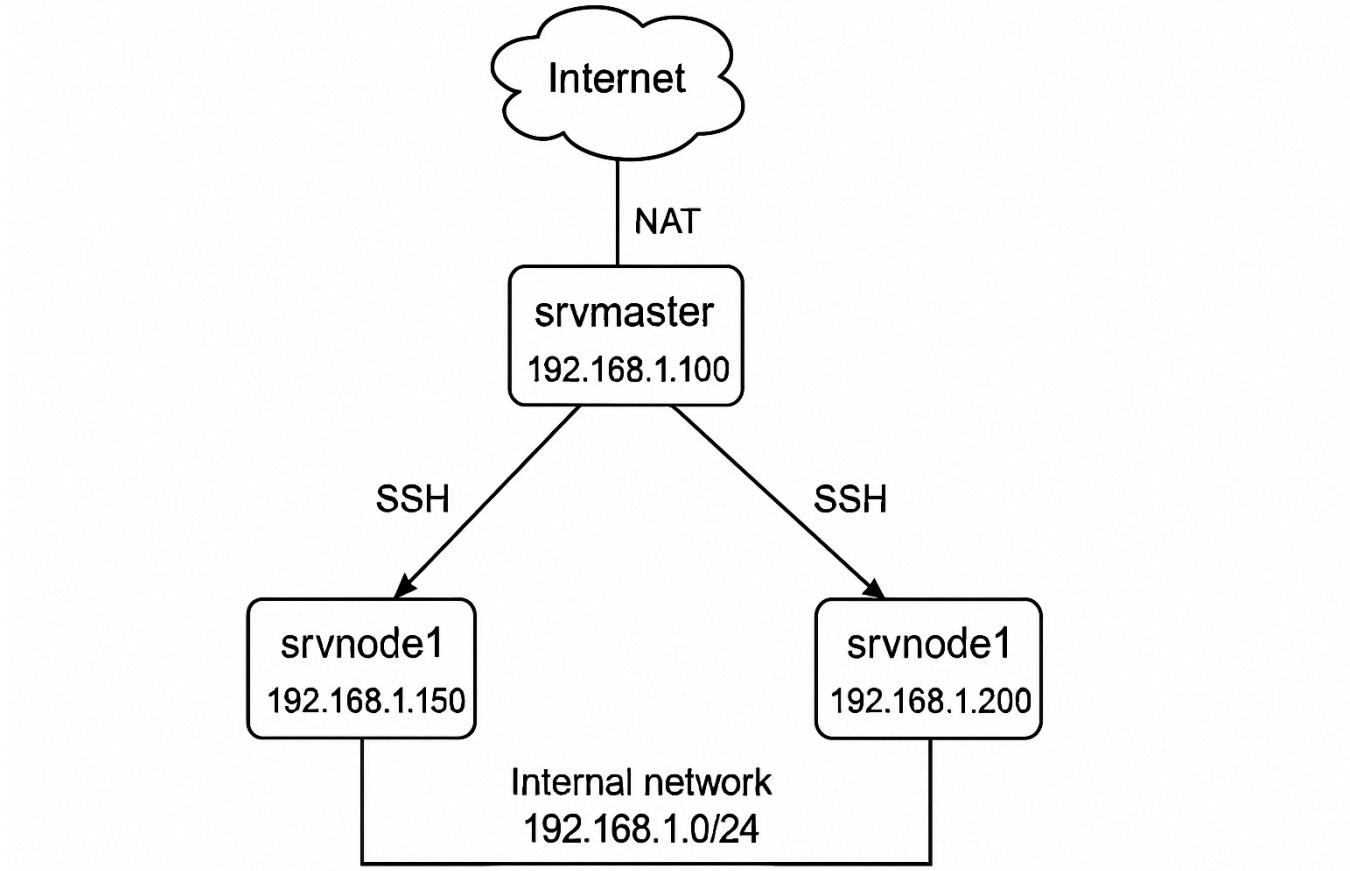
TEAM 7: Aaron Davidson, Kin Johnson, Natalie Voltaire, Maria Fraguas Jover, Nyah Hepburn, Sheniese Aracena-Baez

Three-node Ubuntu cluster step by step

Guide to setting up your three-node Ubuntu cluster (control-server, work-server-1, work-server-2) in VirtualBox with internet access (NAT) and a private internal network (192.168.1.x) for secure command and control via passwordless SSH.



Goal: Create 3 Ubuntu VMs (srvmaster, srvnode1, srvnode2) where:
● Each VM can access the internet independently.
● VMs communicate with each other over a private network (ex: 192.168.1.0/24).
● srvmaster can SSH into srvnode1 and srvnode2 without a password.

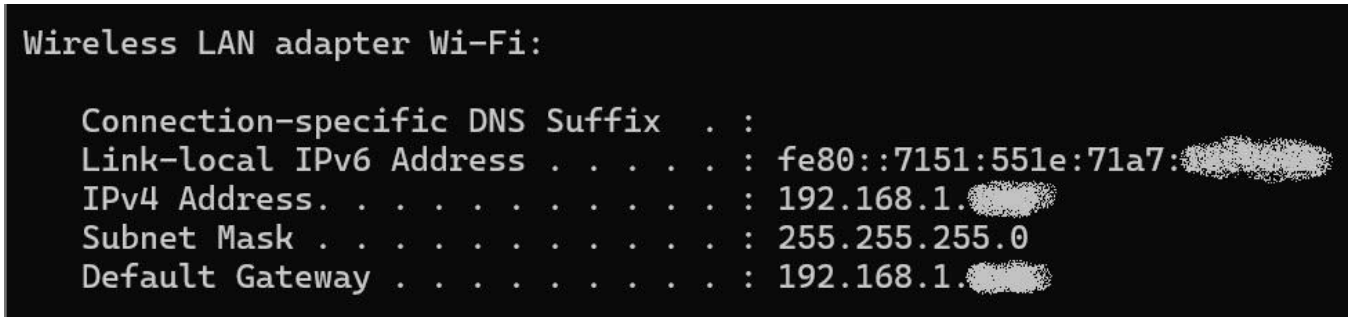
Current Date: Monday, April 14, 2025

Phase 0: Prerequisites & Planning

1. Software:
○ Oracle VirtualBox installed on your host machine.
○ Ubuntu Server ISO image downloaded (Recommended: Latest LTS version, e.g., 22.04 or 24.04). Using Server edition is more lightweight.
2. Naming & User:

