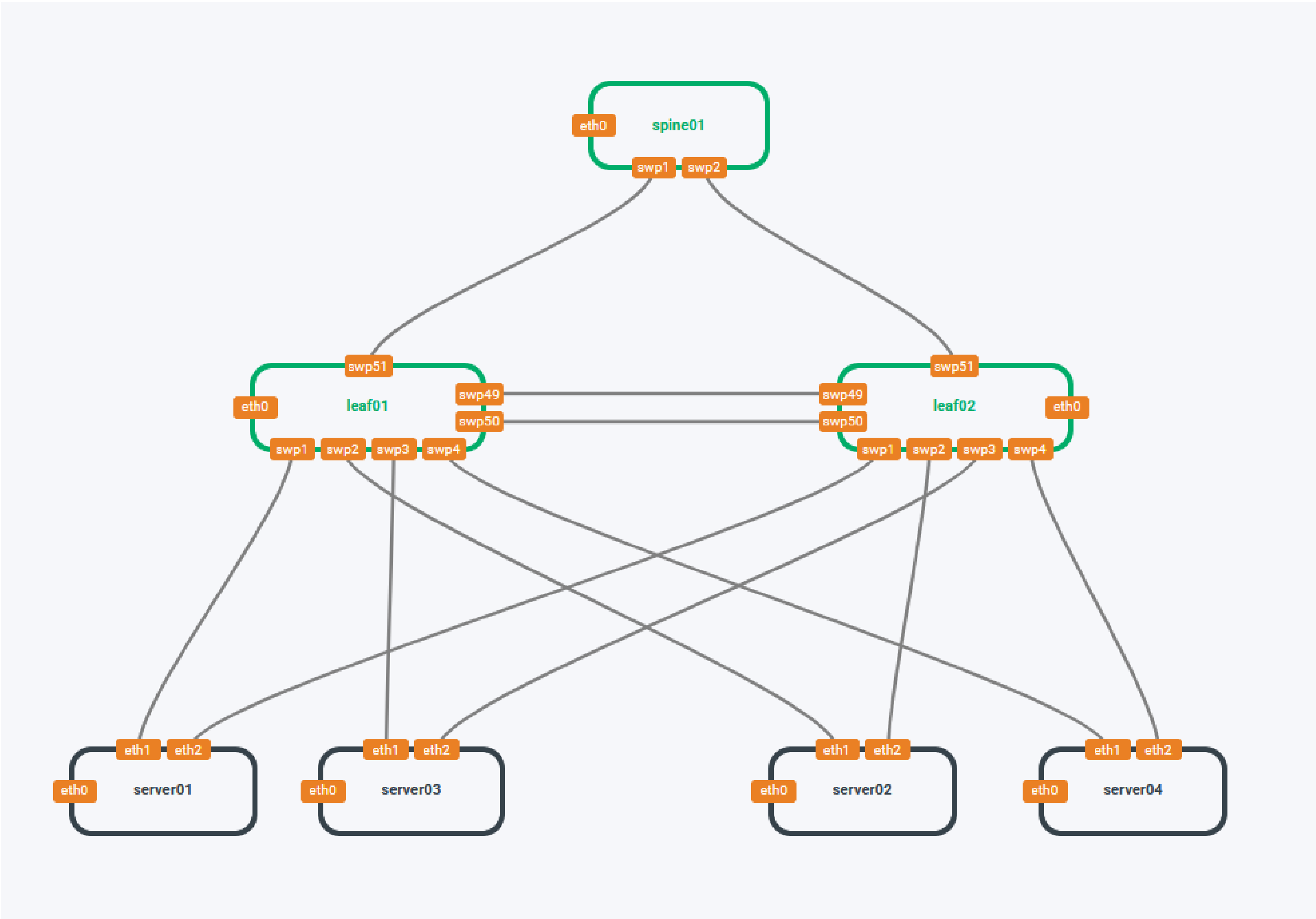
Table of Contents

Cumulus Linux Test Drive: Lab Guide

This document will guide you through some basic Cumulus Linux configuration on the NVIDIA Air platform. You will connect to your Cumulus lab, configure interfaces, and enable BGP.

Before you get started

This lab runs **Cumulus Linux 5.14.0**. The topology used for the lab is as below:



Below are the credentials used in the lab. Note that the `oob-mgmt-server` password needs to be updated on first-login.

System Name	Username	Password
oob-mgmt-server	ubuntu	nvidia
leaf01	cumulus	Cumu1usLinux!
leaf02	cumulus	Cumu1usLinux!
spine01	cumulus	Cumu1usLinux!
server01	ubuntu	nvidia
server02	ubuntu	nvidia
server03	ubuntu	nvidia
server04	ubuntu	nvidia

Lab 1: Verifying Lab Connectivity & initial setup

Let's connect to the out-of-band management server in your lab(oob-mgmt-server). The OOB network connects all your nodes together. Then, we will run an Ansible playbook (start-lab.yml) to prepare the simulation for configuration.

Goals:

> **From your oob-mgmt-server, access your switches via SSH.**
> **Log into your `oob-mgmt-server`.**
> **From your `oob-mgmt-server`, run the setup Ansible playbook** (start-lab.yml)**.**

Procedure:

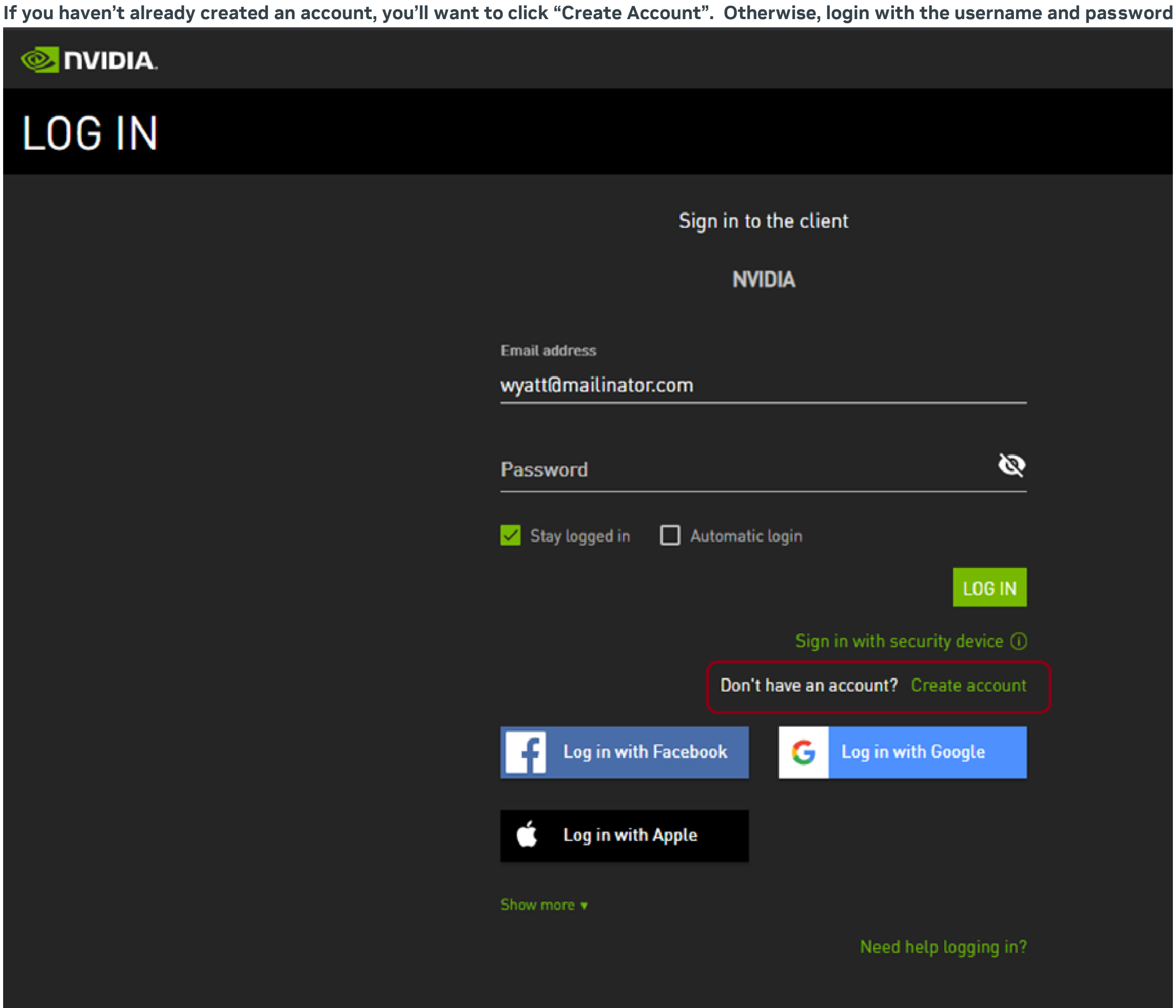
To access your lab workbench you will need to be registered with air.nvidia.com.

Access your NVIDIA AIR workbench

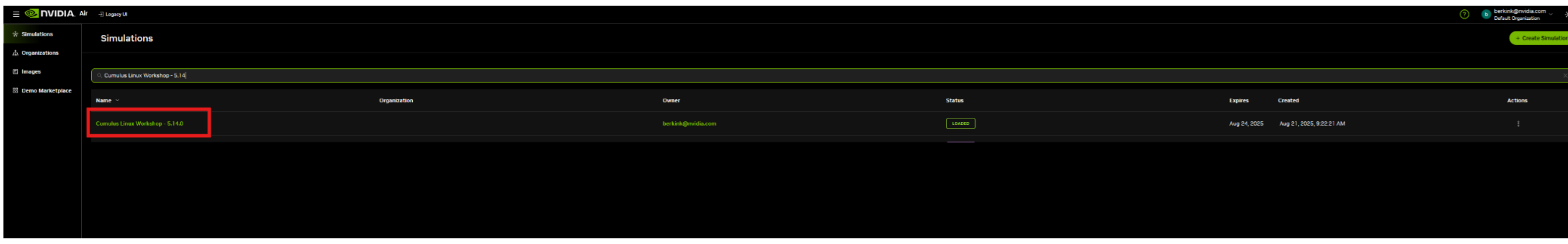
1. **Use a web browser to access and log into https://air.nvidia.com**

Type in the email address you use to sign up for this workshop.

If you haven't already created an account, you'll want to click "Create Account". Otherwise, login with the username and password you previously setup.



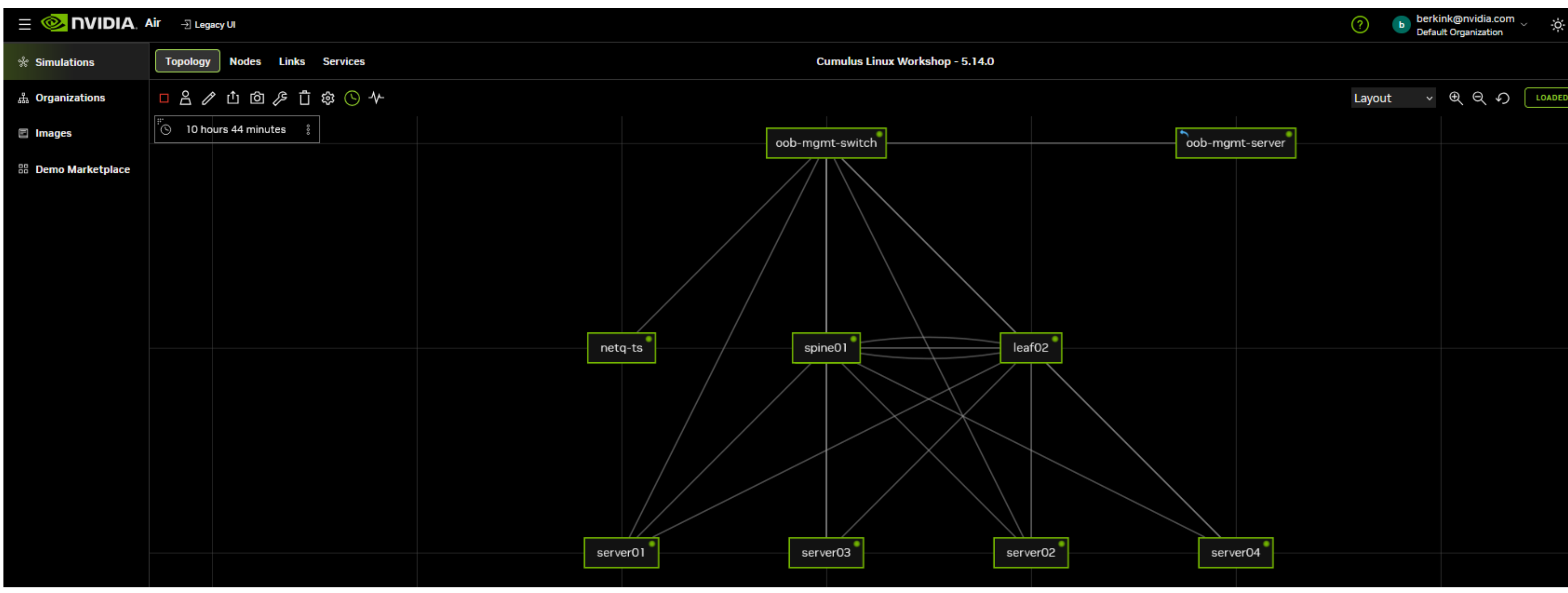
2. Once at the Nvidia Air console, find your NVIDIA Cumulus Linux Workshop simulation



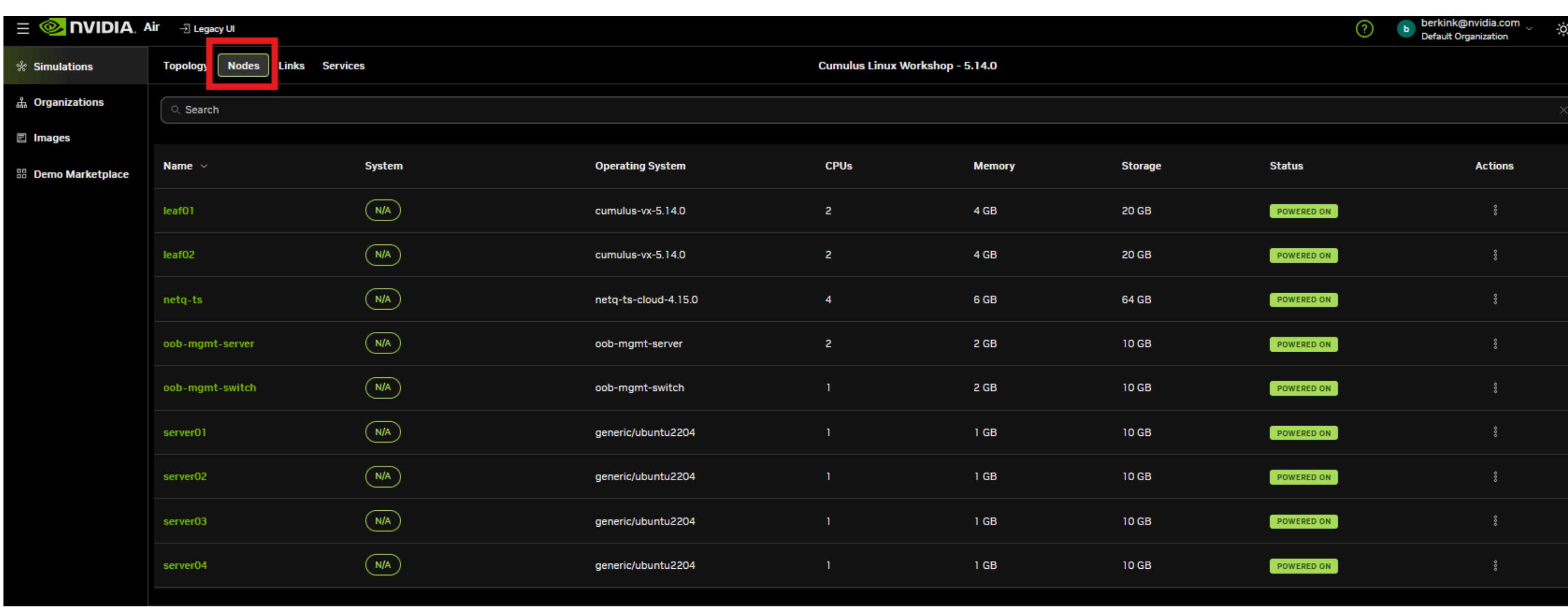
3. Click the “Cumulus Linux Workshop – 5.14.0” simulation to open your simulation console.

Connect to your oob-mgmt-server

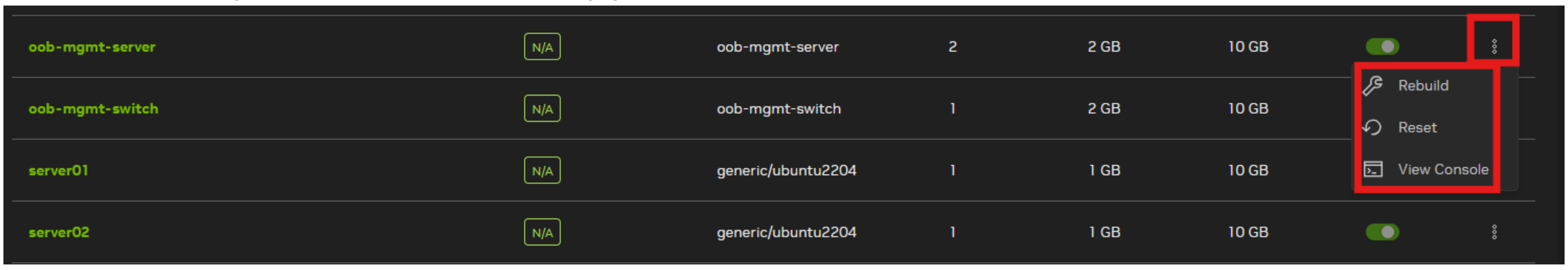
1. NVIDIA AIR GUI will show you a physical connection diagram and Simulation Guide.



2. By clicking on Nodes tab, you will see the list of all nodes in the simulation.



3. **You can also click on any of the nodes in the “Nodes” list to pop out a console window to that device.**



4. **Log into the oob-mgmt-server. You will be asked to change your password on your first login to a new, unique password. First, login with the credentials according to the pre-login banner:**

Username:	**ubuntu**
Password:	**nvidia**

Then, follow the instructions to set a new password. An example is below with the passwords unmasked.

```
oob-mgmt-server login: ubuntu
Password: nvidia
You are required to change your password immediately (administrator enforced).
Changing password for ubuntu.
Current password: nvidia
New password: [your-new-password]
Retype new password: [your-new-password]
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-67-generic x86_64)

   <banner omitted for brevity>

ubuntu@oob-mgmt-server:~$
```

Run the setup playbook

We must run an Ansible playbook to prepare our nodes for the lab.
The GitLab repository for the playbook has already been cloned for you in the `Test-Drive-Automation` directory.

