NVIDIA® Cumulus Linux Virtual Workshop: Lab Guide

NVIDIA Cumulus Linux Virtual Workshop

Built for NVIDIA Cumulus Linux v5.2.0

NVIDIA Cumulus Linux Workshop: Lab Guide

Table of Contents

1: Verifying Lab Connectivity

Objective:

You will confirm that you can access your NVIDIA AIR workbench topology. This involves first visiting your AIR workbench in a web browser. You will connect to your Lab workbench via SSH to an out-of-band management server (oob-mgmt-server), from which you can access your switches.

Goals:

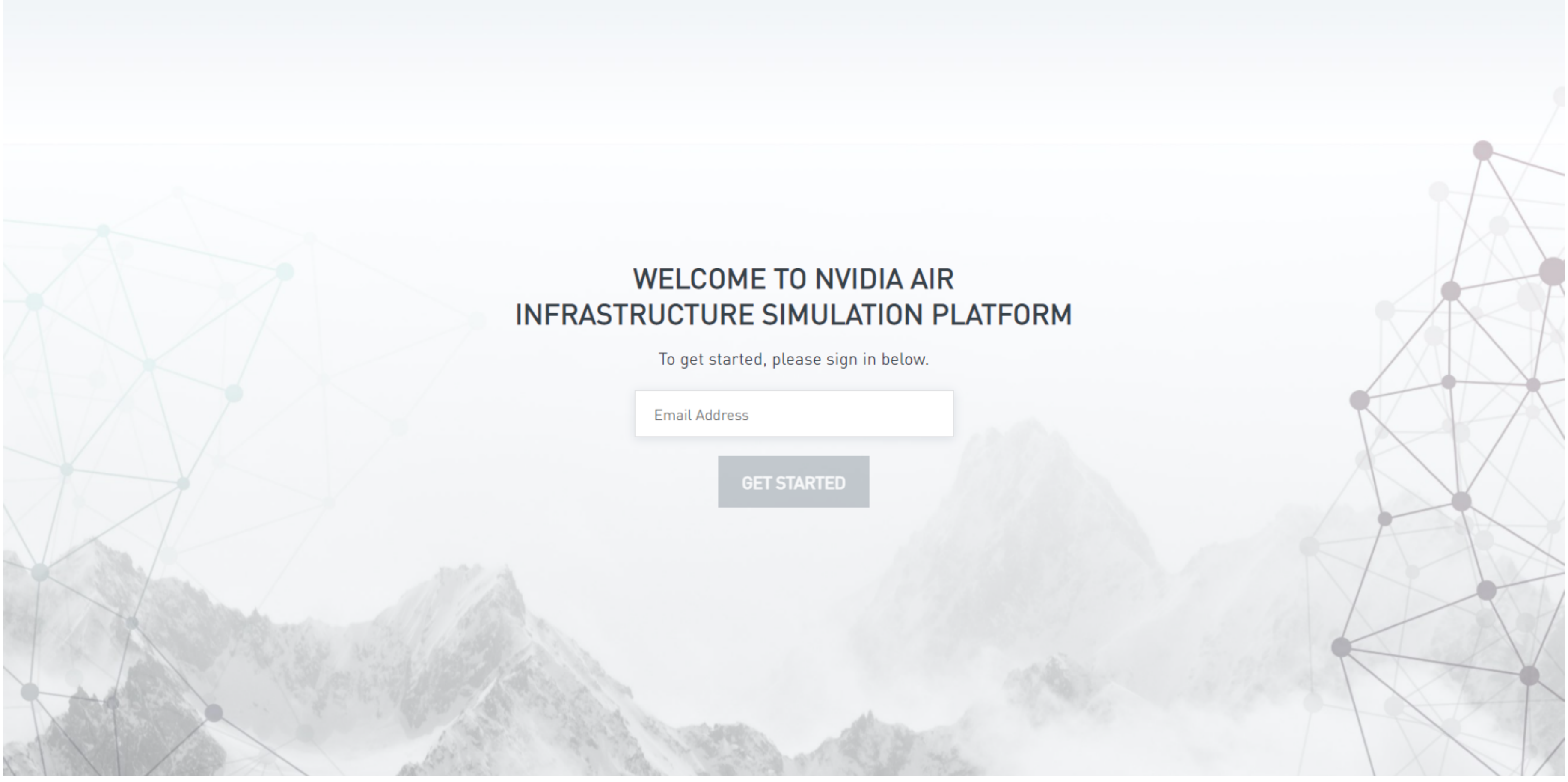
- Learn how to access your Air workbench in a web browser
- Log into your oob-mgmt-server.
- From your oob-mgmt-server, access your switches via SSH.

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**

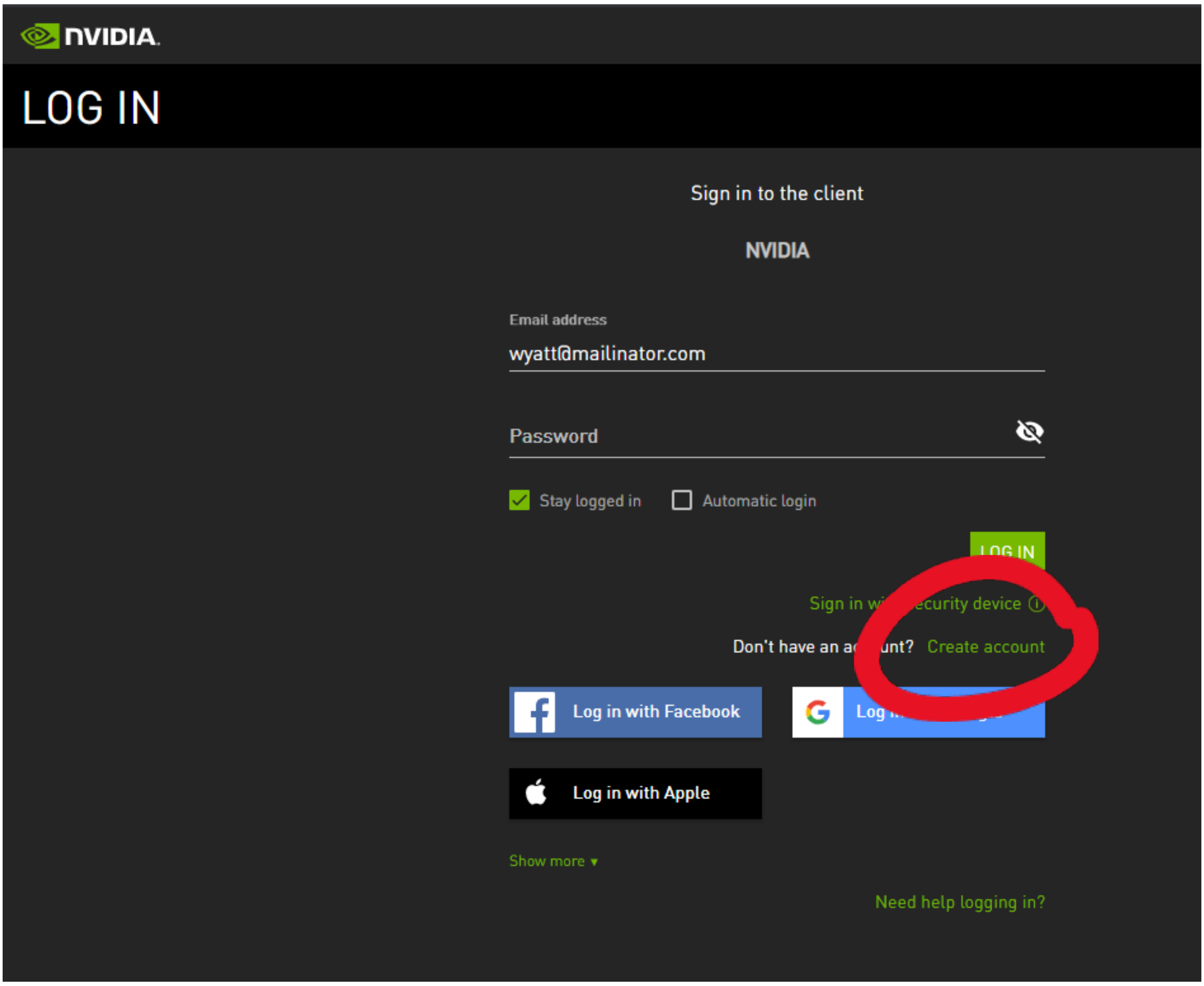


WELCOME TO NVIDIA AIR
INFRASTRUCTURE SIMULATION PLATFORM

To get started, please sign in below.

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.



NVIDIA.

LOG IN

Sign in to the client

NVIDIA

Email address
wyatt@mailinator.com

Password

☑ Stay logged in □ Automatic login

LOG IN

Sign in with security device ⓘ

Don't have an account? Create account

Log in with Facebook

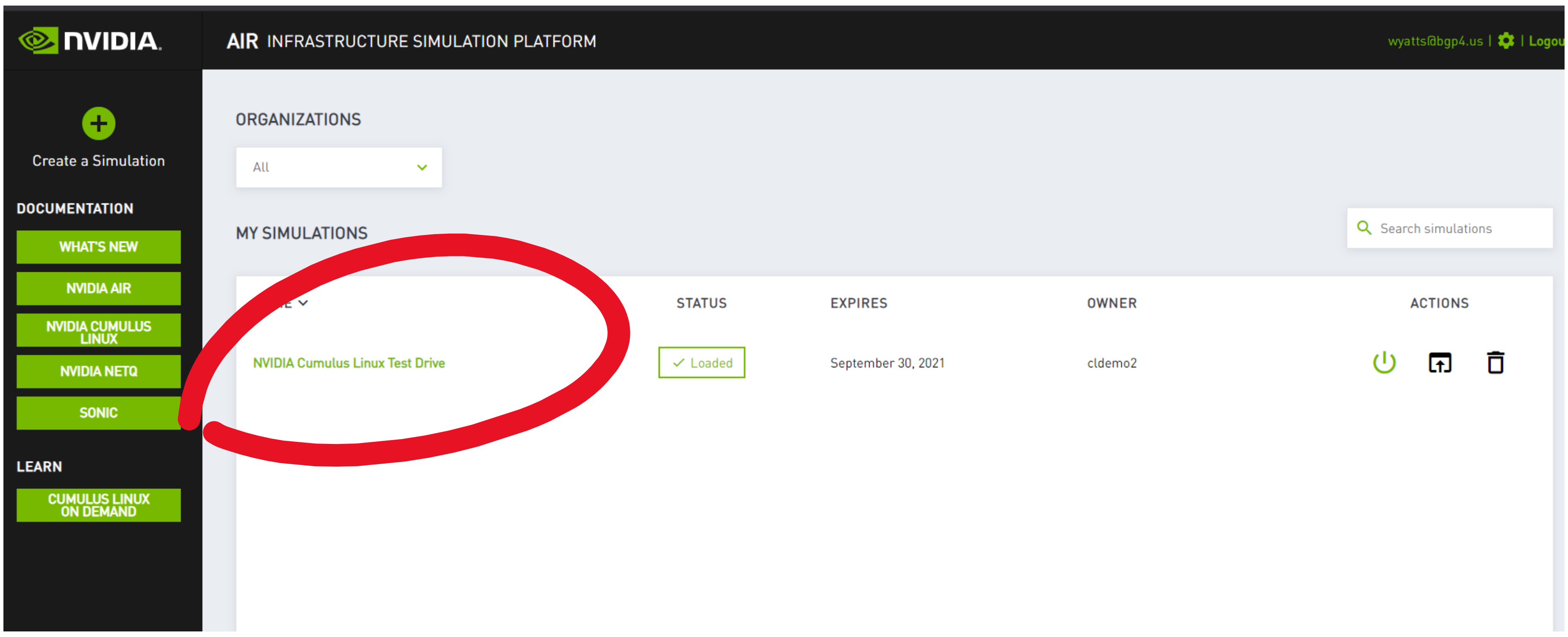
Log in with Google

Log in with Apple

Show more ▾

Need help logging in?

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



NVIDIA. AIR INFRASTRUCTURE SIMULATION PLATFORM wyatts@bgp4.us | | Logou

Create a Simulation

DOCUMENTATION
WHAT'S NEW
NVIDIA AIR
NVIDIA CUMULUS LINUX
NVIDIA NETQ
SONIC

LEARN
CUMULUS LINUX ON DEMAND

ORGANIZATIONS
All

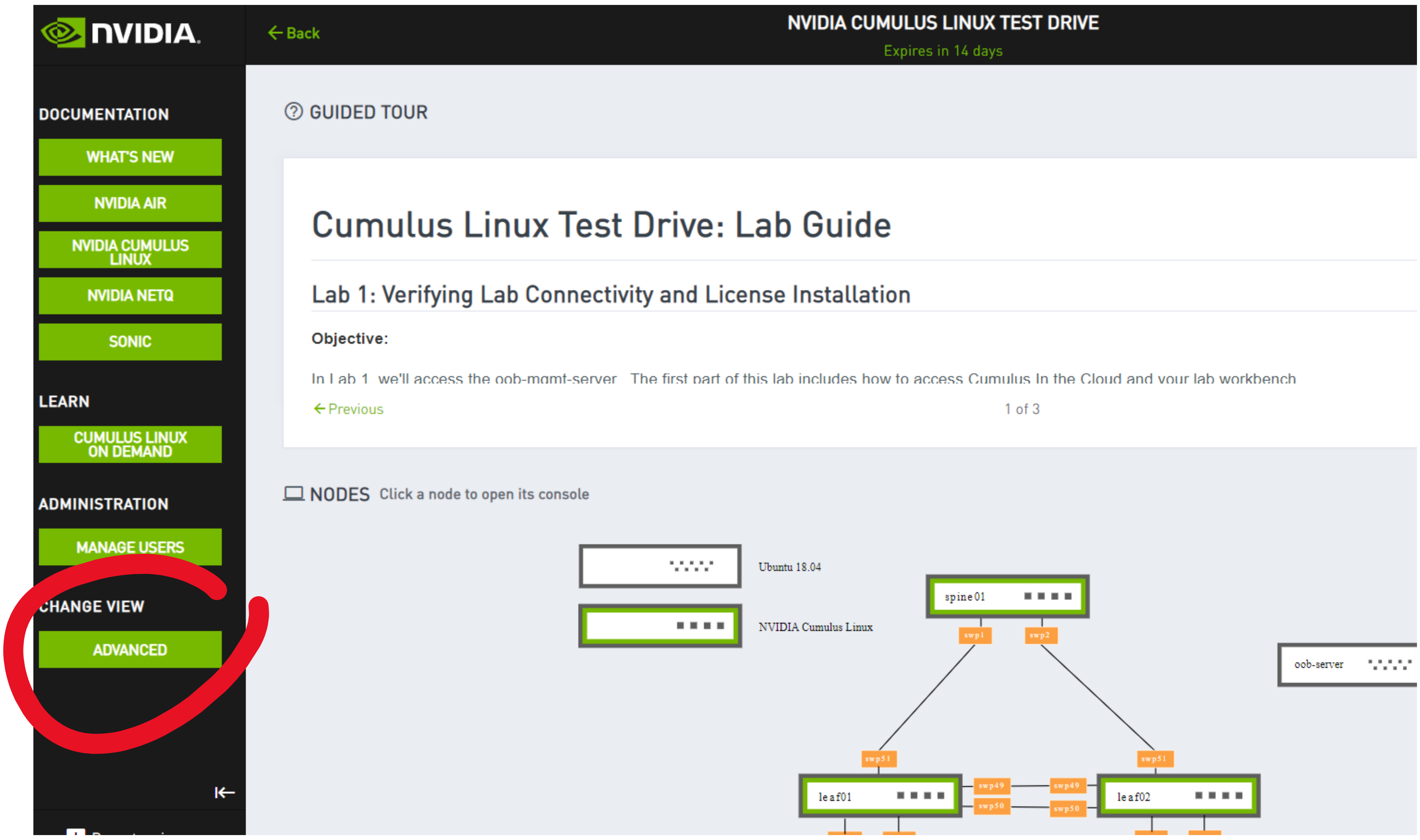
MY SIMULATIONS Search simulations

E ⌄	STATUS	EXPIRES	OWNER	ACTIONS
NVIDIA Cumulus Linux Test Drive	✓ Loaded	September 30, 2021	cldemo2	

3. Click the "NVIDIA Cumulus Linux Test Drive" simulation to open your simulation console. If you do not have this simulation, it has either not been provisioned yet, as it is not close enough to the event, or you have used a different email address than the email we have for you.

Connect to your oob-mgmt-server

1. Find and click the “Advanced” button in the lower left of the simulation console:



NVIDIA CUMULUS LINUX TEST DRIVE

Expires in 14 days

GUIDED TOUR

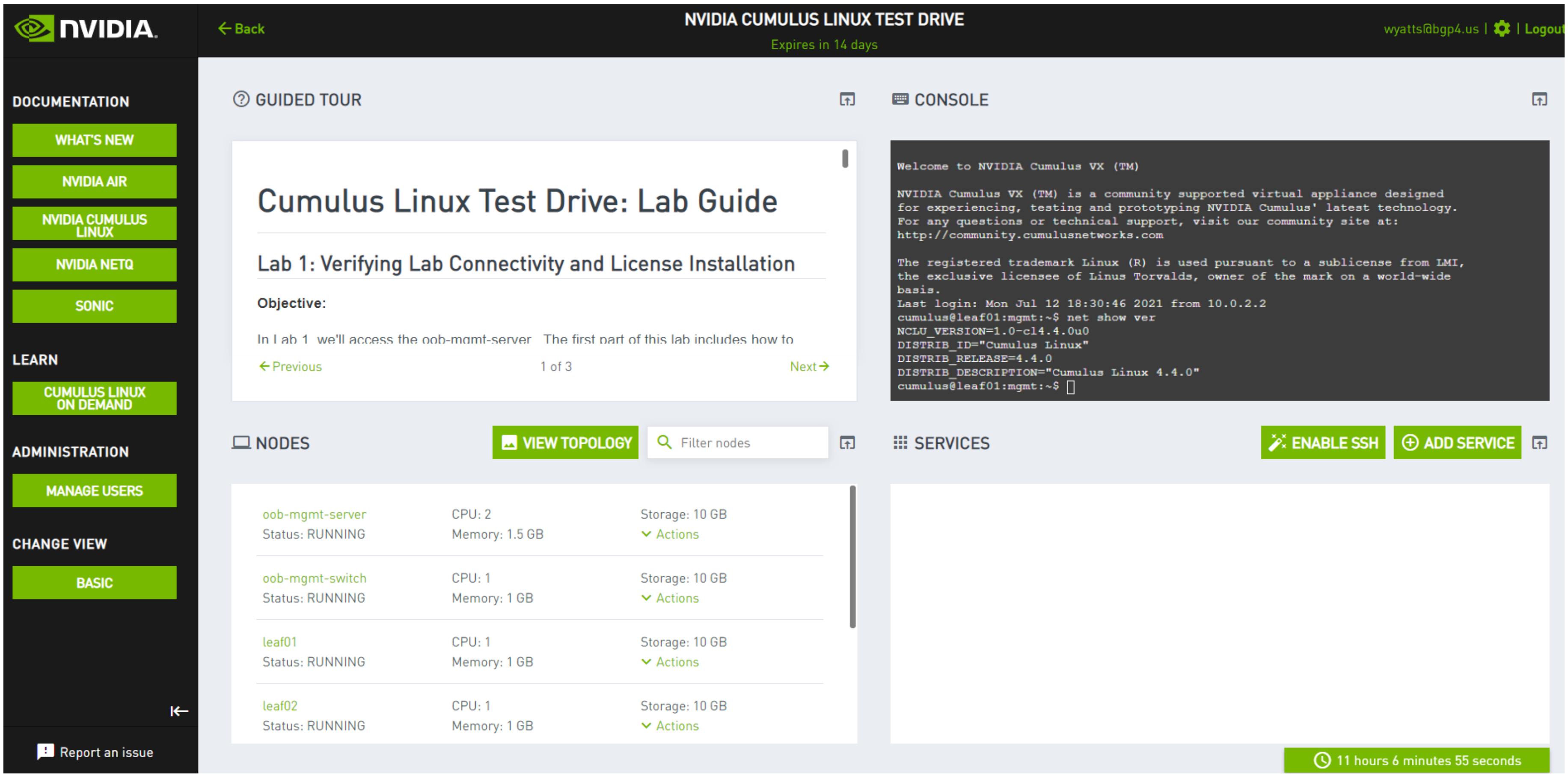
Cumulus Linux Test Drive: Lab Guide

Lab 1: Verifying Lab Connectivity and License Installation

Objective:

In Lab 1, we'll access the oob-mgmt-server. The first part of this lab includes how to access Cumulus In the Cloud and your lab workbench

2. The Advanced view presents you with your console connection to the oob-mgmt-server. Click on the pop-out icon to pop out your oob-mgmt-server console to allow you to resize and position for your convenience.