○ VM/Hostnames: srvmaster, srvnode1, srvnode2 (whatever you choose, keep it consistent throughout the tutorial.
○ Consistent Username: Choose a username for administration on all nodes (e.g., clusteradmin).
3. IP Addressing Scheme:
○ Adapter 1 (Internet): Will use NAT mode. VirtualBox provides DHCP (typically 10.0.2.x). No static configuration needed within Ubuntu for this adapter.
○ Adapter 2 (Internal): Will use Internal Network mode named cluster-net.
Static IPs will be assigned within Ubuntu:
■ srvmaster: 192.168.1.100
■ srvnode1: 192.168.1.150
■ srvnode2: 192.168.1.200
■ Subnet Mask: 255.255.255.0 (represented as /24)
4. (CRITICAL) Check for Host IP Conflict:
○ On your host machine (Windows/Mac/Linux), check your own network IP address (ipconfig or ip addr).
○ If your host machine's network already uses 192.168.1.x: You MUST choose a different private range for your internal network (e.g., 192.168.100.100, 192.168.100.150, 192.168.100.200 with /24). If you proceed with 192.168.1.x in this case, you will have network problems. Adjust the static IPs in the steps below if necessary.

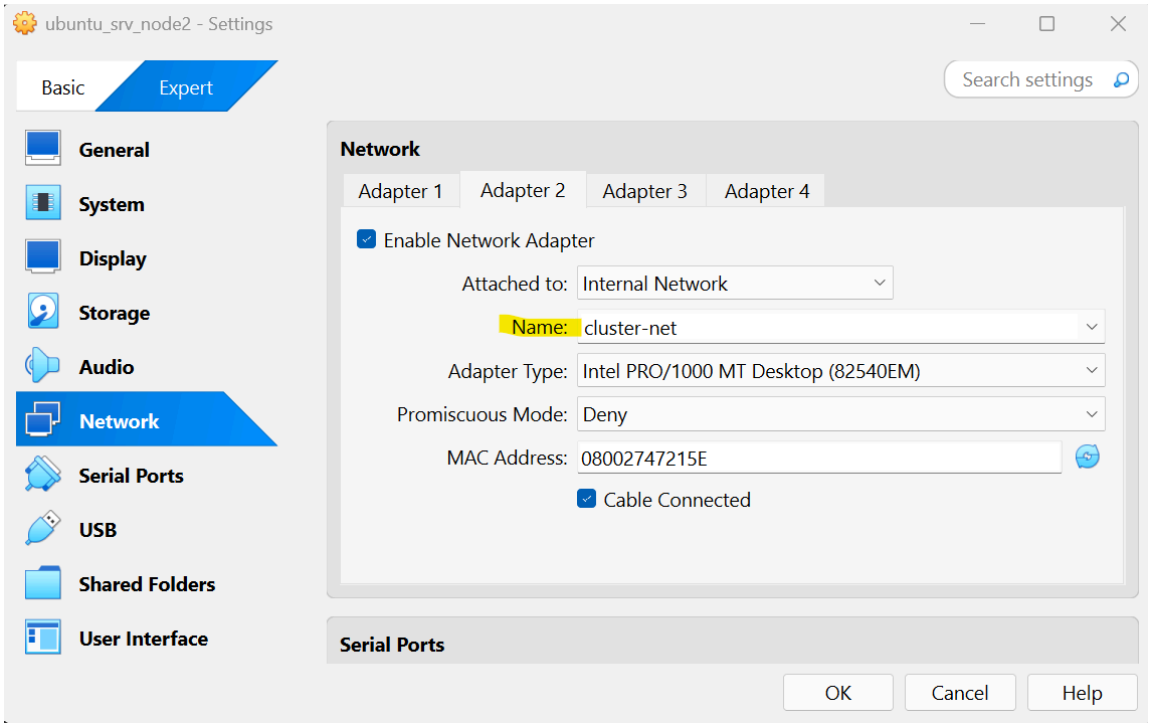
Below is an example from the command line of a host IPv4 address that is 192.168.1.x



In this case, you can use the suggested private range above.

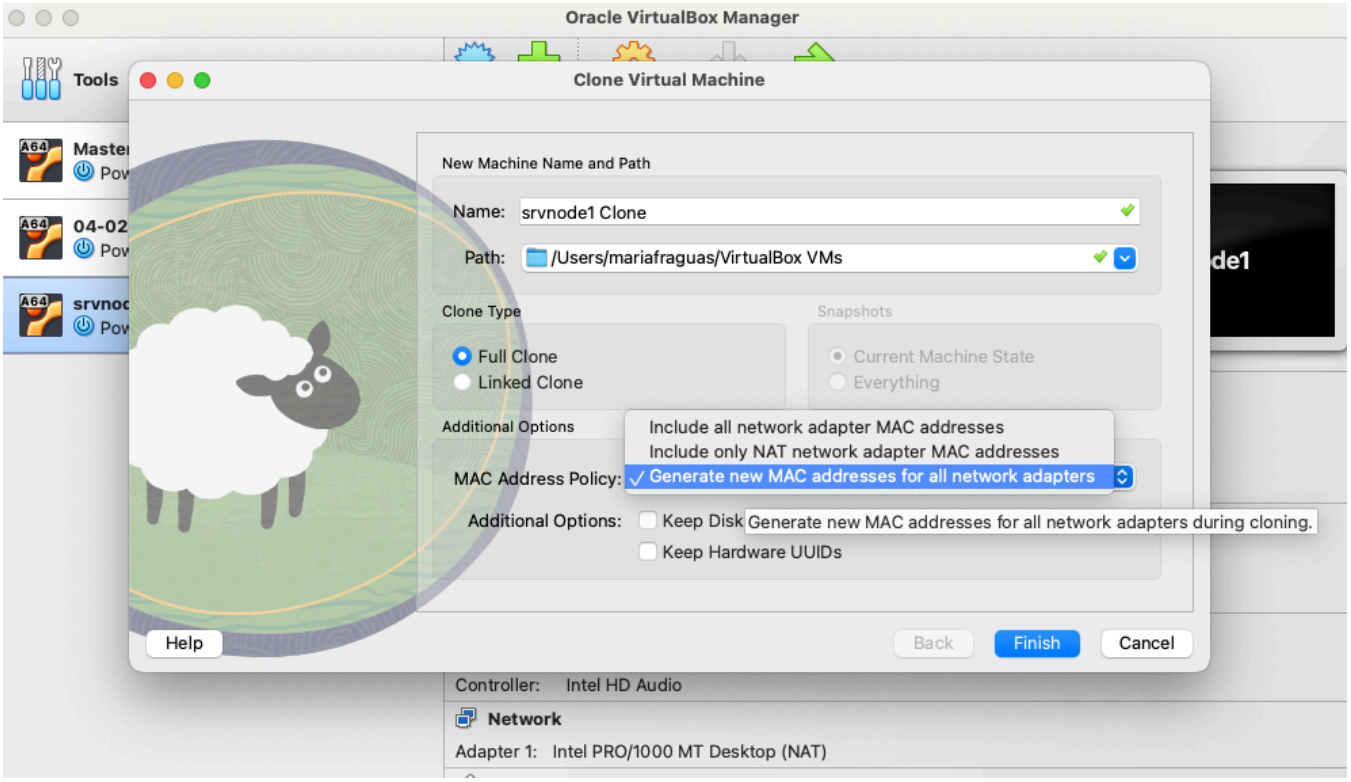
Phase 1: VirtualBox Network Setup (Internal Network Identifier)