To run the playbook:

1. **In the `oob-mgmt-server`, change to the `Test-Drive-Automation` directory.**

```
ubuntu@oob-mgmt-server:~$ cd Test-Drive-Automation
ubuntu@oob-mgmt-server:~/Test-Drive-Automation$
```

2. **Perform a `git pull` to sync/fetch changes (if any).**

```
ubuntu@oob-mgmt-server:~/Test-Drive-Automation$ git pull
Already up-to-date.
ubuntu@oob-mgmt-server:~/Test-Drive-Automation$
```

3. **Run the ‘start-lab.yml’ Ansible playbook.**

```
ubuntu@oob-mgmt-server:~/Test-Drive-Automation$ ansible-playbook start-lab.yml
[WARNING]: Invalid characters were found in group names but not replaced, use
-vvvv to see details

PLAY [host] *********************************************************************

TASK [Setting up the test hosts config] *******************************************
Monday 29 July 2024  12:00:42 +0000 (0:00:00.026)       0:00:00.026 ***********
ok: [server01]
ok: [server02]
ok: [server03]
ok: [server04]


TASK [install traceroute] ****************************************************************
Monday 29 July 2024  12:00:44 +0000 (0:00:01.480)       0:00:01.506 ***********
ok: [server01]
ok: [server02]
ok: [server03]
ok: [server03]


TASK [flush arp] ***************************************************************************
Monday 29 July 2024  12:00:46 +0000 (0:00:01.879)       0:00:03.386 ***********
changed: [server01]
```

```
changed: [server02]
changed: [server03]
changed: [server04]


PLAY RECAP *********************************************************************
server01              : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server02              : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server03              : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server04              : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

Monday 29 July 2024  12:00:47 +0000 (0:00:01.077)       0:00:04.464 ***********
===============================================================================
install traceroute ------------------------------------------------------ 1.88s
Setting up the test hosts config ---------------------------------------- 1.48s
flush arp --------------------------------------------------------------- 1.08s
```

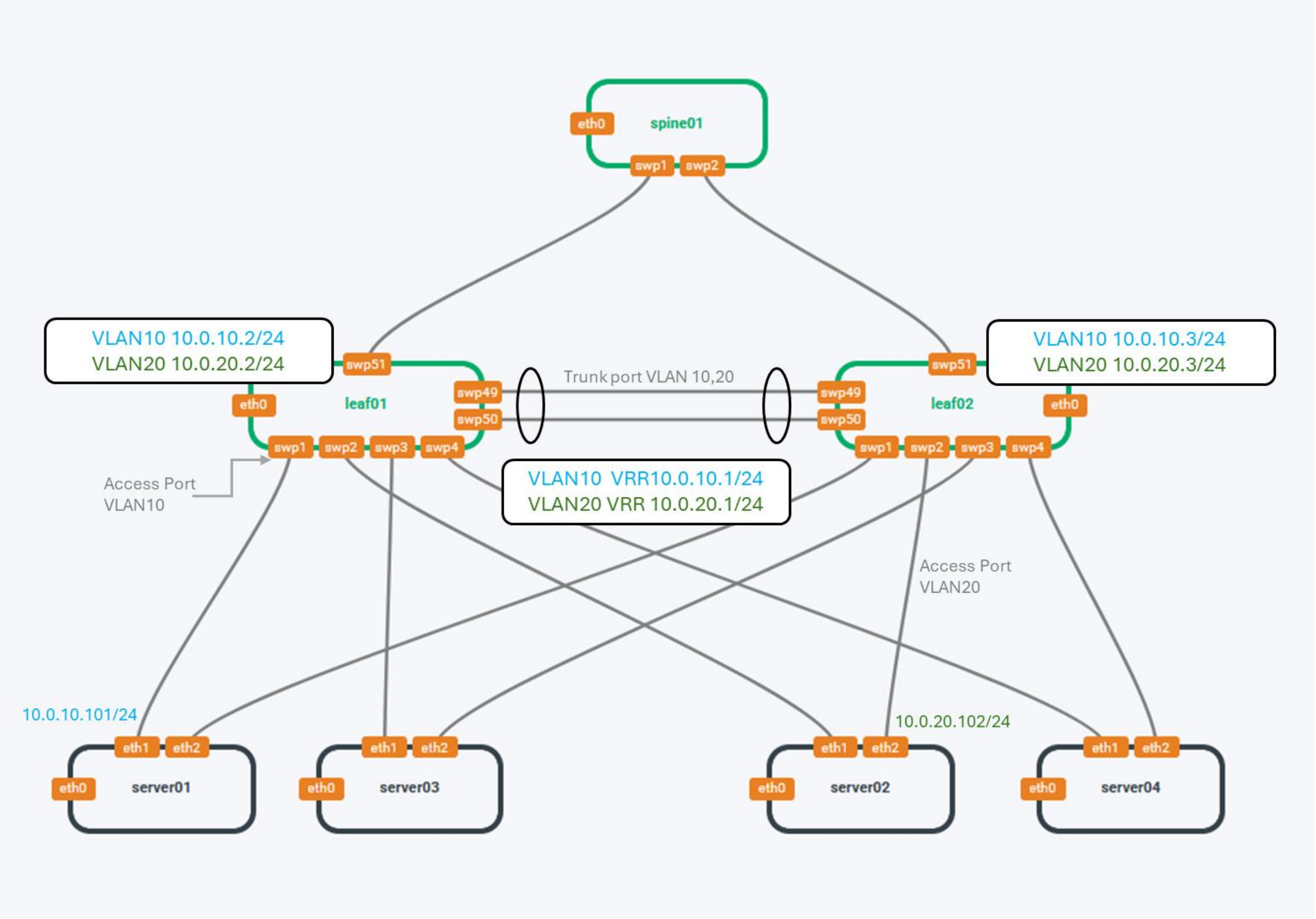
This concludes Lab 1.

Lab 2: Interface Configuration

Objective:

First, we configure a bond between `leaf01` and `leaf02`. We configure this bond as a trunk to pass `vlan10` and `vlan20`. We configure access ports between leafs and servers. Because `Server01` and `Server02` are in different subnets, `leaf01` and `leaf02` are configured to route for each vlan using VRR to provide high availability gateways for each vlan.

This lab assumes you have completed [Lab 1: Verifying Lab Connectivity & Setup]



Dependencies on other Labs:

- Lab1

Goals:

- Configure loopback addresses for `leaf01` and `leaf02`
- Configure a bond between `leaf01` and `leaf02`
- Configure a bridge
- Create a trunk port and access port
- Configure SVIs on `leaf01` and `leaf02`
- Configure VRR addresses on `leaf01` and `leaf02`

Configure loopback addresses on leaf01 and leaf02

Interface Configuration Details		
	leaf01	leaf02
Loopback IP	10.255.255.1/32	10.255.255.2/32

1. **On leaf01, assign 10.255.255.1/32 to `lo` interface. Apply the configuration.**

```
cumulus@leaf01:mgmt:~$ nv set interface lo ip address 10.255.255.1/32
cumulus@leaf01:mgmt:~$ nv set system hostname leaf01
cumulus@leaf01:mgmt:~$ nv config apply
/etc/cumulus/switchd.d/kernel_route_offload_flags.conf has been manually changed since the last save.  These changes WILL be
overwritten.
/etc/cumulus/ports.conf has been manually changed since the last save.  These changes WILL be overwritten.
/etc/ntp.conf has been manually changed since the last save.  These changes WILL be overwritten.
/etc/ptp4l.conf has been manually changed since the last save.  These changes WILL be overwritten.
/etc/network/interfaces has been manually changed since the last save.  These changes WILL be overwritten.
/etc/frr/frr.conf has been manually changed since the last save.  These changes WILL be overwritten.
/etc/frr/daemons has been manually changed since the last save.  These changes WILL be overwritten.
The frr service will need to be restarted because the list of router services has changed.  This will disrupt traffic.
/etc/hostname has been manually changed since the last save.  These changes WILL be overwritten.
/etc/hosts has been manually changed since the last save.  These changes WILL be overwritten.
Are you sure? [y/N] Y
cumulus@leaf01:mgmt:~$
```

2. **On leaf01, assign 10.255.255.2/32 to `lo` interface. Apply the configuration.**

```
cumulus@leaf02:mgmt:~$ nv set interface lo ip address 10.255.255.2/32
```

```
cumulus@leaf02:mgmt:~$ nv set system hostname leaf02
cumulus@leaf02:mgmt:~$ nv config apply
/etc/cumulus/switchd.d/kernel_route_offload_flags.conf has been manually changed since the last save.  These changes WILL be
overwritten.
/etc/cumulus/ports.conf has been manually changed since the last save.  These changes WILL be overwritten.
/etc/ntp.conf has been manually changed since the last save.  These changes WILL be overwritten.
/etc/ptp4l.conf has been manually changed since the last save.  These changes WILL be overwritten.
/etc/network/interfaces has been manually changed since the last save.  These changes WILL be overwritten.
/etc/frr/frr.conf has been manually changed since the last save.  These changes WILL be overwritten.
/etc/frr/daemons has been manually changed since the last save.  These changes WILL be overwritten.
The frr service will need to be restarted because the list of router services has changed.  This will disrupt traffic.
/etc/hostname has been manually changed since the last save.  These changes WILL be overwritten.
/etc/hosts has been manually changed since the last save.  These changes WILL be overwritten.
Are you sure? [y/N] Y
cumulus@leaf02:mgmt:~$
```

Verify loopback IP address configuration

3. **On leaf01, check that the address has been applied.**

```
cumulus@leaf01:mgmt:~$ nv show interface lo

                    operational       applied
----------------------  -----------------  ---------------
type                  loopback          loopback
router
  ospf
    enable                            off
  pim
    enable                            off
  ospf6
    enable                            off
neighbor
  [ipv4]
  [ipv6]
ip
  igmp
    enable                            off
  ipv4
    forward                            on
  ipv6
    enable                            on
    forward                            on
  vrf                              default
  [address]           10.255.255.1/32    10.255.255.1/32
  [address]           127.0.0.1/8
  [address]           ::1/128
link
  mtu                 65536
  state               up
  stats
    in-bytes            776.02 KB
    in-pkts             11960
    in-drops            0
    in-errors           0
    out-bytes           776.02 KB
    out-pkts            11960
    out-drops           0
    out-errors          0
    carrier-transitions  0
  mac                 00:00:00:00:00:00
  protodown           disabled
  oper-status         unknown
  admin-status        up
ifindex               1
```

4. **On leaf02, check that the address has been applied.**

```
cumulus@leaf02:mgmt:~$ nv show interface lo
                    operational       applied
----------------------  ----------------  ---------------

type                  loopback          loopback
router
  ospf
    enable                            off
  pim
    enable                            off
  ospf6
    enable                            off
neighbor
  [ipv4]
  [ipv6]
ip
  igmp
    enable                            off
  ipv4
    forward                            on
  ipv6
    enable                            on
    forward                            on
```

```
  vrf                 default
  [address]      10.255.255.2/32   10.255.255.2/32
  [address]      127.0.0.1/8
  [address]      ::1/128
 link
  mtu           65536
  state         up
  stats
   in-bytes       792.01 KB
   in-pkts        12208
   in-drops       0
   in-errors      0
   out-bytes       792.01 KB
   out-pkts        12208
   out-drops       0
   out-errors      0
   carrier-transitions 0
  mac           00:00:00:00:00:00
  protodown        disabled
  oper-status      unknown
  admin-status       up
 ifindex          1
```

Important things to observe:
- Loopback has user-defined IP address as well as home address assigned to it
- Loopback has a predefined default configuration on NVIDIA Cumulus Linux. Make sure not to delete it.
- Applied is what you have configured.
- Operational is what is currently running on the switch.
- Pending (not shown here) is what you have configured, but not applied with "nv config apply"

Configure bond between leaf01 and leaf02

Create a bond `bond0` with members `swp49` and `swp50`. Apply to the running configuration.

Bond Configuration Details		
	leaf01	**leaf02**
Bond name	bond0	bond0
Bond members	swp49,swp50	swp49,swp50

5. **On leaf01, create a bond with members `swp49` and `swp50`.**

```
cumulus@leaf01:mgmt:~$ nv set interface bond0 bond member swp49-50
cumulus@leaf01:mgmt:~$ nv config apply
```

6. **On leaf02, create a bond with members `swp49` and `swp50`.**

```
cumulus@leaf02:mgmt:~$ nv set interface bond0 bond member swp49-50
cumulus@leaf02:mgmt:~$ nv config apply
```

7. **On leaf01 and leaf02, check status of the bond between two switches. Verify that the bond is operational by checking the status of the bond and its members.**

```
cumulus@leaf01:mgmt:~$ nv show interface bond0
                 operational            applied
------------------------  ----------------------  ----------
type                 bond                 bond
router
 pbr
  [map]
 ospf
  enable                              off
 pim
  enable                              off
 adaptive-routing
  enable                              off
 ospf6
  enable                              off
lldp
 dcbx-pfc-tlv          off
 dcbx-ets-config-tlv      off
 dcbx-ets-recomm-tlv      off
 state               enabled
 [neighbor]
bond
 down-delay          0                 0
 lacp-bypass         off               off
 lacp-rate          fast              fast
 mode               lacp              lacp
 up-delay           0                 0
 [member]           swp49             swp49
 [member]           swp50             swp50
 mlag
```

```
   enable                              off
bridge
  [domain]            br_default            br_default
evpn
  multihoming
   uplink                              off
   segment
    enable                             off
ptp
  enable              off                  off
[acl]
neighbor
  [ipv4]
  [ipv6]
sflow
  state                                enabled
parent                br_default
ip
  vrrp
   enable                              off
  igmp
   enable                              off
  neighbor-discovery
   enable                              on
   router-advertisement
    enable                             off
   home-agent
    enable                             off
   [rdnss]
   [dnssl]
   [prefix]
  ipv4
   forward                              on
  ipv6
   enable                              on
   forward                              on
  vrf                                default
  [gateway]
link
  auto-negotiate         off
  duplex              full                 full
  speed               2G                    auto
  mac-address            48:b0:2d:a8:ea:7b
  fec                                  auto
  mtu                 9216                  9216
  [flag]              broadcast
  [flag]              multicast
  [flag]              master
  [flag]              up
  [flag]              lower-up
  state               up                   up
  flap-protection
   enable                              on
  stats
   in-bytes            6.36 KB
   in-pkts             45
   in-drops             0
   in-errors            0
   out-bytes            6.00 KB
   out-pkts             48
   out-drops             1
   out-errors            0
   carrier-transitions    1
   carrier-up-count       1
   carrier-down-count    0
  protodown              disabled
  oper-status            up
  admin-status           up
  oper-status-last-change  2025/08/21 11:38:23.614
ifindex               18
cumulus@leaf01:mgmt:~$
```

```
cumulus@leaf02:mgmt:~$ nv show interface bond0
                      operational            applied
-------------------------  -----------------------  ----------
type                  bond                  bond
router
  pbr
   [map]
  ospf
   enable                              off
  pim
   enable                              off
  adaptive-routing
   enable                              off
  ospf6
   enable                              off
lldp
  dcbx-pfc-tlv           off
  dcbx-ets-config-tlv     off
  dcbx-ets-recomm-tlv      off
  state                enabled
```

```
  [neighbor]
bond
  down-delay          0                  0
  lacp-bypass         off                off
  lacp-rate           fast               fast
  mode                lacp               lacp
  up-delay            0                  0
  [member]            swp49              swp49
  [member]            swp50              swp50
  mlag
    enable                               off
bridge
  [domain]            br_default         br_default
evpn
  multihoming
    uplink                               off
    segment
      enable                             off
ptp
  enable              off                off
[acl]
neighbor
  [ipv4]
  [ipv6]
sflow
  state                                  enabled
parent                br_default
ip
  vrrp
    enable                               off
  igmp
    enable                               off
  neighbor-discovery
    enable                               on
    router-advertisement
      enable                             off
    home-agent
      enable                             off
    [rdnss]
    [dnssl]
    [prefix]
  ipv4
    forward                              on
  ipv6
    enable                               on
    forward                              on
  vrf                                    default
  [gateway]
link
  auto-negotiate      off
  duplex              full               full
  speed               2G                 auto
  mac-address         48:b0:2d:d5:21:b4
  fec                                    auto
  mtu                 9216               9216
  [flag]              broadcast
  [flag]              multicast
  [flag]              master
  [flag]              up
  [flag]              lower-up
  state               up                 up
  flap-protection
    enable                               on
  stats
    in-bytes          39.51 KB
    in-pkts           355
    in-drops          0
    in-errors         0
    out-bytes         37.06 KB
    out-pkts          292
    out-drops         1
    out-errors        0
    carrier-transitions   1
    carrier-up-count      1
    carrier-down-count    0
  protodown           disabled
  oper-status         up
  admin-status        up
  oper-status-last-change  2025/08/21 11:38:23.845
ifindex               18
cumulus@leaf02:mgmt:~$
```

Important things to observe:
The speed of the bond is the cumulative speed of all member interfaces

Configure bridge and access ports on leaf01 and leaf02

Bridge Configuration Details		
	leaf01	leaf02
Bridge vlans	10,20	10,20
Bridge members	bond0,swp1	bond0,swp2
Bridge access port	swp1	swp2
Bridge access vlan	10	20

8. On leaf01, create vlans `vlan10` & `vlan20` on `br_default`.

```
cumulus@leaf01:mgmt:~$ nv set bridge domain br_default vlan 10,20
```

9. On leaf01, add `swp1` and `bond0` as a member to the bridge. *Note: The name `bond0` is case sensitive in all places.*

```
cumulus@leaf01:mgmt:~$ nv set interface swp1,bond0 bridge domain br_default
```

10. On leaf01, Configure `swp1` (connecting to server01) as an access port for vlan 10.

```
cumulus@leaf01:mgmt:~$ nv set interface swp1 bridge domain br_default access 10
```

11. On leaf01, commit the changes.

```
cumulus@leaf01:mgmt:~$ nv config apply
```

12. On leaf02, repeat the same steps but use swp2 as the access port towards the server (server02).

```
cumulus@leaf02:mgmt:~$ nv set bridge domain br_default vlan 10,20
cumulus@leaf02:mgmt:~$ nv set interface swp2,bond0 bridge domain br_default
cumulus@leaf02:mgmt:~$ nv set interface swp2 bridge domain br_default access 20
cumulus@leaf02:mgmt:~$ nv config apply
```

A code snippet is provided for easy copy and pasting into Cumulus:

```
nv set bridge domain br_default vlan 10,20
nv set interface swp2,bond0 bridge domain br_default
nv set interface swp2 bridge domain br_default access 20
nv config apply
```

Verify bridge configuration on leaf01 and leaf02

13. On leaf01, verify the configuration by checking that `swp1` and `bond0` are part of the bridge.

```
cumulus@leaf01$ nv show bridge domain br_default vlan

Vlan  Ptp State  Source IP  VNI
----  ---------  ---------  ---
10    off        0.0.0.0
20    off        0.0.0.0
```

14. On leaf02, verify the same configuration on leaf02 by checking that `swp2` and `bond0` are part of the bridge.

```
cumulus@leaf02$ nv show bridge domain br_default vlan

Vlan  Ptp State  Source IP  VNI
----  ---------  ---------  ---
10    off        0.0.0.0
20    off        0.0.0.0
```

Important things to observe: > Vlan information has not been completed yet	
On leaf01: • swp1 should be an access port in vlan 10 • BOND0 should be a trunk for vlan10 and vlan20, with a native vlan of 1 (PVID)	On leaf02: • swp2 should be an access port in vlan 20 • BOND0 should be a trunk for vlan10 and vlan20, with a native vlan of 1 (PVID)

Configure SVI and VRR on leaf01 and leaf02

VRR Configuration details		
	leaf01	leaf02
VLAN10 real IP address	10.0.10.2/24	10.0.10.3/24
VLAN10 VRR IP address	10.0.10.1/24	10.0.10.1/24
VLAN10 VRR MAC address	00:00:00:00:1a:10	00:00:00:00:1a:10
VLAN20 real IP address	10.0.20.2/24	10.0.20.3/24
VLAN20 VRR IP address	10.0.20.1/24	10.0.20.1/24
VLAN20 VRR MAC address	00:00:00:00:1a:20	00:00:00:00:1a:20
SERVER01 vlan	10	10
SERVER02 vlan	20	20

