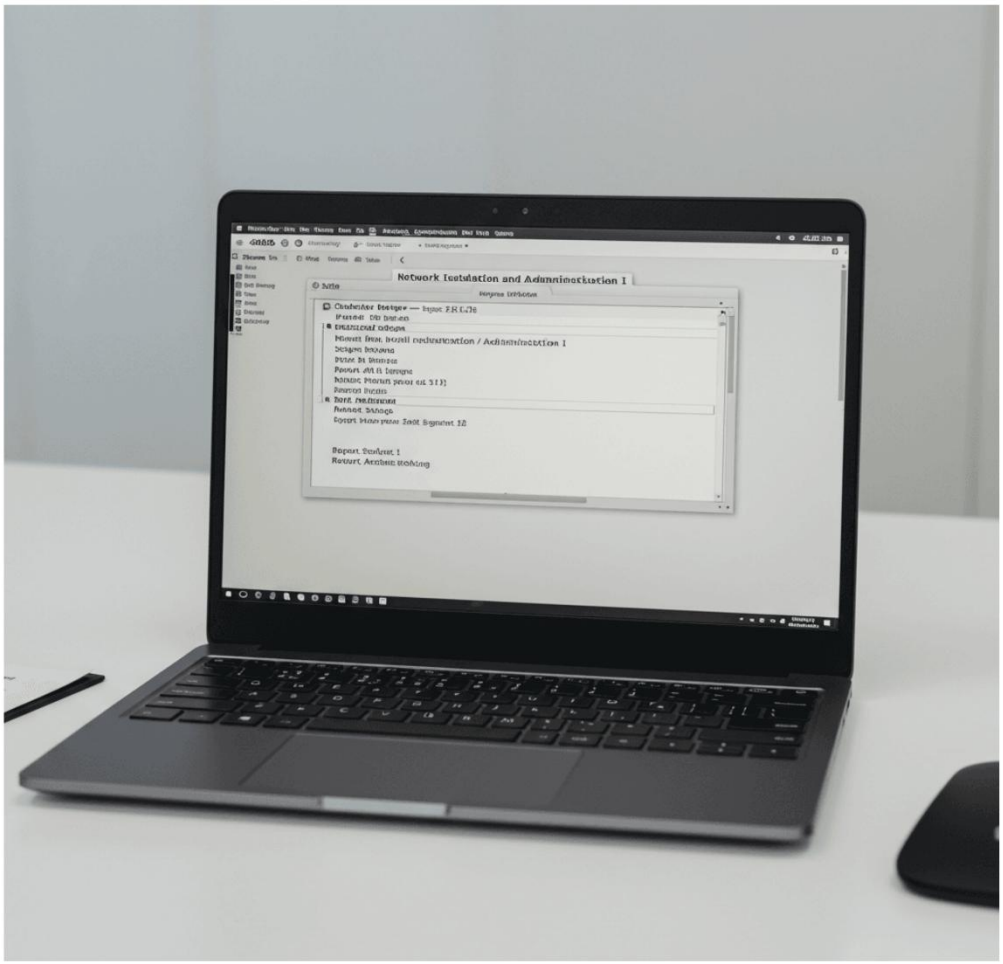
420-635-AB – NETWORK
INSTALLATION AND ADMINISTRATION I

ASSIGNMENT 1
Report : Linux Network Configuration, Partitioning, and Services Setup



PREPARED BY: GUILLERMO PADILLA KEYMOLE

John Abbott College | Network Administration Program

Table of Contents

Table of Figures (captions)

Introduction

This assignment is all about setting up and managing a small Linux-based network from scratch, just like a network admin would in the real world. The main focus is on configuring basic network settings, working with partitions and file systems, managing users and permissions, and setting up essential network services like SSH, NFS, and SAMBA.

These are key building blocks for any Linux system. Knowing how to configure network settings ensures machines can talk to each other. Partitioning and using logical volumes helps manage storage properly. Setting up users and permissions keeps things secure and organized. Finally, configuring services like SSH, NFS, and SAMBA allows for remote access and file sharing across different systems.

To get it all done, we'll be using virtual machines running AlmaLinux (for the server) and Ubuntu (for the client). Everything will be tested and configured within that environment using the terminal and system tools built into each distro.

This report will walk through each step with commands, screenshots, and notes. The goal is to get familiar with how everything works together in a basic client-server setup.

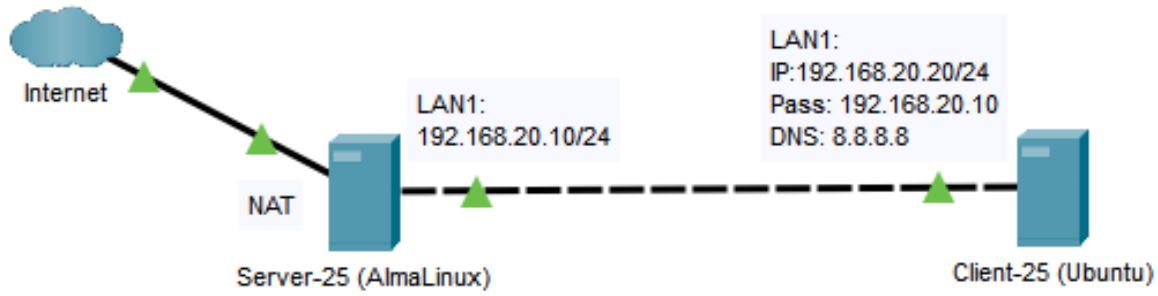


Figure 1 Liinux-based Network Topology

PART 1 – Server Configuration

1- Topology

Step 1: Download the Virtual Machines

I downloaded the two required virtual machine files from the link provided in the assignment instructions:

- **AlmaLinux-Assignment1.ova**
- **Ubuntu-Assignment1.ova**

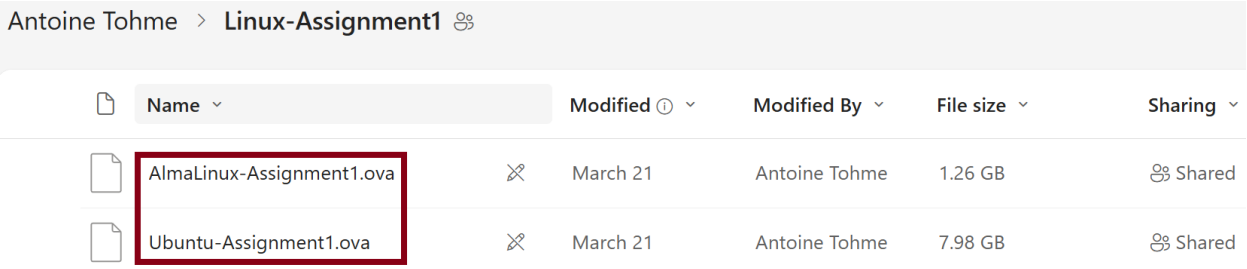


Figure 2 Virtual Machine Download Files

Step 2: Import AlmaLinux Server into VMware

1. I double-clicked on the AlmaLinux-Assignment1.ova file.

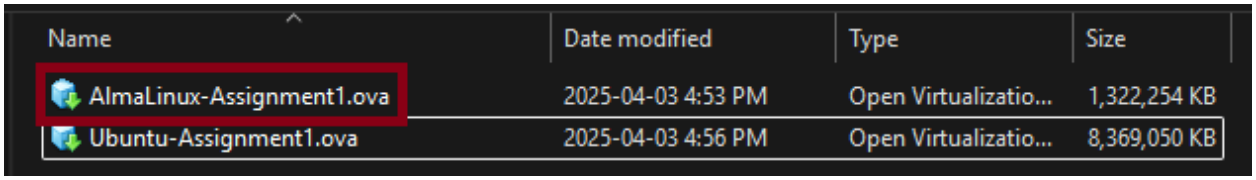


Figure 3 OVA Files

2. In the **VMware Import Virtual Machine** dialog box, I named the virtual machine **Server-7**, where **7** is my remote PC number.
3. I kept the default path and clicked **Import** to begin the import process.