● While the "Internal Network" is created implicitly when you use its name in VM settings, ensure you use the exact same name for all VMs. Let's call it cluster-net.
No specific IP configuration is done at the global VirtualBox level for Internal Networks.



Phase 2: Virtual Machine Creation & Initial Configuration

First create 1 srvmaster, then clone it to create node1 and 2, and only then you will install ubuntu desktop on the srvmaster.
Note: Installing the GUI is optional. This may take extra time in installing and starting the virtual machine. The idea of using the ubuntu server edition is to ease the load on your computer and minimize load time.



Make sure you click on “generate new MAC address” when cloning.

(Repeat these steps for EACH of the 3 VMs: srvmaster, srvnode1, srvnode2)

1. Open VirtualBox Manager.
2. Click New.
3. Name: Enter the specific VM name (e.g., srvmaster).
4. ISO Image: Select your downloaded Ubuntu Server ISO.
5. (Optional but Recommended): Check "Skip Unattended Installation" to have full control during install.
6. Hardware:
   - ○ Base Memory: Allocate sufficient RAM (e.g., 2048 MB or more).
   - ○ Processors: Allocate 2 CPUs if possible.
7. Virtual Hard Disk:
   - ○ Create a new virtual hard disk.
   - ○ Set size (e.g., 25 GB or more). Use default VDI type and Dynamically allocated is fine.
8. Review Summary & Click Finish.
9. Configure Network Adapters (BEFORE first boot):
   - ○ Select the newly created VM in VirtualBox Manager.
   - ○ Click Settings -> Network.
   - ○ Adapter 1 Tab:
     - ■ Check Enable Network Adapter.
     - ■ Attached to: Select NAT.
   - ○ Adapter 2 Tab:
     - ■ Check Enable Network Adapter.
     - ■ Attached to: Select Internal Network.
     - ■ Name: Enter cluster-net (must be identical for all 3 VMs).

○ Click OK.