NVIDIA CUMULUS LINUX TEST DRIVE

Expires in 14 days

GUIDED TOUR

Cumulus Linux Test Drive: Lab Guide

Lab 1: Verifying Lab Connectivity and License Installation

Objective:

In Lab 1, we'll access the oob-mgmt-server. The first part of this lab includes how to

CONSOLE

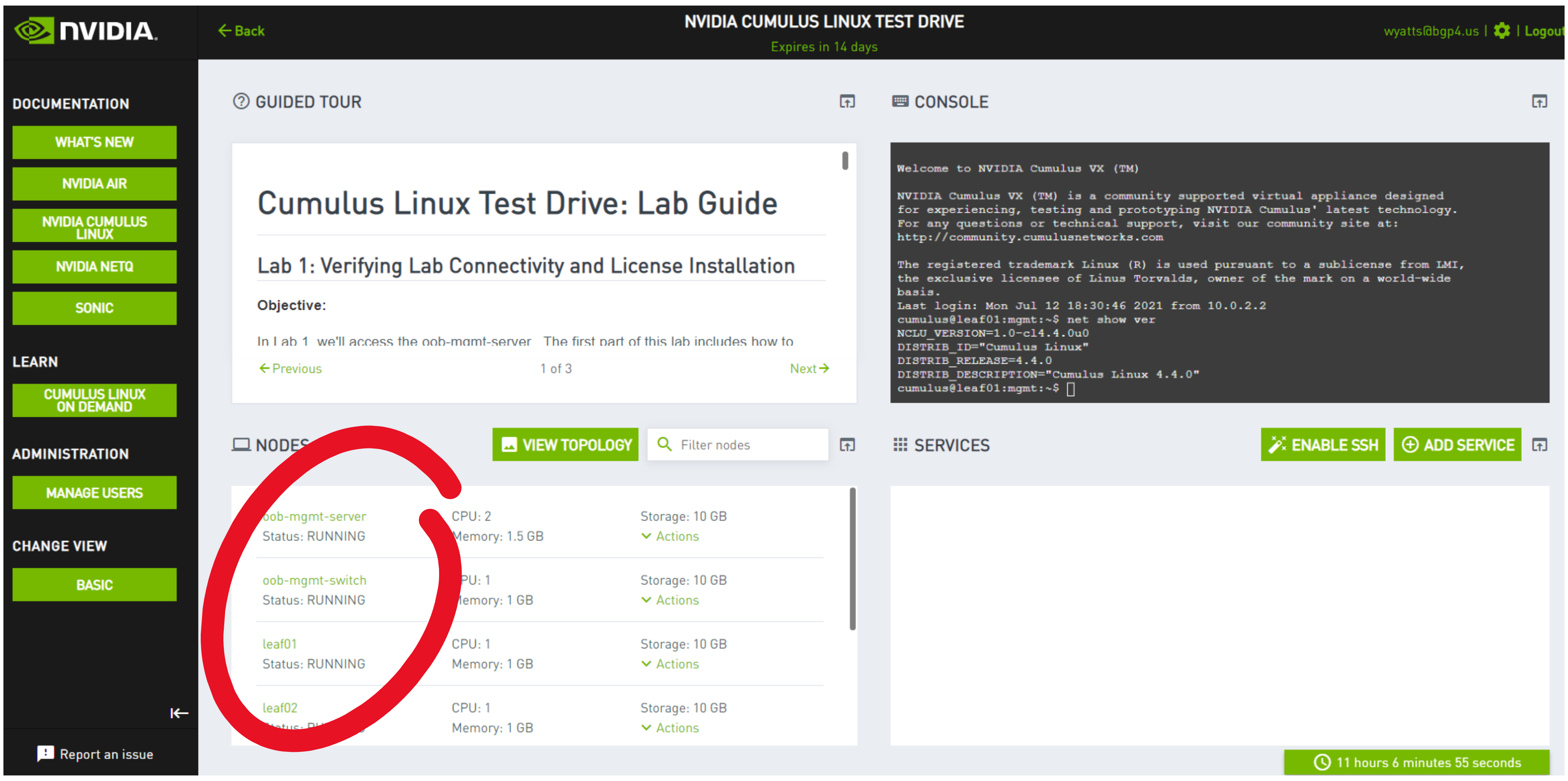
```
Welcome to NVIDIA Cumulus VX (TM)

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

The registered trademark Linux (R) is used pursuant to a sublicense from LMI,
the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide
basis.
Last login: Mon Jul 12 18:30:46 2021 from 10.0.2.2
cumulus@leaf01:mgmt:~$ net show ver
NCLU_VERSION=1.0-cl4.4.0u0
DISTRIB_ID="Cumulus Linux"
DISTRIB_RELEASE=4.4.0
DISTRIB_DESCRIPTION="Cumulus Linux 4.4.0"
cumulus@leaf01:mgmt:~$ 
```

NODES

oob-mgmt-server Status: RUNNING	CPU: 2 Memory: 1.5 GB	Storage: 10 GB Actions
oob-mgmt-switch Status: RUNNING	CPU: 1 Memory: 1 GB	Storage: 10 GB Actions
leaf01 Status: RUNNING	CPU: 1 Memory: 1 GB	Storage: 10 GB Actions
leaf02 Status: RUNNING	CPU: 1 Memory: 1 GB	Storage: 10 GB Actions

SERVICES

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



Cumulus Linux Test Drive: Lab Guide

Lab 1: Verifying Lab Connectivity and License Installation

Objective:

In Lab 1, we'll access the oob-mgmt-server. The first part of this lab includes how to

← Previous 1 of 3 Next →

```
Welcome to NVIDIA Cumulus VX (TM)

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

The registered trademark Linux (R) is used pursuant to a sublicense from LMI,
the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide
basis.
Last login: Mon Jul 12 18:30:46 2021 from 10.0.2.2
cumulus@leaf01:mgmt:~$ net show ver
NCLU_VERSION=1.0-cl4.4.0u0
DISTRIB_ID="Cumulus Linux"
DISTRIB_RELEASE=4.4.0
DISTRIB_DESCRIPTION="Cumulus Linux 4.4.0"
cumulus@leaf01:mgmt:~$ 
```

oob-mgmt-server CPU: 2 Storage: 10 GB
Status: RUNNING Memory: 1.5 GB Actions

oob-mgmt-switch PU: 1 Storage: 10 GB
Status: RUNNING emory: 1 GB Actions

leaf01 CPU: 1 Storage: 10 GB
Status: RUNNING Memory: 1 GB Actions

leaf02 CPU: 1 Storage: 10 GB
Memory: 1 GB Actions

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 unhidden. *Your new password* will be entered as the last two password entries

```
Ubuntu 18.04.5 LTS oob-mgmt-server ttyS0

oob-mgmt-server login: ubuntu
Password: nvidia
You are required to change your password immediately (root enforced)
Changing password for cumulus.
(current) UNIX password: nvidia
Enter new UNIX password: [your-new-password]
Retype new UNIX password: [your-new-password]
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-124-generic x86_64)

    <banner omitted for brevity>


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

Run the setup playbook

1. Change directories to the folder named “Test-Drive-Automation” from the user cumulus home directory.

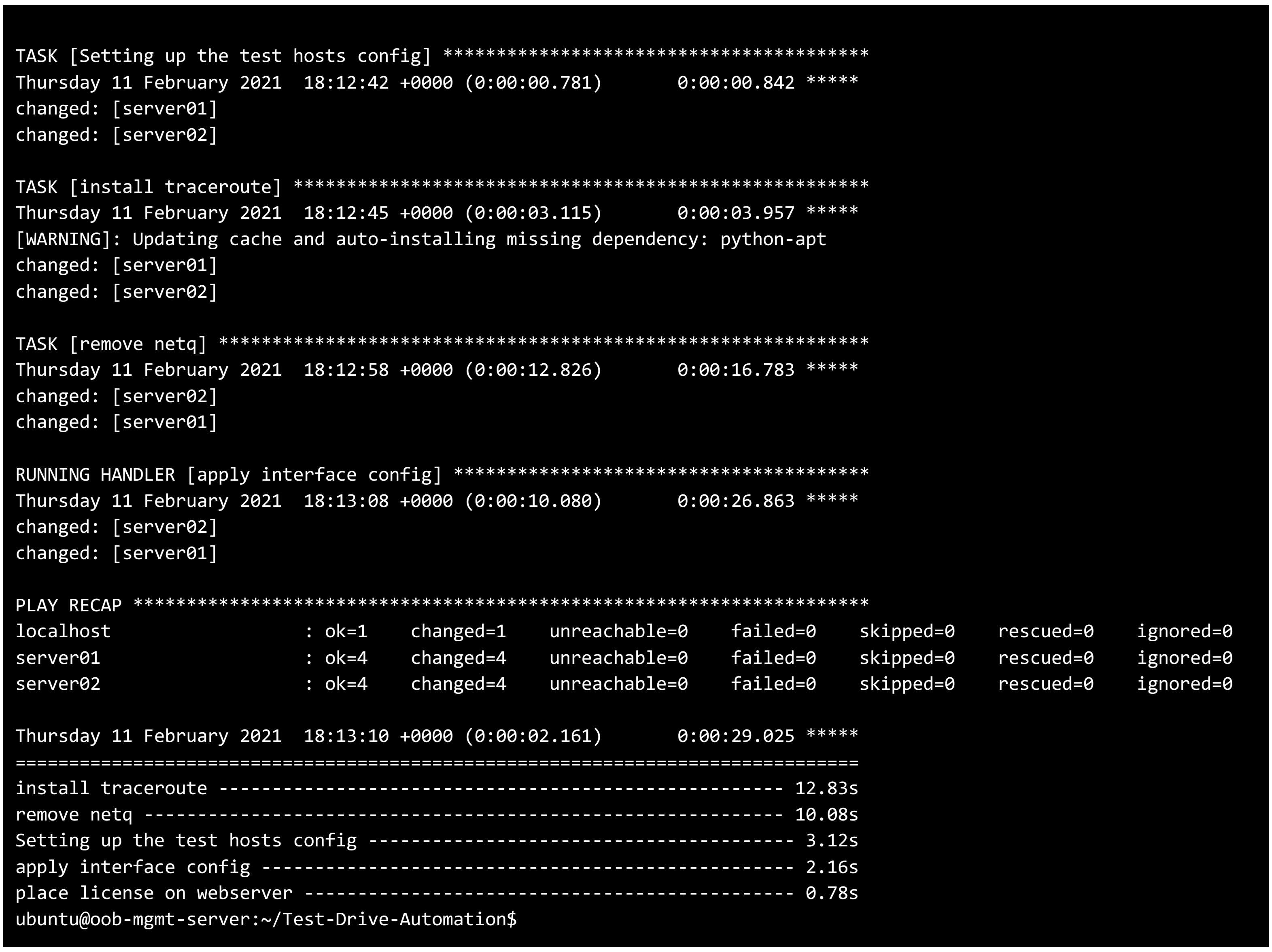
```
ubuntu@oob-mgmt-server:~$ cd Test-Drive-Automation
ubuntu@oob-mgmt-server:~/Test-Drive-Automation$
```

2. Perform a ‘git pull’ to sync/fetch changes

```
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, useml
-vvvv to see details

PLAY [localhost] ***************************************************************

TASK [place license on webserver] **********************************************
Thursday 11 February 2021  18:12:41 +0000 (0:00:00.061)       0:00:00.061 *****
changed: [localhost]

PLAY [server01:server02] *******************************************************
```



```
TASK [Setting up the test hosts config] ****************************************
Thursday 11 February 2021  18:12:42 +0000 (0:00:00.781)       0:00:00.842 *****
changed: [server01]
changed: [server02]

TASK [install traceroute] ******************************************************
Thursday 11 February 2021  18:12:45 +0000 (0:00:03.115)       0:00:03.957 *****
[WARNING]: Updating cache and auto-installing missing dependency: python-apt
changed: [server01]
changed: [server02]

TASK [remove netq] *************************************************************
Thursday 11 February 2021  18:12:58 +0000 (0:00:12.826)       0:00:16.783 *****
changed: [server02]
changed: [server01]

RUNNING HANDLER [apply interface config] ***************************************
Thursday 11 February 2021  18:13:08 +0000 (0:00:10.080)       0:00:26.863 *****
changed: [server02]
changed: [server01]

PLAY RECAP *********************************************************************
localhost                  : ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server01                   : ok=4    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server02                   : ok=4    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

Thursday 11 February 2021  18:13:10 +0000 (0:00:02.161)       0:00:29.025 *****
===============================================================================
install traceroute ----------------------------------------------------- 12.83s
remove netq ------------------------------------------------------------ 10.08s
Setting up the test hosts config ---------------------------------------- 3.12s
apply interface config -------------------------------------------------- 2.16s
place license on webserver ---------------------------------------------- 0.78s
ubuntu@oob-mgmt-server:~/Test-Drive-Automation$
```

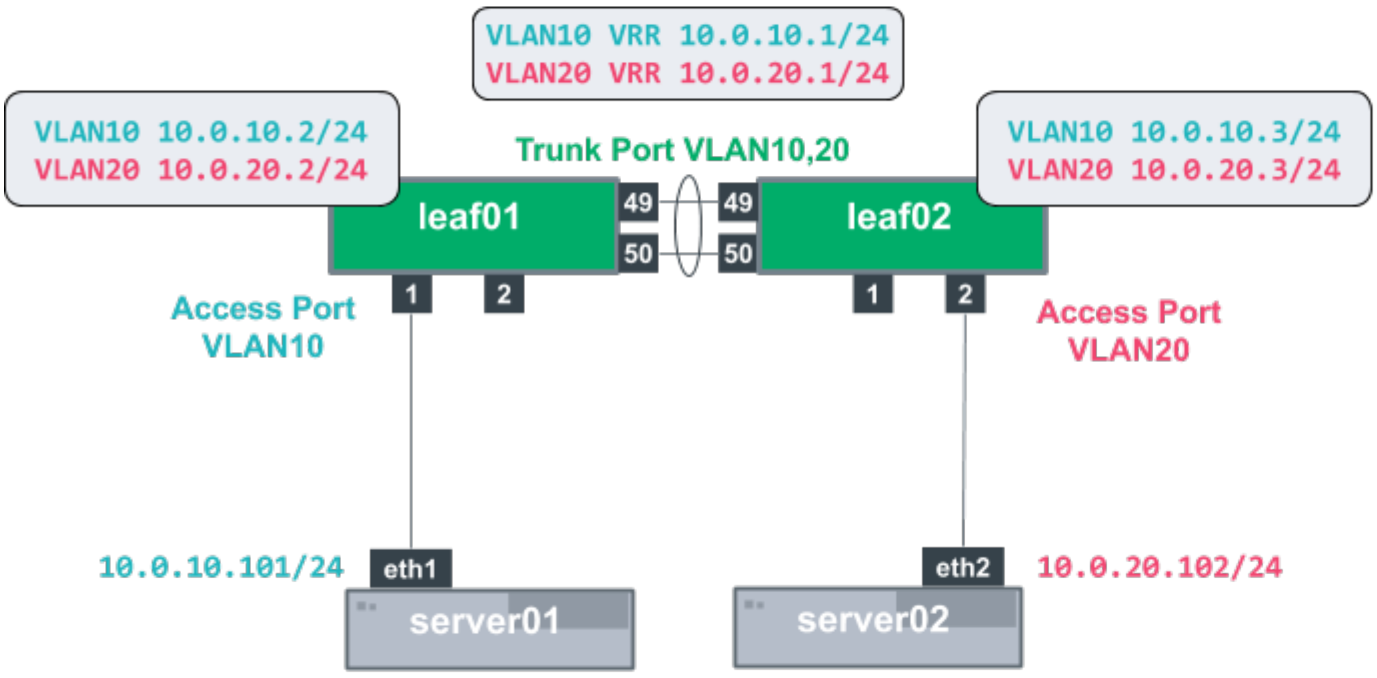
This concludes Lab 1.

2: Interface Configuration

Objective:

This lab will configure several types of interfaces. First, a bond will be configured between leaf01 and leaf02. The bond will be configured as a trunk to pass vlan10 and vlan20. Connections between leafs and servers will be configured as access ports. Server01 and Server02 will be in different subnets, so leaf01 and leaf02 will be configured to route for each vlan using VRR to provide high availability gateways for each vlan.

By the end of this lab, we'll have the following topology implemented:



Dependencies on other Labs:

- None

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

Procedure:

Enable NVUE

1. **On leaf01**: Assign an ip address to the loopback interface.

Configure loopback addresses on leaf01 and leaf02

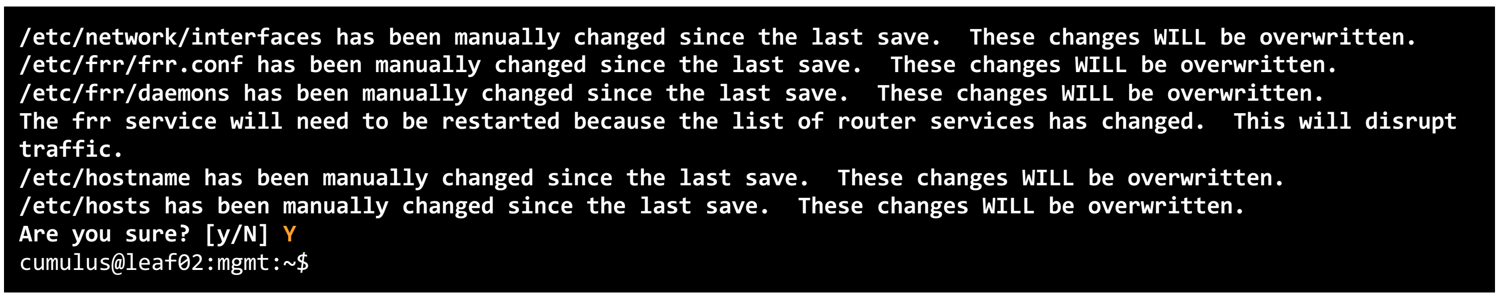
Interface Configuration Details		
Interface↓ \ Switch→	leaf01	leaf02
Loopback IP	10.255.255.1/32	10.255.255.2/32

1. **On leaf01**: Assign an ip address to the loopback interface.