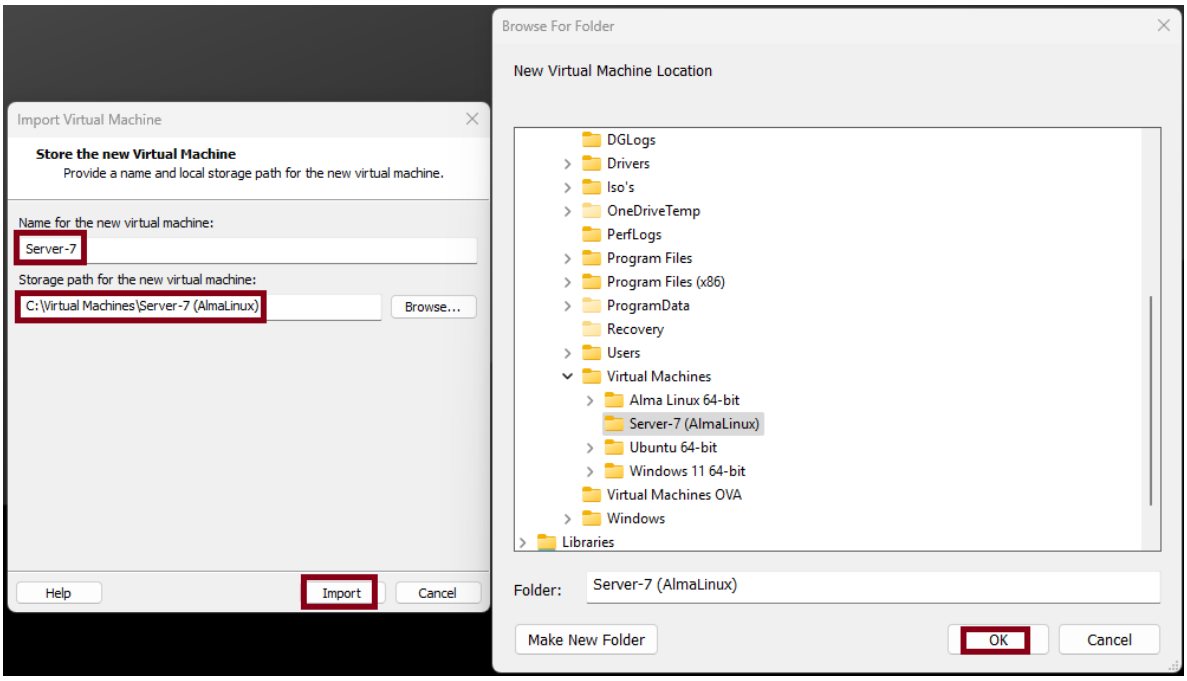


Figure 4 Importing AlmaLinux VM and naming it Server-7

Step 3: Import Ubuntu Client into VMware

1. I repeated the same steps with the Ubuntu-Assignment1.ova file.
2. I named the virtual machine **Client-7** (with 7 as my remote PC number).
3. Clicked **Import** to complete the process.

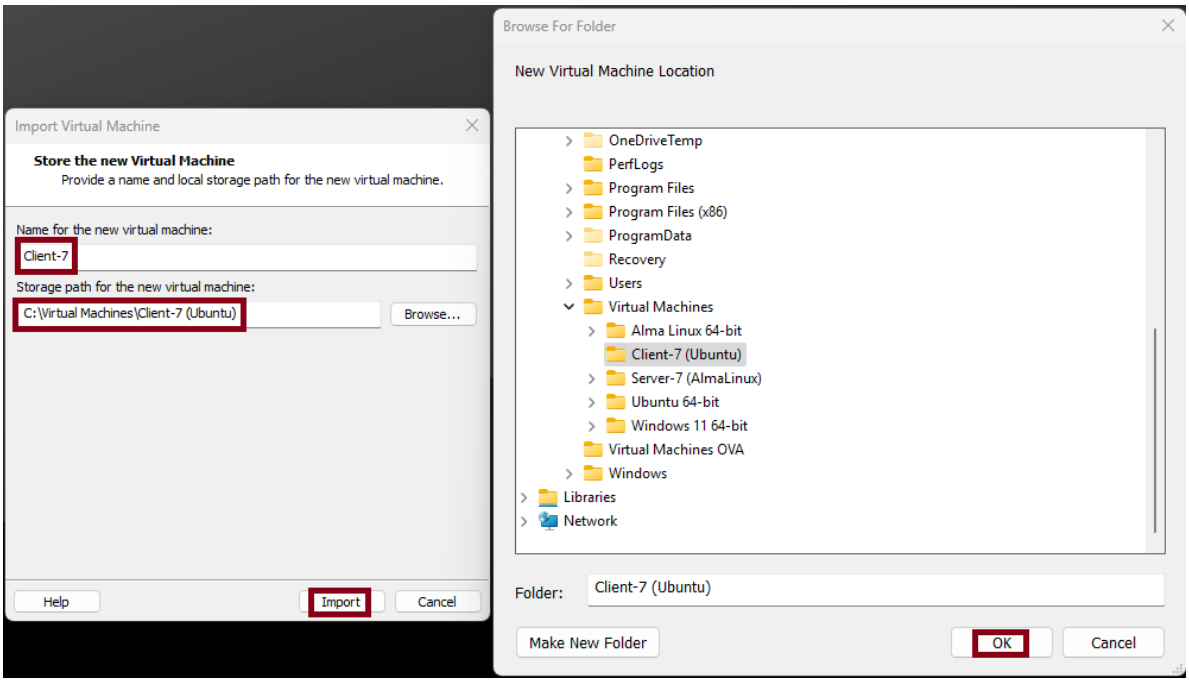


Figure 5 Importing Ubuntu VM and naming it Client-7

Step 4: Setting Up the LAN Connection

To match the network topology provided in the assignment, I configured a LAN segment between the two virtual machines using VMware Workstation. This allows the **Ubuntu client** and **AlmaLinux server** to communicate directly within an isolated local network. The AlmaLinux server also uses NAT for internet access.