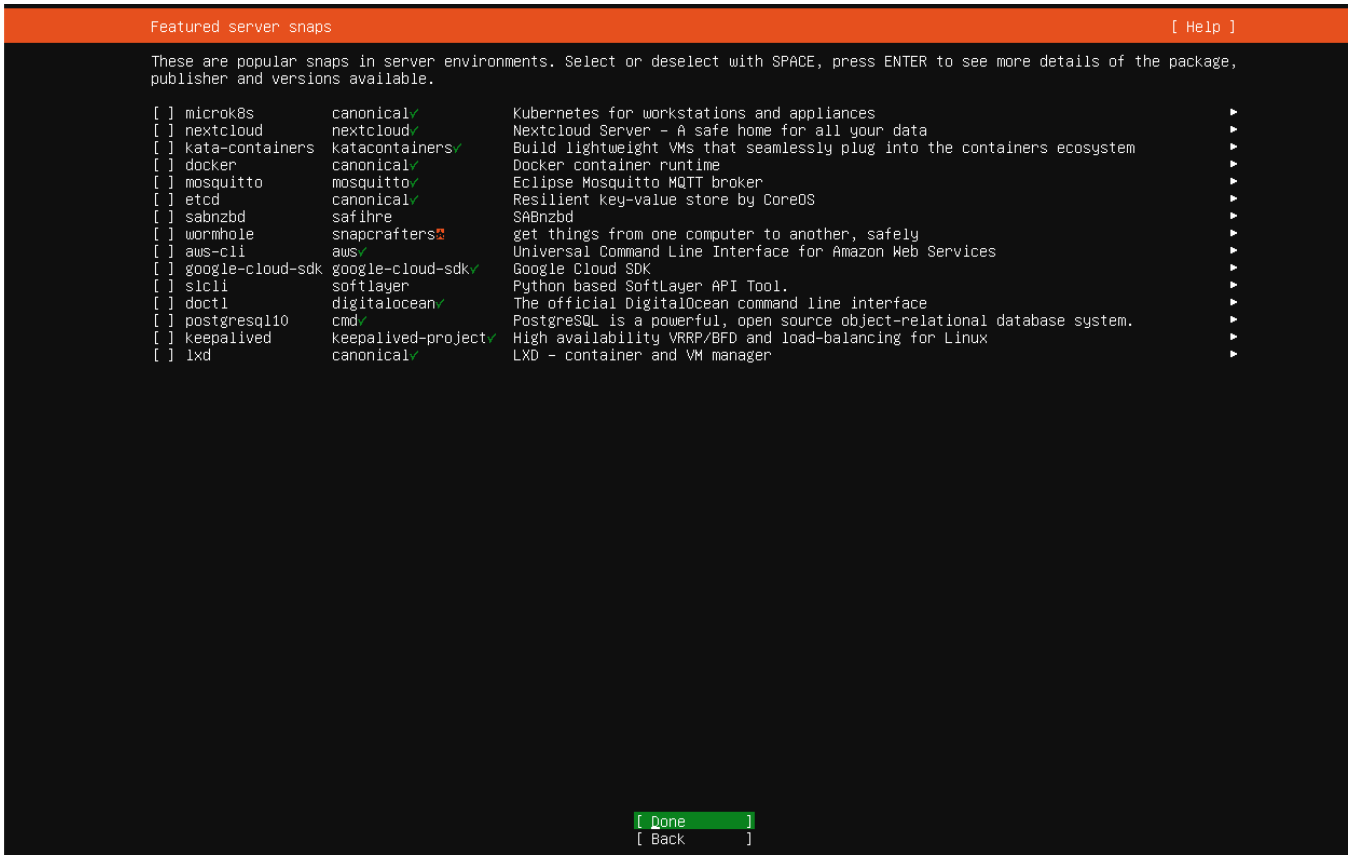
Phase 3: Ubuntu Server Installation

(Perform these steps on EACH of the 3 VMs, starting them one by one)
1. Start the VM. It should boot from the selected ISO.
2. Select "Try or Install Ubuntu Server".
3. Language Selection: Choose your language.
4. Keyboard Configuration: Select your keyboard layout.
5. Installation Type: Choose "Ubuntu Server" (not minimized, unless you prefer).
6. Network Connections:
○ The installer should show two network interfaces (e.g., enp0s3 and enp0s8).
○ One interface (enp0s3, connected to NAT) should automatically get an IP address via DHCP (like 10.0.2.15/24). This is correct and needed for the installation process.
○ The other interface (enp0s8, connected to Internal Network) will likely show no IP configuration. This is also correct for now.
○ Do not make changes here. Select Done.
7. Configure Proxy: Leave blank unless you use one. Select Done.
8. Configure Ubuntu Archive Mirror: Use the default mirror. Select Done.
9. Storage Configuration:
○ Select "Use an entire disk".
○ Select the virtual disk created earlier.
○ Ensure "Set up this disk as an LVM group" is unchecked for simplicity (unless you need LVM).
○ Select Done. Confirm the destructive action in the next prompt.
10. Profile Setup:
○ Your name: Enter your name.
○ Your server's name: Enter the correct hostname (srvmaster, srvnode1, srvnode2).
○ Pick a username: Enter your chosen admin username (e.g., clusteradmin). Make sure this is the same one for all servers, it is a pain to edit later.
○ Choose a password: Enter a strong password.usually yes, for this, enter a short and easy password because you will have to type it in often.
○ Confirm your password: Re-enter the password.
○ Select Done.

11. SSH Setup:
○ IMPORTANT: Check the box for Install OpenSSH server.
○ Do not import SSH identities at this stage.

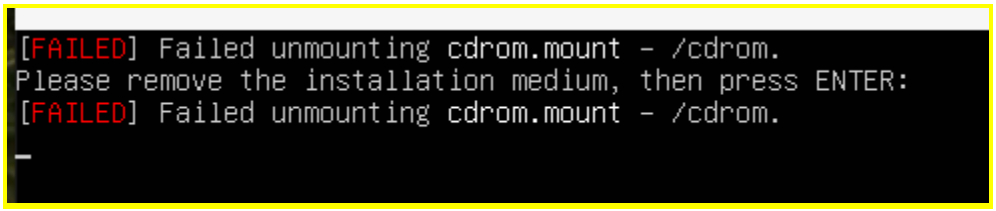
○ Select Done.

12. Server Snaps: Select any desired snaps (optional, can be added later). Select Done.



13. Installation: Wait for the installation process to complete.

14. Reboot: When installation finishes, select Reboot Now. VirtualBox may prompt you to "remove the installation medium" - press Enter if so. You might get an error that says you didn't remove the installation medium. Don't worry. Just press enter.



Phase 4: Post-Installation Configuration (Static IPs & Host Resolution)

(Perform these steps on EACH of the 3 VMs after logging in)

1. Log in using the clusteradmin username and password created during installation. Run `sudo apt update && sudo apt upgrade` (just get used to doing this every time you login.)

2. Identify Network Interface Names:

○ Run: ip a

○ Note the names of your two interfaces. Typically enp0s3 (for NAT, has 10.0.2.x IP) and enp0s8 (for Internal Network, currently unconfigured or APIPA). Use your actual interface names in the next step.

```
clusteradmin@srvnode1:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:e5:a4:61 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s8
       valid_lft 86213sec preferred_lft 86213sec
    inet6 fd00::a00:27ff:fee5:a461/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 86214sec preferred_lft 14214sec
    inet6 fe80::a00:27ff:fee5:a461/64 scope link
       valid_lft forever preferred_lft forever
clusteradmin@srvnode1:~$
```

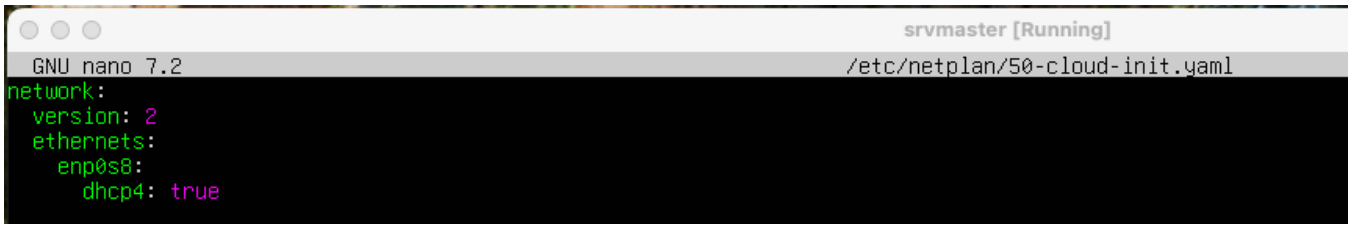
3. Configure Netplan for Static Internal IP:

○ Find the netplan config file: ls /etc/netpilan/ (often 00-installer-config.yaml).

○ Edit the file: sudo nano /etc/netplan/50-cloud-init.yaml (replace filename if different).

```
clusteradmin@srvnode1:~$ ls /etc/netplan/
50-cloud-init.yaml
clusteradmin@srvnode1:~$ sudo nano /etc/netplan/50-cloud-init.yaml _
```

When you press enter, your screen will look something like this:

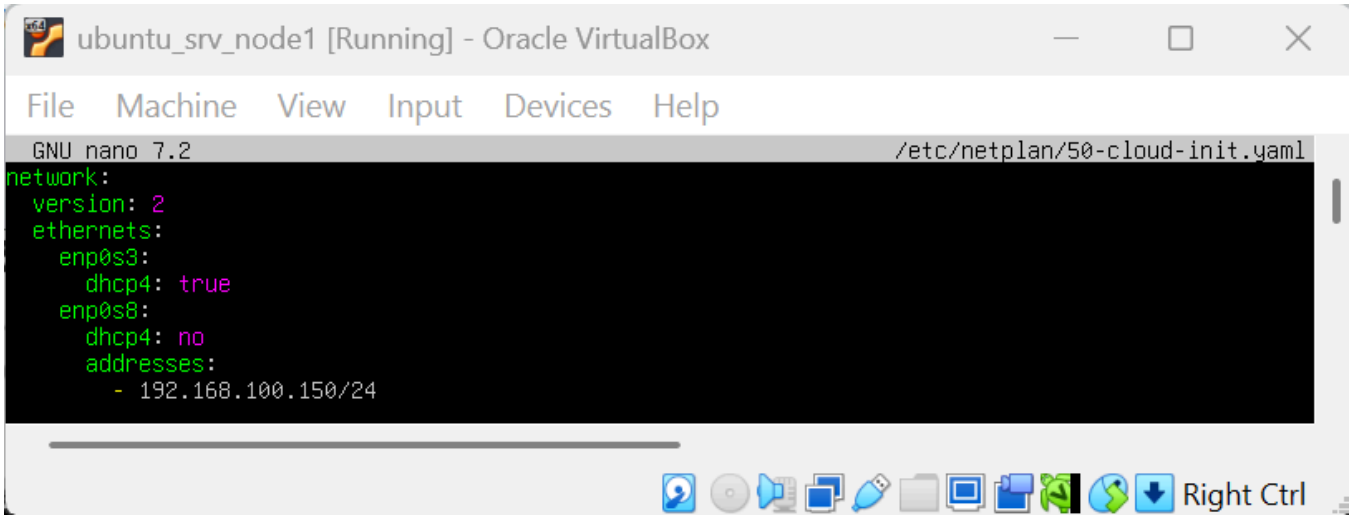


○ Make the file content match the template below, adjusting the interface names (enp0s3, enp0s8) and the static IP address for each specific node. This file is sensitive to spacing, I recommend you use tab.

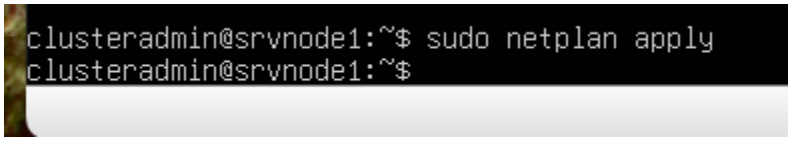
Template (Modify IP and Interface Names per node):

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3: # <<< YOUR NAT INTERFACE NAME (e.g., enp0s3)
      dhcp4: true
    enp0s8: # <<< YOUR INTERNAL NETWORK INTERFACE NAME (e.g., enp0s8)
      dhcp4: no
      addresses:
        - 192.168.1.100/24 # <<< USE .100 for master, .150 for work-server-1, .200 for work-server-2
      # Note: No gateway is configured for the internal interface
  version: 2
```

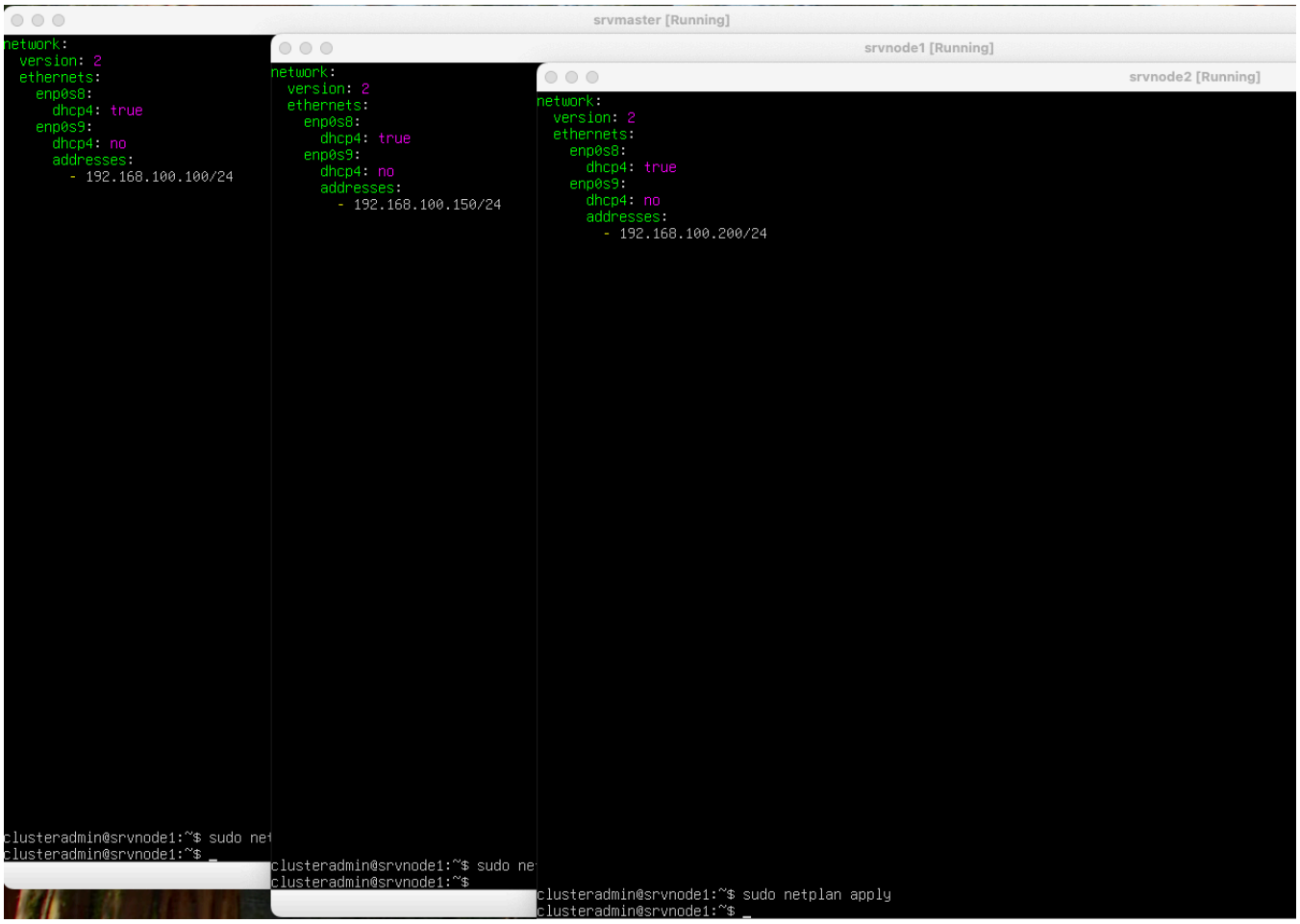
Below you can see what it will look like on your machine after you edit:



4. Apply Netplan Changes:
○ Save the file (Ctrl+O, Enter in nano) and Exit (Ctrl+X).
○ Apply: `sudo netplan apply` (Watch for errors). (No news is good news, see below.)



See all 3 machines configured below. Note that my enp0s looks different than yours, to match the results I got when I ran ip a:



5. Verify IP Configuration:

○ Run ip a again. Check that your internal interface (enp0s8) now has the correct static IP (192.168.1.100, .150, or .200). The NAT interface (enp0s3) should still have its DHCP address (10.0.2.x).