```
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 leaf02**: Assign an ip address to the loopback interface.

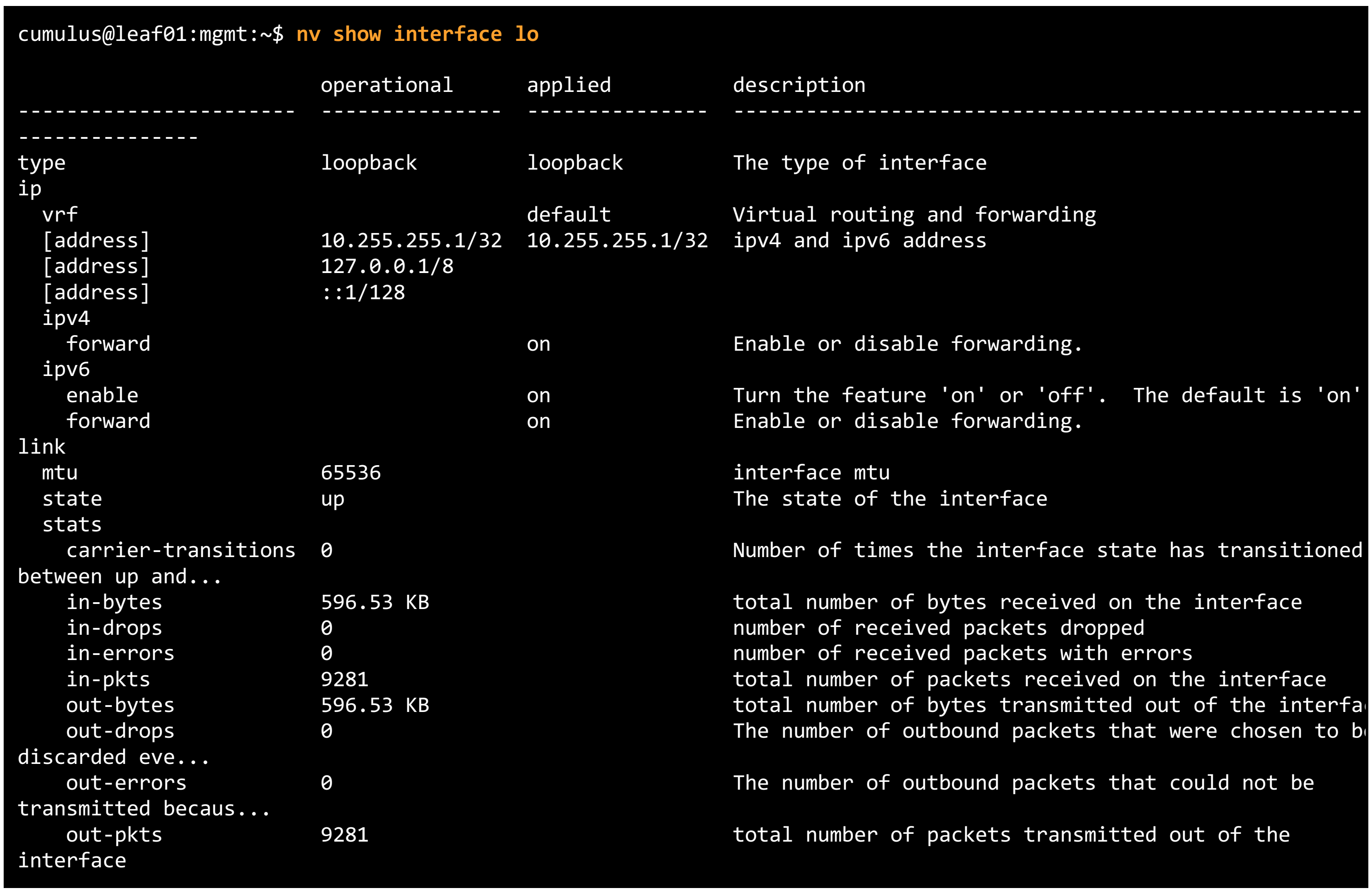
```
cumulus@leaf02:mgmt:~$ nv set interface lo ip address 10.255.255.2/32
cumulus@leaf02:mgmt:~$ nv set system hostname leaf02
cumulus@leaf02:mgmt:~$ nv config apply
/etc/cumulus/switchd.d/kernel_route_offload_flags.conf has been manually changed since the last save.  These
changes WILL be overwritten.
/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:~$
```

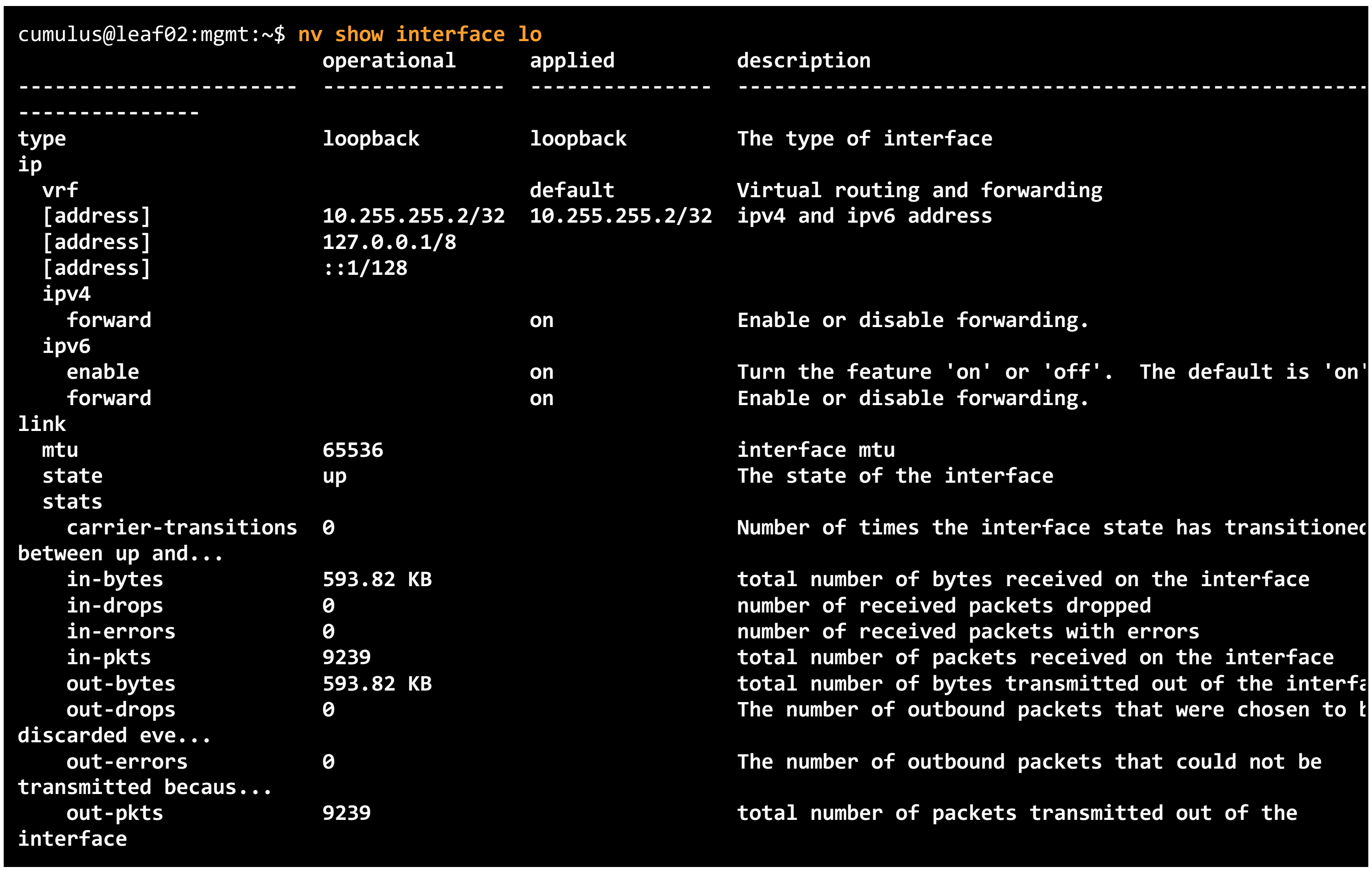
Verify loopback IP address configuration

3. **On leaf01**: Check that the address has been applied.



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

                         operational      applied          description
-----------------------  ---------------  ---------------  ----------------------------------------------------
---------------
type                     loopback         loopback         The type of interface
ip
  vrf                                     default          Virtual routing and forwarding
  [address]              10.255.255.1/32  10.255.255.1/32  ipv4 and ipv6 address
  [address]              127.0.0.1/8
  [address]              ::1/128
  ipv4
    forward                               on               Enable or disable forwarding.
  ipv6
    enable                                on               Turn the feature 'on' or 'off'.  The default is 'on'
    forward                               on               Enable or disable forwarding.
link
  mtu                    65536                             interface mtu
  state                  up                                The state of the interface
  stats
    carrier-transitions  0                                 Number of times the interface state has transitioned
between up and...
    in-bytes             596.53 KB                         total number of bytes received on the interface
    in-drops             0                                 number of received packets dropped
    in-errors            0                                 number of received packets with errors
    in-pkts              9281                              total number of packets received on the interface
    out-bytes            596.53 KB                         total number of bytes transmitted out of the interfa
    out-drops            0                                 The number of outbound packets that were chosen to b
discarded eve...
    out-errors           0                                 The number of outbound packets that could not be
transmitted becaus...
    out-pkts             9281                              total number of packets transmitted out of the
interface
```

4. **On leaf02**: Check that the address has been applied.



```
cumulus@leaf02:mgmt:~$ nv show interface lo
                         operational      applied          description
-----------------------  ---------------  ---------------  ----------------------------------------------------
---------------
type                     loopback         loopback         The type of interface
ip
  vrf                                     default          Virtual routing and forwarding
  [address]              10.255.255.2/32  10.255.255.2/32  ipv4 and ipv6 address
  [address]              127.0.0.1/8
  [address]              ::1/128
  ipv4
    forward                               on               Enable or disable forwarding.
  ipv6
    enable                                on               Turn the feature 'on' or 'off'.  The default is 'on'
    forward                               on               Enable or disable forwarding.
link
  mtu                    65536                             interface mtu
  state                  up                                The state of the interface
  stats
    carrier-transitions  0                                 Number of times the interface state has transitioned
between up and...
    in-bytes             593.82 KB                         total number of bytes received on the interface
    in-drops             0                                 number of received packets dropped
    in-errors            0                                 number of received packets with errors
    in-pkts              9239                              total number of packets received on the interface
    out-bytes            593.82 KB                         total number of bytes transmitted out of the interfa
    out-drops            0                                 The number of outbound packets that were chosen to b
discarded eve...
    out-errors           0                                 The number of outbound packets that could not be
transmitted becaus...
    out-pkts             9239                              total number of packets transmitted out of the
interface
```

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

Bond Configuration Details		
Bond↓ \ Switch→	leaf01	leaf02
Bond name	BOND0	BOND0
Bond members	swp49,swp50	swp49,swp50

5. On leaf01: Create a bond with members swp49 and swp50.

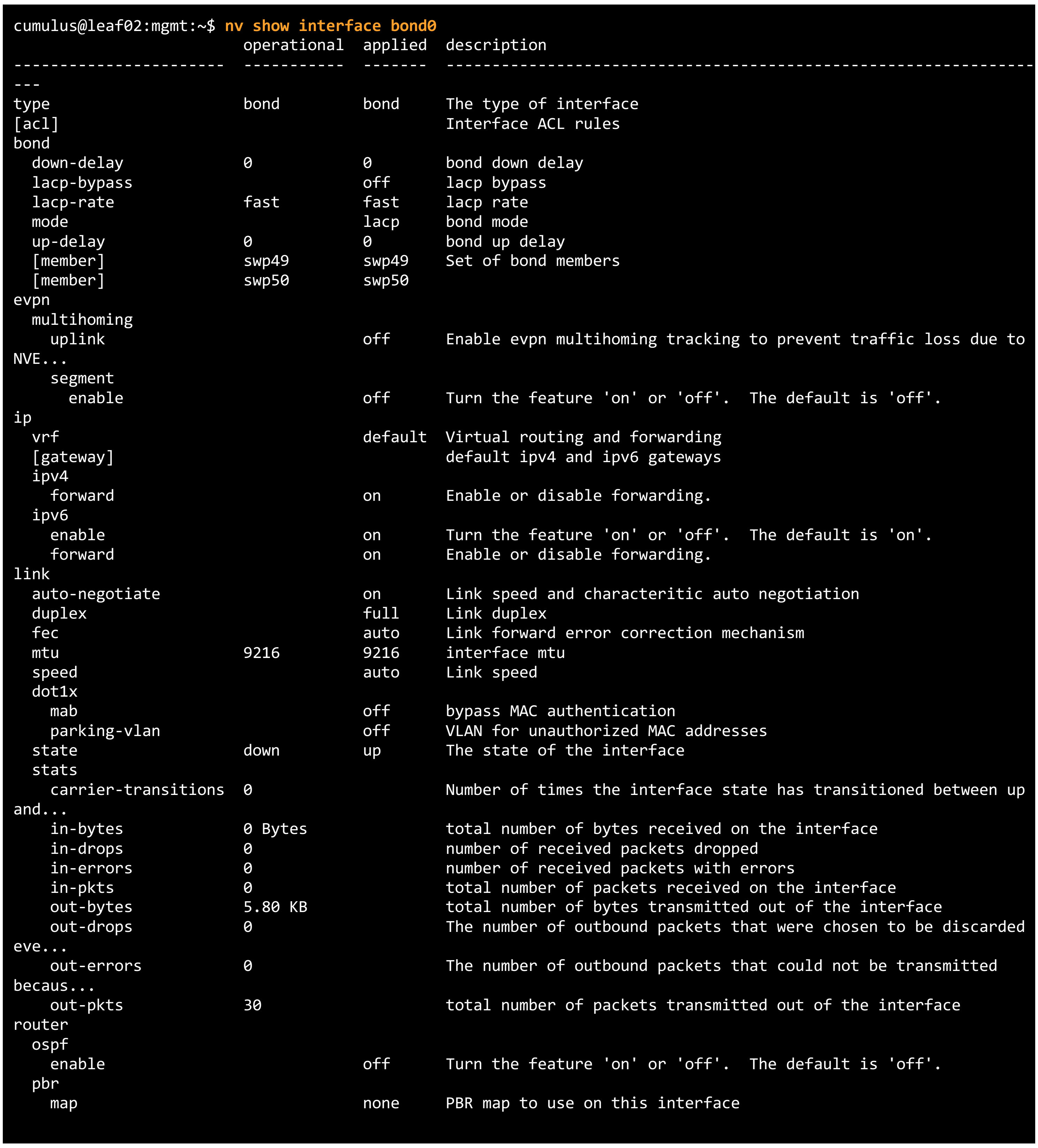
```
cumulus@leaf01:mgmt:~$ nv set interface bond0 bond member swp49-50
cumulus@leaf01:mgmt:~$ nv config apply
```

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. See the red highlighted output below to verify that your lab output matches.

```
cumulus@leaf01:mgmt:~$ nv show interface bond0
                         operational  applied  description
-----------------------  -----------  -------  ----------------------------------------------------------------
---
type                     bond         bond     The type of interface
[acl]                                          Interface ACL rules
bond
  down-delay             0            0        bond down delay
  lacp-bypass                         off      lacp bypass
  lacp-rate              fast         fast     lacp rate
  mode                                lacp     bond mode
  up-delay               0            0        bond up delay
  [member]               swp49        swp49    Set of bond members
  [member]               swp50        swp50
evpn
  multihoming
    uplink                            off      Enable evpn multihoming tracking to prevent traffic loss due to
NVE...
    segment
      enable                          off      Turn the feature 'on' or 'off'.  The default is 'off'.
ip
  vrf                                 default  Virtual routing and forwarding
  [gateway]                                    default ipv4 and ipv6 gateways
  ipv4
    forward                           on       Enable or disable forwarding.
  ipv6
    enable                            on       Turn the feature 'on' or 'off'.  The default is 'on'.
    forward                           on       Enable or disable forwarding.
link
  auto-negotiate                      on       Link speed and characteritic auto negotiation
  duplex                              full     Link duplex
  fec                                 auto     Link forward error correction mechanism
  mtu                    9216         9216     interface mtu
  speed                               auto     Link speed
  dot1x
    mab                               off      bypass MAC authentication
    parking-vlan                      off      VLAN for unauthorized MAC addresses
  state                  down         up       The state of the interface
  stats
    carrier-transitions  0                     Number of times the interface state has transitioned between up
and...
    in-bytes             0 Bytes               total number of bytes received on the interface
    in-drops             0                     number of received packets dropped
    in-errors            0                     number of received packets with errors
    in-pkts              0                     total number of packets received on the interface
    out-bytes            5.80 KB               total number of bytes transmitted out of the interface
    out-drops            0                     The number of outbound packets that were chosen to be discarded
eve...
    out-errors           0                     The number of outbound packets that could not be transmitted
becaus...
    out-pkts             30                    total number of packets transmitted out of the interface
router
  ospf
    enable                            off      Turn the feature 'on' or 'off'.  The default is 'off'.
  pbr
    map                               none     PBR map to use on this interface
```



```
cumulus@leaf02:mgmt:~$ nv show interface bond0
                         operational  applied  description
-----------------------  -----------  -------  ----------------------------------------------------------------
---
type                     bond         bond     The type of interface
[acl]                                          Interface ACL rules
bond
  down-delay             0            0        bond down delay
  lacp-bypass                         off      lacp bypass
  lacp-rate              fast         fast     lacp rate
  mode                                lacp     bond mode
  up-delay               0            0        bond up delay
  [member]               swp49        swp49    Set of bond members
  [member]               swp50        swp50
evpn
  multihoming
    uplink                            off      Enable evpn multihoming tracking to prevent traffic loss due to
NVE...
    segment
      enable                          off      Turn the feature 'on' or 'off'.  The default is 'off'.
ip
  vrf                                 default  Virtual routing and forwarding
  [gateway]                                    default ipv4 and ipv6 gateways
  ipv4
    forward                           on       Enable or disable forwarding.
  ipv6
    enable                            on       Turn the feature 'on' or 'off'.  The default is 'on'.
    forward                           on       Enable or disable forwarding.
link
  auto-negotiate                      on       Link speed and characteritic auto negotiation
  duplex                              full     Link duplex
  fec                                 auto     Link forward error correction mechanism
  mtu                    9216         9216     interface mtu
  speed                               auto     Link speed
  dot1x
    mab                               off      bypass MAC authentication
    parking-vlan                      off      VLAN for unauthorized MAC addresses
  state                  down         up       The state of the interface
  stats
    carrier-transitions  0                     Number of times the interface state has transitioned between up
and...
    in-bytes             0 Bytes               total number of bytes received on the interface
    in-drops             0                     number of received packets dropped
    in-errors            0                     number of received packets with errors
    in-pkts              0                     total number of packets received on the interface
    out-bytes            5.80 KB               total number of bytes transmitted out of the interface
    out-drops            0                     The number of outbound packets that were chosen to be discarded
eve...
    out-errors           0                     The number of outbound packets that could not be transmitted
becaus...
    out-pkts             30                    total number of packets transmitted out of the interface
router
  ospf
    enable                            off      Turn the feature 'on' or 'off'.  The default is 'off'.
  pbr
    map                               none     PBR map to use on this interface
```

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		
Bridge↓ \ Switch→	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 a bridge with vlans 10 and 20.