15. On leaf01, create an SVI for `vlan10`.

```
cumulus@leaf01:mgmt:~$ nv set interface vlan10 ip address 10.0.10.2/24
```

16. On leaf01, create an SVI for `vlan20`.

```
cumulus@leaf01:mgmt:~$ nv set interface vlan20 ip address 10.0.20.2/24
```

17. On leaf01, apply a VRR address for `vlan10`.

```
cumulus@leaf01:mgmt:~$ nv set interface vlan10 ip vrr address 10.0.10.1/24
cumulus@leaf01:mgmt:~$ nv set interface vlan10 ip vrr mac-address 00:00:00:00:1a:10
cumulus@leaf01:mgmt:~$ nv set interface vlan10 ip vrr state up
```

18. On leaf01, apply a VRR address for `vlan20`.

```
cumulus@leaf01:mgmt:~$ nv set interface vlan20 ip vrr address 10.0.20.1/24
cumulus@leaf01:mgmt:~$ nv set interface vlan20 ip vrr mac-address 00:00:00:00:1a:20
cumulus@leaf01:mgmt:~$ nv set interface vlan20 ip vrr state up
```

19. On leaf01, commit the changes.

```
cumulus@leaf01:mgmt:~$ nv config apply
```

20. On leaf02, repeat steps these steps.

```
cumulus@leaf02:mgmt:~$ nv set interface vlan10 ip address 10.0.10.3/24
cumulus@leaf02:mgmt:~$ nv set interface vlan20 ip address 10.0.20.3/24
cumulus@leaf02:mgmt:~$ nv set interface vlan10 ip vrr address 10.0.10.1/24
cumulus@leaf02:mgmt:~$ nv set interface vlan10 ip vrr mac-address 00:00:00:00:1a:10
cumulus@leaf02:mgmt:~$ nv set interface vlan10 ip vrr state up
cumulus@leaf02:mgmt:~$ nv set interface vlan20 ip vrr address 10.0.20.1/24
cumulus@leaf02:mgmt:~$ nv set interface vlan20 ip vrr mac-address 00:00:00:00:1a:20
cumulus@leaf02:mgmt:~$ nv set interface vlan20 ip vrr state up
cumulus@leaf02:mgmt:~$ nv config apply
```

A code snippet is provided for easy copy and pasting into Cumulus:

```
nv set interface vlan10 ip address 10.0.10.3/24
nv set interface vlan10 ip vrr address 10.0.10.1/24
nv set interface vlan10 ip vrr mac-address 00:00:00:00:1a:10
nv set interface vlan10 ip vrr state up
nv set interface vlan20 ip address 10.0.20.3/24
nv set interface vlan20 ip vrr address 10.0.20.1/24
nv set interface vlan20 ip vrr mac-address 00:00:00:00:1a:20
nv set interface vlan20 ip vrr state up
nv config apply
```

Verify connectivity

21. On server01, ping the VRR gateway address for `vlan10`.

```
ubuntu@server01:~$ ping 10.0.10.1
PING 10.0.10.1 (10.0.10.1) 56(84) bytes of data.
64 bytes from 10.0.10.1: icmp_seq=1 ttl=64 time=0.686 ms
64 bytes from 10.0.10.1: icmp_seq=2 ttl=64 time=0.922 ms
^C
--- 10.0.10.1 ping statistics ---
```

```
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.686/0.804/0.922/0.118 ms
```

22. **On server01, ping the SVI interface IP Address of `leaf01` for `vlan10`.**

```
ubuntu@server01:~$ ping 10.0.10.2
PING 10.0.10.2 (10.0.10.2) 56(84) bytes of data.
64 bytes from 10.0.10.2: icmp_seq=1 ttl=64 time=0.887 ms
64 bytes from 10.0.10.2: icmp_seq=2 ttl=64 time=0.835 ms
^C
--- 10.0.10.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.835/0.861/0.887/0.026 ms
```

23. **On server01, ping the SVI interface IP Address of `leaf02` for `vlan10`.**

```
ubuntu@server01:~$ ping 10.0.10.3
PING 10.0.10.3 (10.0.10.3) 56(84) bytes of data.
64 bytes from 10.0.10.3: icmp_seq=1 ttl=64 time=0.528 ms
64 bytes from 10.0.10.3: icmp_seq=2 ttl=64 time=0.876 ms
^C
--- 10.0.10.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.528/0.702/0.876/0.174 ms
```

24. **On server01, check the IP neighbor table to view each MAC address. You can also use the `arp -a` command.**

```
ubuntu@server01:~$ ip neighbor show
192.168.200.1 dev eth0 lladdr 44:38:39:00:00:11 REACHABLE
10.0.10.1 dev eth1 lladdr 00:00:00:00:1a:10 STALE
10.0.10.2 dev eth1 lladdr 44:38:39:00:00:05 STALE
10.0.10.3 dev eth1 lladdr 44:38:39:00:00:0b STALE
fe80::4638:39ff:fe00:5 dev eth1 lladdr 44:38:39:00:00:05 router STALE
fe80::4638:39ff:fe00:12 dev eth0 lladdr 44:38:39:00:00:12 router STALE
fe80::4638:39ff:fe00:b dev eth1 lladdr 44:38:39:00:00:0b router REACHABLE
```

25. **On server02, repeat the same connectivity tests in step 21-24 from server02 to switch IP addresses.**

```
ubuntu@server02:~$ ping 10.0.20.1
PING 10.0.20.1 (10.0.20.1) 56(84) bytes of data.
64 bytes from 10.0.20.1: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 10.0.20.1: icmp_seq=2 ttl=64 time=0.672 ms
^C
--- 10.0.20.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.672/0.949/1.226/0.277 ms
```

```
ubuntu@server02:~$ ping 10.0.20.2
PING 10.0.20.2 (10.0.20.2) 56(84) bytes of data.
64 bytes from 10.0.20.2: icmp_seq=1 ttl=64 time=0.735 ms
64 bytes from 10.0.20.2: icmp_seq=2 ttl=64 time=1.02 ms
^C
--- 10.0.20.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.735/0.882/1.029/0.147 ms
```

```
ubuntu@server02:~$ ping 10.0.20.3
PING 10.0.20.3 (10.0.20.3) 56(84) bytes of data.
64 bytes from 10.0.20.3: icmp_seq=1 ttl=64 time=0.993 ms
64 bytes from 10.0.20.3: icmp_seq=2 ttl=64 time=1.08 ms
^C
--- 10.0.20.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.993/1.040/1.087/0.047 ms
```

```
ubuntu@server02:~$ ip neighbor show
192.168.200.1 dev eth0 lladdr 44:38:39:00:00:11 REACHABLE
10.0.20.2 dev eth2 lladdr 44:38:39:00:00:05 REACHABLE
10.0.20.3 dev eth2 lladdr 44:38:39:00:00:0b REACHABLE
10.0.20.1 dev eth2 lladdr 00:00:00:00:1a:20 STALE
fe80::4638:39ff:fe00:5 dev eth2 lladdr 44:38:39:00:00:05 router STALE
fe80::4638:39ff:fe00:12 dev eth0 lladdr 44:38:39:00:00:12 router STALE
fe80::4638:39ff:fe00:b dev eth2 lladdr 44:38:39:00:00:0b router STALE
```

Important things to observe:
> **Pings to the VRR and unique SVI IP addresses should all be successful for all Vlans**

26. **On server01 and server02, ping to verify connectivity between server01 and server02.**

```
ubuntu@server01:~$ ping 10.0.20.102
PING 10.0.20.102 (10.0.20.102) 56(84) bytes of data.
64 bytes from 10.0.20.102: icmp_seq=1 ttl=63 time=0.790 ms
64 bytes from 10.0.20.102: icmp_seq=2 ttl=63 time=1.35 ms
^C
--- 10.0.20.102 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.790/1.070/1.351/0.282 ms
```

```
ubuntu@server02:~$ ping 10.0.10.101
PING 10.0.10.101 (10.0.10.101) 56(84) bytes of data.
64 bytes from 10.0.10.101: icmp_seq=1 ttl=63 time=1.08 ms
64 bytes from 10.0.10.101: icmp_seq=2 ttl=63 time=1.36 ms
^C
--- 10.0.10.101 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.089/1.225/1.361/0.136 ms
```

27. On server01 and server02, traceroute to server02.

```
ubuntu@server01:~$ traceroute 10.0.20.102
traceroute to 10.0.20.102 (10.0.20.102), 30 hops max, 60 byte packets
 1  10.0.10.1 (10.0.10.1)  1.628 ms  1.672 ms  1.855 ms
 2  10.0.20.102 (10.0.20.102)  7.947 ms  7.973 ms  8.155 ms
cumulus@server01:~$
```

```
ubuntu@server02:~$ traceroute 10.0.10.101
traceroute to 10.0.10.101 (10.0.10.101), 30 hops max, 60 byte packets
 1  10.0.20.1 (10.0.20.1)  2.813 ms  2.776 ms  3.307 ms
 2  10.0.10.101 (10.0.10.101)  9.199 ms  7.836 ms  7.766 ms
cumulus@server02:~$
```

Verify MAC address table on leaf01 and leaf02

28. On leaf01 and leaf02, verify that MAC addresses are learned correctly.

```
cumulus@leaf01:mgmt:~$ nv show bridge domain br_default mac-table

entry-id  MAC address        vlan  interface   remote-dst  src-vni  entry-type  last-update  age
--------  -----------------  ----  ----------  ----------  -------  ----------  -----------  -------
1         48:b0:2d:db:95:44  10    swp1                                         0:00:00      0:01:43
2         48:b0:2d:f8:3d:4c        swp1                             permanent   0:16:37      0:16:37
3         48:b0:2d:46:ba:d2  20    bond0                                        0:00:29      0:07:15
4         00:00:00:00:1a:20  20    bond0                                        0:00:39      0:10:08
5         44:38:39:22:01:c8  20    bond0                                        0:00:24      0:10:08
6         44:38:39:22:01:c8  10    bond0                                        0:01:43      0:04:22
7         44:38:39:22:01:c8  1     bond0                                        0:14:17      0:14:19
8         48:b0:2d:44:7b:d7  1     bond0                                        0:00:27      0:16:27
9         48:b0:2d:4e:b6:de  1     bond0                                        0:00:27      0:16:27
10        48:b0:2d:8b:5f:13  1     bond0                            permanent   0:16:37      0:16:37
11        48:b0:2d:8b:5f:13        bond0                            permanent   0:16:37      0:16:37
12        00:00:00:00:1a:10        br_default                       permanent
13        44:38:39:22:01:80  20    br_default                       permanent   0:10:34      0:10:34
14        00:00:00:00:1a:10  10    br_default                       permanent   0:10:34      0:10:34
15        44:38:39:22:01:80  10    br_default                       permanent   0:10:34      0:10:34
16        44:38:39:22:01:80  1     br_default                       permanent   0:16:37      0:16:37
17        44:38:39:22:01:80        br_default                       permanent   0:16:37      0:16:37
cumulus@leaf01:mgmt:~$
```

```
cumulus@leaf02:mgmt:~$ nv show bridge domain br_default mac-table

entry-id  MAC address        vlan  interface   remote-dst  src-vni  entry-type  last-update  age
--------  -----------------  ----  ----------  ----------  -------  ----------  -----------  -------
1         48:b0:2d:46:ba:d2  20    swp2                                         0:00:30      0:02:11
2         48:b0:2d:cc:6c:c4        swp2                             permanent   0:16:04      0:16:04
3         48:b0:2d:db:95:44  10    bond0                                        0:01:07      0:07:25
4         44:38:39:22:01:80  20    bond0                                        0:02:11      0:02:11
5         44:38:39:22:01:80  10    bond0                                        0:01:09      0:12:17
6         48:b0:2d:19:9b:8a  1     bond0                                        0:00:09      0:15:39
7         48:b0:2d:8b:5f:13  1     bond0                                        0:00:00      0:16:02
8         48:b0:2d:44:7b:d7  1     bond0                            permanent   0:16:04      0:16:04
9         48:b0:2d:44:7b:d7        bond0                            permanent   0:16:04      0:16:04
10        00:00:00:00:1a:10        br_default                       permanent
11        00:00:00:00:1a:20        br_default                       permanent
12        00:00:00:00:1a:20  20    br_default                       permanent   0:11:50      0:11:50
13        44:38:39:22:01:c8  20    br_default                       permanent   0:11:50      0:11:50
14        44:38:39:22:01:c8  10    br_default                       permanent   0:11:50      0:11:50
15        00:00:00:00:1a:10  10    br_default                       permanent   0:11:50      0:11:50
16        44:38:39:22:01:c8  1     br_default                       permanent   0:16:04      0:16:04
17        44:38:39:22:01:c8        br_default                       permanent   0:16:04      0:16:04
cumulus@leaf02:mgmt:~$
```

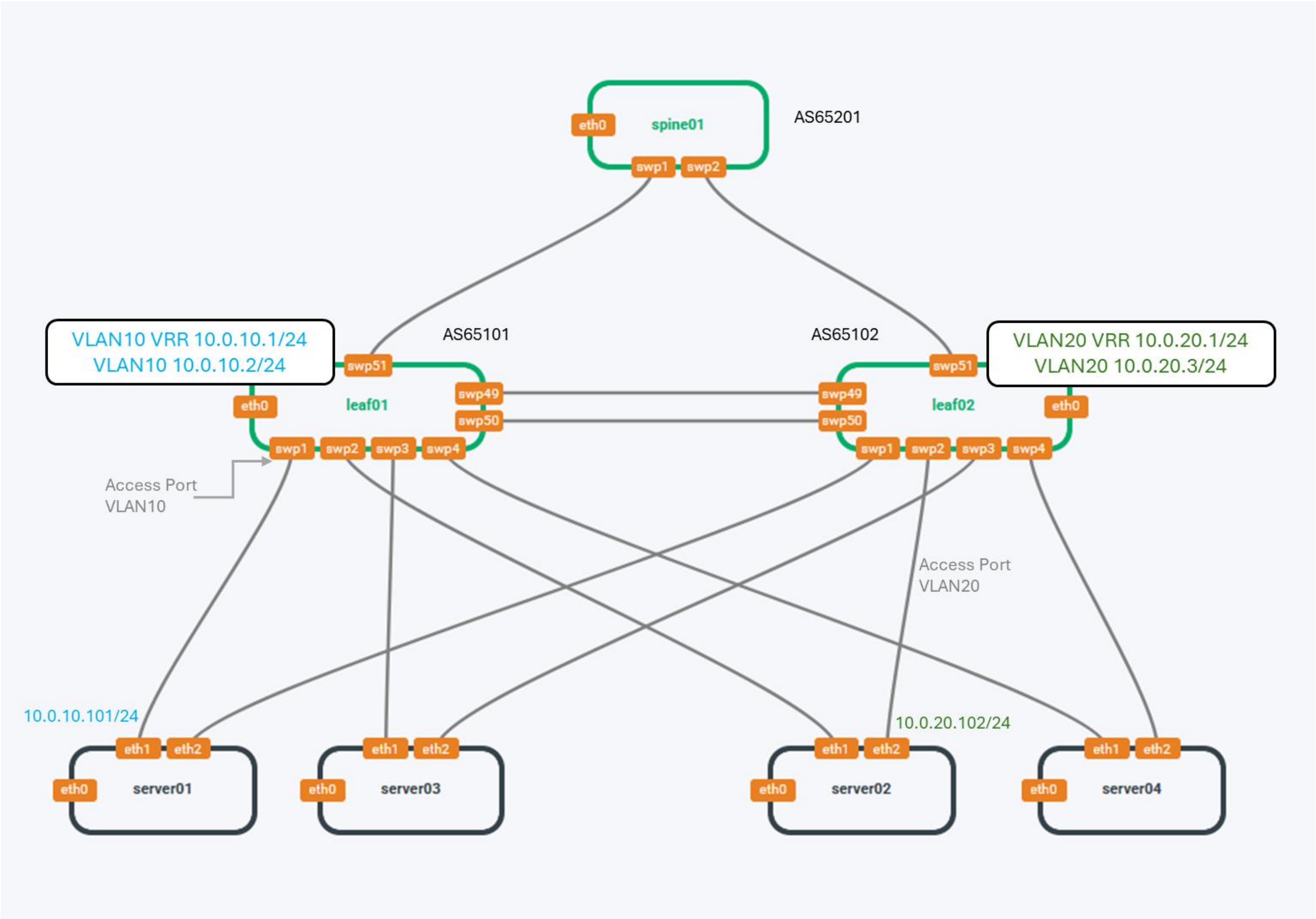
Important things to observe:

> **MAC addresses of servers should be learned on `bond0` and `swp` interface of switch**

This concludes Lab 2.

Lab 3: BGP Unnumbered and VRF-Lite configuration

This lab will configure BGP unnumbered between the leaf01/leaf02 to spine01. This configuration will share the IP addresses of the loopback interfaces on each device as well as the `vlan10` and `vlan20` subnets on the leaf01 and leaf02 devices.
As a second step this lab covers VRF-lite configuration in combination with BGP (unnumbered) peering to carry VRF specific prefixes.
This lab assumes you have completed [Lab 1: Verifying Lab Connectivity & Setup]



Dependencies on other Labs:

- Lab1

Goals:

- Configure BGP unnumbered on spine01, leaf01 and leaf02
- Advertise loopback addresses into BGP
- Advertise SVI subnets of leaf switches into BGP
- Verify BGP peering
- Verify BGP route advertisements
- Verify routed connectivity and path between servers
- Configure VRF-lite across leaf01, spine01 and leaf02
- Configure BGP peering between VRF member subinterfaces
- Advertise prefixes into VRF
- Verify connectivity

Procedure:

Run setup playbook

We must run another Ansible playbook to prepare our nodes for the lab.

1. **On oob-mgmt-server, run the playbook named 'lab3.yml'. Even if you fully completed Lab2, you must run this playbook.**

```
ubuntu@oob-mgmt-server:~/Test-Drive-Automation$ ansible-playbook lab3.yml

[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [Getting you ready for lab3] ***************************************************************************************

TASK [Dropping in config] ***************************************************************************************************
Wednesday 11 May 2022  17:32:27 +0000 (0:00:00.018)       0:00:00.018 *********
ok: [server01]
ok: [server02]
ok: [server03]
ok: [server04]

PLAY [Getting you ready for lab3] ***************************************************************************************

TASK [Drop the nvue yaml] *************************************************************************************************
Wednesday 11 May 2022  17:32:28 +0000 (0:00:01.362)       0:00:01.381 *********
changed: [leaf01]
changed: [leaf02]
```

```
RUNNING HANDLER [nvue config replace] **********************************************************************************************
Wednesday 11 May 2022  17:32:29 +0000 (0:00:01.063)       0:00:02.445 *********
changed: [leaf01]
changed: [leaf02]
RUNNING HANDLER [nvue config apply] ***********************************************************************************************
Wednesday 11 May 2022  17:32:30 +0000 (0:00:01.088)       0:00:03.533 *********
changed: [leaf01]
changed: [leaf02]

PLAY RECAP ************************************************************************************************************************
leaf01              : ok=3   changed=3   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
leaf02              : ok=3   changed=3   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
server01            : ok=1   changed=0   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
server02            : ok=1   changed=0   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
server03            : ok=1   changed=0   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
server04            : ok=1   changed=0   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0

Wednesday 11 May 2022  17:32:35 +0000 (0:00:05.243)       0:00:08.777 *********
===============================================================================
nvue config apply ---------------------------------------------------------------------------------------------------- 5.24s
Dropping in config --------------------------------------------------------------------------------------------------- 1.36s
nvue config replace -------------------------------------------------------------------------------------------------- 1.09s
Drop the nvue yaml --------------------------------------------------------------------------------------------------- 1.06s
```

Configure Loopback on Spine

Configure the `lo` address for spine0`. We already configured loopbacks for leaf01 and leaf01 in the previous lab.