LAN Segment in VMware Workstation

For Server-7 (AlmaLinux Server):

1. Selected **Server-7** in VMware Workstation.
2. Clicked on **"Edit virtual machine settings."**
3. Under the **Hardware** tab, clicked on **"Add."**
4. Chose **"Network Adapter"** as the hardware type and clicked **"Finish."**

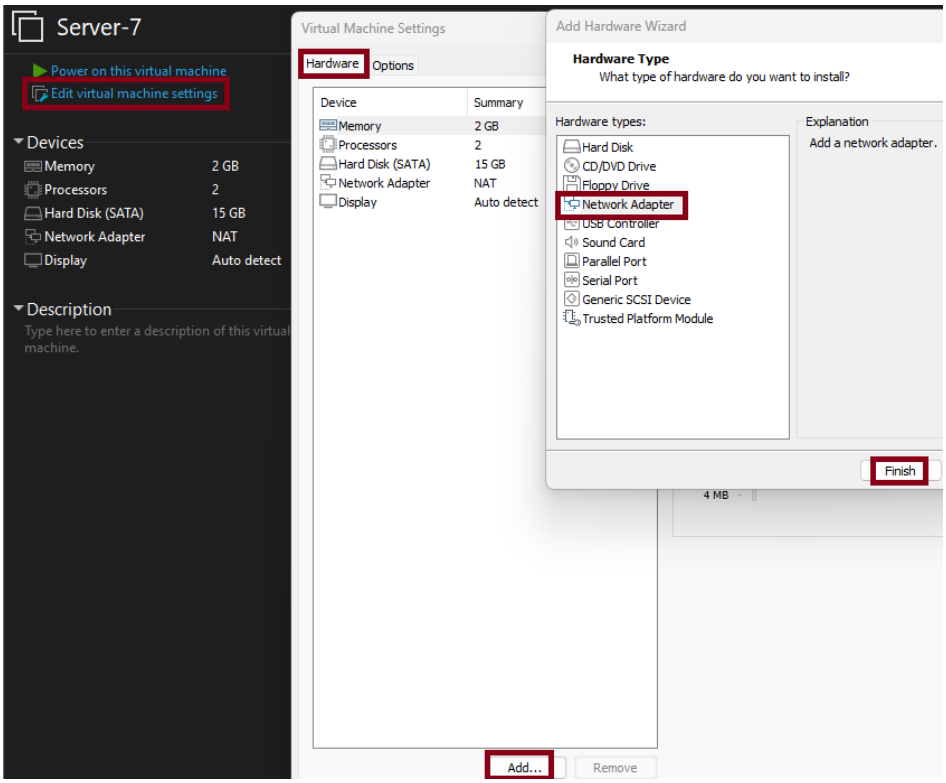


Figure 6 VM network adapter settings

5. Selected the newly added network adapter and clicked on **"LAN Segments."**
6. In the **Global LAN Segments** window, clicked on **"Add."**
7. Entered the name **"LAN1"** for the new segment and clicked **OK.**

8. Used the **dropdown menu** to select the newly created **"LAN1"** segment, then clicked **OK** to apply the settings.

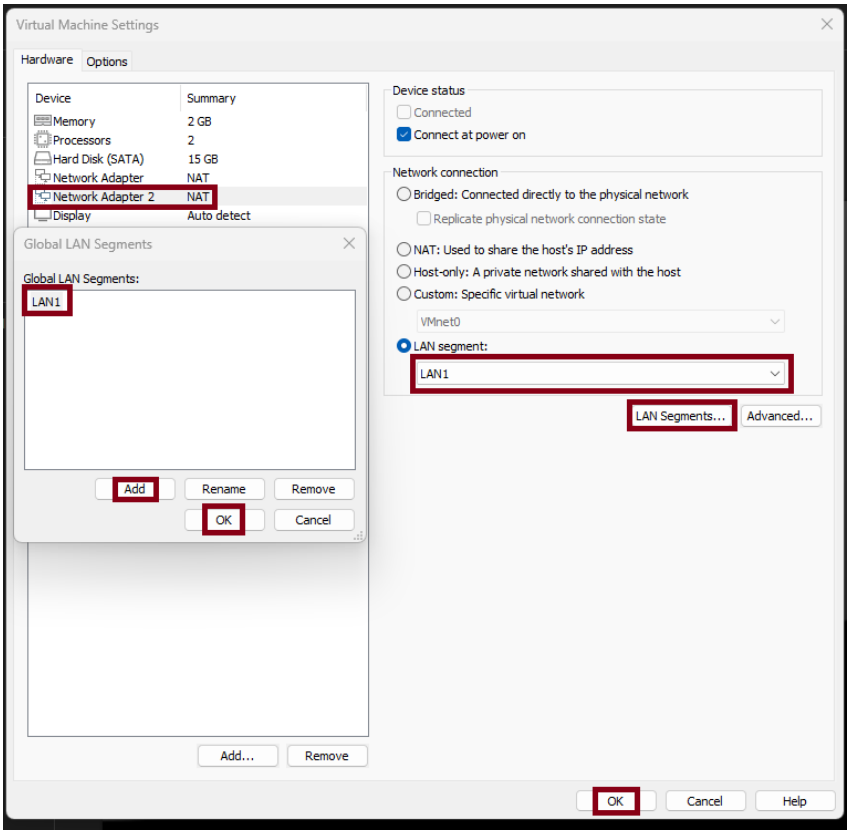


Figure 7 VM LAN Segment Creation

For Client-7 (Ubuntu):

1. Selected **Client-7** in VMware Workstation.
2. Clicked on **"Edit virtual machine settings."**
3. Under the **Hardware** tab, selected the existing **Network Adapter.**
4. Chose **"LAN Segment"** as the network connection type.
5. From the dropdown menu, selected **"LAN1."**
6. Clicked **OK** to save and apply the settings.