```
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: Make swp1 (connecting to server01) 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
```

Note: The section below is provided for easier copying and pasting.

```
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 on leaf01 by checking that swp1 and BOND0 are part of the bridge.

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

      multicast.snooping.querier.source-ip  ptp.enable  Summary
----  ------------------------------------  ----------  -------
+ 10  0.0.0.0                               on
+ 20  0.0.0.0                               on
```

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

      multicast.snooping.querier.source-ip  ptp.enable  Summary
----  ------------------------------------  ----------  -------
+ 10  0.0.0.0                               on
+ 20  0.0.0.0                               on
```

Important things to observe:

> Vlan information has not been completed yet
>

On leaf01:	On leaf02:
• 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)	• 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

Setting↓ \ Switch→	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 vlan 20.

```
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
```

Note: The section below is provided for easier copying and pasting.

```
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
```

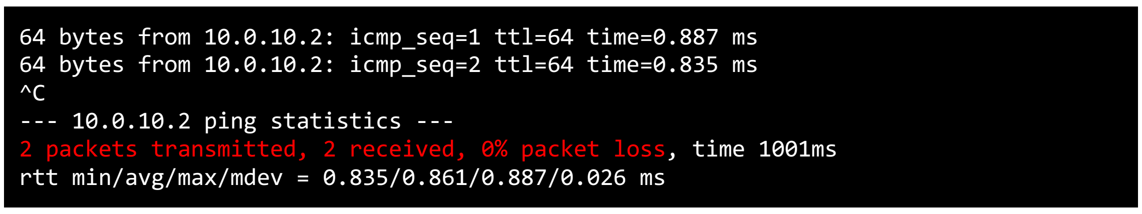
Test VRR connectivity

21. On server01: Test connectivity from server01 to the VRR gateway address. The login and password on the servers is ubuntu/nvidia

```
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
```

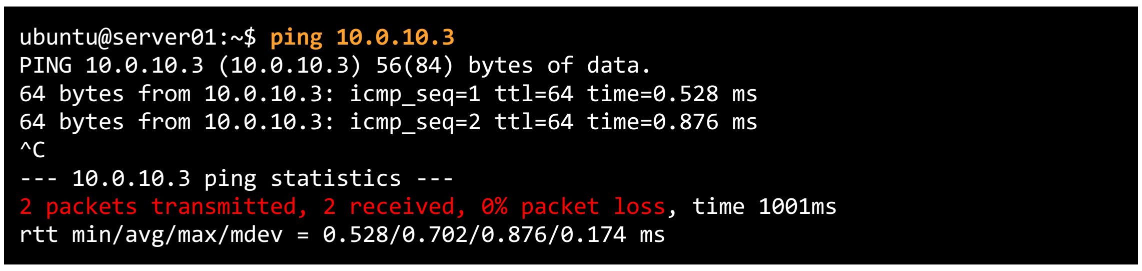
22. On server01: Test connectivity from server01 to leaf01 real IP address.

```
ubuntu@server01:~$ ping 10.0.10.2
PING 10.0.10.2 (10.0.10.2) 56(84) bytes of data.
```



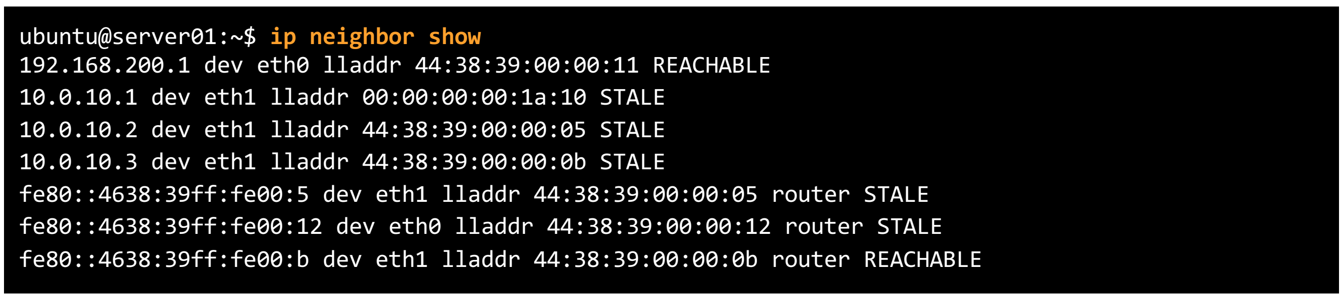
```
64 bytes from 10.0.10.2: icmp_seq=1 ttl=64 time=0.887 ms
64 bytes from 10.0.10.2: icmp_seq=2 ttl=64 time=0.835 ms
^C
--- 10.0.10.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.835/0.861/0.887/0.026 ms
```

23. On server01: Test connectivity from server01 to leaf02 real IP address.



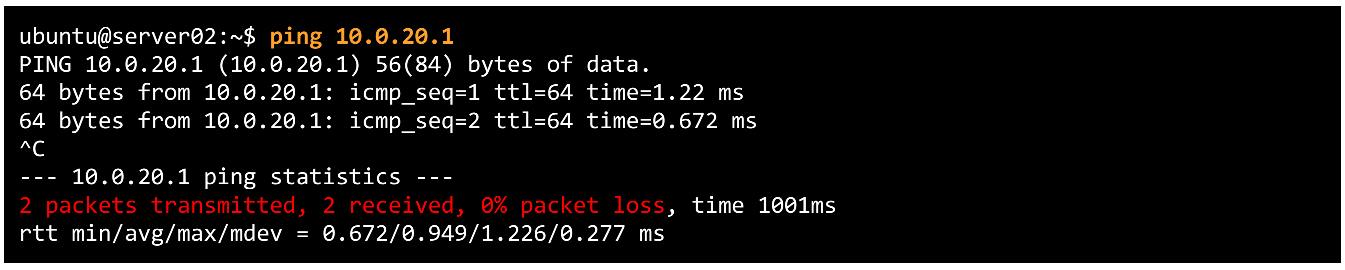
```
ubuntu@server01:~$ ping 10.0.10.3
PING 10.0.10.3 (10.0.10.3) 56(84) bytes of data.
64 bytes from 10.0.10.3: icmp_seq=1 ttl=64 time=0.528 ms
64 bytes from 10.0.10.3: icmp_seq=2 ttl=64 time=0.876 ms
^C
--- 10.0.10.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms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

24. On server01: Check the IP neighbor table which is similar to the ARP table, to view each MAC address. It also includes ipv6 neighbors. The arp table could also be evaluated using the “arp -a” command.

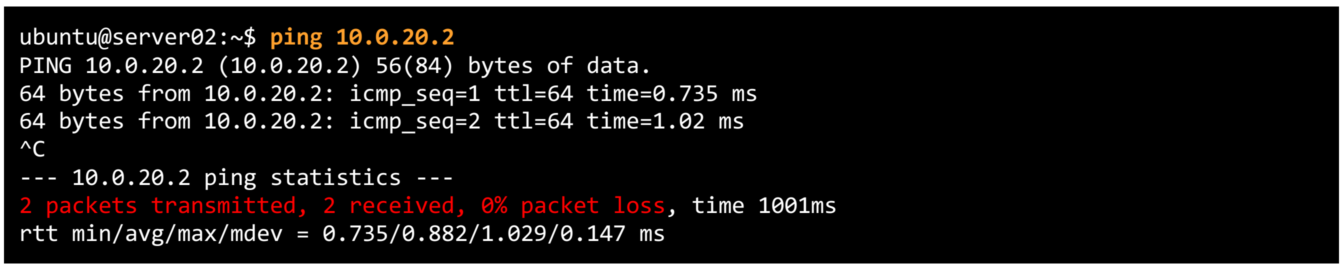


```
ubuntu@server01:~$ ip neighbor show
192.168.200.1 dev eth0 lladdr 44:38:39:00:00:11 REACHABLE
10.0.10.1 dev eth1 lladdr 00:00:00:00:1a:10 STALE
10.0.10.2 dev eth1 lladdr 44:38:39:00:00:05 STALE
10.0.10.3 dev eth1 lladdr 44:38:39:00:00:0b STALE
fe80::4638:39ff:fe00:5 dev eth1 lladdr 44:38:39:00:00:05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

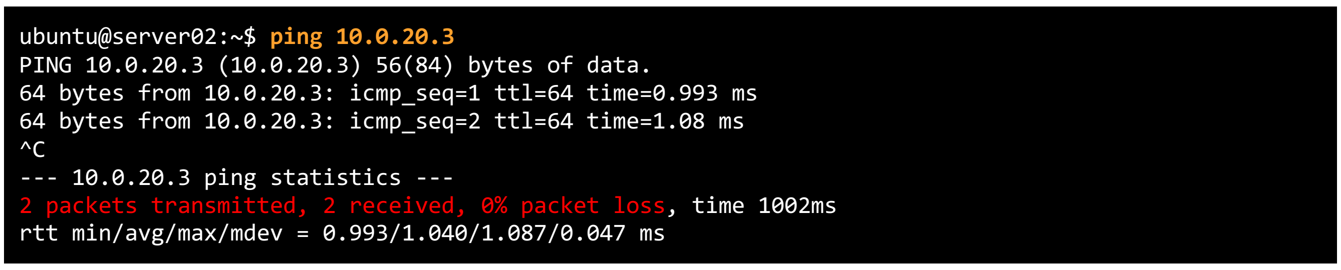
25. On server02: Repeat the same connectivity tests in step 10-13 from server02 to switch IP addresses.



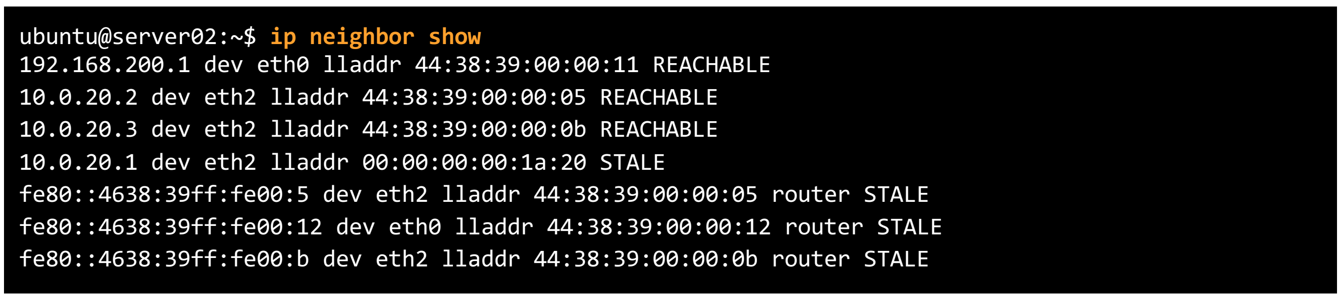
```
ubuntu@server02:~$ ping 10.0.20.1
PING 10.0.20.1 (10.0.20.1) 56(84) bytes of data.
64 bytes from 10.0.20.1: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 10.0.20.1: icmp_seq=2 ttl=64 time=0.672 ms
^C
--- 10.0.20.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt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

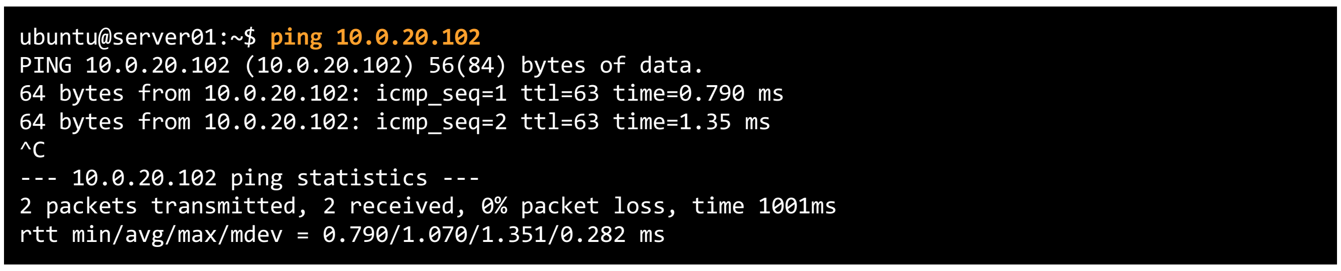


```
ubuntu@server02:~$ ip neighbor show
192.168.200.1 dev eth0 lladdr 44:38:39:00:00:11 REACHABLE
10.0.20.2 dev eth2 lladdr 44:38:39:00:00:05 REACHABLE
10.0.20.3 dev eth2 lladdr 44:38:39:00:00:0b REACHABLE
10.0.20.1 dev eth2 lladdr 00:00:00:00:1a:20 STALE
fe80::4638:39ff:fe00:5 dev eth2 lladdr 44:38:39:00:00:05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

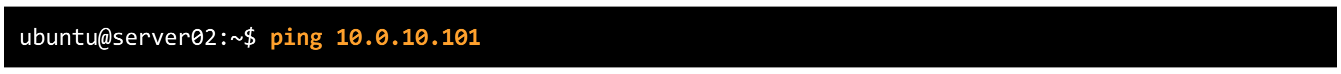
Important things to observe:

> Pings to the VRR and unique SVI IP addresses should all be successful for all Vlans

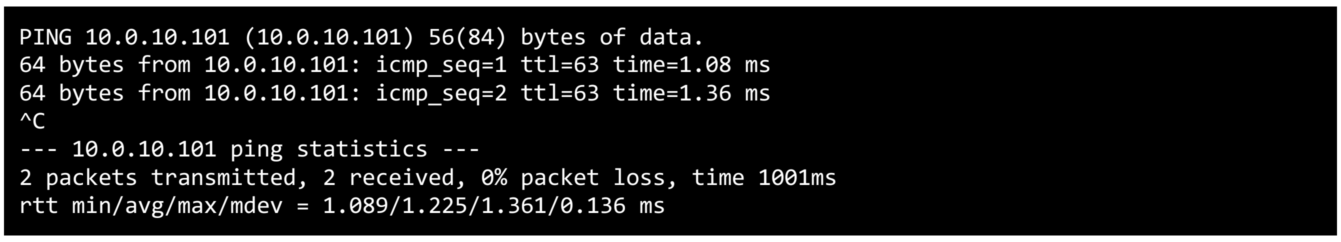
26. On server01 and server02: Ping to verify connectivity between server01 and server02.



```
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
```

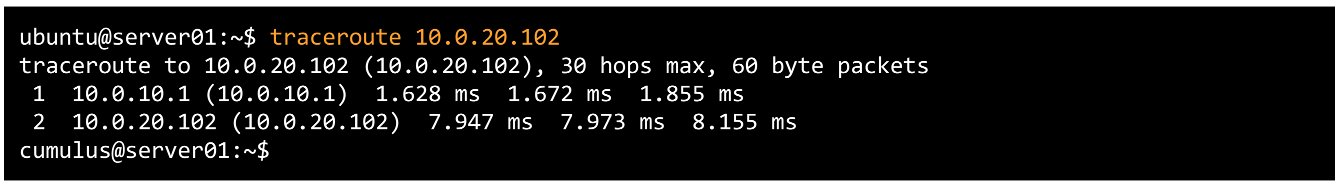


```
ubuntu@server02:~$ ping 10.0.10.101
```