Loopback Configuration			
	leaf01	leaf02	spine01
Loopback IP address	10.255.255.1/32	10.255.255.2/32	10.255.255.101/32

2. **On spine01, assign 10.255.255.101/32 to `lo`. Apply to the running configuration.**

```
cumulus@spine01:mgmt:~$ nv set interface lo ip address 10.255.255.101/32
cumulus@spine01:mgmt:~$ nv set system hostname spine01
cumulus@spine01:mgmt:~$ nv config apply
```

Configure BGP unnumbered

3. **On spine01, configure a BGP Autonomous System (AS) number for the routing instance. Multipath-relax is typically configured to more easily accommodate load sharing via ECMP.**

```
cumulus@spine01:mgmt:~$ nv set vrf default router bgp autonomous-system 65201
cumulus@spine01:mgmt:~$ nv set vrf default router bgp path-selection multipath aspath-ignore on
cumulus@spine01:mgmt:~$ nv set router bgp router-id 10.255.255.101
```

4. **On spine01, configure BGP peering on `swp1` towards leaf01 and `swp2` towards leaf02.**

```
cumulus@spine01:mgmt:~$ nv set vrf default router bgp neighbor swp1 remote-as external
cumulus@spine01:mgmt:~$ nv set vrf default router bgp neighbor swp2 remote-as external
cumulus@spine01:mgmt:~$ nv set interface swp1 link state up
cumulus@spine01:mgmt:~$ nv set interface swp2 link state up
```

5. **On spine01, apply to the running configuration.**

```
cumulus@spine01:mgmt:~$ nv config apply
```

6. **On leaf01, repeat the configuration with the corresponding AS number and router-id for leaf01.**

```
cumulus@leaf01:mgmt:~$ nv set vrf default router bgp autonomous-system 65101
cumulus@leaf01:mgmt:~$ nv set vrf default router bgp path-selection multipath aspath-ignore on
cumulus@leaf01:mgmt:~$ nv set vrf default router bgp router-id 10.255.255.1
cumulus@leaf01:mgmt:~$ nv set vrf default router bgp neighbor swp51 remote-as external
cumulus@leaf01:mgmt:~$ nv set interface swp51 link state up
cumulus@leaf01:mgmt:~$ nv config apply
```

A code snippet is provided for easy copy and pasting into Cumulus:

```
nv set vrf default router bgp autonomous-system 65101
nv set vrf default router bgp path-selection multipath aspath-ignore on
nv set vrf default router bgp router-id 10.255.255.1
nv set vrf default router bgp neighbor swp51 remote-as external
nv set interface swp51 link state up
nv config apply
```

7. **On leaf02, repeat the configuration with the corresponding AS number and router-id for leaf02.**

```
cumulus@leaf02:mgmt:~$ nv set vrf default router bgp autonomous-system 65102
cumulus@leaf02:mgmt:~$ nv set vrf default router bgp path-selection multipath aspath-ignore on
cumulus@leaf02:mgmt:~$ nv set vrf default router bgp router-id 10.255.255.2
cumulus@leaf02:mgmt:~$ nv set vrf default router bgp neighbor swp51 remote-as external
cumulus@leaf02:mgmt:~$ nv set interface swp51 link state up
cumulus@leaf02:mgmt:~$ nv config apply
```

A code snippet is provided for easy copy and pasting into Cumulus:

```
nv set vrf default router bgp autonomous-system 65102
nv set vrf default router bgp path-selection multipath aspath-ignore on
nv set vrf default router bgp router-id 10.255.255.2
nv set vrf default router bgp neighbor swp51 remote-as external
nv set interface swp51 link state up
nv config apply
```

Verify BGP connectivity between fabric nodes

Verify BGP peers are connected. We can use `vtysh` to FRR CLI output or NVUE commands.
Learn more about [vtysh](https://docs.frrouting.org/projects/dev-guide/en/latest/vtysh.html).

1. **On spine01, verify BGP peering between spine and leaf switches.**

```
cumulus@spine01:mgmt:~$ sudo vtysh -c "show ip bgp summary"

IPv4 Unicast Summary (VRF default):
BGP router identifier 10.255.255.101, local AS number 65201 vrf-id 0
BGP table version 5
RIB entries 9, using 1728 bytes of memory
Peers 2, using 40 KiB of memory

Neighbor       V       AS   MsgRcvd   MsgSent   TblVer  InQ OutQ  Up/Down State/PfxRcd   PfxSnt Desc
leaf01(swp1)   4    65101       83        83        0    0    0 00:03:51            0        0 N/A
leaf02(swp2)   4    65102       78        78        0    0    0 00:03:38            0        0 N/A

Total number of neighbors 2
```

or

```
cumulus@spine01:mgmt:~$ nv show vrf default router bgp neighbor brief

AS - Remote Autonomous System, PeerEstablishedTime - Peer established time in
UTC format, UpTime - Last connection reset time in days,hours:min:sec, Afi-Safi
- Address family, PfxSent - Transmitted prefix counter, PfxRcvd - Recieved
prefix counter

Neighbor  AS     State        PeerEstablishedTime   UpTime   MsgRcvd  MsgSent  Afi-Safi      PfxSent  PfxRcvd
--------  -----  -----------  --------------------  -------  -------  -------  ------------  -------  -------
swp1      65101  established  2025-05-09T11:26:39Z  0:01:34  36       36       ipv4-unicast  0        0
swp2      65102  established  2025-05-09T11:26:39Z  0:01:34  36       36       ipv4-unicast  0        0
```

2. **On leaf01, verify BGP peering between leaf and spine**

```
cumulus@leaf01:mgmt:~$ sudo vtysh -c "show ip bgp summary"

IPv4 Unicast Summary (VRF default):
BGP router identifier 10.255.255.1, local AS number 65101 vrf-id 0
BGP table version 5
RIB entries 9, using 1728 bytes of memory
Peers 1, using 20 KiB of memory

Neighbor        V       AS   MsgRcvd   MsgSent   TblVer  InQ OutQ  Up/Down State/PfxRcd   PfxSnt Desc
spine01(swp51)  4    65201       92        93        0    0    0 00:04:18            0        0 N/A

Total number of neighbors 1
```

or

```
cumulus@leaf01:mgmt:~$ nv show vrf default router bgp neighbor brief

AS - Remote Autonomous System, Uptime - BGP session up time, ResetTime - Last
connection reset time, Afi-Safi - Address family, PfxSent - Transmitted prefix
counter, PfxRcvd - Recieved prefix counter

Neighbor  AS     State       Uptime   ResetTime  MsgRcvd  MsgSent  Afi-Safi      PfxSent  PfxRcvd
--------  -----  ----------- -------  ---------  -------  -------  ------------  -------  -------
swp51     65201  established 0:22:05  0:22:07    447      447      ipv4-unicast  0        0
```

1. **On leaf02, verify BGP peering between leaf and spine**

```
cumulus@leaf02:mgmt:~$ sudo vtysh -c "show ip bgp summary"

IPv4 Unicast Summary (VRF default):
BGP router identifier 10.255.255.2, local AS number 65102 vrf-id 0
BGP table version 5
RIB entries 9, using 1728 bytes of memory
Peers 1, using 20 KiB of memory

Neighbor        V         AS   MsgRcvd   MsgSent   TblVer  InQ OutQ  Up/Down State/PfxRcd   PfxSnt Desc
cumulus(swp51)  4      65201        96        97        0    0    0 00:04:31            0        0 N/A

Total number of neighbors 1
```

Or

```
cumulus@leaf02:mgmt:~$ nv show vrf default router bgp neighbor brief

AS - Remote Autonomous System, PeerEstablishedTime - Peer established time in
UTC format, UpTime - Last connection reset time in days,hours:min:sec, Afi-Safi
- Address family, PfxSent - Transmitted prefix counter, PfxRcvd - Recieved
prefix counter

Neighbor  AS     State        PeerEstablishedTime   UpTime   MsgRcvd  MsgSent  Afi-Safi      PfxSent  PfxRcvd
--------  -----  -----------  --------------------  -------  -------  -------  ------------  -------  -------
swp51     65201  established  2025-05-09T11:26:39Z  1:07:03  1346     1347     ipv4-unicast  0        0
```

Important things to observe:

> **The BGP neighbor shows the hostname of the BGP peer**
> **Only the peer is up, no routes are being advertised yet**
> **The BGP router identifier uses the loopback address**
> **You can use either NVUE (`nv`) commands or `vtysh` to observe BGP peering.**

Advertise Loopback and SVI subnets into fabric

Routing Advertisement Configuration			
	leaf01	**leaf02**	**spine01**
Subnets to be advertised	**10.255.255.1/32 10.0.10.0/24**	**10.255.255.2/32 10.0.20.0/24**	**10.255.255.101/32**

2. **On spine01, advertise the loopback address into BGP. Apply to the running configuration.**

```
cumulus@spine01:mgmt:~$ nv set vrf default router bgp address-family ipv4-unicast network 10.255.255.101/32
cumulus@spine01:mgmt:~$ nv config apply
```

3. **On leaf01, advertise loopback address into BGP.**

```
cumulus@leaf01:mgmt:~$ nv set vrf default router bgp address-family ipv4-unicast network 10.255.255.1/32
```

4. **On leaf01, advertise subnet for `VLAN10`.**

```
cumulus@leaf01:mgmt:~$ nv set vrf default router bgp address-family ipv4-unicast network 10.0.10.0/24
```

5. **On leaf01, apply to the running configuration.**

```
cumulus@leaf01:mgmt:~$ nv config apply
```

6. **On leaf02, repeat configuration steps [2-5] with the corresponding IP subnets.**

```
cumulus@leaf02:mgmt:~$ nv set vrf default router bgp address-family ipv4-unicast network 10.255.255.2/32
cumulus@leaf02:mgmt:~$ nv set vrf default router bgp address-family ipv4-unicast network 10.0.20.0/24
cumulus@leaf02:mgmt:~$ nv config apply
```

A code snippet is provided for easy copy and pasting into Cumulus:

```
nv set vrf default router bgp address-family ipv4-unicast network 10.255.255.2/32
nv set vrf default router bgp address-family ipv4-unicast network 10.0.20.0/24
nv config apply
```

Verify BGP prefixes

7. **On spine01, verify routes are learned.**

```
cumulus@spine01:mgmt:~$ sudo vtysh -c "show ip bgp ipv4 unicast"

BGP table version is 5, local router ID is 10.255.255.101, vrf id 0
Default local pref 100, local AS 65201
Status codes:  s suppressed, d damped, h history, u unsorted, * valid, > best, = multipath,
           i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network         Next Hop          Metric LocPrf Weight Path
*> 10.0.10.0/24     swp1                 0           0 65101 i
*> 10.0.20.0/24     swp2                 0           0 65102 i
*> 10.255.255.1/32  swp1                 0           0 65101 i
*> 10.255.255.2/32  swp2                 0           0 65102 i
*> 10.255.255.101/32
                0.0.0.0(spine01)
                                  0       32768 i

Displayed  5 routes and 5 total paths
```

Or

```
cumulus@spine01:mgmt:~$ nv show vrf default router bgp address-family ipv4-unicast route

PathCount - Number of paths present for the prefix, MultipathCount - Number of
paths that are part of the ECMP, DestFlags - * - bestpath-exists, w - fib-wait-
for-install, s - fib-suppress, i - fib-installed, x - fib-install-failed

Prefix            PathCount  MultipathCount  DestFlags
-----------------  ---------  --------------  ---------
10.0.10.0/24       1          1               *
10.0.20.0/24       1          1               *
10.255.255.1/32    1          1               *
10.255.255.2/32    1          1               *
10.255.255.101/32  1          1               *
```

Important things to observe:

> **AS PATH identifies where routes are originating**
> **NEXT HOP is the interface and not an IP address because of BGP unnumbered**
> **Where the next hop is equal to 0.0.0.0, that route is originated locally.**

Verify server connectivity

8. **On Server01, ping to Server02 (10.0.20.102)**

```
ubuntu@server01:~$ ping 10.0.20.102
PING 10.0.20.102 (10.0.20.102) 56(84) bytes of data.
64 bytes from 10.0.20.102: icmp_seq=1 ttl=61 time=9.86 ms
64 bytes from 10.0.20.102: icmp_seq=2 ttl=61 time=5.96 ms
64 bytes from 10.0.20.102: icmp_seq=3 ttl=61 time=5.80 ms
^C
--- 10.0.20.102 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 5.806/7.211/9.864/1.877 ms
```

9. **On Server01, traceroute to Server02. Identify the hops.**

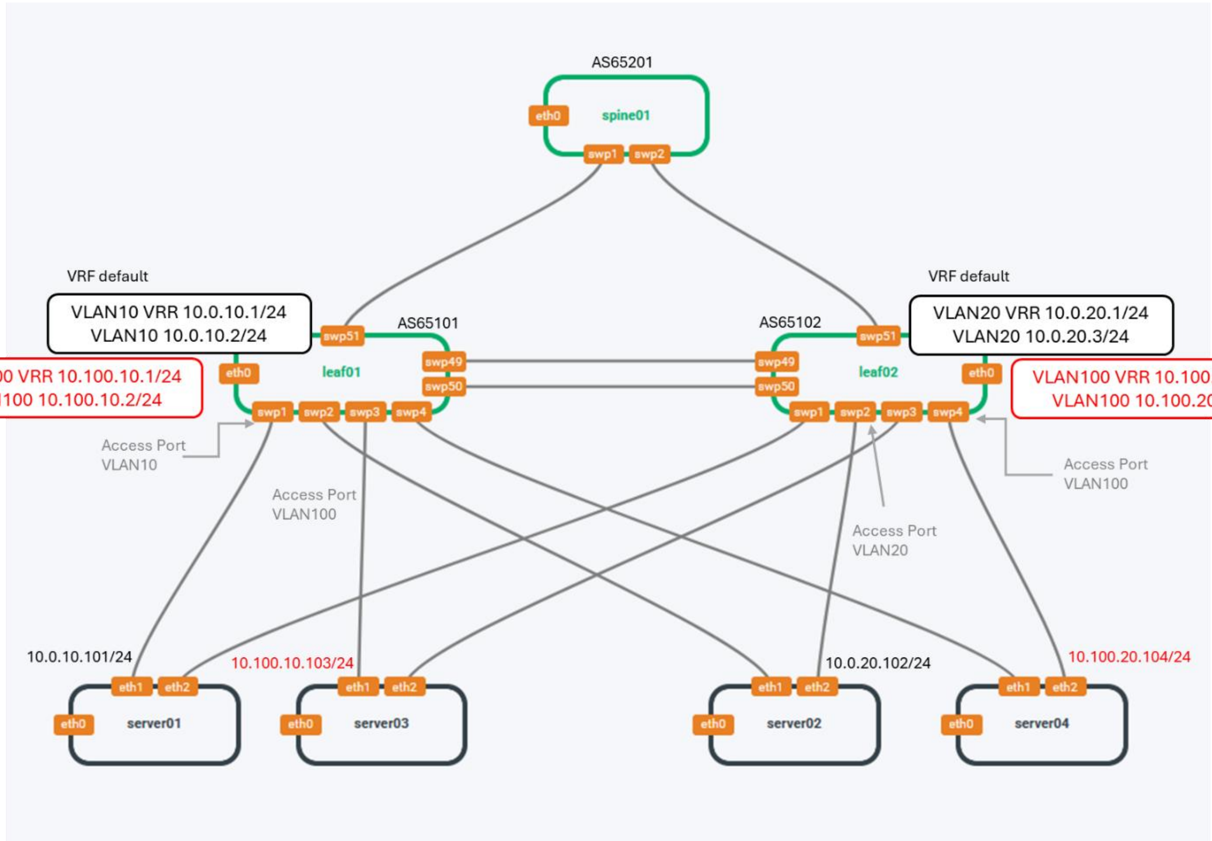
```
ubuntu@server01:~$ traceroute 10.0.20.102
traceroute to 10.0.20.102 (10.0.20.102), 30 hops max, 60 byte packets
 1  10.0.10.1 (10.0.10.1)  1.280 ms  1.389 ms  1.553 ms
 2  10.255.255.101 (10.255.255.101)  4.702 ms  4.679 ms  4.789 ms
 3  10.255.255.2 (10.255.255.2)  8.438 ms  8.877 ms  9.476 ms
 4  10.0.20.102 (10.0.20.102)  9.541 ms  9.766 ms  13.549 ms
cumulus@server01:~$
```

Important things to observe:

> **With Unnumbered interfaces, traceroute (ICMP source interface) packets come from the loopback ipv4 address of the node.**

Start configuring VRF and member interfaces

We will add our changes on top of the BGP configuration we have done in the previous section of this lab. This part is a continuation of lab3, therefore configuring BGP in previous steps [1-17] is a prerequisite to continue with this section of the lab.
By the end of this section, we will be configuring the following topology:



10. **To start VRF-lite section of the lab, first create a VRF named 'RED', create a separate VLAN (100) and SVI (VLAN100) that will be mapped to this VRF and configure interfaces.**
We will use server03 and server04 to verify the connectivity inside vrf RED.