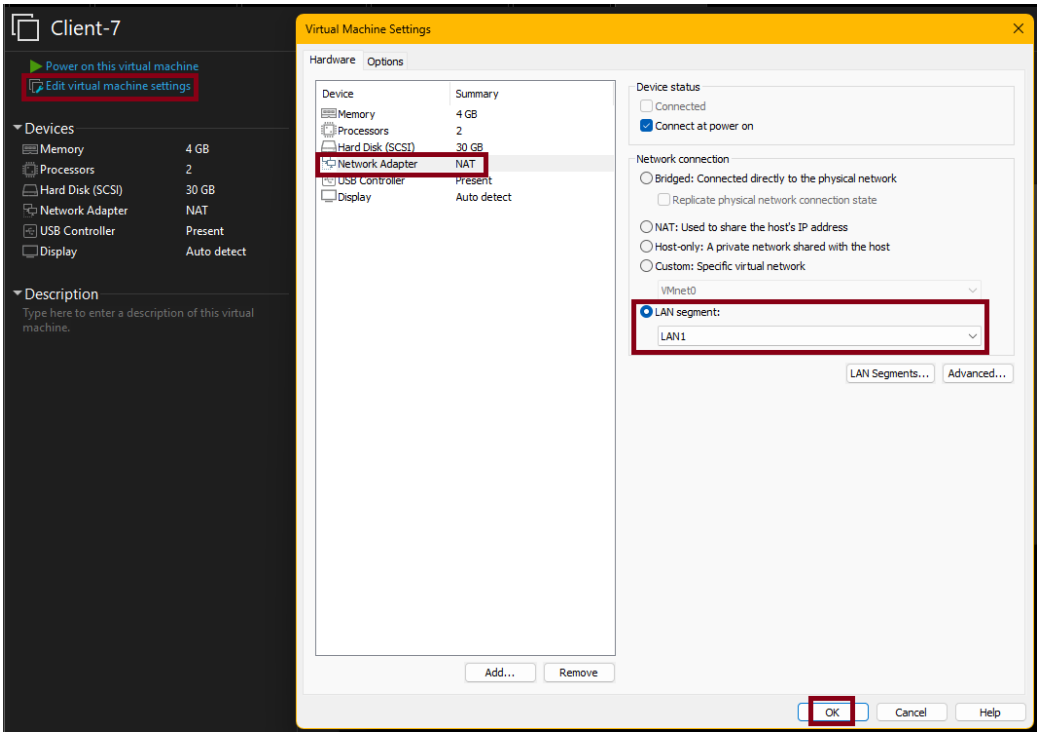


Figure 8 VM LAN Segment Selection

Step 5: Network Configuration

To start configuring Server-7 (AlmaLinux VM), I first identified the DHCP-assigned IP address directly from the VM console. I then used that temporary IP to connect via SSH from my host machine, making it easier to work in the CLI. Once connected, I changed the hostname and applied a static IP using nmcli.

Identify Temporary DHCP IP (Server-7)

1. After powering on Server-7 (AlmaLinux VM), I logged in using the credentials provided in the assignment.
2. I ran the following command to check the current network status and identify the DHCP-assigned IP address: **“nmcli”.**
3. The output showed that the system was connected via interface **ens160**, and the IP address assigned by DHCP was: **“192.168.5.134/24”.**
4. To confirm the connection name for future nmcli use, I also ran: **“nmcli connection show”.**

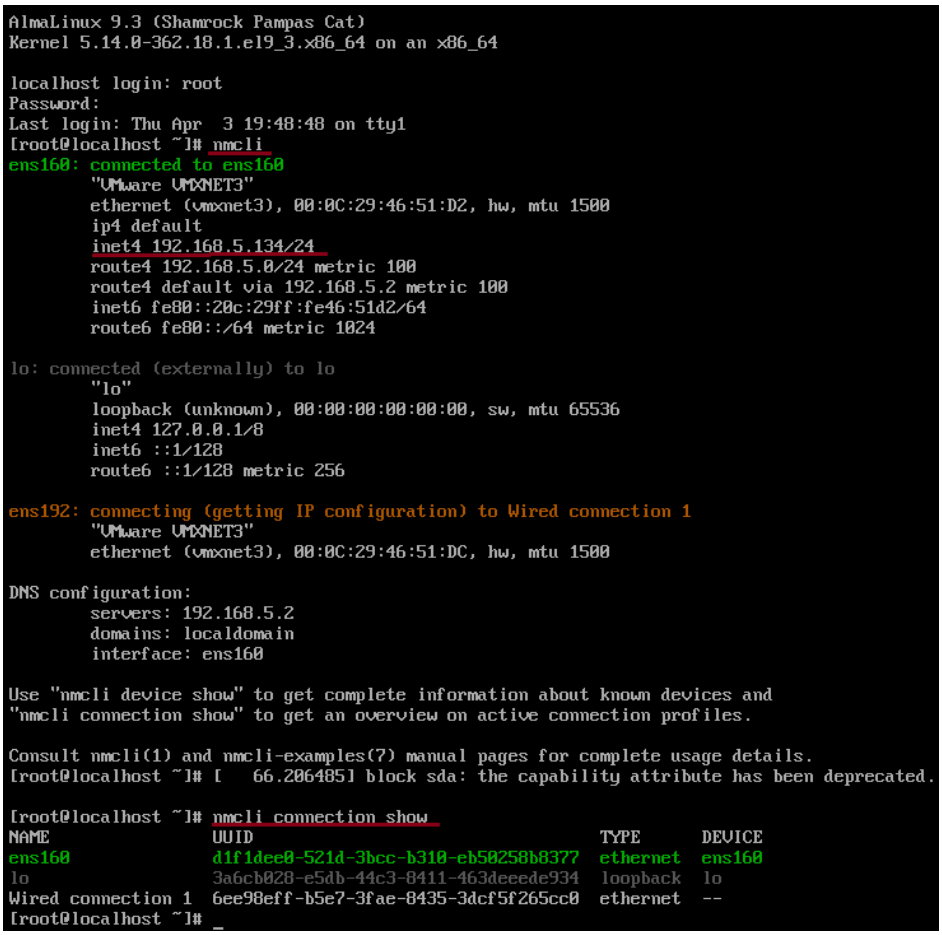


Figure 9 DHCP IP address and active network connection confirmed via "nmcli"

Connect to Server-7 via SSH from Host PC

With the DHCP IP address identified on the server (192.168.5.134), I used **SSH** from my host PC to remotely access and configure the Server-7. This approach is more convenient, especially since the server has no GUI.

SSH connection steps:

1. On my host PC (PowerShell on Windows), I opened a terminal.
2. Entered the following command to connect via SSH: **"ssh root@192.168.5.134"**
3. When prompted, entered the password: **"alma"**
4. Once connected, I ran the following command to verify the hostname: **"hostname"**

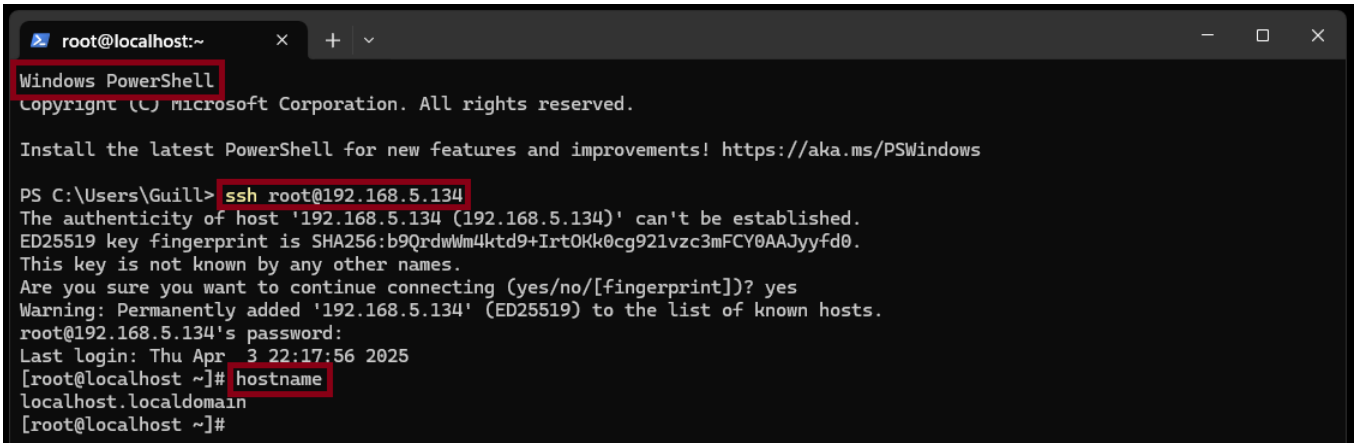


Figure 10 SSH login from host PC to Server-7 using DHCP IP

Change Hostname Server-7

While connected to Server-7 through SSH, I changed the hostname from the default localhost.localdomain to Server-7.