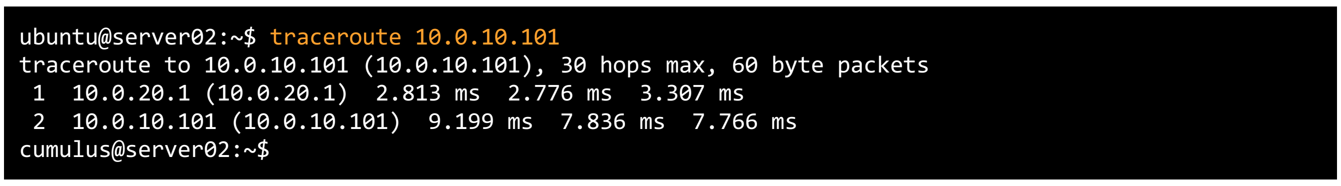


```
PING 10.0.10.101 (10.0.10.101) 56(84) bytes of data.
64 bytes from 10.0.10.101: icmp_seq=1 ttl=63 time=1.08 ms
64 bytes from 10.0.10.101: icmp_seq=2 ttl=63 time=1.36 ms
^C
--- 10.0.10.101 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.089/1.225/1.361/0.136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

27. On server01 and server02: traceroute to server02.



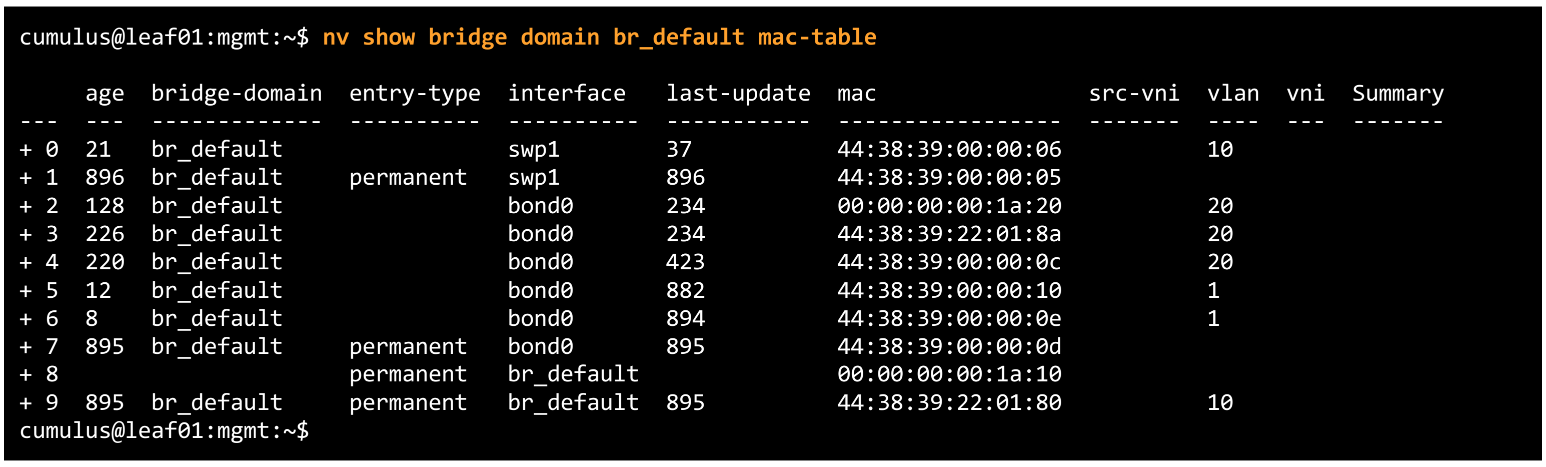
```
ubuntu@server01:~$ traceroute 10.0.20.102
traceroute to 10.0.20.102 (10.0.20.102), 30 hops max, 60 byte packets
 1  10.0.10.1 (10.0.10.1)  1.628 ms  1.672 ms  1.855 ms
 2  10.0.20.102 (10.0.20.102)  7.947 ms  7.973 ms  8.155 ms
cumulus@server01:~$
```



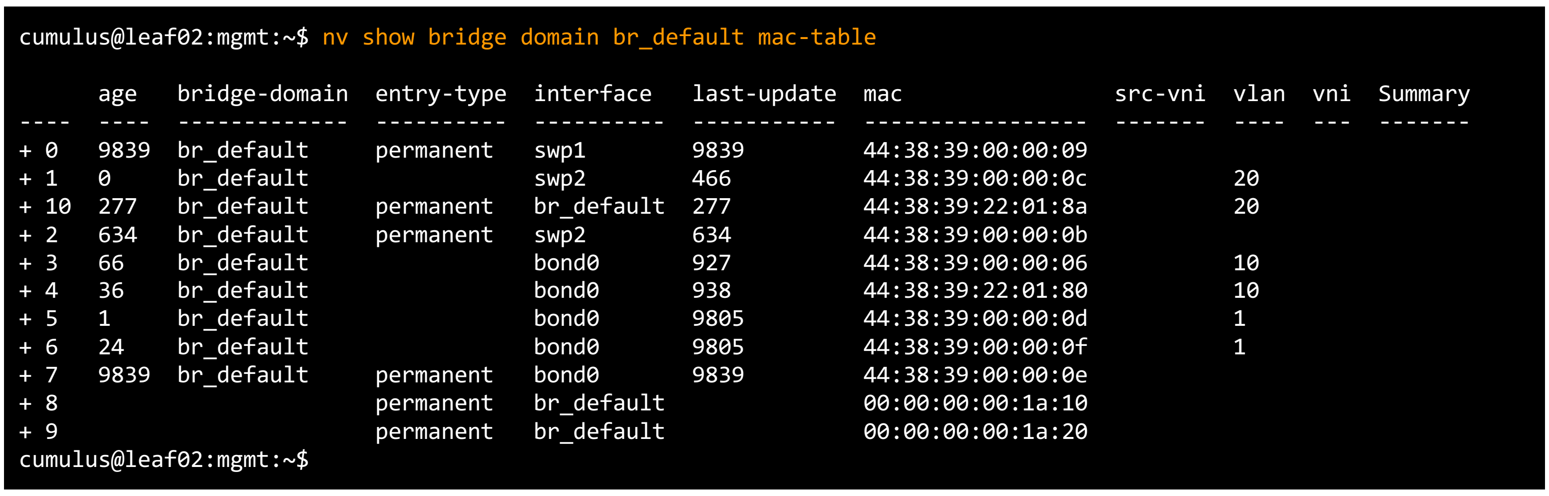
```
ubuntu@server02:~$ traceroute 10.0.10.101
traceroute to 10.0.10.101 (10.0.10.101), 30 hops max, 60 byte packets
 1  10.0.20.1 (10.0.20.1)  2.813 ms  2.776 ms  3.307 ms
 2  10.0.10.101 (10.0.10.101)  9.199 ms  7.836 ms  7.766 ms
cumulus@server02:~$
```

Verify MAC address table on leaf01 and leaf02

28. On leaf01 and leaf02 : Check to verify that the MAC addresses are learned correctly.



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

     age  bridge-domain  entry-type  interface   last-update  mac                src-vni  vlan  vni  Summary
---  ---  -------------  ----------  ----------  -----------  -----------------  -------  ----  ---  -------
+ 0  21   br_default                 swp1        37           44:38:39:00:00:06           10
+ 1  896  br_default     permanent   swp1        896          44:38:39:00:00:05
+ 2  128  br_default                 bond0       234          00:00:00:00:1a:20           20
+ 3  226  br_default                 bond0       234          44:38:39:22:01:8a           20
+ 4  220  br_default                 bond0       423          44:38:39:00:00:0c           20
+ 5  12   br_default                 bond0       882          44:38:39:00:00:10           1
+ 6  8    br_default                 bond0       894          44:38:39:00:00:0e           1
+ 7  895  br_default     permanent   bond0       895          44:38:39:00:00:0d
+ 8                      permanent   br_default               00:00:00:00:1a:10
+ 9  895  br_default     permanent   br_default  895          44:38:39:22:01:80           10
cumulus@leaf01:mgmt:~$
```



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

      age   bridge-domain  entry-type  interface   last-update  mac                src-vni  vlan  vni  Summary
----  ----  -------------  ----------  ----------  -----------  -----------------  -------  ----  ---  -------
+ 0   9839  br_default     permanent   swp1        9839         44:38:39:00:00:09
+ 1   0     br_default                 swp2        466          44:38:39:00:00:0c           20
+ 10  277   br_default     permanent   br_default  277          44:38:39:22:01:8a           20
+ 2   634   br_default     permanent   swp2        634          44:38:39:00:00:0b
+ 3   66    br_default                 bond0       927          44:38:39:00:00:06           10
+ 4   36    br_default                 bond0       938          44:38:39:22:01:80           10
+ 5   1     br_default                 bond0       9805         44:38:39:00:00:0d           1
+ 6   24    br_default                 bond0       9805         44:38:39:00:00:0f           1
+ 7   9839  br_default     permanent   bond0       9839         44:38:39:00:00:0e
+ 8                        permanent   br_default               00:00:00:00:1a:10
+ 9                        permanent   br_default               00:00:00:00:1a:20
cumulus@leaf02:mgmt:~$
```

Important things to observe:

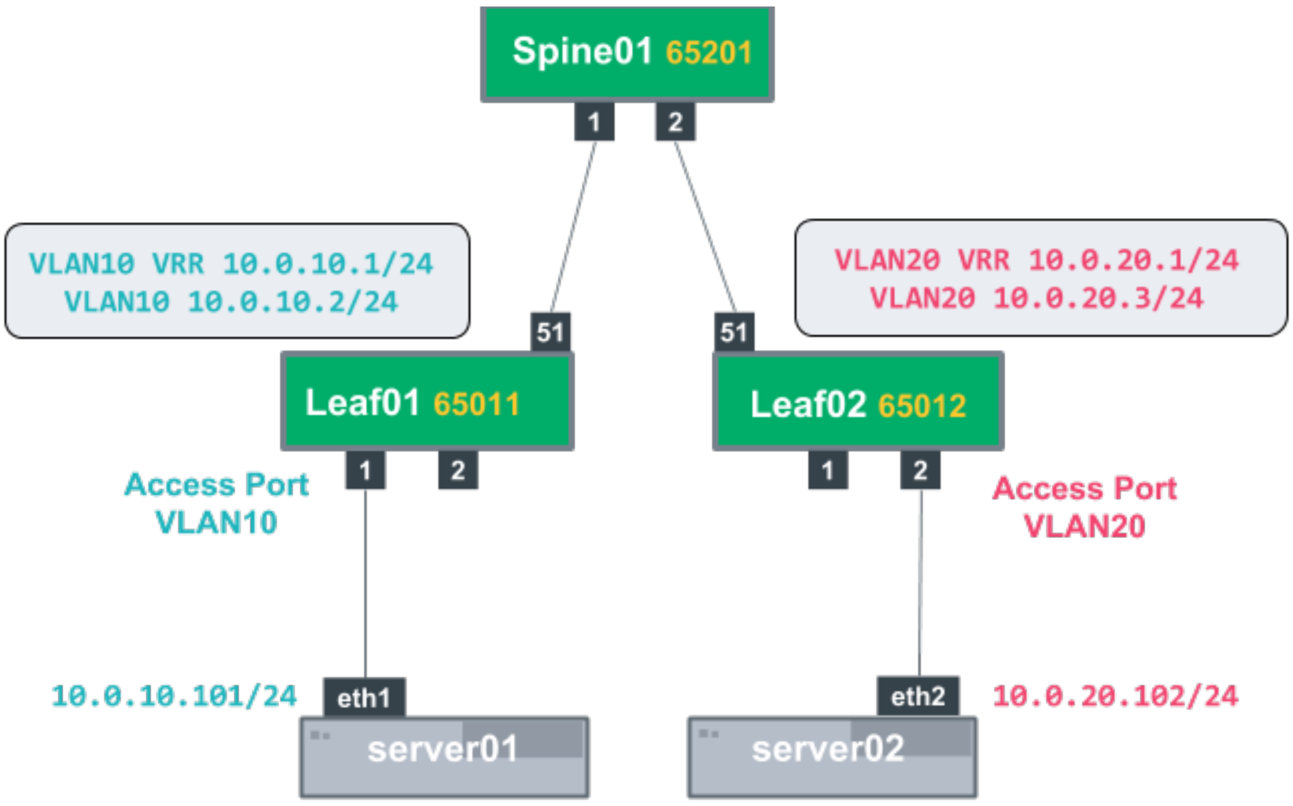
> MAC addresses of servers should be learned on BOND0 and swp interface of switch

This concludes Lab 2.

3: BGP Unnumbered

Objective:

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.



Dependencies on other Labs:

- None. Running Lab3.yml playbook configures all prerequisites.

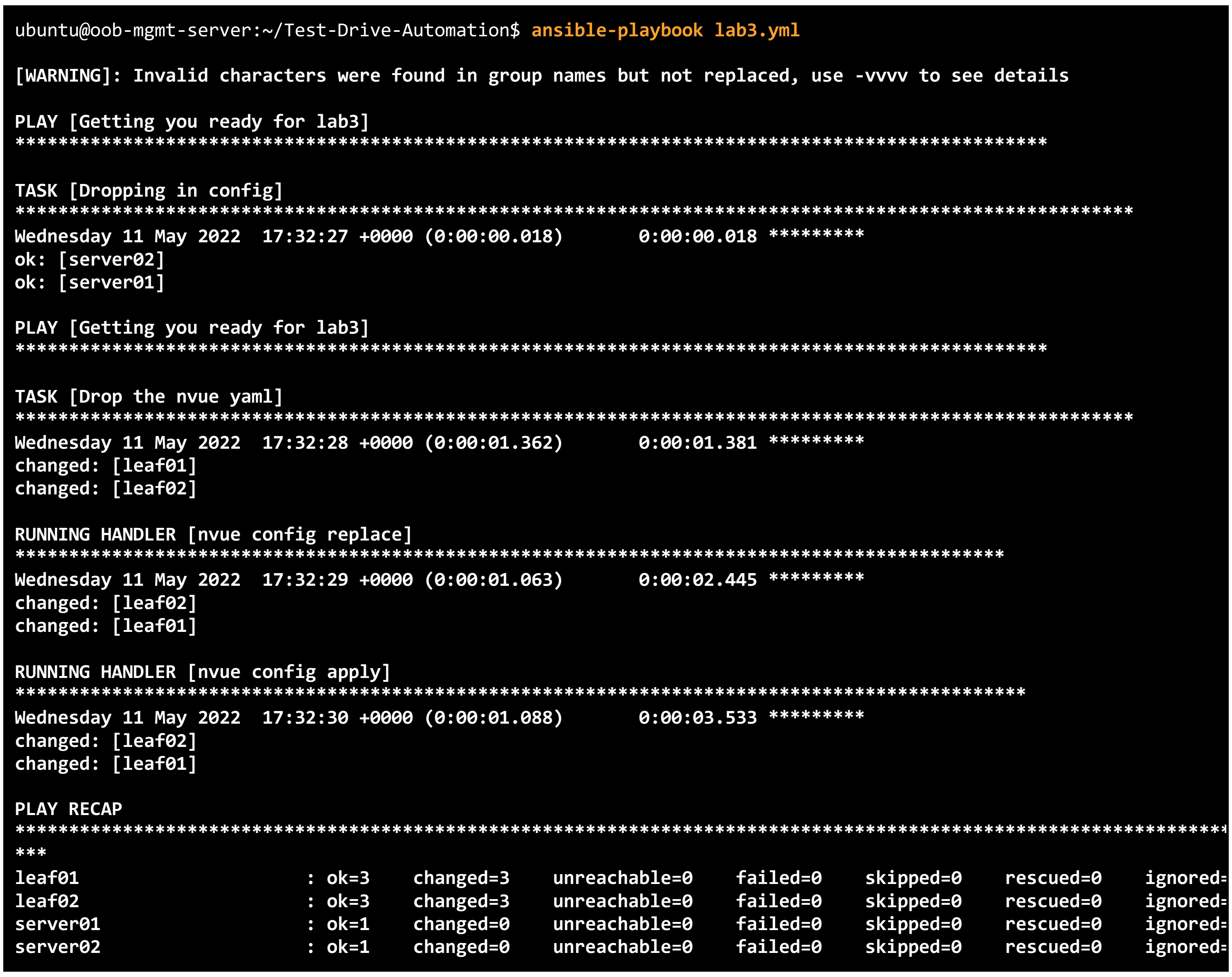
Goals:

- Configure BGP unnumbered on spine01
- Configure BGP unnumbered on leaf01/leaf02
- Advertise loopback addresses into BGP
- Advertise SVI subnets of leafs into BGP
- Verify BGP peering
- Verify BGP route advertisements
- Verify routed connectivity and path between servers

Procedure:

Run Lab3 setup playbook

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: [server02]
ok: [server01]

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: [leaf02]
changed: [leaf01]

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

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

Apply loopback address to spine01

Loopback Configuration			
Configuration↓ \ Switch→	leaf01	leaf02	spine01
Loopback IP address	10.255.255.1/32	10.255.255.2/32	10.255.255.101/32

2. On spine01: Configure a loopback interface

```
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
```

Leaf01 and Leaf02 loopback addresses are already configured.

Configure BGP unnumbered on spine01, leaf01 and leaf02

3. On spine01: Configure a BGP Autonomous System (AS) number for the routing instance.

```
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: Commit the changes.

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

6. On leaf01: Repeat steps 2-5.

```
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
```

For copy/paste convenience:

```
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 steps 2-5.

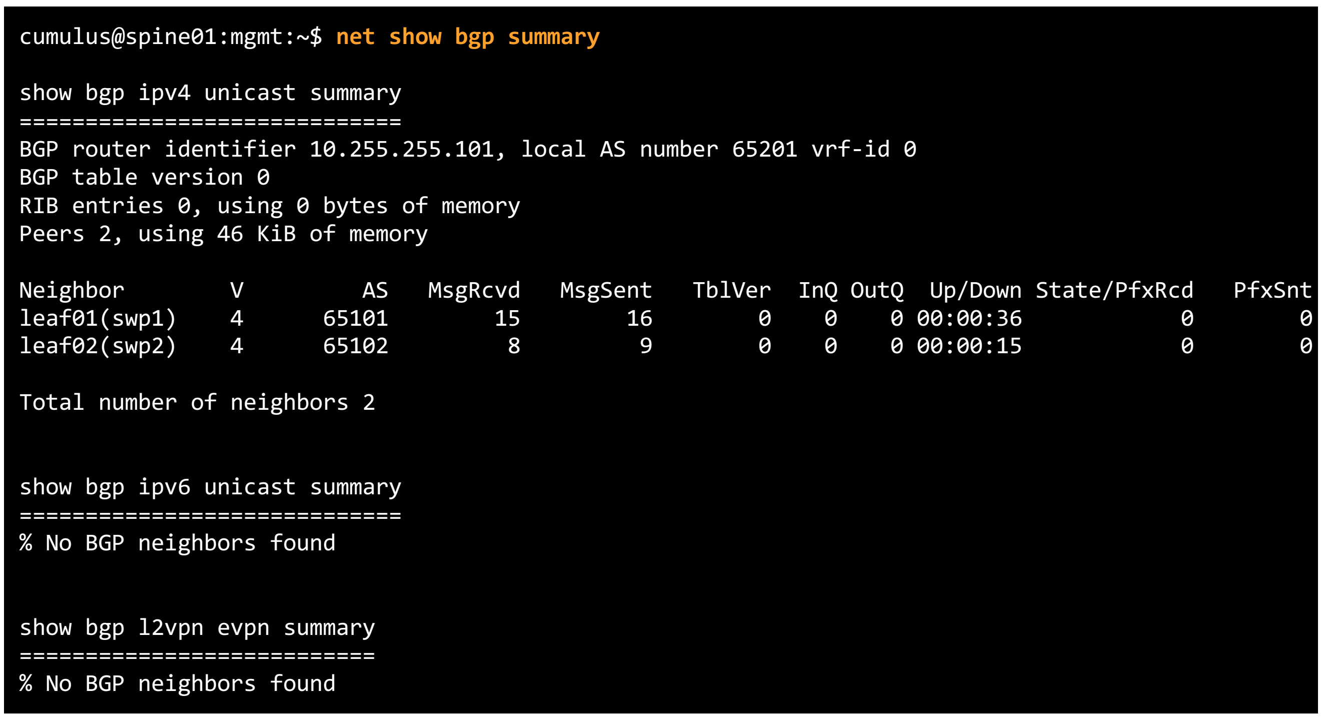
```
cumulus@leaf02:mgmt:~$ nv set vrf default router bgp autonomous-system 65102
cumulus@leaf02:mgmt:~$ nv set vrf default router bgp path-selection multipath aspath-ignore on
cumulus@leaf02:mgmt:~$ nv set vrf default router bgp router-id 10.255.255.2
cumulus@leaf02:mgmt:~$ nv set vrf default router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

For copy/paste convenience:

```
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
```