Leaf01:

```
cumulus@leaf01:mgmt:~$ nv set vrf RED table auto
cumulus@leaf01:mgmt:~$ nv set bridge domain br_default vlan 100
cumulus@leaf01:mgmt:~$ nv set interface swp3 bridge domain br_default access 100
cumulus@leaf01:mgmt:~$ nv set interface vlan100 ip vrr enable on
cumulus@leaf01:mgmt:~$ nv set interface vlan100 ip vrr state up
cumulus@leaf01:mgmt:~$ nv set interface vlan100 type svi
cumulus@leaf01:mgmt:~$ nv set interface vlan100 ip address 10.100.10.2/24
cumulus@leaf01:mgmt:~$ nv set interface vlan100 ip vrr address 10.100.10.1/24
cumulus@leaf01:mgmt:~$ nv set interface vlan100 ip vrr mac-address 00:00:00:00:2a:10
cumulus@leaf01:mgmt:~$ nv set interface swp51.100,vlan100 ip vrf RED
cumulus@leaf01:mgmt:~$ nv set interface swp51.100,vlan100 vlan 100
```

Leaf02:

```
cumulus@leaf02:mgmt:~$ nv set vrf RED table auto
cumulus@leaf02:mgmt:~$ nv set bridge domain br_default vlan 100
cumulus@leaf02:mgmt:~$ nv set interface swp4 bridge domain br_default access 100
cumulus@leaf02:mgmt:~$ nv set interface vlan100 ip vrr enable on
cumulus@leaf02:mgmt:~$ nv set interface vlan100 ip vrr state up
cumulus@leaf02:mgmt:~$ nv set interface vlan100 type svi
cumulus@leaf02:mgmt:~$ nv set interface vlan100 ip address 10.100.20.2/24
cumulus@leaf02:mgmt:~$ nv set interface vlan100 ip vrr address 10.100.20.1/24
cumulus@leaf02:mgmt:~$ nv set interface vlan100 ip vrr mac-address 00:00:00:00:2a:20
cumulus@leaf02:mgmt:~$ nv set interface swp51.100,vlan100 ip vrf RED
cumulus@leaf02:mgmt:~$ nv set interface swp51.100,vlan100 vlan 100
```

Spine01:

```
cumulus@spine01:mgmt:~$ nv set vrf RED table auto
cumulus@spine01:mgmt:~$ nv set interface swp1.100 base-interface swp1
cumulus@spine01:mgmt:~$ nv set interface swp1.100,swp2.100 ip vrf RED
cumulus@spine01:mgmt:~$ nv set interface swp1.100,swp2.100 type sub
cumulus@spine01:mgmt:~$ nv set interface swp1.100,swp2.100 vlan 100
cumulus@spine01:mgmt:~$ nv set interface swp2.100 base-interface swp2
```

Configure BGP routing for VRF-lite

11. **Then configure BGP unnumbered peering for vrf RED, advertise vlan100 SVI interface prefix on both leaf switches:**
Leaf01:

```
cumulus@leaf01:mgmt:~$ nv set vrf RED router bgp address-family ipv4-unicast enable on
cumulus@leaf01:mgmt:~$ nv set vrf RED router bgp address-family ipv4-unicast network 10.100.10.0/24
cumulus@leaf01:mgmt:~$ nv set vrf RED router bgp autonomous-system 65101
cumulus@leaf01:mgmt:~$ nv set vrf RED router bgp enable on
cumulus@leaf01:mgmt:~$ nv set vrf RED router bgp neighbor swp51.100 remote-as external
cumulus@leaf01:mgmt:~$ nv set vrf RED router bgp neighbor swp51.100 type unnumbered
cumulus@leaf01:mgmt:~$ nv set vrf RED router bgp path-selection multipath aspath-ignore on
```

Leaf02:

```
cumulus@leaf02:mgmt:~$ nv set vrf RED router bgp address-family ipv4-unicast enable on
cumulus@leaf02:mgmt:~$ nv set vrf RED router bgp address-family ipv4-unicast network 10.100.20.0/24
cumulus@leaf02:mgmt:~$ nv set vrf RED router bgp autonomous-system 65102
cumulus@leaf02:mgmt:~$ nv set vrf RED router bgp enable on
cumulus@leaf02:mgmt:~$ nv set vrf RED router bgp neighbor swp51.100 remote-as external
cumulus@leaf02:mgmt:~$ nv set vrf RED router bgp neighbor swp51.100 type unnumbered
cumulus@leaf02:mgmt:~$ nv set vrf RED router bgp path-selection multipath aspath-ignore on
```

Spine01:

```
cumulus@spine01:mgmt:~$ nv set vrf RED router bgp autonomous-system 65201
cumulus@spine01:mgmt:~$ nv set vrf RED router bgp enable on
cumulus@spine01:mgmt:~$ nv set vrf RED router bgp neighbor swp1.100 remote-as external
cumulus@spine01:mgmt:~$ nv set vrf RED router bgp neighbor swp1.100 type unnumbered
cumulus@spine01:mgmt:~$ nv set vrf RED router bgp neighbor swp2.100 remote-as external
cumulus@spine01:mgmt:~$ nv set vrf RED router bgp neighbor swp2.100 type unnumbered
cumulus@spine01:mgmt:~$ nv set vrf RED router bgp path-selection multipath aspath-ignore on
```

Verify routing table for each VRF

12. **Verify default and RED vrf routing tables and check if prefixes are in the correct vrf routing tables.**

Leaf01:

```
cumulus@leaf01:mgmt:~$ nv show vrf default router rib ipv4 route

Flags - * - selected, q - queued, o - offloaded, i - installed, S - fib-
selected, x - failed

Route          Protocol  Distance  Uptime             NHGId  Metric  Flags
----------------- --------- -------- -------------------- ----- ------ -----
10.0.10.0/24      connected 0       2025-08-20T00:33:49Z  36    1024   i
              connected 0       2025-08-20T00:33:49Z  37    0      *Si
10.0.10.1/32      local     0       2025-08-20T00:33:49Z  36    0      *Si
10.0.10.2/32      local     0       2025-08-20T00:33:49Z  37    0      *Si
10.0.20.0/24      bgp       20      2025-08-20T07:31:23Z  47    0      *Si
10.255.255.1/32   connected 0       2025-08-20T00:33:48Z  17    0      *Si
              local     0       2025-08-20T00:33:48Z  17    0      i
10.255.255.2/32   bgp       20      2025-08-20T07:31:23Z  47    0      *Si
10.255.255.101/32 bgp       20      2025-08-20T07:31:23Z  47    0      *Si
cumulus@leaf01:mgmt:~$



cumulus@leaf01:mgmt:~$ nv show vrf RED router rib ipv4 route

Flags - * - selected, q - queued, o - offloaded, i - installed, S - fib-
selected, x - failed
```

```
Route          Protocol  Distance  Uptime               NHGId  Metric  Flags
-------------- --------- -------- -------------------- ----- ------ -----
0.0.0.0/0      kernel    255      2025-08-20T08:50:59Z  22    8192   *Si
10.100.10.0/24 connected 0        2025-08-20T08:51:00Z  36    1024   i
               connected 0        2025-08-20T08:51:00Z  37    0      *Si
10.100.10.1/32 local     0        2025-08-20T08:51:00Z  36    0      *Si
10.100.10.2/32 local     0        2025-08-20T08:51:00Z  37    0      *Si
10.100.20.0/24 bgp       20       2025-08-20T08:55:35Z  52    0      *Si
```

Leaf02:

```
cumulus@leaf02:mgmt:~$ nv show vrf default router rib ipv4 route

Flags - * - selected, q - queued, o - offloaded, i - installed, S - fib-
selected, x - failed

Route             Protocol  Distance  Uptime               NHGId  Metric  Flags
----------------- --------- -------- -------------------- ----- ------ -----
10.0.10.0/24      bgp       20       2025-08-20T07:31:22Z  45    0      *Si
10.0.20.0/24      connected 0        2025-08-20T00:33:48Z  36    1024   i
                  connected 0        2025-08-20T00:33:48Z  37    0      *Si
10.0.20.1/32      local     0        2025-08-20T00:33:48Z  36    0      *Si
10.0.20.3/32      local     0        2025-08-20T00:33:48Z  37    0      *Si
10.255.255.1/32   bgp       20       2025-08-20T07:31:22Z  45    0      *Si
10.255.255.2/32   connected 0        2025-08-20T00:33:47Z  17    0      *Si
                  local     0        2025-08-20T00:33:47Z  17    0      i
10.255.255.101/32 bgp       20       2025-08-20T07:31:22Z  45    0      *Si
cumulus@leaf02:mgmt:~$
cumulus@leaf02:mgmt:~$ nv show vrf RED router rib ipv4 route

Flags - * - selected, q - queued, o - offloaded, i - installed, S - fib-
selected, x - failed

Route          Protocol  Distance  Uptime               NHGId  Metric  Flags
-------------- --------- -------- -------------------- ----- ------ -----
0.0.0.0/0      kernel    255      2025-08-20T08:50:59Z  22    8192   *Si
10.100.10.0/24 bgp       20       2025-08-20T08:55:35Z  48    0      *Si
10.100.20.0/24 connected 0        2025-08-20T08:51:00Z  34    1024   i
               connected 0        2025-08-20T08:51:00Z  35    0      *Si
10.100.20.1/32 local     0        2025-08-20T08:51:00Z  34    0      *Si
10.100.20.2/32 local     0        2025-08-20T08:51:00Z  35    0      *Si
cumulus@leaf02:mgmt:~$
```

Spine01:

```
cumulus@leaf01:mgmt:~$ nv show vrf default router rib ipv4 route

Flags - * - selected, q - queued, o - offloaded, i - installed, S - fib-
selected, x - failed

Route             Protocol  Distance  Uptime               NHGId  Metric  Flags
----------------- --------- -------- -------------------- ----- ------ -----
10.0.10.0/24      connected 0        2025-08-20T00:33:49Z  36    1024   i
                  connected 0        2025-08-20T00:33:49Z  37    0      *Si
10.0.10.1/32      local     0        2025-08-20T00:33:49Z  36    0      *Si
10.0.10.2/32      local     0        2025-08-20T00:33:49Z  37    0      *Si
10.0.20.0/24      bgp       20       2025-08-20T07:31:23Z  47    0      *Si
10.255.255.1/32   connected 0        2025-08-20T00:33:48Z  17    0      *Si
                  local     0        2025-08-20T00:33:48Z  17    0      i
10.255.255.2/32   bgp       20       2025-08-20T07:31:23Z  47    0      *Si
10.255.255.101/32 bgp       20       2025-08-20T07:31:23Z  47    0      *Si
cumulus@leaf01:mgmt:~$
cumulus@spine01:mgmt:~$ nv show vrf RED router rib ipv4 route

Flags - * - selected, q - queued, o - offloaded, i - installed, S - fib-
selected, x - failed

Route          Protocol Distance  Uptime               NHGId  Metric  Flags
-------------- -------- -------- -------------------- ----- ------ -----
0.0.0.0/0      kernel   255      2025-08-20T08:50:59Z  24    8192   *Si
10.100.10.0/24 bgp      20       2025-08-20T08:55:34Z  41    0      *Si
10.100.20.0/24 bgp      20       2025-08-20T08:55:34Z  40    0      *Si
```

Verify connectivity

13. **Verify end to end connectivity from servers for the VRF they belong to (intra-VRF):**

```
ubuntu@server01:~$ ping 10.0.20.102 -c 3
PING 10.0.20.102 (10.0.20.102) 56(84) bytes of data.
64 bytes from 10.0.20.102: icmp_seq=1 ttl=61 time=1.84 ms
64 bytes from 10.0.20.102: icmp_seq=2 ttl=61 time=1.97 ms
64 bytes from 10.0.20.102: icmp_seq=3 ttl=61 time=2.06 ms

--- 10.0.20.102 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.837/1.956/2.060/0.091 ms
ubuntu@server01:~$


ubuntu@server02:~$ ping 10.0.10.101 -c 3
PING 10.0.10.101 (10.0.10.101) 56(84) bytes of data.
64 bytes from 10.0.10.101: icmp_seq=1 ttl=61 time=1.56 ms
64 bytes from 10.0.10.101: icmp_seq=2 ttl=61 time=1.64 ms
64 bytes from 10.0.10.101: icmp_seq=3 ttl=61 time=1.64 ms

--- 10.0.10.101 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 1.562/1.612/1.638/0.035 ms
ubuntu@server02:~$

ubuntu@server03:~$ ping 10.100.20.104 -c 3
PING 10.100.20.104 (10.100.20.104) 56(84) bytes of data.
64 bytes from 10.100.20.104: icmp_seq=1 ttl=61 time=1.99 ms
64 bytes from 10.100.20.104: icmp_seq=2 ttl=61 time=1.87 ms
64 bytes from 10.100.20.104: icmp_seq=3 ttl=61 time=1.76 ms

--- 10.100.20.104 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.764/1.875/1.989/0.091 ms
ubuntu@server03:~$

ubuntu@server04:~$ ping 10.100.10.103 -c 3
PING 10.100.10.103 (10.100.10.103) 56(84) bytes of data.
64 bytes from 10.100.10.103: icmp_seq=1 ttl=61 time=2.80 ms
64 bytes from 10.100.10.103: icmp_seq=2 ttl=61 time=2.08 ms
64 bytes from 10.100.10.103: icmp_seq=3 ttl=61 time=2.12 ms

--- 10.100.10.103 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 2.075/2.331/2.800/0.331 ms
ubuntu@server04:~$
```

14. **Verify that prefixes are NOT leaked between VRFs (inter-VRF):**

```
ubuntu@server01:~$ ping 10.100.20.104 -c 3
PING 10.100.20.104 (10.100.20.104) 56(84) bytes of data.
From 10.0.10.1 icmp_seq=1 Destination Net Unreachable
From 10.0.10.1 icmp_seq=2 Destination Net Unreachable
From 10.0.10.1 icmp_seq=3 Destination Net Unreachable

--- 10.100.20.104 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2048ms

ubuntu@server01:~$

ubuntu@server02:~$ ping 10.100.20.104 -c 3
PING 10.100.20.104 (10.100.20.104) 56(84) bytes of data.
From 10.0.20.1 icmp_seq=1 Destination Net Unreachable
From 10.0.20.1 icmp_seq=2 Destination Net Unreachable
From 10.0.20.1 icmp_seq=3 Destination Net Unreachable

--- 10.100.20.104 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2027ms

ubuntu@server02:~$

ubuntu@server03:~$ ping 10.0.10.101 -c 3
PING 10.0.10.101 (10.0.10.101) 56(84) bytes of data.
From 10.100.10.2 icmp_seq=1 Destination Host Unreachable
From 10.100.10.2 icmp_seq=2 Destination Host Unreachable
From 10.100.10.2 icmp_seq=3 Destination Host Unreachable

--- 10.0.10.101 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2025ms

ubuntu@server03:~$

ubuntu@server04:~$ ping 10.0.20.102 -c 3
PING 10.0.20.102 (10.0.20.102) 56(84) bytes of data.
From 10.100.20.2 icmp_seq=1 Destination Host Unreachable
From 10.100.20.2 icmp_seq=2 Destination Host Unreachable
```

```
From 10.100.20.2 icmp_seq=3 Destination Host Unreachable

--- 10.0.20.102 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2060ms

ubuntu@server04:~$
```

Lab 4: NetQ Configuration

Objective:

This lab will guide you through Nvidia NetQ telemetry and monitoring product which will help customers to monitor data center fabric telemetry, and perform a comprehensive list of health checks and validations. NetQ is part of our Cumulus switch portfolio as a monitoring and telemetry tool and is an essential component of Spectrum-X AI fabric solution. It supports Open Telemetry and Grafana data export.
In this lab, we will go through some basic steps of telemetry and validation checks using NetQ

Dependencies on other Labs:

> **None.**

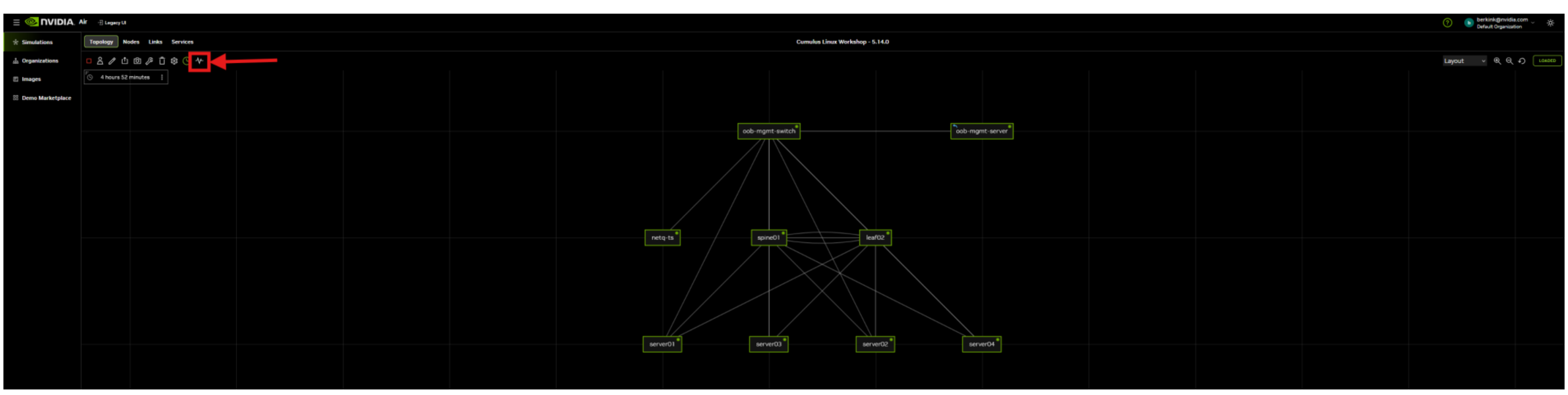
Goals:

> **Login to NetQ SaaS solution and familiarize yourself with the dashboard**
> **Monitor interface counters**
> **Familiarize yourself with the fabric events, BGP events menu**
> **Create validations and view results**

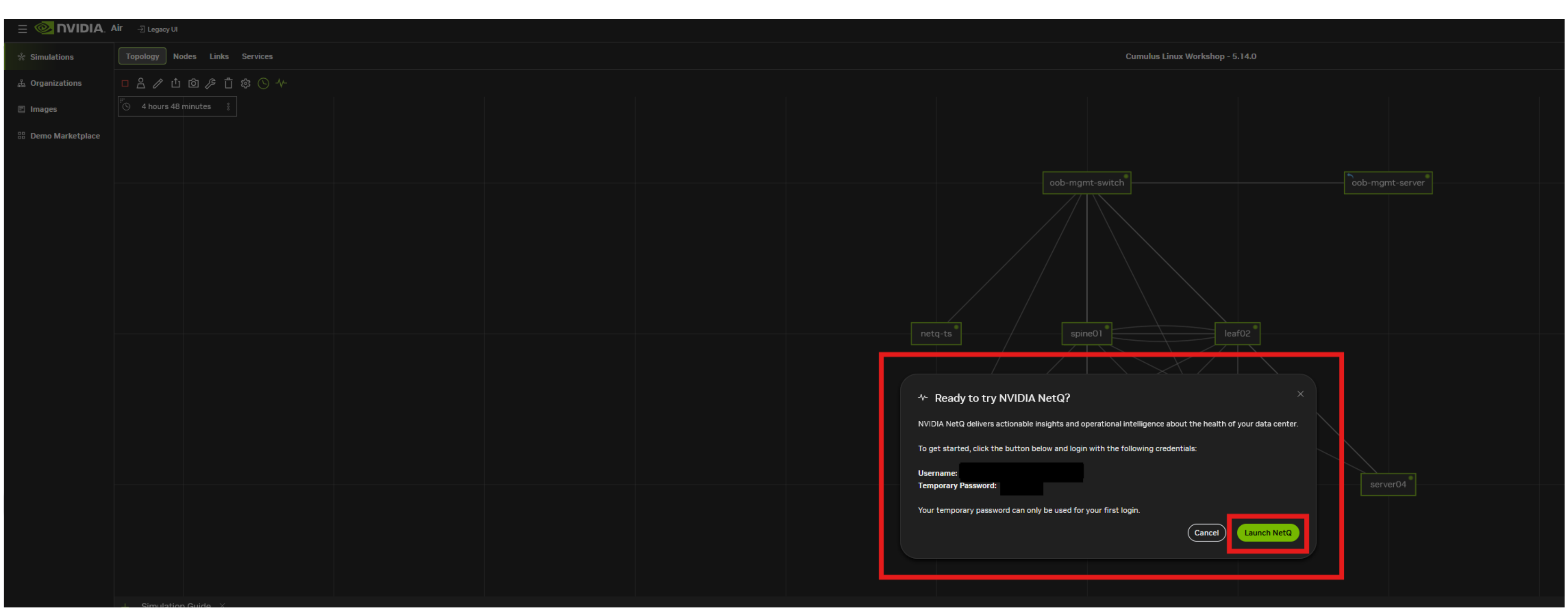
Procedure:

Login

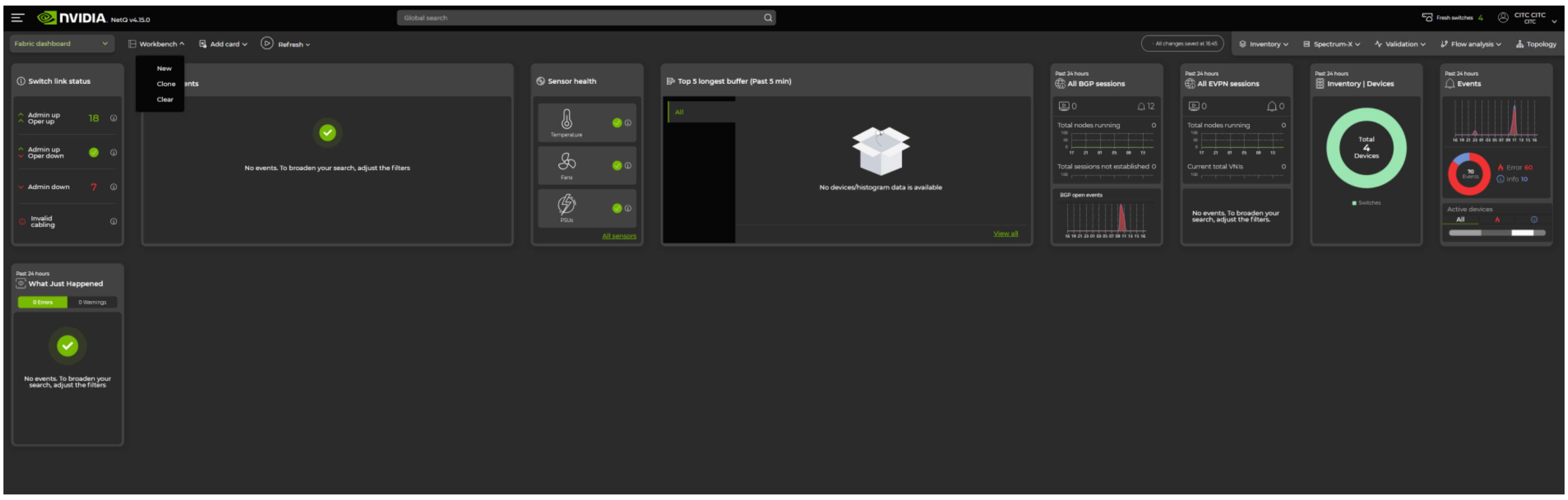
From AIR simulation window, click on the little hearth-beat shaped button to open launch NetQ pop-up.