```
clusteradmin@srvnode1:~$ sudo netplan apply
clusteradmin@srvnode1:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ae:30:d3 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s8
       valid_lft 86080sec preferred_lft 86080sec
    inet6 fd00::a00:27ff:feae:30d3/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 86264sec preferred_lft 14264sec
    inet6 fe80::a00:27ff:feae:30d3/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:8b:39:9b brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.100/24 brd 192.168.100.255 scope global enp0s9
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe8b:399b/64 scope link
       valid_lft forever preferred_lft forever
clusteradmin@srvnode1:~$ _
```

6. Configure Hostname Resolution:

○ Edit the hosts file: sudo nano /etc/hosts

○ Add these three lines (below the existing 127.0.0.1 localhost line):

192.168.1.100 srvmaster.local

192.168.1.150 srvnode1.local

192.168.1.200 srvnode2.local

(Use your chosen internal IP range if you changed it due to conflicts).

○ Save and exit. Ensure this /etc/hosts update is done on ALL THREE nodes.

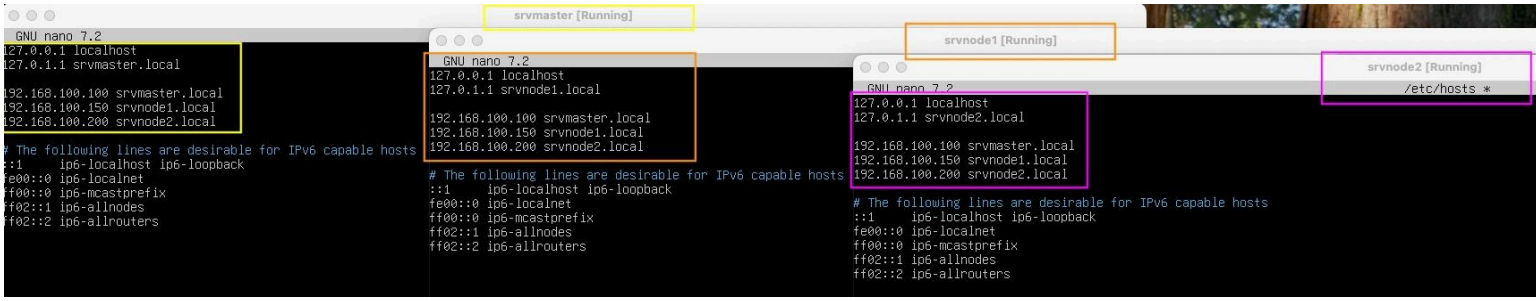
Below is what your hosts file should look like after you've edited.

Remember cat is used to read the file. And I used 192.168.100.x so stick to whatever you used.

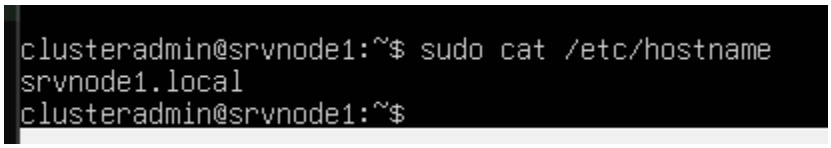
```
clusteradmin@srvnode1:~$ sudo cat /etc/hosts
127.0.0.1 localhost
192.168.100.100 srvmaster.local
192.168.100.150 srvnode1.local
192.168.100.200 srvnode2.local
127.0.1.1 srvnode1.local