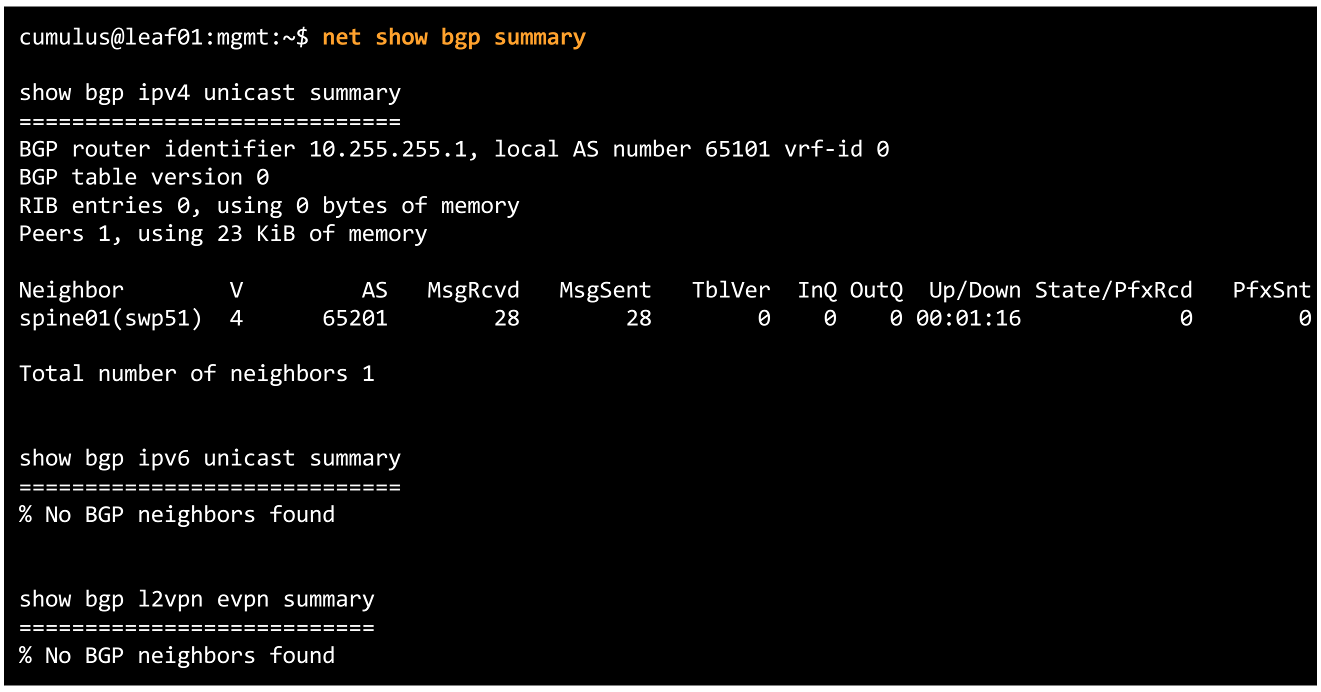
Verify BGP connectivity between fabric nodes

8. On spine01: Verify BGP peering between spine and leafs.

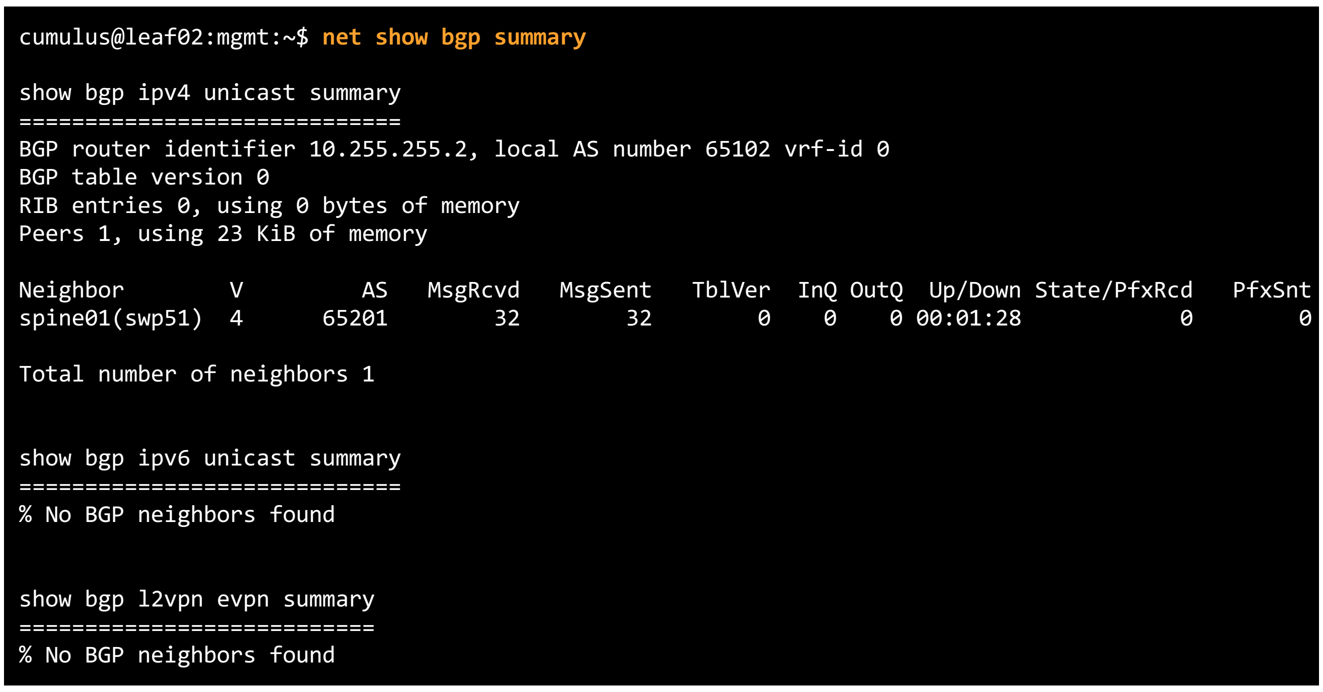


```
cumulus@spine01:mgmt:~$ net show bgp summary

show bgp ipv4 unicast summary
=============================
BGP router identifier 10.255.255.101, local AS number 65201 vrf-id 0
BGP table version 0
RIB entries 0, using 0 bytes of memory
Peers 2, using 46 KiB of memory

Neighbor        V         AS   MsgRcvd   MsgSent   TblVer  InQ OutQ  Up/Down State/PfxRcd   PfxSnt
leaf01(swp1)    4      65101        15        16        0    0    0 00:00:36            0        0
leaf02(swp2)    4      65102         8         9        0    0    0 00:00:15            0        0

Total number of neighbors 2


show bgp ipv6 unicast summary
=============================
% No BGP neighbors found


show bgp l2vpn evpn summary
===========================
% No BGP neighbors found
```

9. On leaf01: Verify BGP peering between leafs and spine



```
cumulus@leaf01:mgmt:~$ net show bgp summary

show bgp ipv4 unicast summary
=============================
BGP router identifier 10.255.255.1, local AS number 65101 vrf-id 0
BGP table version 0
RIB entries 0, using 0 bytes of memory
Peers 1, using 23 KiB of memory

Neighbor        V         AS   MsgRcvd   MsgSent   TblVer  InQ OutQ  Up/Down State/PfxRcd   PfxSnt
spine01(swp51)  4      65201        28        28        0    0    0 00:01:16            0        0

Total number of neighbors 1


show bgp ipv6 unicast summary
=============================
% No BGP neighbors found


show bgp l2vpn evpn summary
===========================
% No BGP neighbors found
```



```
cumulus@leaf02:mgmt:~$ net show bgp summary

show bgp ipv4 unicast summary
=============================
BGP router identifier 10.255.255.2, local AS number 65102 vrf-id 0
BGP table version 0
RIB entries 0, using 0 bytes of memory
Peers 1, using 23 KiB of memory

Neighbor        V         AS   MsgRcvd   MsgSent   TblVer  InQ OutQ  Up/Down State/PfxRcd   PfxSnt
spine01(swp51)  4      65201        32        32        0    0    0 00:01:28            0        0

Total number of neighbors 1


show bgp ipv6 unicast summary
=============================
% No BGP neighbors found


show bgp l2vpn evpn summary
===========================
% No BGP neighbors found
```

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
- Show commands can be a mix of “nv show” and legacy “net show”. Here we are using “net show” commands

Advertise Loopback and SVI subnets from leaf01, leaf02 and spine01 into fabric

Routing Advertisement Configuration	

Routes↓ \ Switch→	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

10. On spine01: Advertise loopback address into BGP.

```
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
```

11. 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
```

12. 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
```

13. On leaf01: Commit the changes.

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

14. On leaf02: Repeat steps xx-xx. Notice the different loopback IP and subnet that is advertised.

```
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
```

```
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 that BGP is advertising the routes

15. On spine01: Check that routes are being learned.

```
cumulus@spine01:mgmt:~$ net show bgp
show 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, * 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

   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                  0         32768 i

Displayed  5 routes and 5 total paths


show bgp ipv6 unicast
=====================
No BGP prefixes displayed, 0 exist
```

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 connectivity and path between server01 and server02

16. 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
```

17. On Server01, traceroute to Server02. Identify all of 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.

This concludes the NVIDIA Cumulus Linux Workshop.

pendix 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/i4AlXVcOQl5tWBCAq1tPWmZHJCPcElFTjeG2wMYMx2Kmb3kYwrLcwfTk06avziBhjMwIprFiupWCkykRPm4IOHkiW
DS/htpZfBdwuIFXV4MQtCiD9zUhLi0Uq0Is+lVtE1Q0/38N7sSa7FHaVNpDpJOQf3PYVdfhk/BG19WQlyKMUSjOaRrAHUIckiQs2H5Wm198ciKkgl4AxoDM9QB+flcCl3We52ei5tWqV8CgLeh
hrdjEXn+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 important to 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.

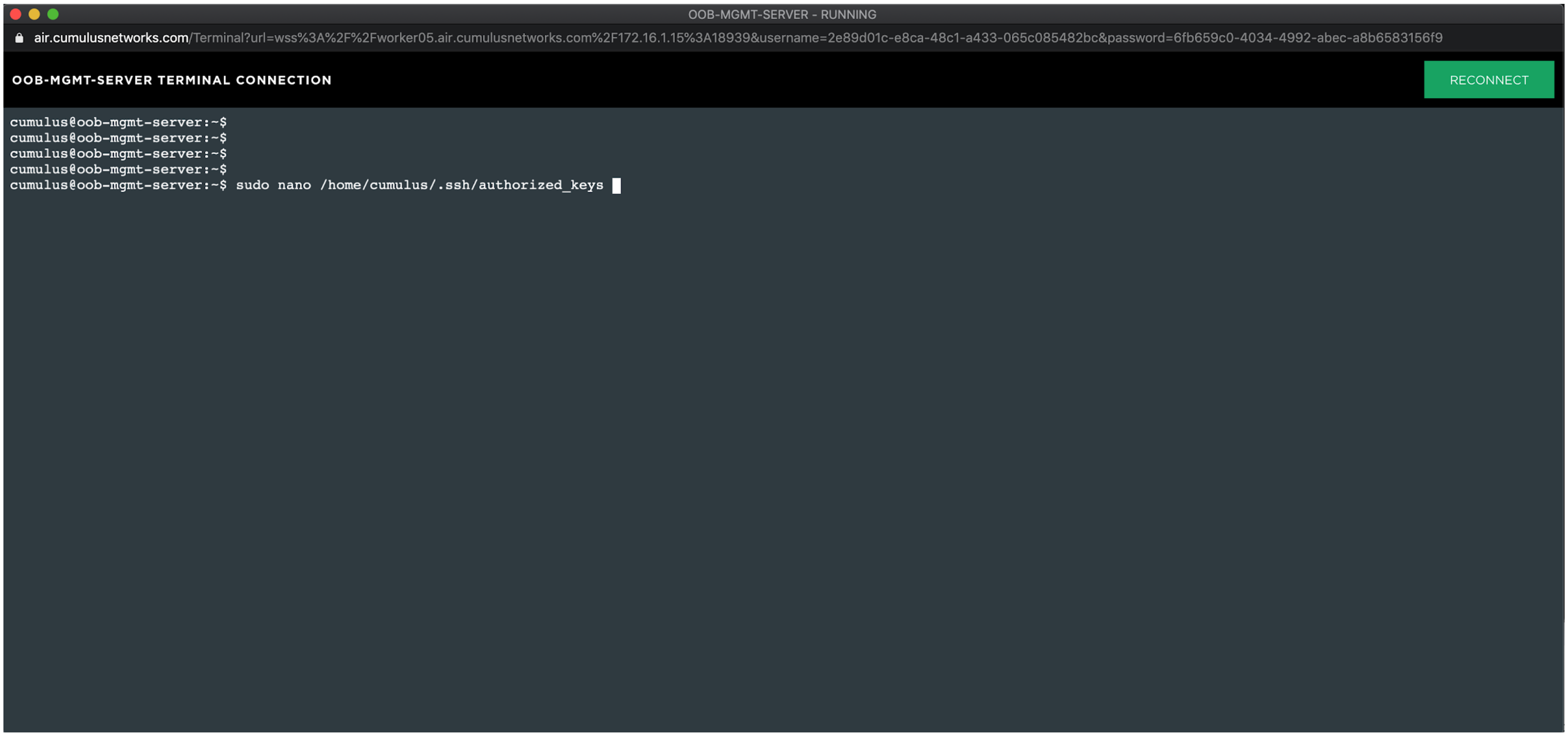
Windows (using putty): https://www.ssh.com/ssh/putty/windows/puttygen
Mac: https://docs.joyent.com/public-cloud/getting-started/ssh-keys/generating-an-ssh-key-manually/manually-generating-your-ssh-key-in-mac-os-x
Ubuntu/linux: https://help.ubuntu.com/community/SSH/OpenSSH/Keys

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/i4AlXVcOQl5tWBCAq1tPWmZHJCPcElFTjeG2wMYMx2Kmb3kYwrLcwfTk06avziBhjMwIprFiupWCkykRPm4IOHk
iWDS/htpZfBdwuIFXV4MQtCiD9zUhLi0Uq0Is+lVtE1Q0/38N7sSa7FHaVNpDpJOQf3PYVdfhk/BG19WQlyKMUSjOaRrAHUIckiQs2H5Wm198ciKkgl4AxoDM9QB+flcCl3We52ei5tWqV8C
gLehhrdjEXn+iXNdkcg+nGka1syUSYntotally+fake+key+MkEFwD9v16SmJYDK67w5RaHTjBS52UoRjnEEN user@hostname.local
```

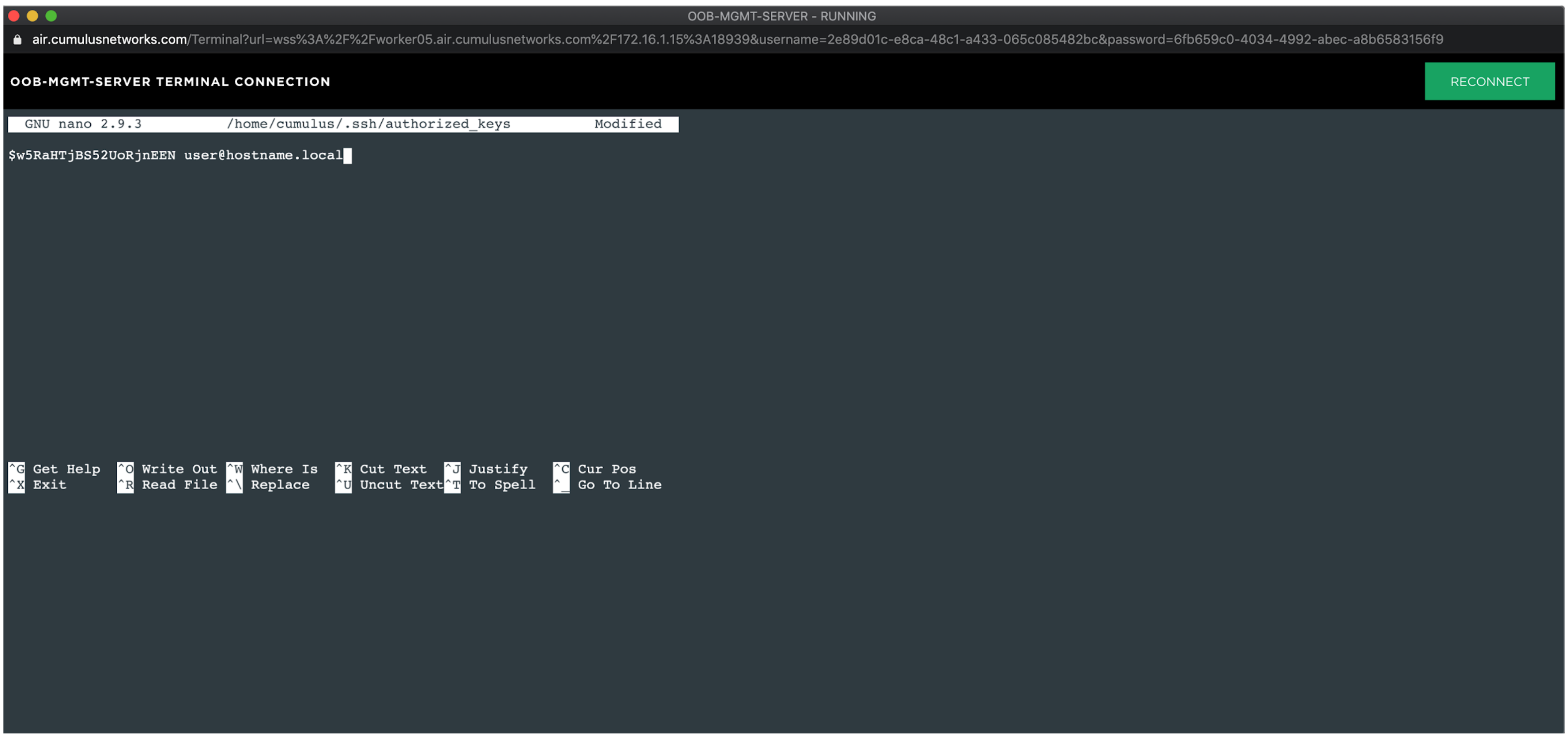
1. **First, you must add your ssh pubkey to the oob-mgmt-server authorized_keys file for the cumulus user. Pop out the console window of the oob-mgmt-server.**

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


OOB-MGMT-SERVER - RUNNING
air.cumulusnetworks.com/Terminal?url=wss%3A%2F%2Fworker05.air.cumulusnetworks.com%2F172.16.1.15%3A18939&username=2e89d01c-e8ca-48c1-a433-065c085482bc&password=6fb659c0-4034-4992-abec-a8b6583156f9
OOB-MGMT-SERVER TERMINAL CONNECTION
RECONNECT