- Commands Used: **"nmcli general hostname Server-7"**
- To confirm the change: **"hostname"**

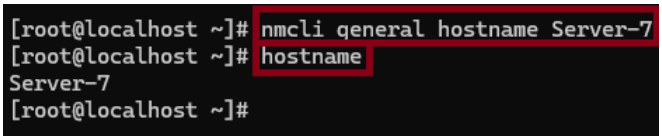


Figure 11 Hostname changed to Server-7 using nmcli

Verify Hostname Change Persists Over New SSH Session

- To confirm that the hostname change was successful and persistent, I logged out of the current SSH session: "exit"
- Then I reconnected using the same command: "ssh root@192.168.5.134"
- The command prompt now displayed root@Server-7, confirming that the new hostname was correctly applied.

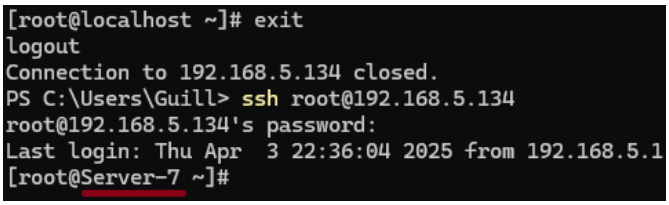


Figure 12 Reconnecting via SSH showing updated hostname in prompt

Configure Static IP on Server-7

Instead of modifying the existing interface ens160, I decided to configure the second adapter ens192, which corresponds to the internal LAN segment in the assignment topology.

Step 1: Check Existing Interface Status

I first checked the current status of all network devices and confirmed that **ens192** was available but disconnected.

```
[root@Server-7 ~]# nmcli general status
STATE      CONNECTIVITY  WIFI-HW  WIFI     WWAN-HW  WWAN
connected  full          missing  enabled  missing  enabled
[root@Server-7 ~]# nmcli device status
DEVICE  TYPE      STATE                   CONNECTION
ens160  ethernet  connected               ens160
lo      loopback  connected (externally)  lo
ens192  ethernet  disconnected            --
[root@Server-7 ~]# nmcli
ens160: connected to ens160
        "VMware VMXNET3"
        ethernet (vmxnet3), 00:0C:29:46:51:D2, hw, mtu 1500
        ip4 default
        inet4 192.168.5.134/24
        route4 default via 192.168.5.2 metric 100
        route4 192.168.5.0/24 metric 100
        inet6 fe80::20c:29ff:fe46:51d2/64
        route6 fe80::/64 metric 1024

lo: connected (externally) to lo
        "lo"
        loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
        inet4 127.0.0.1/8
        inet6 ::1/128
        route6 ::1/128 metric 256

ens192: disconnected
        "VMware VMXNET3"
        1 connection available
        ethernet (vmxnet3), 00:0C:29:46:51:DC, hw, mtu 1500

DNS configuration:
        servers: 192.168.5.2
        domains: localdomain
        interface: ens160
```

Figure 13 Interface Status ens160 & ens192

Step 2: Add New Static Connection on ens192

To match the assignment requirements, I created a new Ethernet connection named **LAN1** using **nmcli** and assigned the static IP **192.168.20.10/24.**

```
[root@Server-7 ~]# nmcli connection add type ethernet con-name LAN1 ifname ens192 ipv4.method manual ipv4.addresses 192.168.20.10/24
Connection 'LAN1' (0dabdfa7-ebdb-4202-8e28-1268bd8b2cef) successfully added.
[root@Server-7 ~]#
```

Figure 14 Static IP connection 'LAN1' added on ens192

Client-7 Configuration – Ubuntu

Change Hostname

After logging into the Ubuntu VM, I changed the hostname to follow the format required by the assignment (Client-X, where X is the remote PC number).

```
atohme@client25:~$ nmcli general hostname Client-7
atohme@client25:~$ hostname
Client-7
atohme@client25:~$
```

Figure 15 Hostname changed to Client-7

Check Network Interface Status

After changing the hostname, I verified that the **NetworkManager** service was running, then checked the current network configuration using nmcli.

```
atohme@Client-7:~$ sudo systemctl status NetworkManager
● NetworkManager.service - Network Manager
     Loaded: loaded (/lib/systemd/system/NetworkManager.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2025-04-04 09:13:46 EDT; 8min ago
       Docs: man:NetworkManager(8)
   Main PID: 569 (NetworkManager)
      Tasks: 3 (limit: 4551)
     Memory: 10.7M
        CPU: 255ms
     CGroup: /system.slice/NetworkManager.service
             └─569 /usr/sbin/NetworkManager --no-daemon

Apr 04 09:18:10 Client-7 NetworkManager[569]: <info>  [1743772690.1563] policy: set-hostname: current hostname was changed outside NetworkManager: 'Client-7'
Apr 04 09:21:47 Client-7 NetworkManager[569]: <info>  [1743772907.1178] policy: auto-activating connection 'Wired connection 1' (42826c07-c567-3970-b510-d6840
Apr 04 09:21:47 Client-7 NetworkManager[569]: <info>  [1743772907.1191] device (ens33): Activation: starting connection 'Wired connection 1' (42826c07-c567-39
Apr 04 09:21:47 Client-7 NetworkManager[569]: <info>  [1743772907.1192] device (ens33): state change: disconnected -> prepare (reason 'none', sys-iface-state:
Apr 04 09:21:47 Client-7 NetworkManager[569]: <info>  [1743772907.1194] manager: NetworkManager state is now CONNECTING
Apr 04 09:21:47 Client-7 NetworkManager[569]: <info>  [1743772907.1195] device (ens33): state change: prepare -> config (reason 'none', sys-iface-state: 'mana
Apr 04 09:21:47 Client-7 NetworkManager[569]: <info>  [1743772907.1240] device (ens33): state change: config -> ip-config (reason 'none', sys-iface-state: 'ma
Apr 04 09:21:47 Client-7 NetworkManager[569]: <info>  [1743772907.1320] dhcp4 (ens33): activation: beginning transaction (timeout in 45 seconds)
Apr 04 09:21:47 Client-7 NetworkManager[569]: <info>  [1743772907.1368] policy: set-hostname: current hostname was changed outside NetworkManager: 'Client-7'
Apr 04 09:21:49 Client-7 NetworkManager[569]: <info>  [1743772909.1027] policy: set-hostname: current hostname was changed outside NetworkManager: 'Client-7'
lines 1-21/21 (END)
atohme@Client-7:~$ nmcli general status
STATE       CONNECTIVITY  WIFI-HW  WIFI     WWAN-HW  WWAN
connecting  limited       enabled  enabled  enabled  enabled
atohme@Client-7:~$ nmcli device status
DEVICE  TYPE      STATE                                  CONNECTION
ens33   ethernet  connecting (getting IP configuration)  Wired connection 1
lo      loopback  unmanaged                              --
atohme@Client-7:~$ nmcli connection show
NAME                UUID                                  TYPE      DEVICE
Wired connection 1  42826c07-c567-3970-b510-d6840e285106  ethernet  ens33
atohme@Client-7:~$ nmcli
ens33: connecting (getting IP configuration) to Wired connection 1
        "Intel 82545EM"
        ethernet (e1000), 00:0C:29:E5:8B:FE, hw, mtu 1500

lo: unmanaged
        "lo"
        loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536

Use "nmcli device show" to get complete information about known devices and
"nmcli connection show" to get an overview on active connection profiles.

Consult nmcli(1) and nmcli-examples(7) manual pages for complete usage details.
atohme@Client-7:~$
```