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
clusteradmin@srvnode1:~$
```

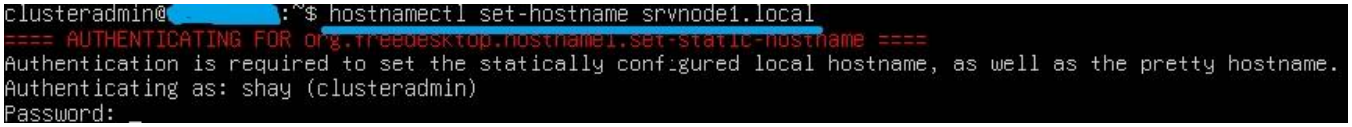
Final product on all 3 machines:



Check your hostname file to make sure it matches the hostname in hosts file.
to edit hostname:
sudo nano /etc/hosts



Edit your hostname by using the `hostnamectl set-hostname srvnode1.local`← last bit can be different depending on which node you're working on, for svrmaster this command would be "`hostnamectl set-hostname srvmaster.local`"
`Remember to reboot afterwards.`
See below:



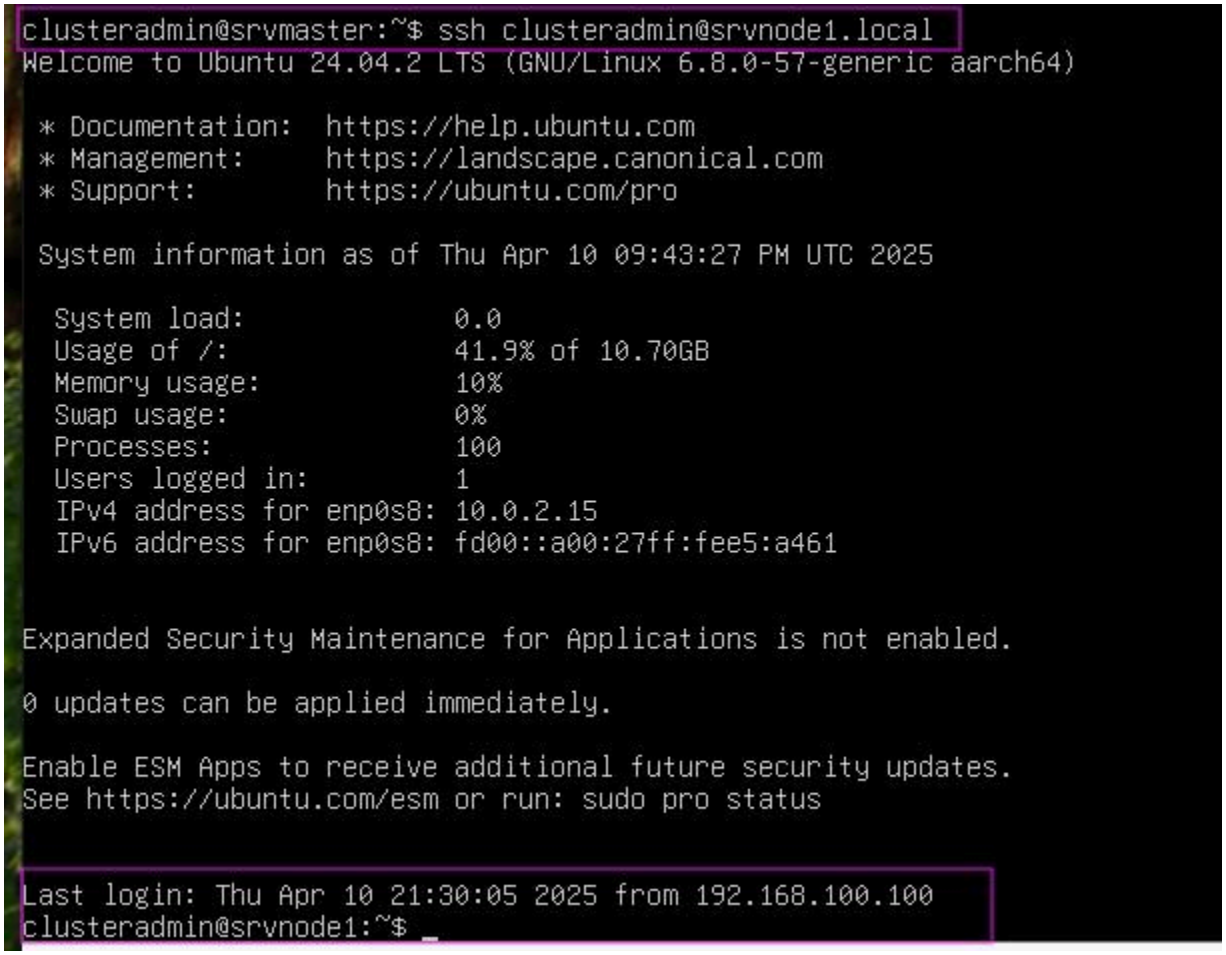
Once you reboot you will see the

Phase 5: SSH Configuration (Passwordless Access from Master)

(Perform these steps ONLY on srvmaster)
1. Log in to srvmaster as clusteradmin.
2. Generate SSH Key Pair:
○ Run: `ssh-keygen -t rsa -b 4096`
○ Press Enter to accept the default file location (~/.ssh/id_rsa).
○ Press Enter twice to create the key with no passphrase.
3. Copy Public Key to Nodes:
○ Copy to srvnode1:
Bash
`ssh-copy-id clusteradmin@srvnode1.local`
■ Type yes if prompted to trust the host key.
■ Enter the clusteradmin password for srvnode1 when prompted.
○ Copy to srvnode2:
Bash
`ssh-copy-id clusteradmin@srvnode2.local`

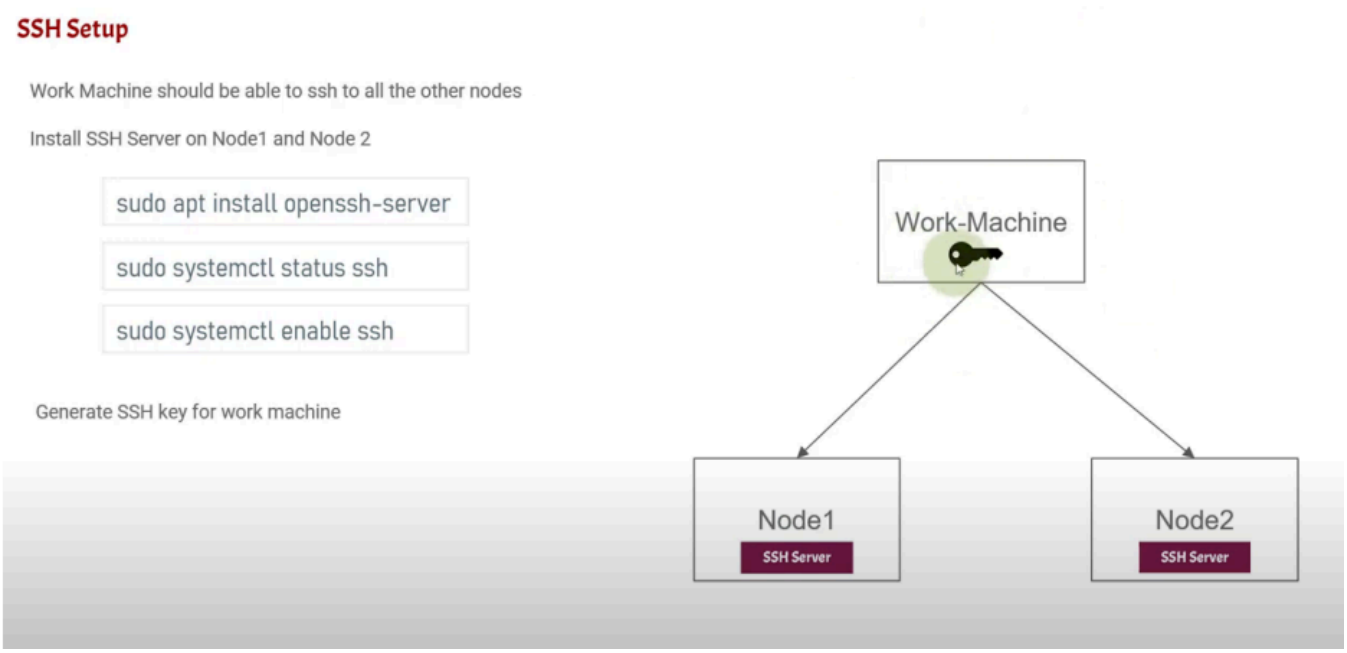
■ Type yes if prompted.
■ Enter the clusteradmin password for srvnode2 when prompted.

Image of logging in to srvnode1 from srvmaster via ssh:



If your username is the same for all 3 machines, you can ssh with ssh “hostname” like so:
`ssh srvnode1.local`

Extra info, only if you haven’t already installed ssh during the Ubuntu Server install.



Phase 6: Verification & Testing

1. IP Configuration Check: On each node, run ip a to confirm interfaces (enp0s3 or similar with 10.0.2.x IP, enp0s8 or similar with correct static 192.168.1.x IP).
2. Internet Access Check: On each node: ping -c 3 google.com (Should succeed).
3. Internal Network Ping (IP):
○ From srvmaster: ping -c 3 192.168.1.150 and ping -c 3 192.168.1.200 (Should succeed).
4. Internal Network Ping (Hostname):
○ From srvmaster: ping -c 3 srvnode1 and ping -c 3 srvnode2 (Should succeed).
○ From srvnode1: ping -c 3 srvmaster (Should succeed).