From the popup, notice your individual Netq SaaS login Username and password and click on Launch NetQ button:

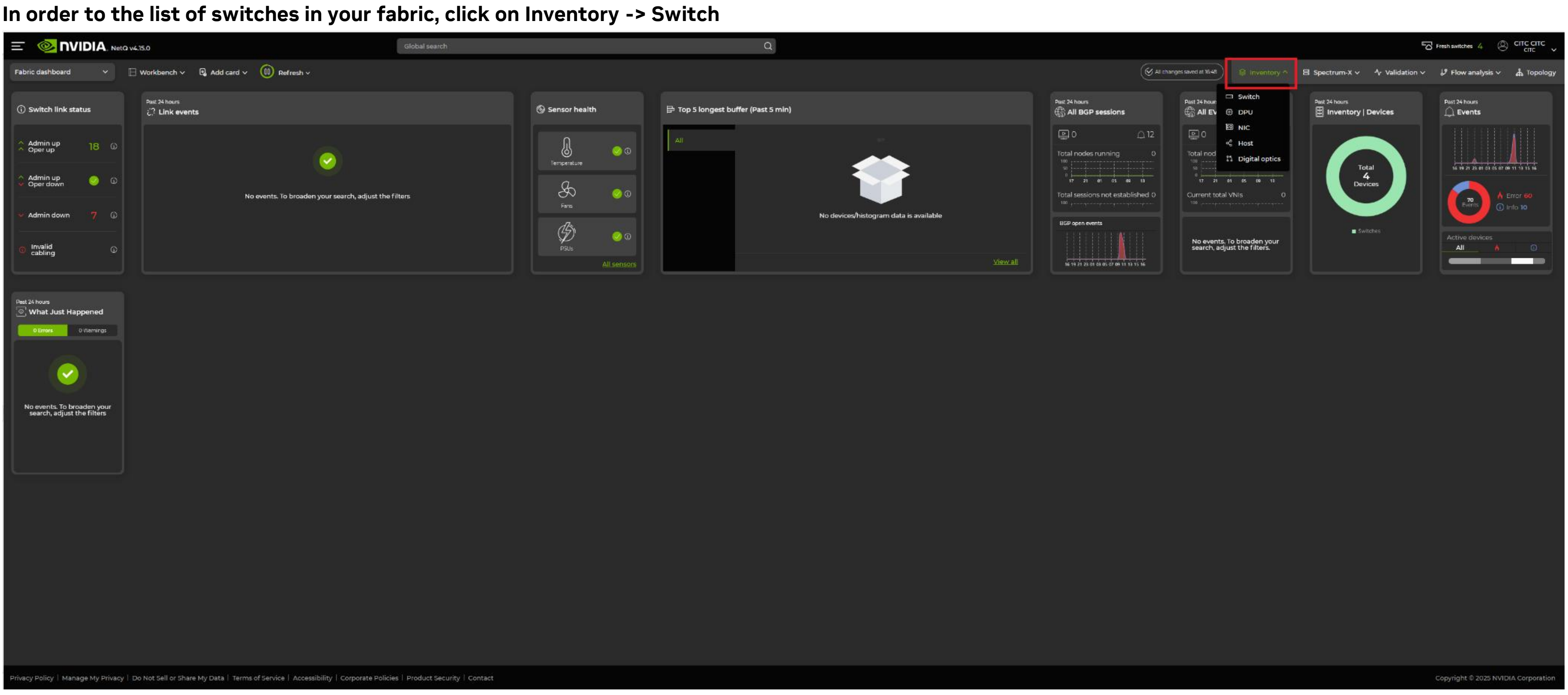


After changing your password and logging in, you will see the NetQ dashboard with all switches in the simulation, already populated in NetQ:

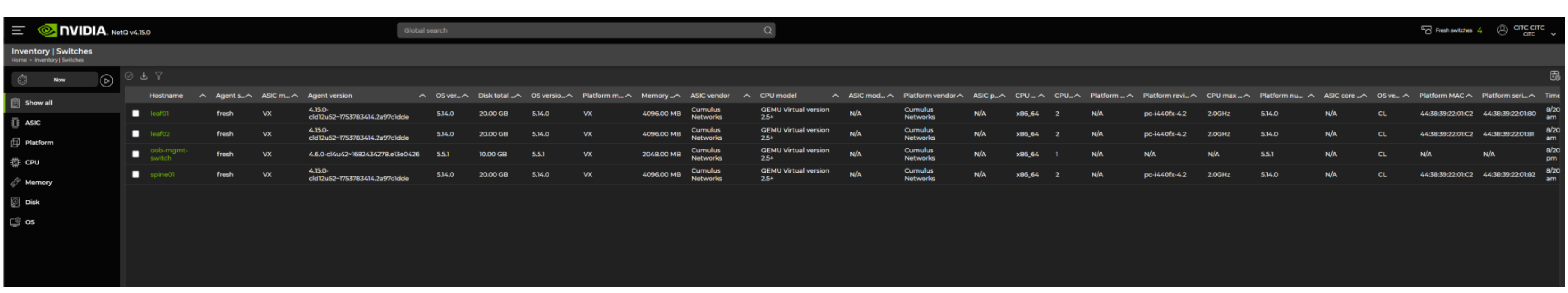


Please notice that NetQ comes with all switches in the simulation already configured. In real life deployments, you need to configure switches and add them into NetQ.

In order to the list of switches in your fabric, click on Inventory -> Switch

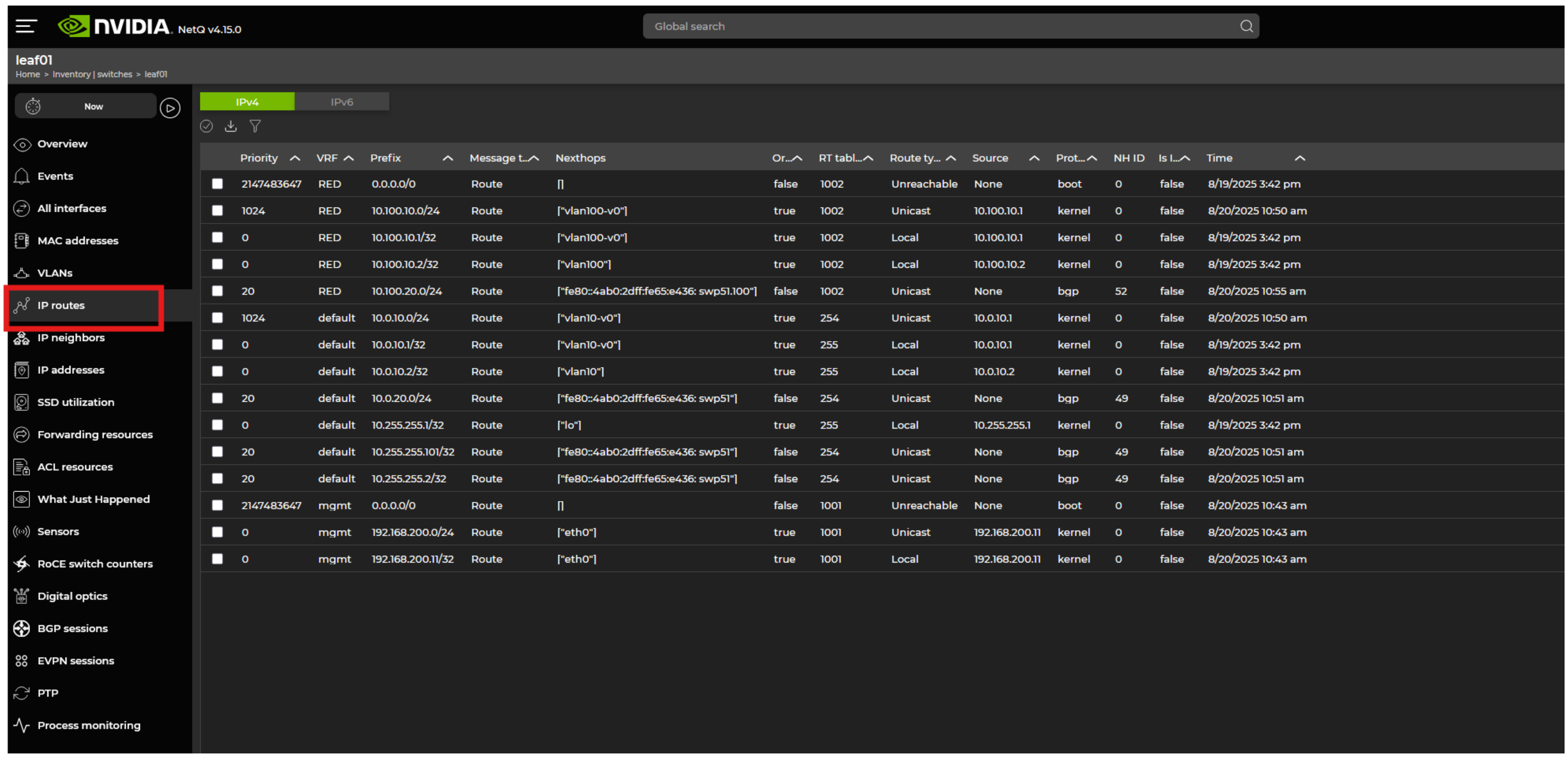


This will take you to Switch inventory list

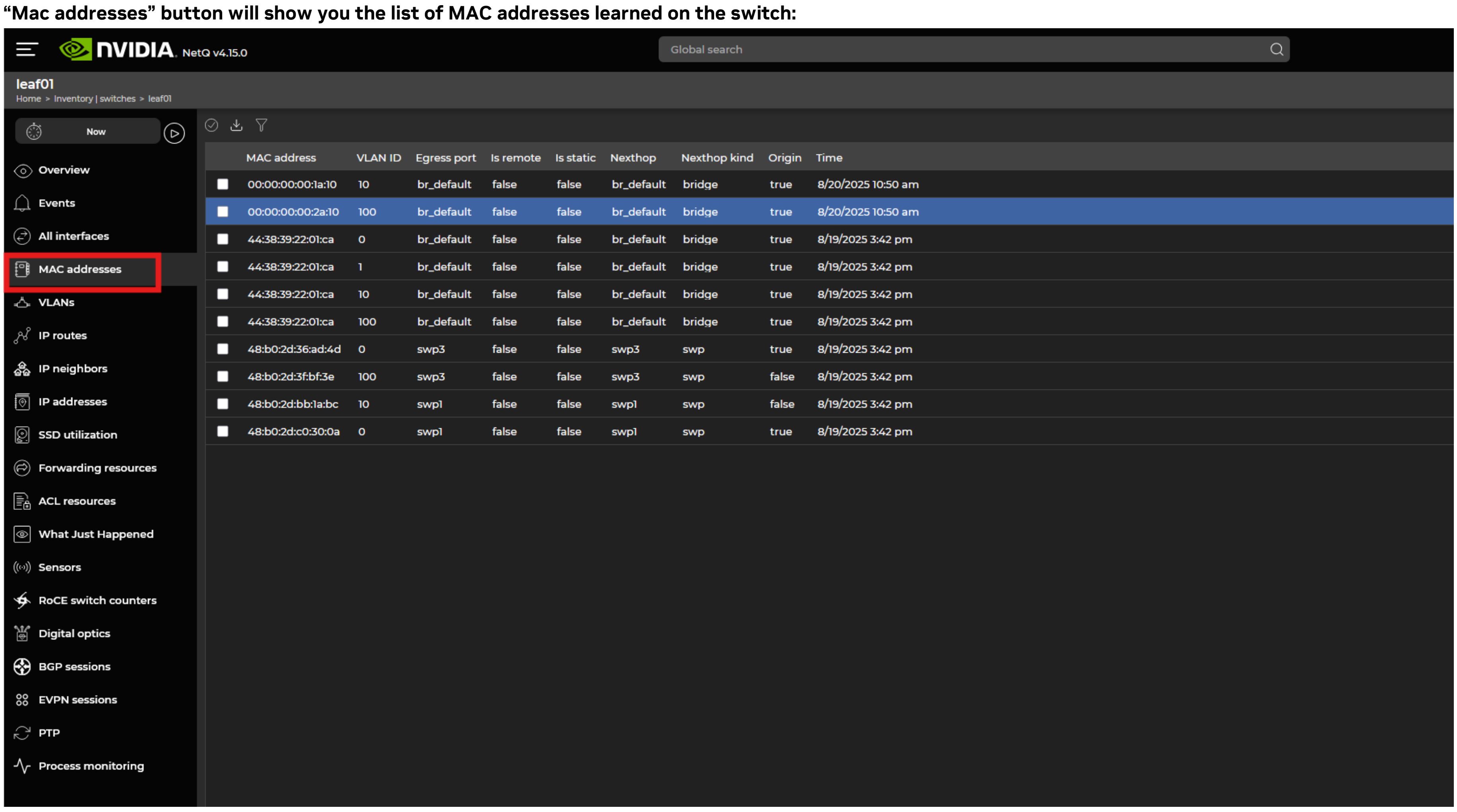


When you click on one of the switches, you will navigate to the switch view, including a list of interfaces configured on the switch and the telemetry collected from the interfaces:

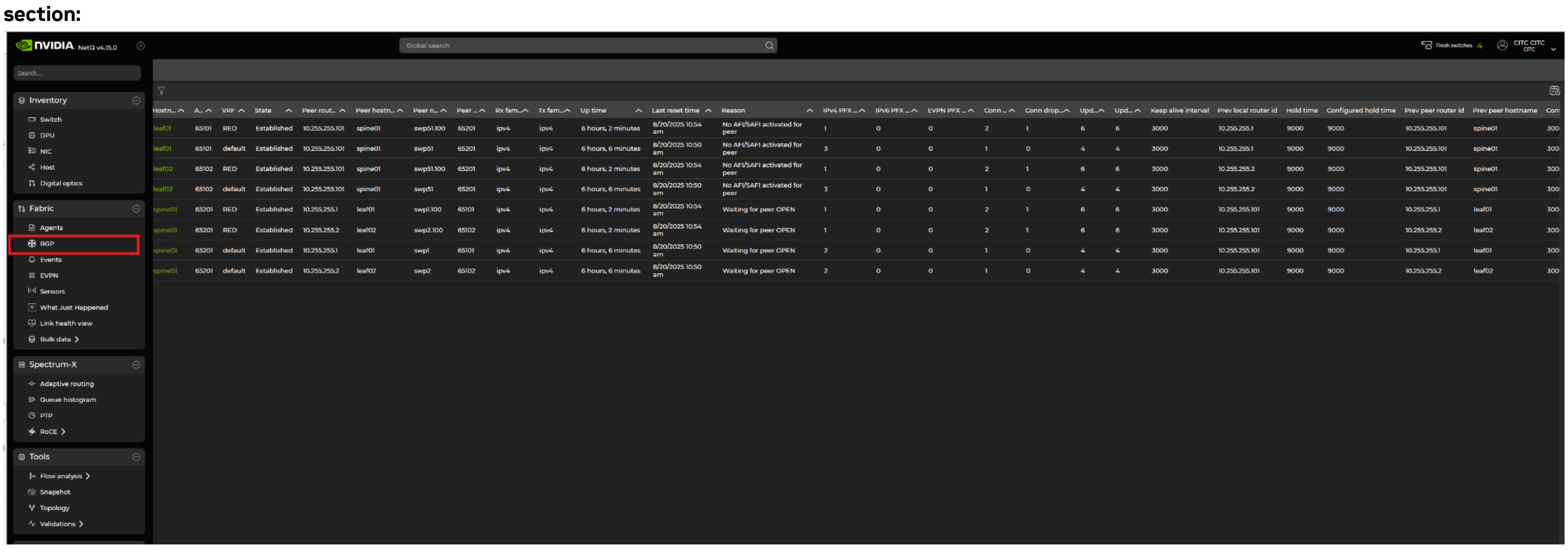
Click on “IP Routes” button to see the list of prefixes in each VRF:



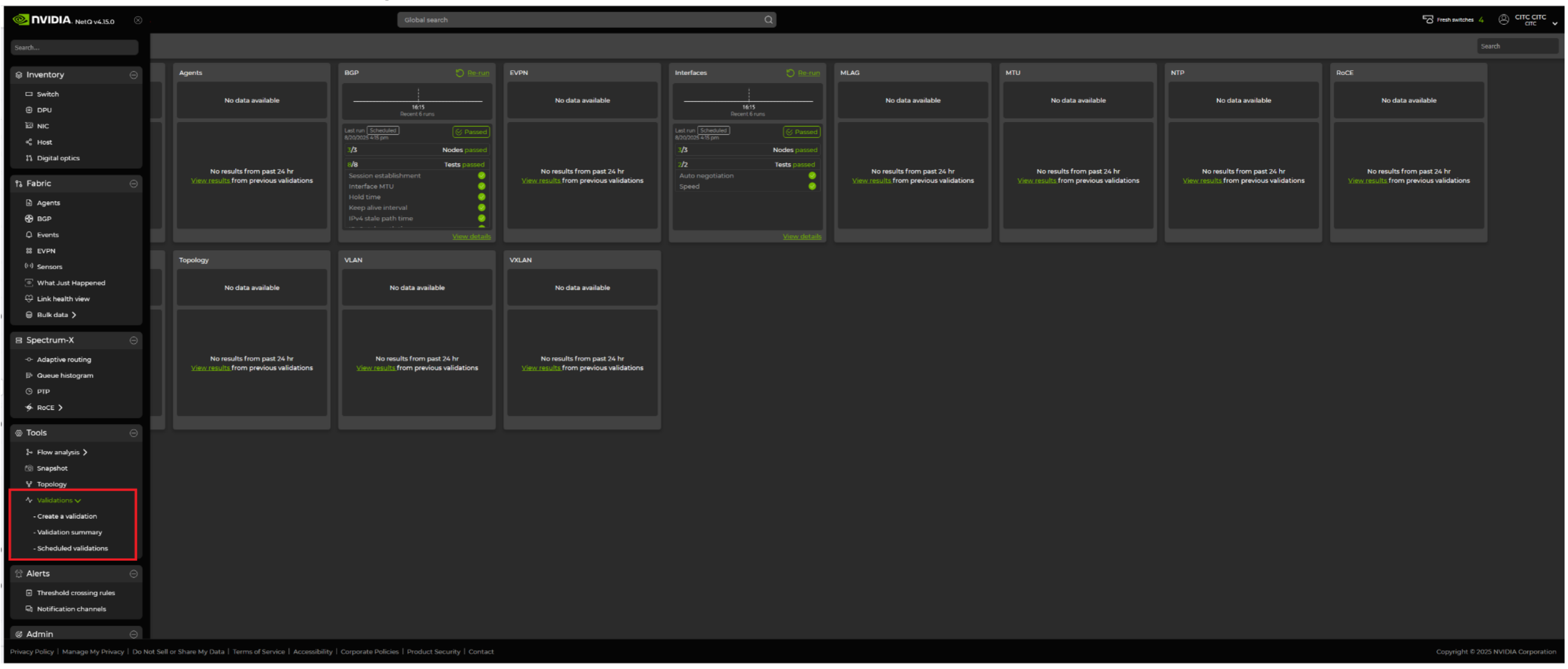
“Mac addresses” button will show you the list of MAC addresses learned on the switch:



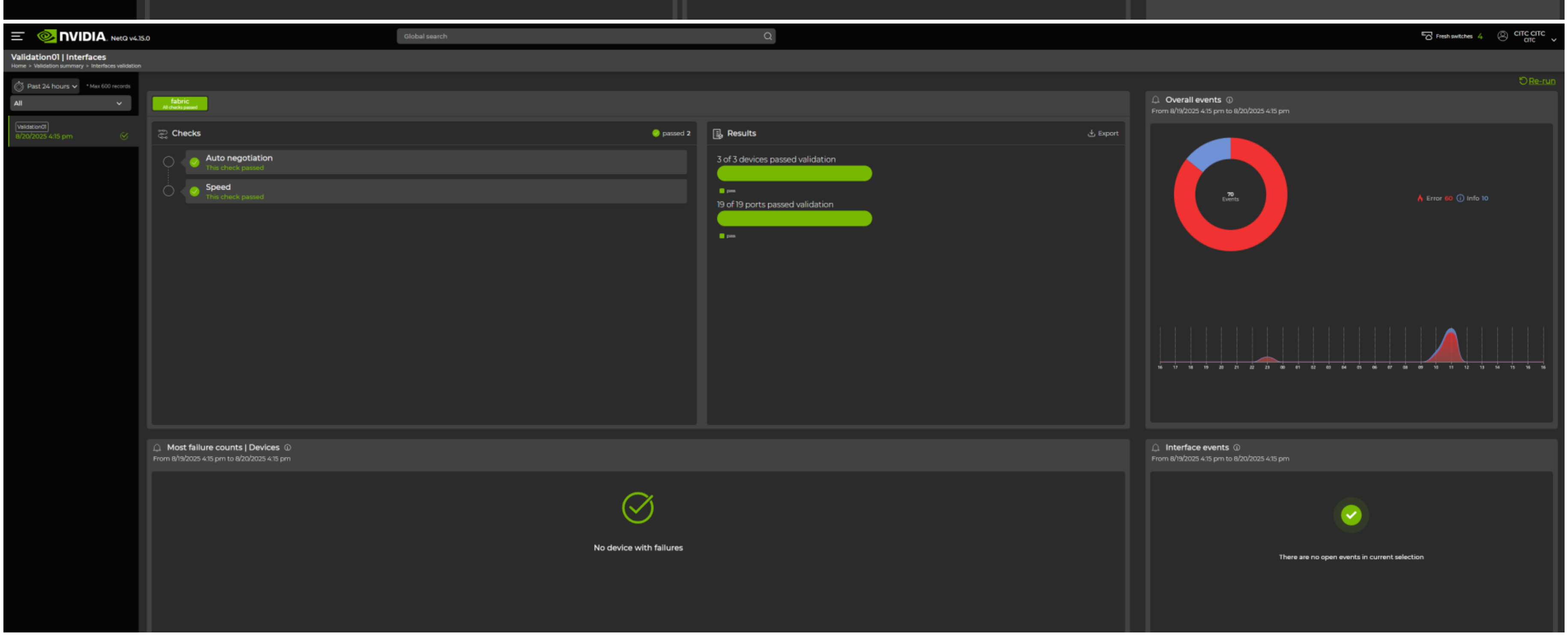
Fabric -> BGP view will take you to the list of all BGP sessions, here you can monitor all the BGP sessions in your fabric and see all up/down events from Events section:



Use Tools-> Validation section to configure validations:



In this example we have validations configured for BGP and Interfaces:



Create your own validations and schedule their running period and observe the results.

This concludes the NVIDIA Cumulus Linux Workshop.

Appendix A: How to use an SSH client to manually connect to the to the lab

SSH Key Prerequisite:

You must have an SSH public and private keypair. Generating one is simple. It is possible you may already have one generated on your box. For *nix users, you may already have a keypair created. Check in your home directory for a .ssh folder that contains id_rsa and id_rsa.pub files.

laptop:~ user$ cat ~/.ssh/id_rsa.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQC7feqSFSAUxpe2qTv77+pEk82C/i4AlXVcOQl5tWBCAq1tPWmZHJCPcElFTjeG2wMYMx2Kmb3kYwrLcwfTk06avziBhjMwIprFiupWC
kykRPm4lOHkiWDS/htpZfBdwuIFXV4MQtCiD9zUhLi0Uq0Is+lVtE1Q0/38N7sSa7FHaVNpDpJOQf3PYVdfhk/BG19WQlyKMUSjOaRrAHUIckiQs2H5Wm198ciKkgl4AxoDM9QB
+flcCl3We52ei5tWqV8CgLehhrdjEXn+iXNdkcg+nGka1syUSYntotally+fake+key+MkEFwD9v16SmJYDK67w5RaHTjBS52UoRjnEEN user@hostname.local

> **Warning:** Please be careful. We can't know in advance if you have existing ssh keys and if you do, if they are used by anything on your machine or for any importar tools in your workflow or line of work. There is risk that ***regenerating default keys may break something*** you do or some other remote logins that depend on your default/existing keys. If you already have default keys, please use them and do not regenerate them.

These links are suggestions only for generating ssh keys. The information on these pages is not owned or controlled by NVIDIA. Use with caution.

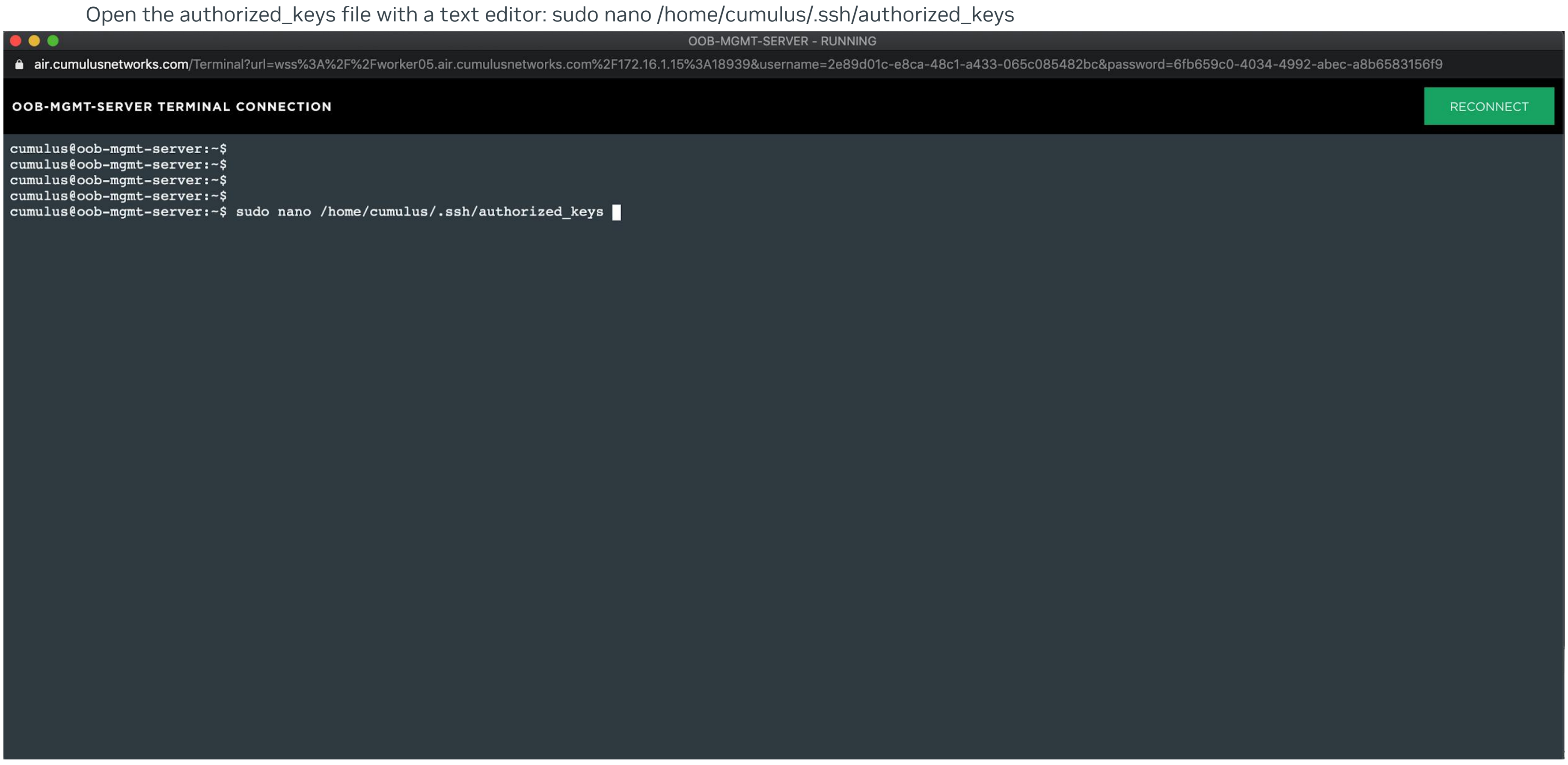
Windows (using putty): https://www.ssh.com/ssh/putty/windows/puttygen
Mac: https://docs.joyent.com/public-cloud/getting-started/ssh-keys/generating-an-ssh-key-manually/manually-generating-your-ssh-key-in-mac-os-x
Ubuntu/linux: https://help.ubuntu.com/community/SSH/OpenSSH/Keys

Your goal is to have an SSH keypair and access to your public key string that looks similar to the key string below. Often the ssh *private key* is password protected. You ought to know the password for your key. NVIDIA/Cumulus does not know the password/passphrase of your ssh private key. If you do not know or have forgotten, you may need to generate a new set of keys for access here.