Figure 16 NetworkManager status and DHCP-based interface (ens33) on Client-7

Remove Default DHCP Connection

To avoid IP conflicts and ensure the system uses the new static IP configuration, I removed the existing DHCP-based connection named Wired connection 1.

```
atohme@Client-7:~$ nmcli connection show
NAME                UUID                                  TYPE      DEVICE
Wired connection 1  42826c07-c567-3970-b510-d6840e285106  ethernet  --
atohme@Client-7:~$ nmcli connection delete Wired\ connection\ 1
Connection 'Wired connection 1' (42826c07-c567-3970-b510-d6840e285106) successfully deleted.
atohme@Client-7:~$
```

Figure 17 Deleted default DHCP connection on Client-7

Create New Static Connection 'LAN1' on Client-7

After removing the default DHCP configuration, I created a new Ethernet connection named LAN1 on interface ens33 to match the assignment's topology

```
atohme@Client-7:~$ nmcli connection add type ethernet ifname ens33 con-name LAN1 ipv4.method manual ipv4.addresses 192.168.20.20/2
4 ipv4.gateway 192.168.20.10 ipv4.dns 8.8.8.8
Connection 'LAN1' (0790e97d-437c-447c-b323-417773833d86) successfully added.
atohme@Client-7:~$ nmcli connection down LAN1 ; nmcli connection up LAN1
Connection 'LAN1' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/11)
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/12)
atohme@Client-7:~$
```

Figure 18 Static connection 'LAN1' created and activated on Client-7

```
atohme@Client-7:~$ nmcli connection show LAN1 | grep IP
GENERAL.IP-IFACE:                       ens33
IP4.ADDRESS[1]:                         192.168.20.20/24
IP4.GATEWAY:                            192.168.20.10
IP4.ROUTE[1]:                           dst = 192.168.20.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[2]:                           dst = 0.0.0.0/0, nh = 192.168.20.10, mt = 20100
IP4.ROUTE[3]:                           dst = 169.254.0.0/16, nh = 0.0.0.0, mt = 1000
IP4.DNS[1]:                             8.8.8.8
IP6.ADDRESS[1]:                         fe80::451d:6e3f:5ff1:47b8/64
IP6.GATEWAY:                            --
IP6.ROUTE[1]:                           dst = fe80::/64, nh = ::, mt = 1024
atohme@Client-7:~$
```

Figure 19 Validating Static IP, Subnet, Gateway, and DNS on LAN1

Verify Connectivity Between Client-7 and Server-7

To confirm that the network configuration was successful and that both machines are properly connected over the LAN segment, I performed a **two-way ping test**, followed by an additional test to verify NAT interface accessibility.

From Client-7 to Server-7

On Client-7, I ran the following command to test connectivity to Server-7's static IP address

```
atohme@Client-7:~$ ping -c 4 192.168.20.10
PING 192.168.20.10 (192.168.20.10) 56(84) bytes of data.
64 bytes from 192.168.20.10: icmp_seq=1 ttl=64 time=0.682 ms
64 bytes from 192.168.20.10: icmp_seq=2 ttl=64 time=0.378 ms
64 bytes from 192.168.20.10: icmp_seq=3 ttl=64 time=0.370 ms
64 bytes from 192.168.20.10: icmp_seq=4 ttl=64 time=0.580 ms

--- 192.168.20.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3068ms
rtt min/avg/max/mdev = 0.370/0.502/0.682/0.133 ms
atohme@Client-7:~$
```

Figure 20 Successful ping from Client-7 to Server-7 (192.168.20.10)

Ping from Server-7 to Client-7

From Server-7, I also ran a ping test to confirm two-way communication by targeting the client's IP

```
[root@Server-7 ~]# ping -c 4 192.168.20.20
PING 192.168.20.20 (192.168.20.20) 56(84) bytes of data.
64 bytes from 192.168.20.20: icmp_seq=1 ttl=64 time=0.504 ms
64 bytes from 192.168.20.20: icmp_seq=2 ttl=64 time=0.321 ms
64 bytes from 192.168.20.20: icmp_seq=3 ttl=64 time=0.405 ms
64 bytes from 192.168.20.20: icmp_seq=4 ttl=64 time=0.466 ms

--- 192.168.20.20 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2998ms
rtt min/avg/max/mdev = 0.321/0.424/0.504/0.069 ms
[root@Server-7 ~]#
```

Figure 21 Successful ping from Server-7 to Client-7 (192.168.20.20)

Ping from Client-7 to Server-7's NAT Interface

To further verify internal routing and ensure the NAT interface on Server-7 (192.168.5.134) was reachable from Client-7, I performed an additional ping:

```
atohme@Client-7:~$ ping -c 4 192.168.5.134
PING 192.168.5.134 (192.168.5.134) 56(84) bytes of data.
64 bytes from 192.168.5.134: icmp_seq=1 ttl=64 time=0.586 ms
64 bytes from 192.168.5.134: icmp_seq=2 ttl=64 time=0.492 ms
64 bytes from 192.168.5.134: icmp_seq=3 ttl=64 time=0.427 ms
64 bytes from 192.168.5.134: icmp_seq=4 ttl=64 time=0.515 ms

--- 192.168.5.134 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3063ms
rtt min/avg/max/mdev = 0.427/0.505/0.586/0.056 ms
atohme@Client-7:~$
```

Figure 22 Successful ping from Client-7 to Server-7's NAT interface (192.168.5.134)

Step 6: Enable Routing and Configure Firewall on Server-7

Before allowing Client-7 to access external resources (e.g., DNS 8.8.8.8 or websites), I configured Server-7 to act as a router by enabling **IP forwarding** and assigning its network interfaces to the appropriate **firewall zones**.