```
cumulus@oob-mgmt-server:~$
cumulus@oob-mgmt-server:~$
cumulus@oob-mgmt-server:~$
cumulus@oob-mgmt-server:~$
cumulus@oob-mgmt-server:~$ sudo nano /home/cumulus/.ssh/authorized_keys
```

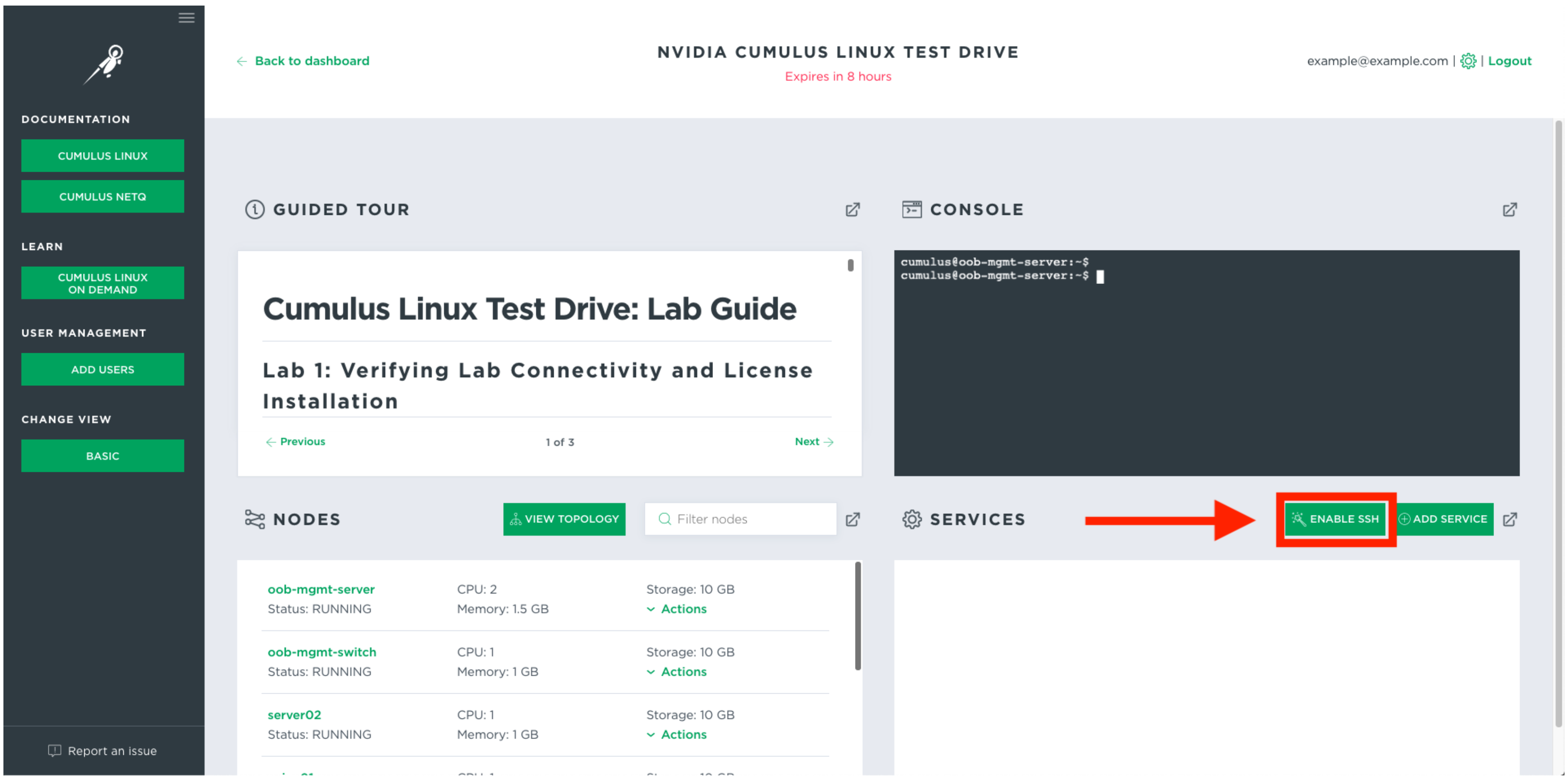
   Paste in your ssh pubkey string. It is a long single line string that doesn't wrap in nano.


OOB-MGMT-SERVER - RUNNING
air.cumulusnetworks.com/Terminal?url=wss%3A%2F%2Fworker05.air.cumulusnetworks.com%2F172.16.1.15%3A18939&username=2e89d01c-e8ca-48c1-a433-065c085482bc&password=6fb659c0-4034-4992-abec-a8b6583156f9
OOB-MGMT-SERVER TERMINAL CONNECTION
RECONNECT

```
  GNU nano 2.9.3           /home/cumulus/.ssh/authorized_keys          Modified

$w5RaHTjBS52UoRjnEEN user@hostname.local









^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

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.



Cumulus Linux Test Drive: Lab Guide

Lab 1: Verifying Lab Connectivity and License Installation

← Previous 1 of 3 Next →

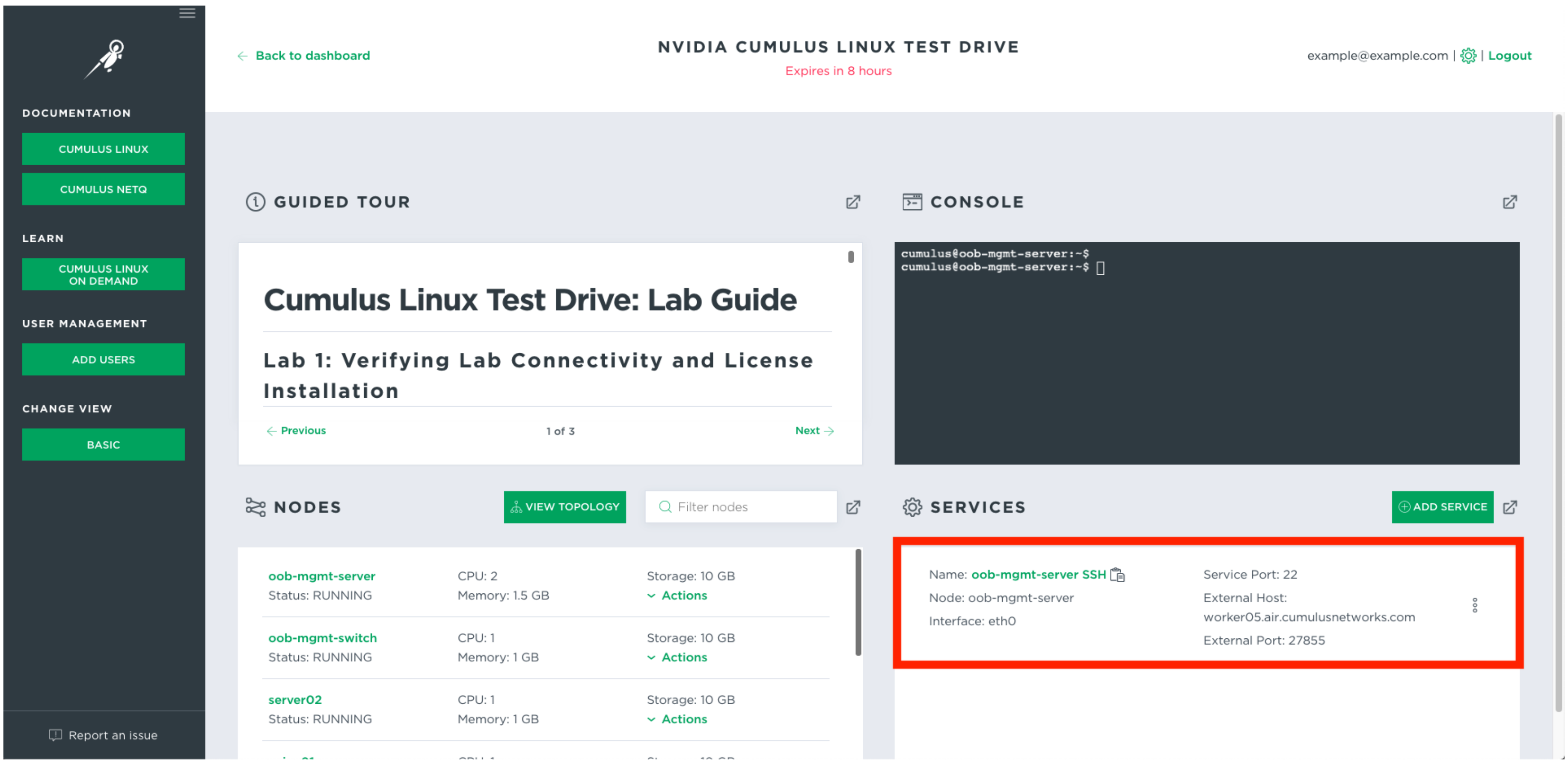
```
cumulus@oob-mgmt-server:~$
cumulus@oob-mgmt-server:~$
```

oob-mgmt-server
Status: RUNNING CPU: 2
Memory: 1.5 GB Storage: 10 GB
Actions

oob-mgmt-switch
Status: RUNNING CPU: 1
Memory: 1 GB Storage: 10 GB
Actions

server02
Status: RUNNING CPU: 1
Memory: 1 GB Storage: 10 GB
Actions

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



Cumulus Linux Test Drive: Lab Guide

Lab 1: Verifying Lab Connectivity and License Installation

← Previous 1 of 3 Next →

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

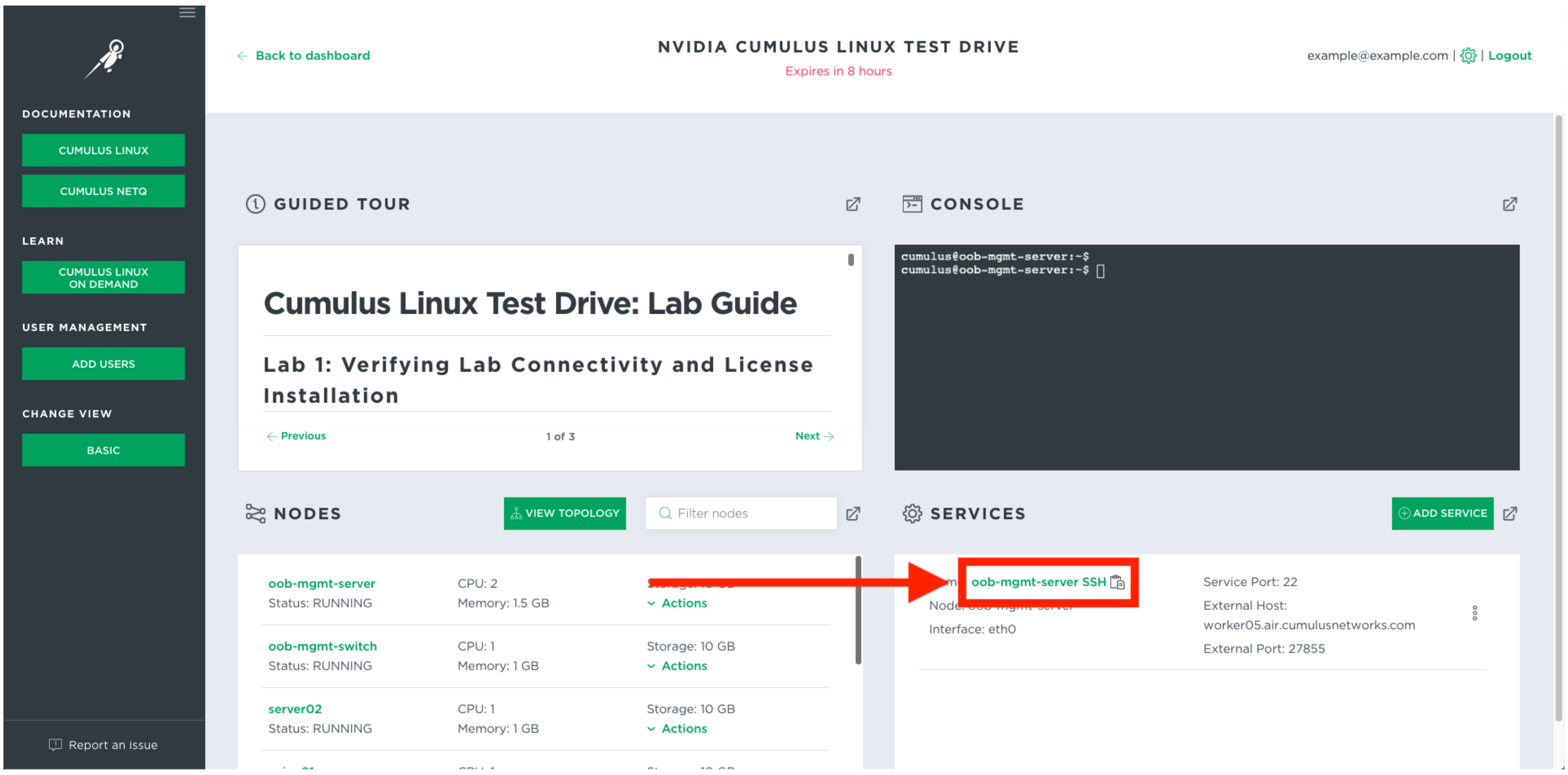
oob-mgmt-server Status: RUNNING CPU: 2 Memory: 1.5 GB Storage: 10 GB Actions

oob-mgmt-switch Status: RUNNING CPU: 1 Memory: 1 GB Storage: 10 GB Actions

server02 Status: RUNNING CPU: 1 Memory: 1 GB Storage: 10 GB Actions

Name: oob-mgmt-server SSH
Node: oob-mgmt-server
Interface: eth0
Service Port: 22
External Host: worker05.air.cumulusnetworks.com
External Port: 27855

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.



Cumulus Linux Test Drive: Lab Guide

Lab 1: Verifying Lab Connectivity and License Installation

← Previous 1 of 3 Next →

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

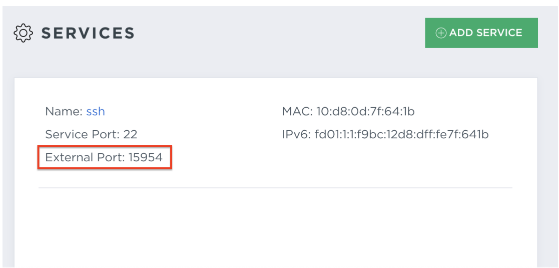
oob-mgmt-server Status: RUNNING CPU: 2 Memory: 1.5 GB Actions

oob-mgmt-switch Status: RUNNING CPU: 1 Memory: 1 GB Storage: 10 GB Actions

server02 Status: RUNNING CPU: 1 Memory: 1 GB Storage: 10 GB Actions

oob-mgmt-server SSH
Interface: eth0
Service Port: 22
External Host: worker05.air.cumulusnetworks.com
External Port: 27855

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:

Manual SSH Connection Details		 SERVICES ADD SERVICE Name: ssh Service Port: 22 External Port: 15954 MAC: 10:d8:0d:7f:64:1b IPv6: fd01:1:1:f9bc:12d8:dff:fe7f:641b
Username	cumulus	
~~Password~~	If prompted for a password, you are being asked for the password/passphrase for *your ssh private key*	
Server Hostname	air.cumulusnetworks.com	
SSH Port	*Use "External Port" in Services box on UI* --------------------->	

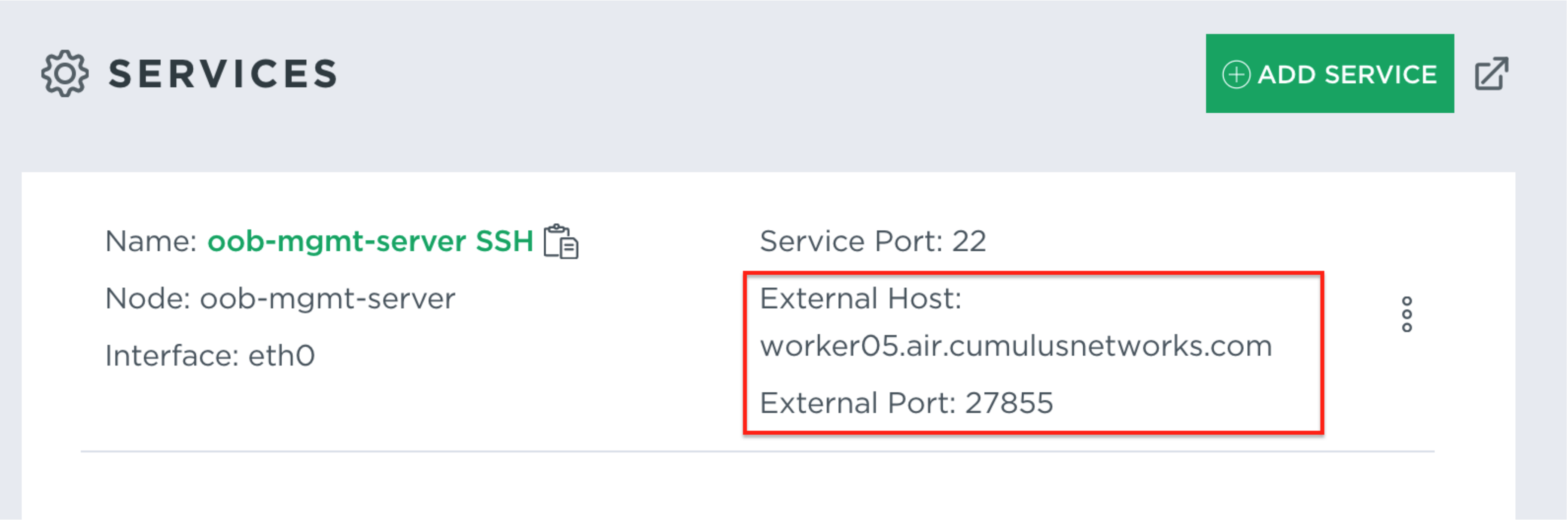
Note: This SSH connection ***does not use the default destination TCP port 22***. Ensure that the external port is specified in your SSH client.
Note: If you are prompted for a password, it is the password to access your SSH private key. This is not a password being requested from the server for authentication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

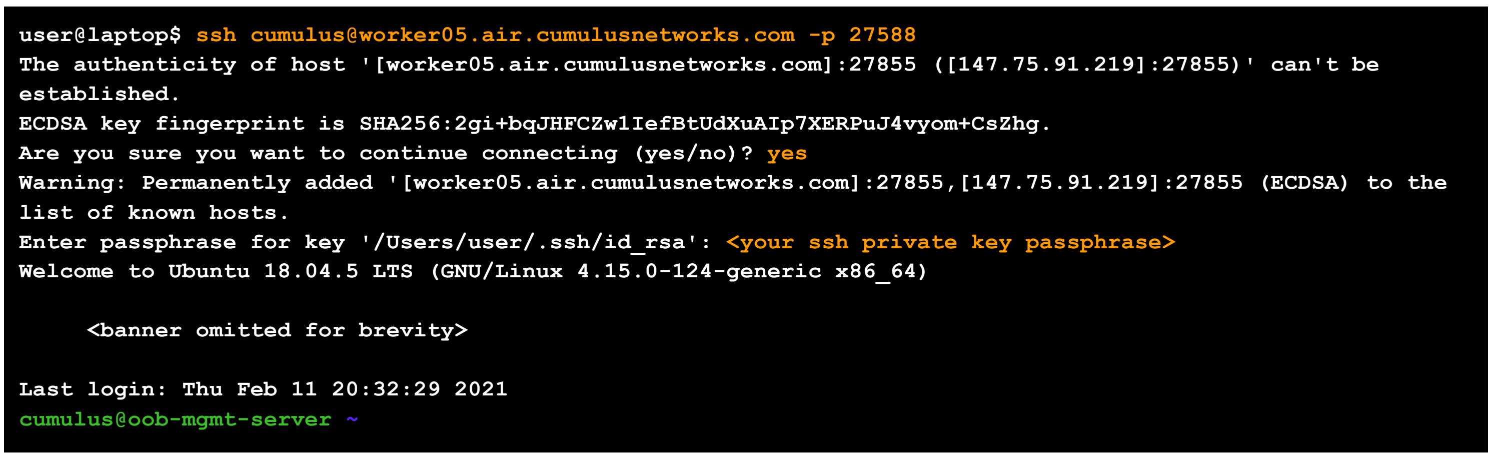
Linux/Mac OS:

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



Name: oob-mgmt-server SSH
Node: oob-mgmt-server
Interface: eth0
Service Port: 22
External Host: worker05.air.cumulusnetworks.com
External Port: 27855

Example)

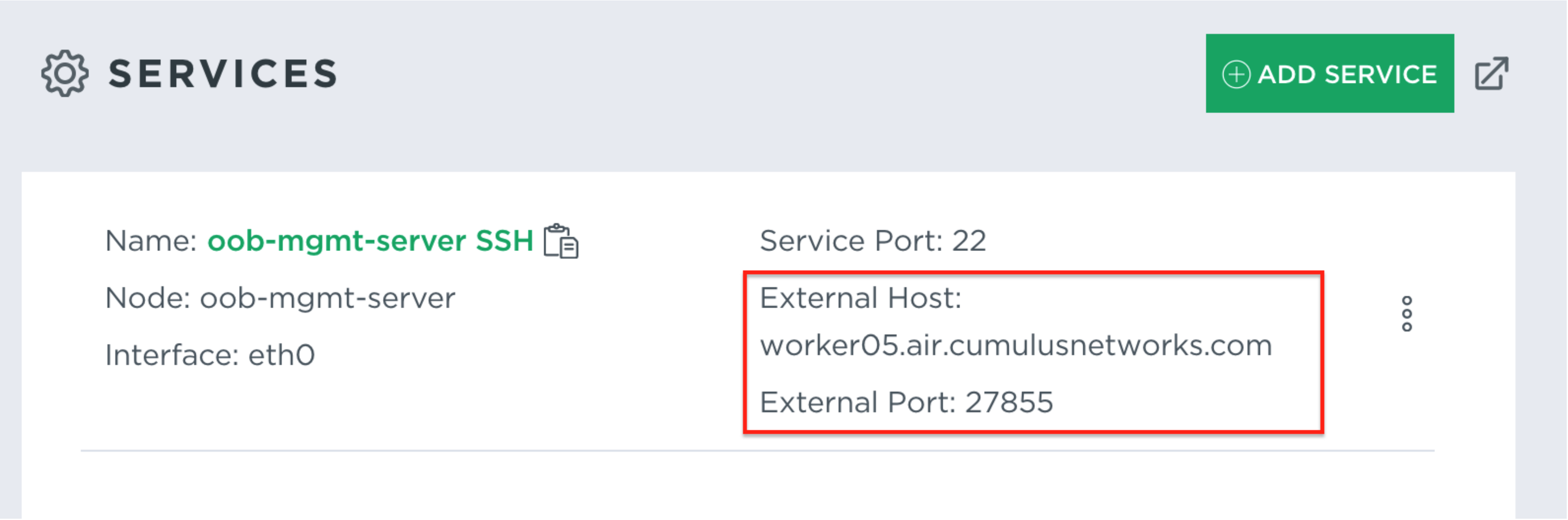


```
user@laptop$ ssh cumulus@worker05.air.cumulusnetworks.com -p 27588
The authenticity of host '[worker05.air.cumulusnetworks.com]:27855 ([147.75.91.219]:27855)' can't be
established.
ECDSA key fingerprint is SHA256:2gi+bqJHFCZw1IefBtUdXuAIp7XERPuJ4vyom+CsZhg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[worker05.air.cumulusnetworks.com]:27855,[147.75.91.219]:27855 (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
cumulus@oob-mgmt-server ~
```

Windows using PuTTY:

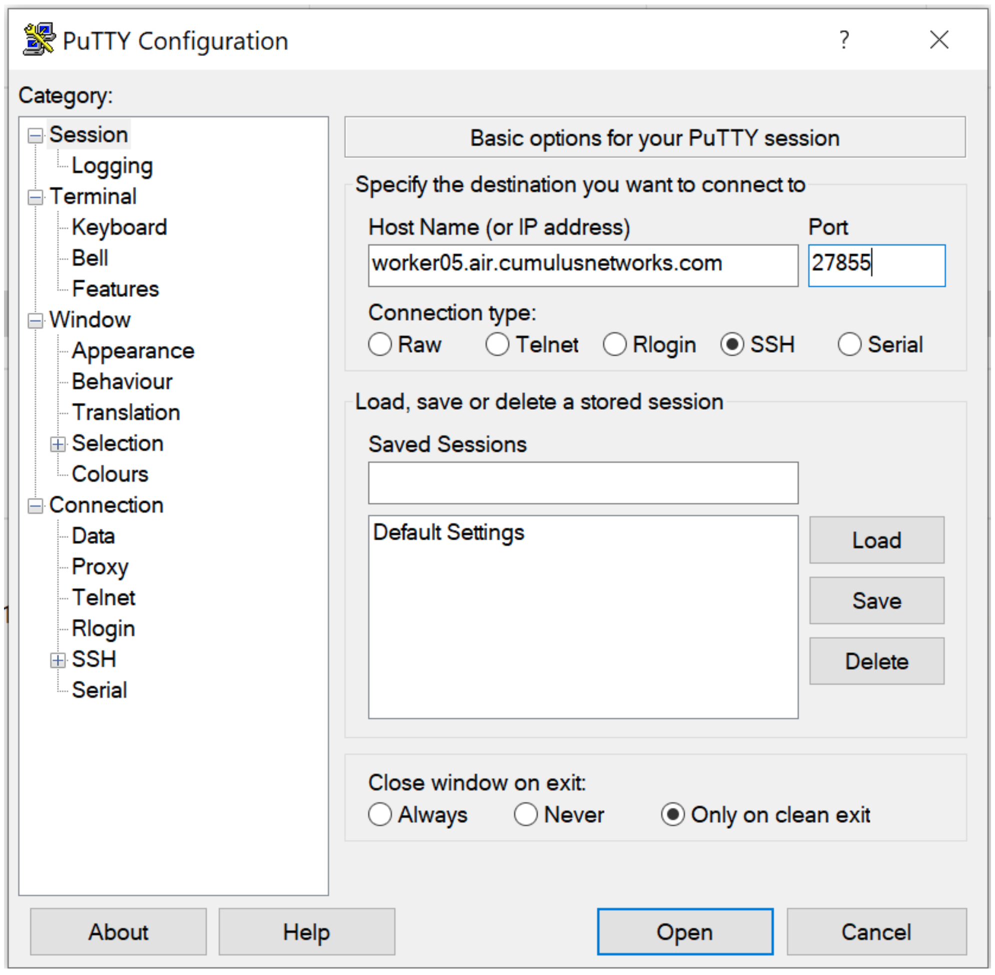
You must tell PuTTY the SSH private key to use for this connection. The public key should be on the oob-mgmt-server in the authorized_keys file. Now your client must use your SSH private key. Here is an example: https://devops.ionos.com/tutorials/use-ssh-keys-with-putty-on-windows/#connect-to-server-with-private-key

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



Name: oob-mgmt-server SSH
Node: oob-mgmt-server
Interface: eth0
Service Port: 22
External Host: worker05.air.cumulusnetworks.com
External Port: 27855

In your PuTTY Connection Info:



Specify the destination you want to connect to

Host Name (or IP address): worker05.air.cumulusnetworks.com

Port: 27855

Connection type:
○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

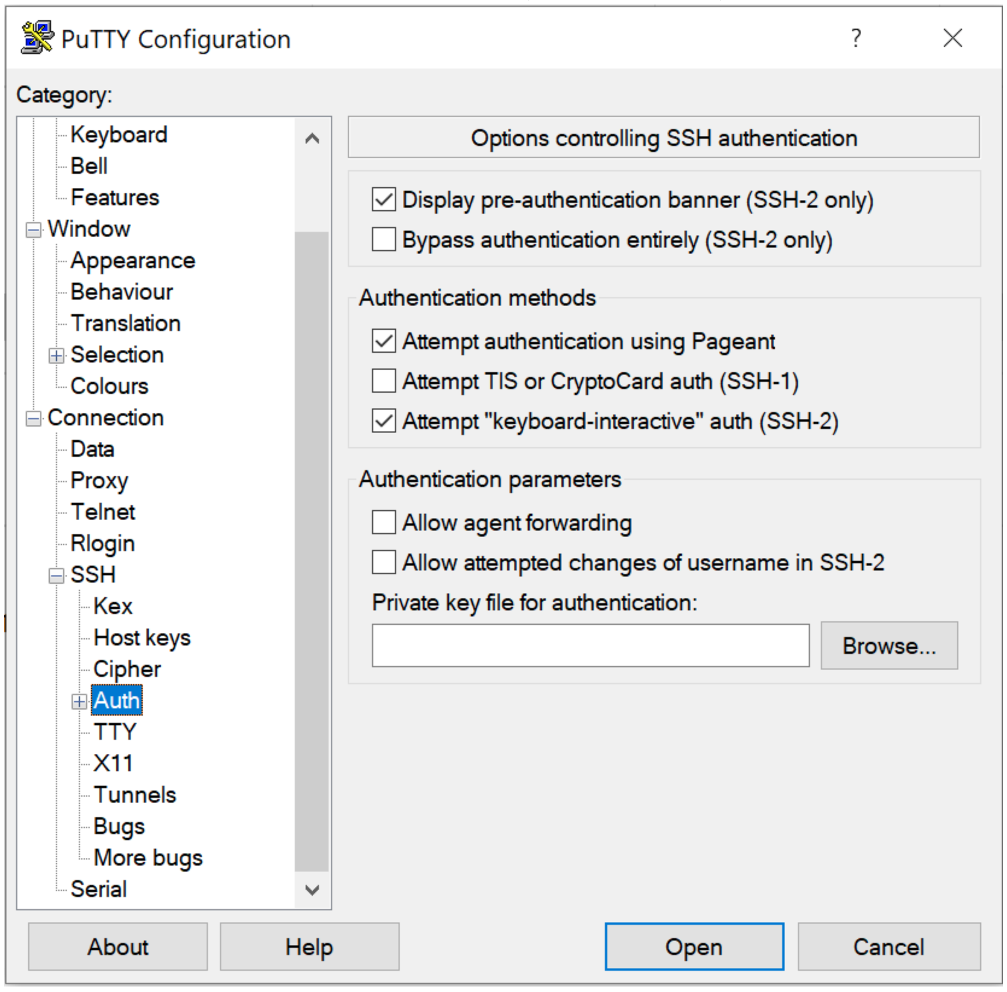
Saved Sessions

Default Settings

Close window on exit:
○ Always ○ Never ● Only on clean exit

```
Hostname: workerXX.air.cumulusnetworks.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.



☑ Display pre-authentication banner (SSH-2 only)

☐ Bypass authentication entirely (SSH-2 only)

Authentication methods

☑ Attempt authentication using Pageant

☐ Attempt TIS or CryptoCard auth (SSH-1)

☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

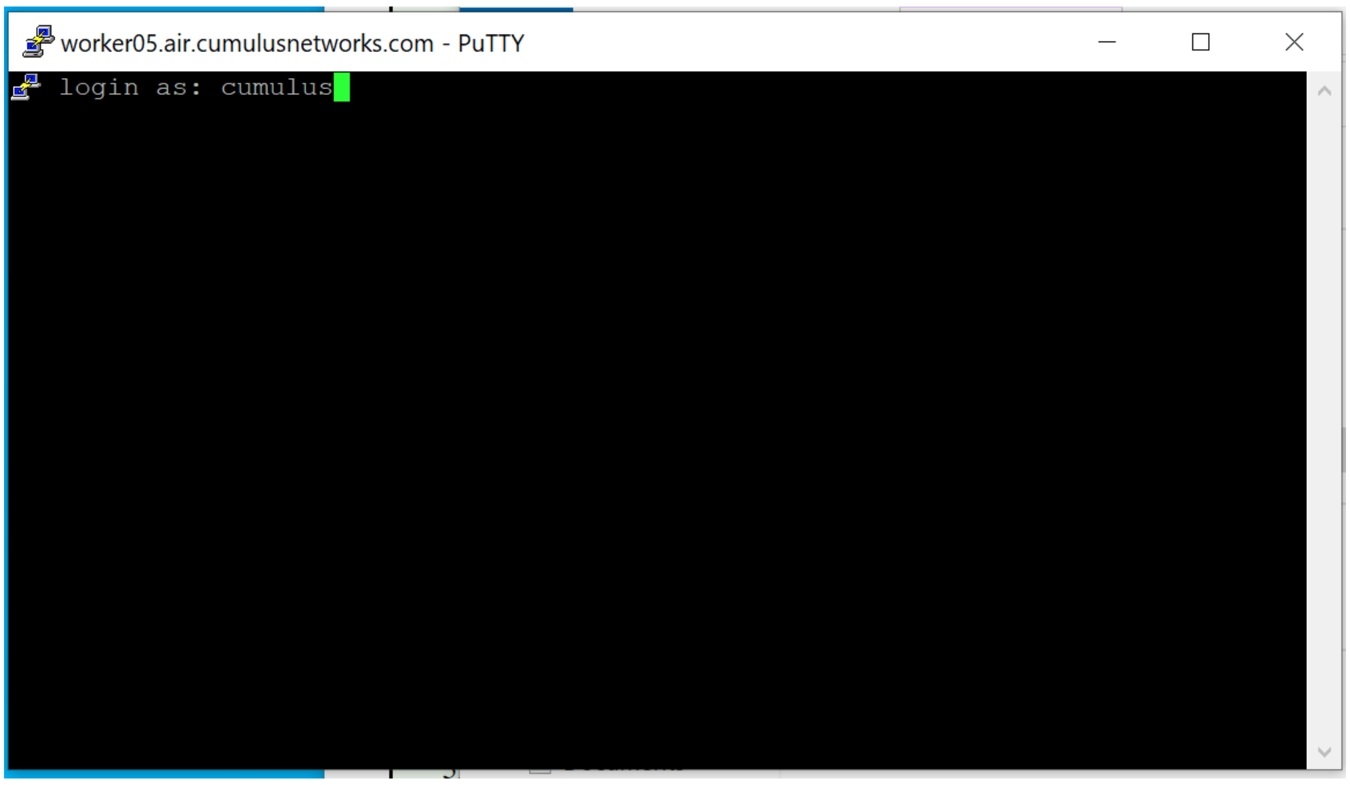
☐ Allow agent forwarding

☐ Allow attempted changes of username in SSH-2

Private key file for authentication:

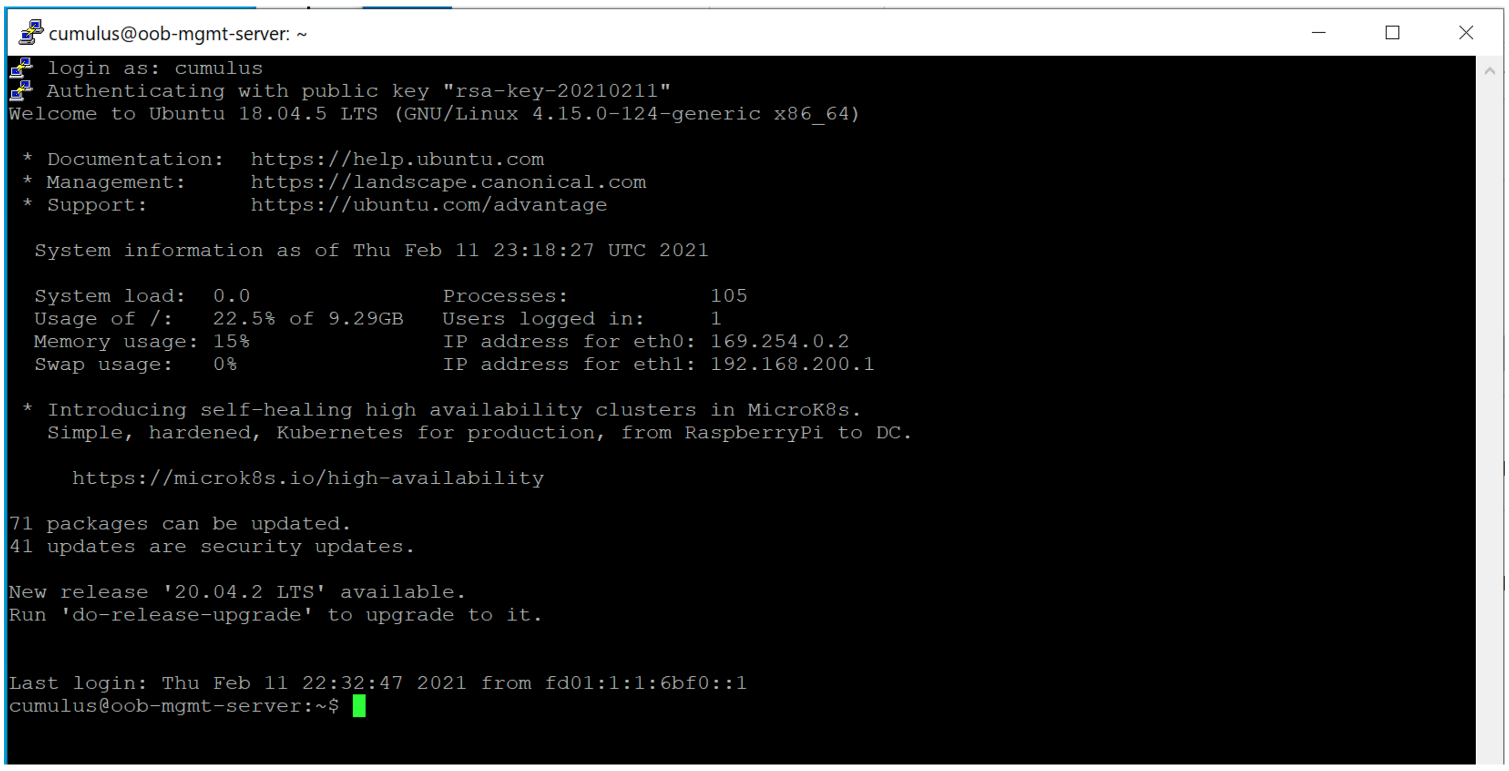
Click Open.

When prompted, login as user cumulus



```
login as: cumulus
```

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



```
login as: cumulus
Authenticating with public key "rsa-key-20210211"
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-124-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Feb 11 23:18:27 UTC 2021

  System load:  0.0               Processes:           105
  Usage of /:   22.5% of 9.29GB   Users logged in:     1
  Memory usage: 15%               IP address for eth0: 169.254.0.2
  Swap usage:   0%                IP address for eth1: 192.168.200.1

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

     https://microk8s.io/high-availability

71 packages can be updated.
41 updates are security updates.

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Thu Feb 11 22:32:47 2021 from fd01:1:1:6bf0::1
cumulus@oob-mgmt-server:~$
```

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

Last updated: Jan 10th 2022

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/.