```
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 5.953/8.979/13.496/3.254 ms
clusteradmin@srvnode2:~$ ping -c 3 192.168.100.100
PING 192.168.100.100 (192.168.100.100) 56(84) bytes of data.
64 bytes from 192.168.100.100: icmp_seq=1 ttl=64 time=0.403 ms
64 bytes from 192.168.100.100: icmp_seq=2 ttl=64 time=0.393 ms
64 bytes from 192.168.100.100: icmp_seq=3 ttl=64 time=0.460 ms

--- 192.168.100.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2061ms
rtt min/avg/max/mdev = 0.393/0.418/0.460/0.029 ms
clusteradmin@srvnode2:~$ ping -c 3 192.168.100.150
PING 192.168.100.150 (192.168.100.150) 56(84) bytes of data.
64 bytes from 192.168.100.150: icmp_seq=1 ttl=64 time=0.445 ms
64 bytes from 192.168.100.150: icmp_seq=2 ttl=64 time=0.442 ms
64 bytes from 192.168.100.150: icmp_seq=3 ttl=64 time=0.380 ms

--- 192.168.100.150 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2059ms
rtt min/avg/max/mdev = 0.380/0.422/0.445/0.029 ms
clusteradmin@srvnode2:~$ _
```

5. Passwordless SSH Check:
○ From srvmaster:
Bash
ssh clusteradmin@srvnode1.local 'hostname'
ssh clusteradmin@srvnode2.local 'hostname'
Note again: If your username is the same for all 3 machines, you can leave out the username “clusteradmin”

```
clusteradmin@srvmaster:~$ ssh srvnode1.local "hostname"
srvnode1.local
```

■ Each command should immediately return the hostname of the respective work servers (srvnode1 or srvnode2) without asking for a password.
6. (Optional) Basic Security: Consider enabling the UFW firewall on each node (it’s a long command): `sudo ufw enable`, then `sudo ufw allow from 192.168.1.0/24 to any port 22 proto tcp` (to ensure SSH still works over the internal network). Keep systems updated: `sudo apt update && sudo apt upgrade -y`

Conclusion:
If all verification steps are successful, you now have a working 3-node Ubuntu cluster in VirtualBox. srvmaster can securely control srvnode1 and srvnode2 via passwordless SSH over the dedicated 192.168.1.x internal network, and all nodes retain internet access via their separate NAT interface. You are ready to deploy your

Challenges and Solutions

1. SSH via Hostname Failure

Issue:
SSH access using the hostname was unsuccessful. Potential causes include:

- The hostname not being updated.
- A mismatch between the hostname and entries in the `/etc/hosts` file.
- An incorrect command or hostname used.

Resolution:

- Open each VM and verify the hostname using `hostnamectl`.
- Update the hostname with: hostnamectl set-hostname <new-hostname>
- Confirm consistency across `/etc/hosts` and hostname settings.

2. SSH via IP Address Failure

Issue:
SSH access using the IP address failed. Likely causes include:

- An incorrect IP address configured in the Netplan file.
- An IP address conflict with another device on the network.

Resolution:

- Review IP configuration in the Netplan file on each VM.
- Verify current IP settings using `ip a` or `ip addr`.
- Check for conflicting IPs on the host machine using `ipconfig` (Windows) or `ifconfig/ip a` (Linux).

3. Netplan Configuration Errors

Issue:
Errors encountered while updating the Netplan configuration, typically due to:

- Improper indentation or spacing within the YAML file.

Resolution:

- Re-enter the Netplan configuration carefully.
- Validate syntax using: sudo netplan apply
- Review any error messages and ensure correct spacing (YAML is space-sensitive).

4. Virtual Machine Setup Challenges

Issue:
When cloning virtual machines, the failure to select **"Generate new MAC address"** for NAT adapters caused network conflicts.

Resolution:

- Power off the VM.
- Navigate to the VM settings, select **"Advanced"** > **"MAC Address"** > click **"Refresh"** to generate a new MAC address.
- Apply changes and restart the VM.

5. SSH Error 22 – Connection Refused

Issue:
Attempting to SSH results in **Error 22: Connection refused**. This often means the SSH **port 22 is blocked or closed**. Common causes include:

- UFW firewall rules on the destination VM blocking port 22.
- Cloud or virtualization platform firewall/security group blocking inbound SSH traffic on port 22.

Resolution:
Allow SSH traffic through the internal network by running the following command on the destination VM:

```
sudo ufw allow from 192.168.1.0/24 to any port 22 proto tcp
```

This enables port 22 for SSH access from other internal network machines, like srvmaster.

Ubuntu Cluster Setup Troubleshooting Checklist

Network & Virtual Machine Setup

_Confirm **VirtualBox** is installed with at least Ubuntu Server ISO 22.04/24.04 downloaded.

_When cloning VMs, did you **generate a new MAC address**?

_Are **network adapters set up properly**?

- Adapter 1: NAT
- Adapter 2: Internal Network (`cluster-net`)

_Did you **check for IP conflicts** with your host machine's IP? If `192.168.1.x` is in use, use a different private range.

During Ubuntu Installation

_Is **OpenSSH server** selected during installation?

_Did each VM get the correct **hostname** and **consistent admin username** (e.g., `clusteradmin`)?

Static IP Setup (Netplan)

_Did you correctly identify interfaces (`ip a`) before editing Netplan?

_Is Netplan configuration **YAML-indentation correct** (use spaces, not tabs)?

_Did you use the correct static IPs?

- srvmaster: `192.168.1.100`
- srvnode1: `192.168.1.150`
- srvnode2: `192.168.1.200`

_After `sudo netplan apply`, did it return without error?

_Did you verify IPs using `ip a` after applying Netplan?

Hostname Resolution

_Is the `/etc/hosts` file updated on **all nodes** with:

```
192.168.1.100 srvmaster.local
192.168.1.150 srvnode1.local
192.168.1.200 srvnode2.local
```

_Did you **reboot** after updating hostnames?

Passwordless SSH Setup

_On srvmaster, was the SSH key generated? (`ssh-keygen -t rsa -b 4096`)

_Were keys copied to nodes using `ssh-copy-id clusteradmin@srvnodeX.local`?

_Can you SSH into nodes without a password? (e.g., `ssh srvnode1.local`)

_Did you verify SSH over IP and hostname both work?

Testing & Verification

_Can each VM **ping google.com** (internet access via NAT)?

_Can srvmaster ping srvnode1 and srvnode2 by:

- IP (e.g., `ping 192.168.1.150`)
- Hostname (e.g., `ping srvnode1.local`)

_Did you verify passwordless SSH returns the hostname?