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQC7feqSFSAUxpe2qTv77+pEk82C/i4AlXVcOQl5tWBCAq1tPWmZHJCPcElFTjeG2wMYMx2Kmb3kYwrLcwfTk06avziBhjMwIprFiup
WCkykRPm4lOHkiWDS/htpZfBdwuIFXV4MQtCiD9zUhLi0Uq0Is+lVtE1Q0/38N7sSa7FHaVNpDpJOQf3PYVdfhk/BG19WQlyKMUSjOaRrAHUIckiQs2H5Wm198ciKkgl4AxoD
M9QB+flcCl3We52ei5tWqV8CgLehhrdjEXn+iXNdkcg+nGka1syUSYntotally+fake+key+MkEFwD9v16SmJYDK67w5RaHTjBS52UoRjnEEN user@hostname.local
```

1. First, you must add your ssh pubkey to the oob-mgmt-server authorized_keys file for the cumulus user. Pop out the console window of the oob-mgmt-server.

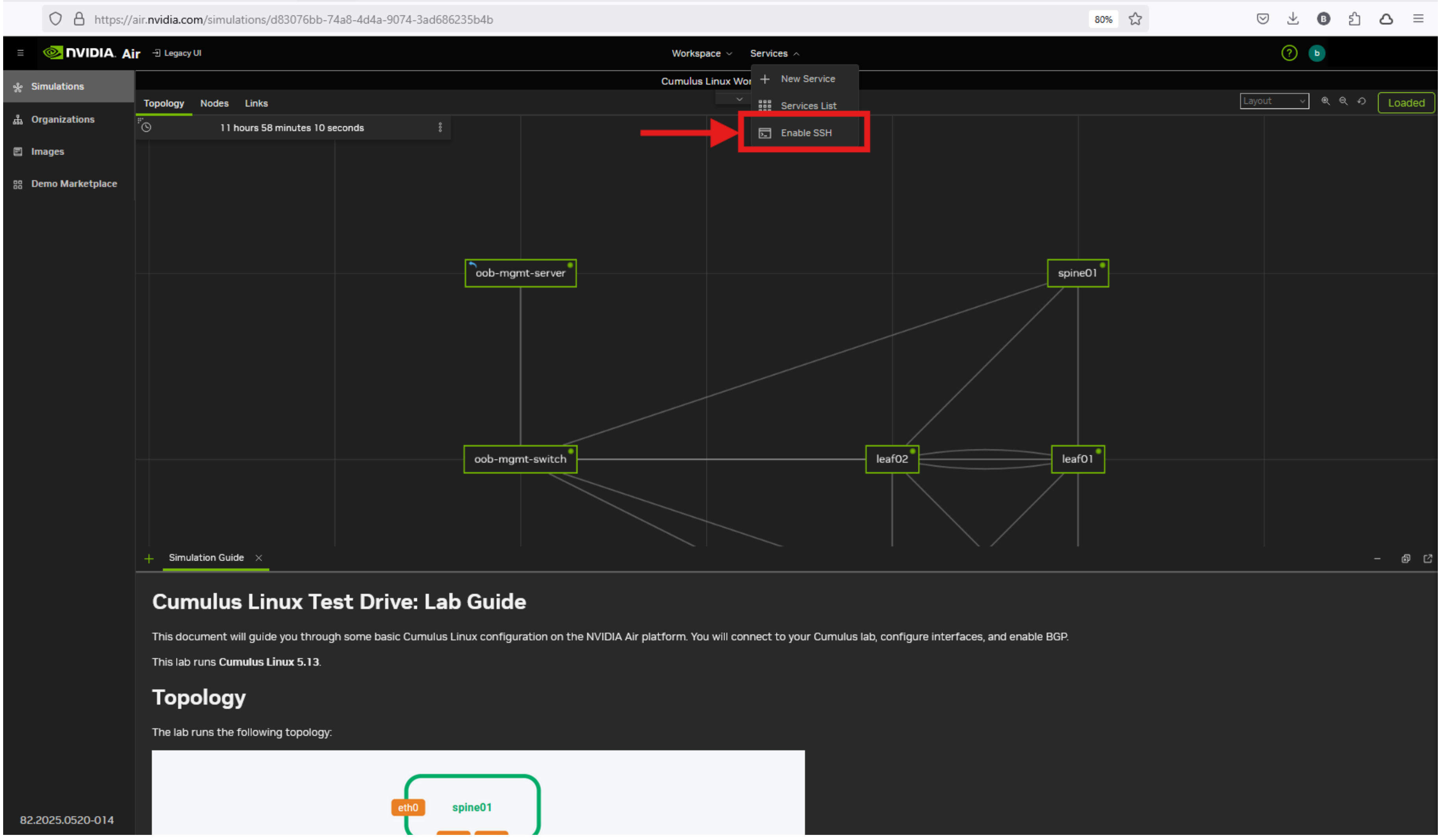
Open the authorized_keys file with a text editor: sudo nano /home/cumulus/.ssh/authorized_keys



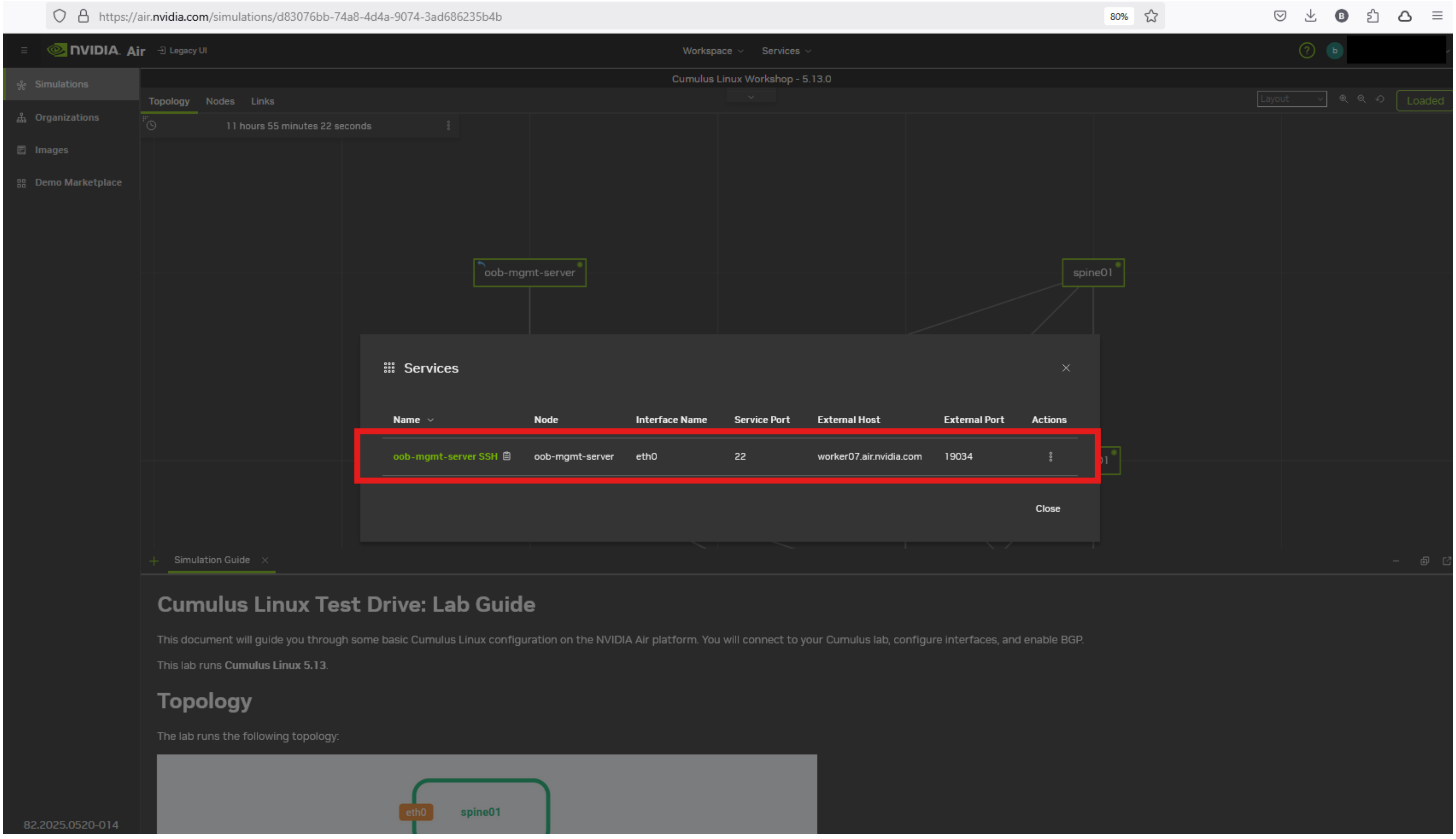
Paste in your ssh pubkey string. It is a long single line string that doesn't wrap in nano.

Save and quit. In nano, save is [ctrl + o], then confirm the filename with [return]. Quit is [ctrl + x]

2. Next, click the "Enable SSH" button under the Services window to expose the SSH service on the oob-mgmt-server to the Internet.



This will cause the SSH service information to populate in the "Services" Panel



Next, click on the hyperlink for the SSH service. If your web browser is configured with an application to handle SSH URLs, then clicking on the link from your browser will automatically launch the application to handle the SSH connection and connect with the correct username, IP address, and port number.

If your browser is unable to handle the SSH URL to automatically launch a default program for SSH, follow the additional steps below to connect manually with your SSH client:

<table>
<tr><th colspan="2">Manual SSH Connection Details</th><td rowspan="5">Service Port: 22
External Host: worker05.air.nvidia.com
External Port: 25230</td></tr>
<tr><td>Username</td><td>ubuntu</td></tr>
<tr><td><s>Password</s></td><td>If prompted for a password, you are being asked for the password/passphrase for your ssh private key</td></tr>
<tr><td>Server Hostname</td><td>Use “External Host” in services box on UI ---------------------></td></tr>
<tr><td>SSH Port</td><td>Use “External Port” in Services box on UI --------------------></td></tr>
</table>

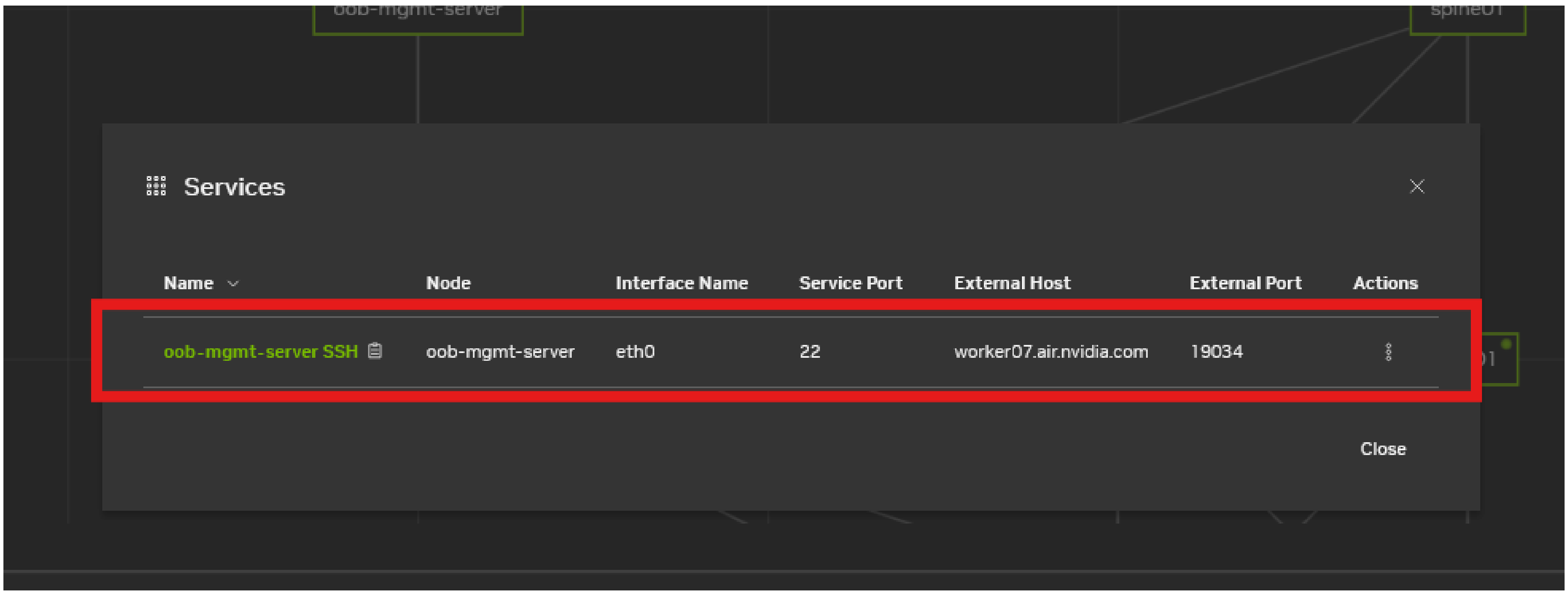
Note: This SSH connection ***does not use the default destination TCP port 22***. Ensure that the external port is specified in your SSH client.
Note: If you are prompted for a password, it is the password to access your SSH private key. This is not a password being requested from the server for authentication against the cumulus user.

To connect via SSH manually, you must have an SSH client installed.

- Windows users: Download PuTTY from https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
- Mac users: Use the *Terminal* application.
- Linux users: Open a Bash shell.

Linux/Mac OS:

The SSH command will follow a format similar to: **ssh ubuntu@workerXX.air.nvidia.com -p XXXXX**
You just need to find your worker and the port number from the information in the AIR UI Services panel:



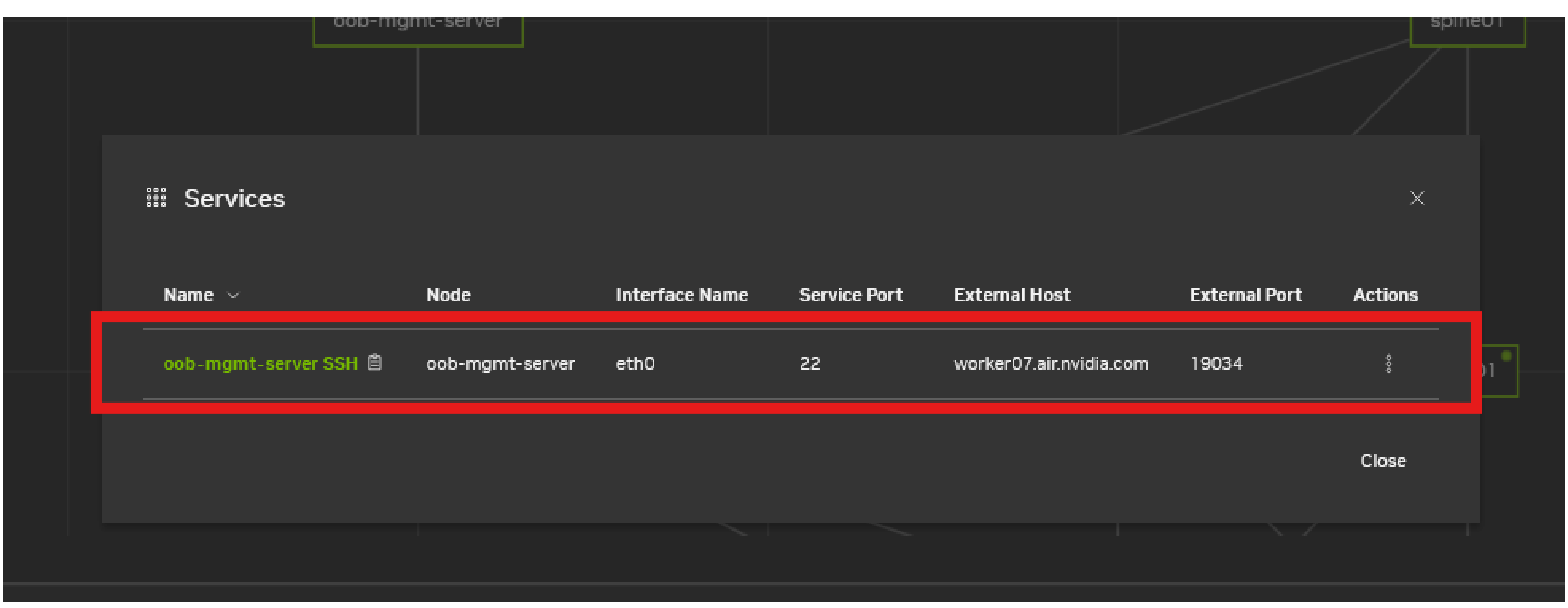
Example:

```
user@laptop$ ssh ubuntu@worker07.air.nvidia.com -p 19034
The authenticity of host '[worker07.air.nvidia.com]:19034 ([147.75.91.219]:19034)' can't be established.
ECDSA key fingerprint is SHA256:2gi+bqJHFCZw1lefBtUdXuAIp7XERPuJ4vyom+CsZhg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[worker07.air.nvidia.com]:19034,[147.75.91.219]:19034 (ECDSA) to the list of known hosts.
Enter passphrase for key '/Users/user/.ssh/id_rsa': <your ssh private key passphrase>
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-124-generic x86_64)

   <banner omitted for brevity>

Last login: Thu Feb 11 20:32:29 2021
ubuntu@oob-mgmt-server ~
```

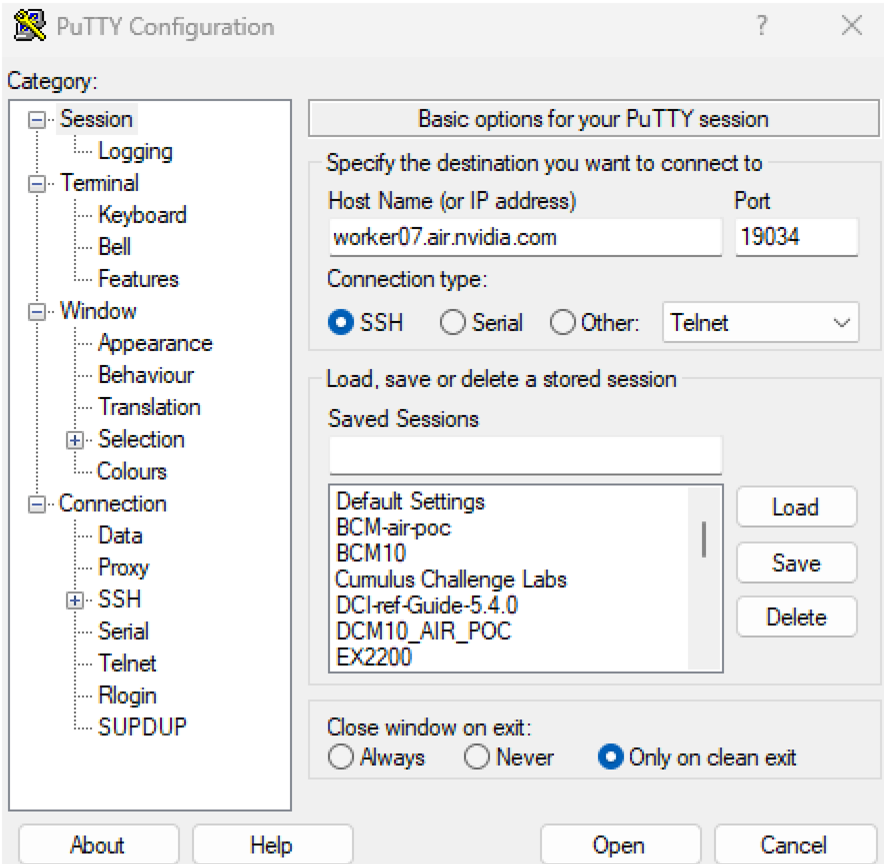
Windows using PuTTY:

You must tell PuTTY the SSH private key to use for this connection. The public key should be on the oob-mgmt-server in the authorized_keys file. Now your client must use your SSH private key. Here is an example: https://devops.ionos.com/tutorials/use-ssh-keys-with-putty-on-windows/#connect-to-server-with-private-key

The SSH session will be found in the Services Pane of the AIR UI

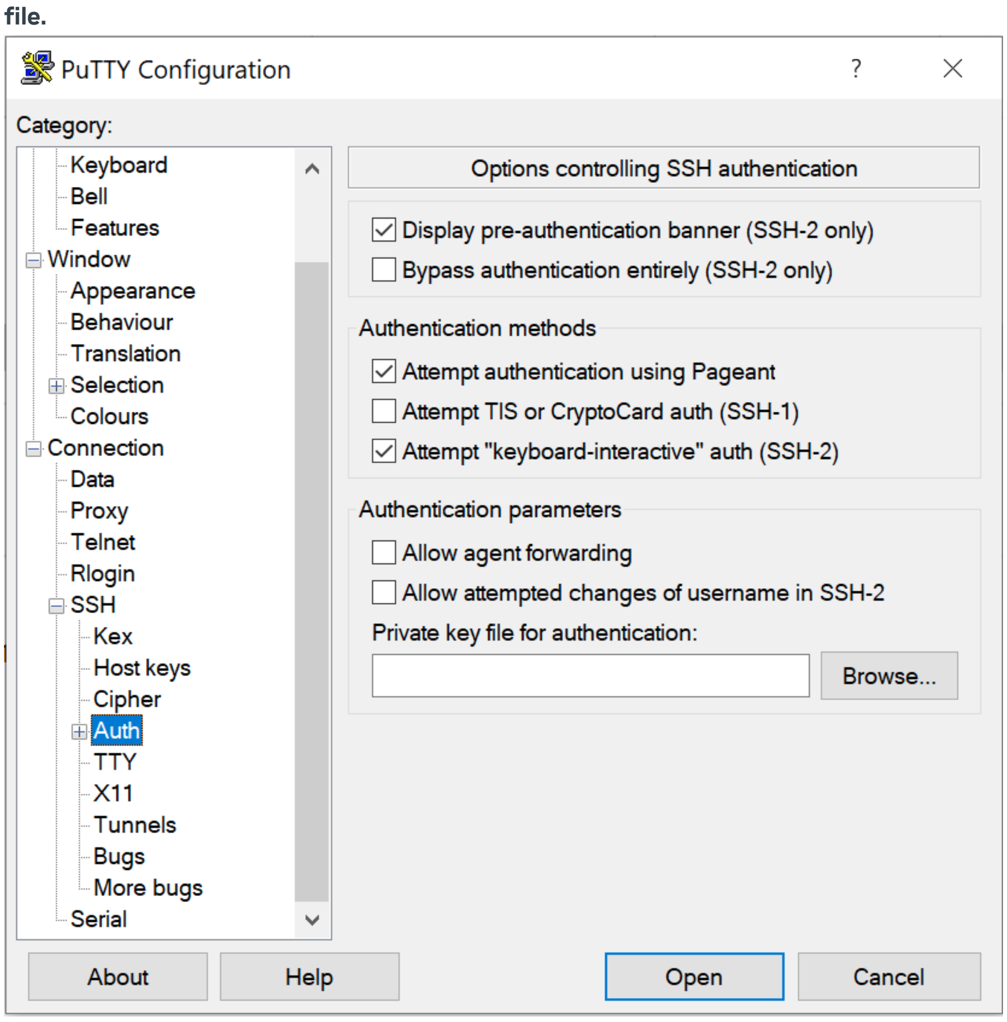


In your PuTTY Connection Info:



Hostname: workerXX.air.nvidia.com ["External Host" in Services pane on AIR UI]
Port: ["External Port" in Services pane on AIR UI]

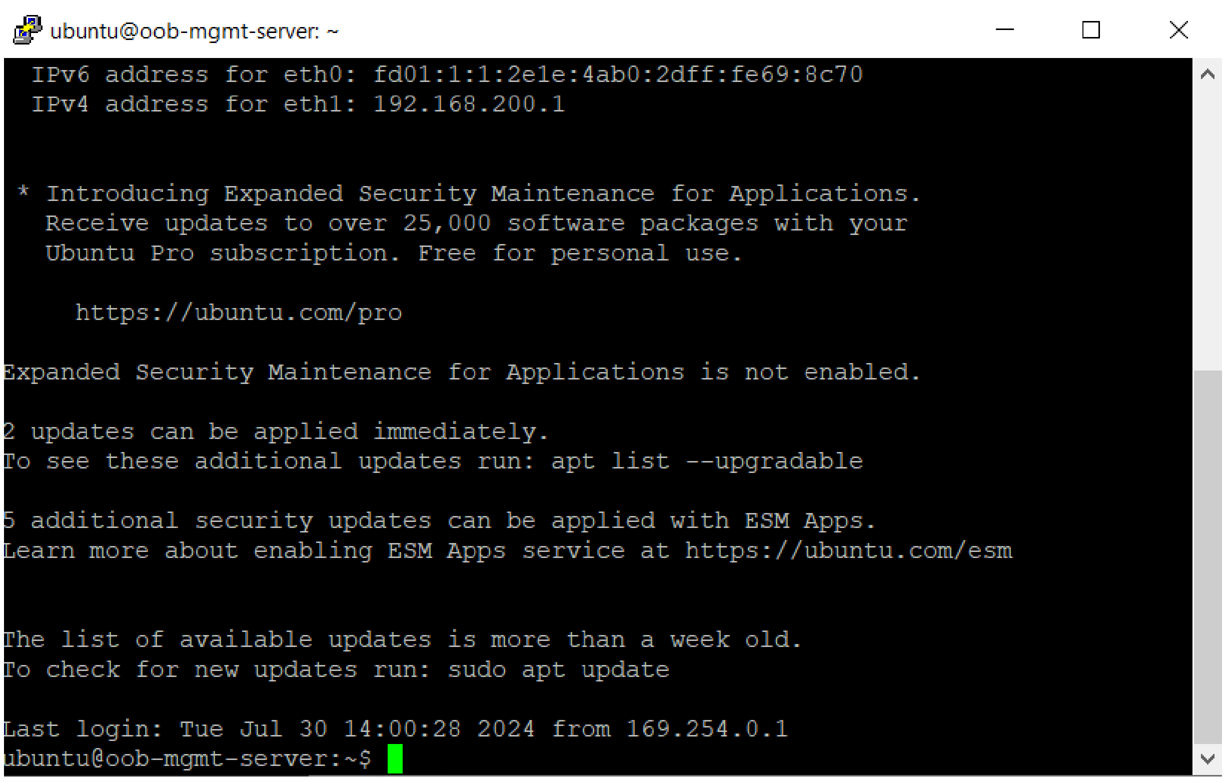
You must also tell PuTTY to use a private key (discussed earlier) in Connection -> SSH -> Auth Click the Browse button to pop out a box to point to the private key file.



Click Open.

When prompted, login as user ubuntu

You will be authenticated using your ssh key and may have to provide a password/passphrase if one was used to save the private key



You now have an SSH session to your workbench, and you will be at the BASH prompt on the oob-mgmt-server.

Last updated: May 22nd 2025

About NVIDIA (Cumulus Networks was acquired by NVIDIA in June 2020)

NVIDIA is leading the transformation of bringing web-scale networking to enterprise cloud. Its network switch operating system, NVIDIA © Cumulus Linux, is the only solution that allows you to affordably build and efficiently operate your network like the world's largest data center operators, unlocking vertical network stacks. By allowing operators to use standard hardware components, NVIDIA Cumulus Linux offers unprecedented operational speed and agility, at the industry's most competitive cost. For more information visit https://www.nvidia.com/en-us/networking/ethernet-switching/.

© 2022 NVIDIA Corporation. All rights reserved. NVIDIA, the NVIDIA logo, and NVIDIA © Cumulus Linux, NVIDIA © Cumulus NetQ are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.