Enable IP Forwarding

To allow packet forwarding between interfaces (LAN and NAT), I enabled IPv4 forwarding in the kernel.

Temporarily (effective immediately): sysctl -w net.ipv4.ip_forward=1

Permanently (persists after reboot): echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf

```
[root@Server-7 ~]# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
[root@Server-7 ~]# echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf
[root@Server-7 ~]#
```

Figure 23 IP forwarding enabled and persisted in sysctl.conf

Rename Network Connections for Clarity

To simplify firewall configuration and clearly identify interfaces, I renamed the network connections

```
[root@Server-7 ~]# nmcli connection show
NAME                UUID                                  TYPE      DEVICE
ens160              d1f1dee0-521d-3bcc-b310-eb50258b8377  ethernet  ens160
LAN1                0dabdfa7-ebdb-4202-8e28-1268bd8b2cef  ethernet  ens192
lo                  3a6cb028-e5db-44c3-8411-463deeede934  loopback  lo
Wired connection 1  6ee98eff-b5e7-3fae-8435-3dcf5f265cc0  ethernet  --
[root@Server-7 ~]# nmcli connection modify ens160 connection.id NAT
[root@Server-7 ~]# nmcli connection show
NAME                UUID                                  TYPE      DEVICE
NAT                 d1f1dee0-521d-3bcc-b310-eb50258b8377  ethernet  ens160
LAN1                0dabdfa7-ebdb-4202-8e28-1268bd8b2cef  ethernet  ens192
lo                  3a6cb028-e5db-44c3-8411-463deeede934  loopback  lo
Wired connection 1  6ee98eff-b5e7-3fae-8435-3dcf5f265cc0  ethernet  --
[root@Server-7 ~]#
```

Figure 24 Network connection renamed for clarity (ens160 → NAT)

Assign Firewall Zone to NAT Interface

I assigned the NAT connection (associated with interface ens160) to the **external** firewall zone, which is appropriate for internet-facing traffic.

1. Listed available zones
2. Assigned the external zone to the NAT connection
3. Restarted the connection to apply changes

```
[root@Server-7 ~]# firewall-cmd --get-zones
block dmz drop external home internal nm-shared public trusted work
[root@Server-7 ~]# nmcli connection modify NAT con.zone external
[root@Server-7 ~]# nmcli connection down NAT ; nmcli connection up NAT
Connection 'NAT' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/2)
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/43)
[root@Server-7 ~]#
```

Figure 25 NAT interface assigned to external firewall zone

Assign Firewall Zone to LAN Interface

I assigned the LAN1 connection (linked to ens192) to the **nm-shared** zone, which is suitable for internal/private traffic. To finish, a verification to confirm interfaces were assigned properly.

```
[root@Server-7 ~]# nmcli connection modify LAN1 con.zone nm-shared
[root@Server-7 ~]# nmcli connection down LAN1 ; nmcli connection up LAN1
Connection 'LAN1' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/42)
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/45)
[root@Server-7 ~]# firewall-cmd --get-active-zones
external
  interfaces: ens160
nm-shared
  interfaces: ens192
[root@Server-7 ~]#
```

Figure 26 Firewall zones assigned: NAT to external, LAN1 to nm-shared

Step 7: Verify Internet Access from Client-7

With Server-7 now acting as a router for the LAN, I tested external connectivity from Client-7 to confirm that NAT and routing were configured correctly.

Ping External IP (Google DNS)

On Client-7, I ran the following command to ping Google's public DNS server

```
atohme@Client-7:~$ ping -c 3 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=127 time=11.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=127 time=8.38 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=127 time=8.20 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 8.200/9.330/11.413/1.474 ms
atohme@Client-7:~$
```

Figure 27 Client-7 successfully pinging external IP (8.8.8.8)

Ping External Website (e.g., google.com)

To confirm that DNS resolution was also working, I pinged a public website

```
atohme@Client-7:~$ ping -c 3 www.google.com
PING www.google.com (192.178.57.36) 56(84) bytes of data.
64 bytes from pnqroa-aj-in-f4.1e100.net (192.178.57.36): icmp_seq=1 ttl=127 time=8.62 ms
64 bytes from pnqroa-aj-in-f4.1e100.net (192.178.57.36): icmp_seq=2 ttl=127 time=8.35 ms
64 bytes from pnqroa-aj-in-f4.1e100.net (192.178.57.36): icmp_seq=3 ttl=127 time=8.37 ms

--- www.google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 8.350/8.446/8.619/0.122 ms
atohme@Client-7:~$
```

Figure 28 Client-7 successfully pinging external website (google.com)

2- Partitioning

This section covers the creation and temporary mounting of partitions on both Client-7 and Server-7 to prepare for later file sharing and testing.

Ubuntu Client (Client-7) Partitioning

Step 1: Attach New Virtual Disk (3 GB)

Using VMware settings:

- Powered off the **Client-7 VM**
- Added a new **3 GB SATA** virtual hard drive

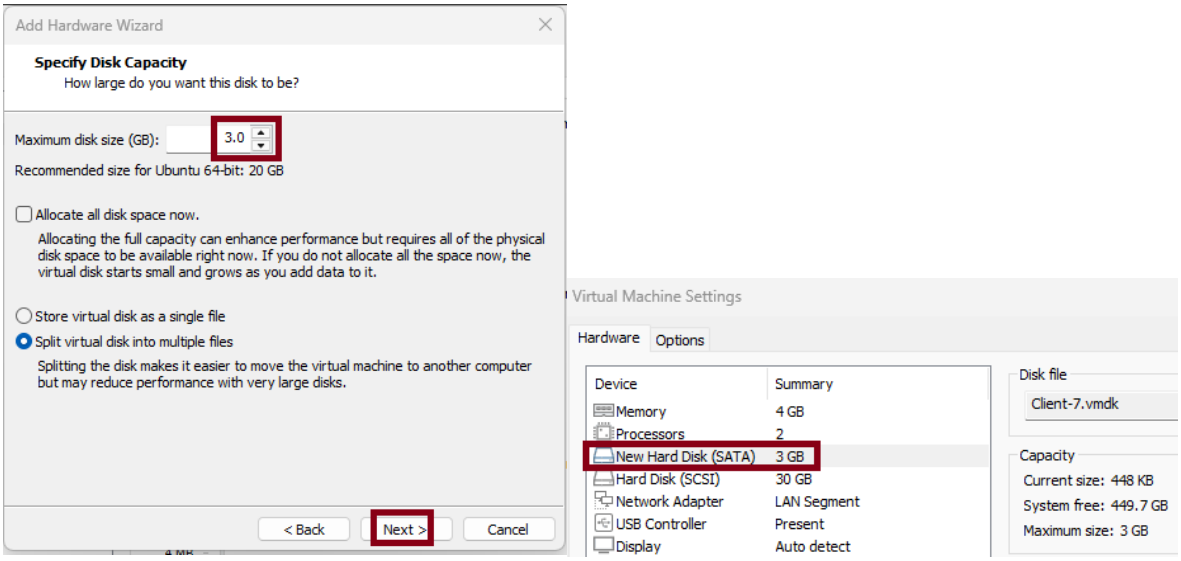


Figure 29 3 GB SATA disk added to Client-7 in VMware

Step 2: Identify the New Disk on Client-7

After powering the VM back on, I verified that the new disk was recognized by the system and listed all disk devices

```
atohme@Client-7:~$ sudo fdisk -l
Disk /dev/sda: 30 GiB, 32212254720 bytes, 62914560 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 40037ABF-8572-4823-8B55-06F6A8D5BEDE

Device        Start      End  Sectors  Size Type
/dev/sda1      2048     4095     2048    1M BIOS boot
/dev/sda2      4096  2004991  2000896  977M EFI System
/dev/sda3   2004992  6004735  3999744  1.9G Linux swap
/dev/sda4   6004736  8005631  2000896  977M Linux filesystem
/dev/sda5   8005632 17770495  9764864  4.7G Linux filesystem
/dev/sda6  17770496 62912511 45142016 21.5G Linux filesystem


Disk /dev/sdb: 3 GiB  3221225472 bytes, 6291456 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
atohme@Client-7:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda      8:0    0   30G  0 disk
├─sda1   8:1    0    1M  0 part
├─sda2   8:2    0  977M  0 part /boot/efi
├─sda3   8:3    0  1.9G  0 part [SWAP]
├─sda4   8:4    0  977M  0 part /boot
├─sda5   8:5    0  4.7G  0 part /home
└─sda6   8:6    0 21.5G  0 part /
sdb      8:16   0    3G  0 disk
atohme@Client-7:~$
```

Figure 30 New unpartitioned disk detected (/dev/sdb)

Step 3: Create a 2 GB Partition on /dev/sdb

Using fdisk, I created a **2 GB standard primary partition** on the newly added 3 GB SATA drive.

Steps performed in fdisk:

- Created a **new primary partition**
- Set size to **+2G**
- Wrote the changes to the partition table

```
atohme@Client-7:~$ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.37.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xc803c308.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-6291455, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-6291455, default 6291455): +2G

Created a new partition 1 of type 'Linux' and of size 2 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

Figure 31 2 GB primary partition created on /dev/sdb using fdisk

Step 4: Format Partition as ext4

Once the 2 GB partition was created, I formatted it with the **ext4 filesystem**, as required by the assignment.

```
Disk /dev/sdb: 3 GiB, 3221225472 bytes, 6291456 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc803c308

Device     Boot Start     End Sectors Size Id Type
/dev/sdb1        2048 4196351 4194304   2G 83 Linux
atohme@Client-7:~$ sudo mkfs.ext4 /dev/sdb1
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 524288 4k blocks and 131072 inodes
Filesystem UUID: 39d2fbf5-5aa6-478a-851c-f6c084592233
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

atohme@Client-7:~$
```

```
atohme@Client-7:~$ lsblk -f
NAME   FSTYPE FSVER LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINTS
sda
├─sda1
├─sda2 vfat   FAT32       005B-F52C                               969M     1% /boot/efi
├─sda3 swap   1           6299a0e6-8315-4bc5-ab11-771e29f5330b                [SWAP]
├─sda4 ext4   1.0         f349ffd8-da7e-4776-9a9f-76ae75e57a77  680.6M    21% /boot
├─sda5 ext4   1.0         adad363f-98f9-4020-8265-54dce36548b3    4.1G     3% /home
└─sda6 ext4   1.0         51e730d1-39a0-4e0a-b305-8fb8f83f4148   13.4G    31% /
sdb
└─sdb1 ext4   1.0         39d2fbf5-5aa6-478a-851c-f6c084592233
atohme@Client-7:~$
```

Figure 32 /dev/sdb1 formatted as ext4

Step 5: Mount ext4 Partition Temporarily on /Part1

After formatting /dev/sdb1 as ext4, I created a directory under my user's home folder and mounted the partition there.

```
atohme@Client-7:~$ mkdir /home/atohme/Part1
atohme@Client-7:~$ sudo mount /dev/sdb1 /home/atohme/Part1
atohme@Client-7:~$ df -Th
Filesystem     Type   Size  Used Avail Use% Mounted on
tmpfs          tmpfs  387M  1.8M  386M   1% /run
/dev/sda6      ext4    22G  6.6G   14G  33% /
tmpfs          tmpfs  1.9G     0  1.9G   0% /dev/shm
tmpfs          tmpfs  5.0M     0  5.0M   0% /run/lock
/dev/sda4      ext4   944M  198M  681M  23% /boot
/dev/sda5      ext4   4.6G  149M  4.2G   4% /home
/dev/sda2      vfat   976M  6.1M  969M   1% /boot/efi
tmpfs          tmpfs  387M   96K  387M   1% /run/user/1000
/dev/sdb1      ext4   2.0G   24K  1.8G   1% /home/atohme/Part1
atohme@Client-7:~$
```

Note. This mounted the 2 GB partition temporarily so it can be used during validation or file-sharing configuration later.

Figure 33 /dev/sdb1 mounted at /home/atohme/Part1 on Client-7

AlmaLinux Server (Server-7) – LVM Partitioning

In this section, I set up **logical volume management (LVM)** using two newly attached SATA disks. This will be used later for file sharing between Server-7 and Client-7

Attach Two New 5 GB SATA Disks

Using VMware:

- Powered off **Server-7**
- Added **two new 5 GB SATA** virtual disks

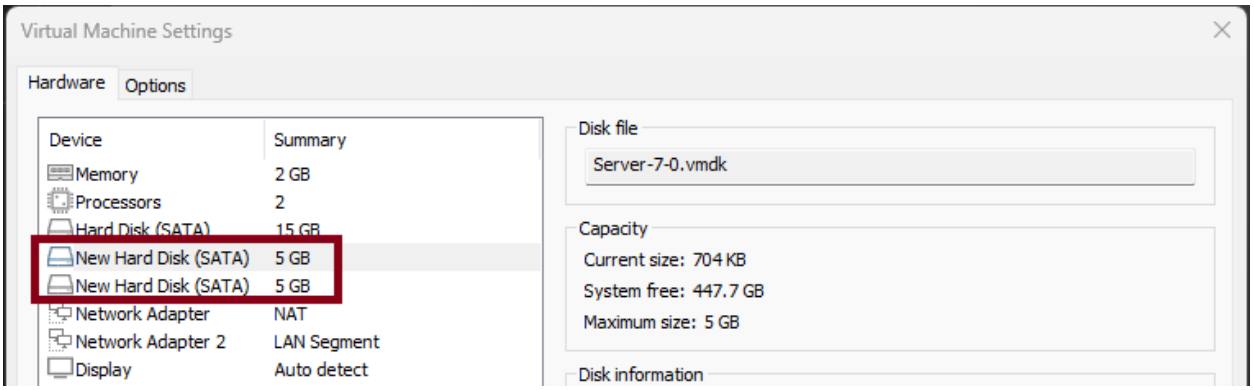


Figure 34 Two 5 GB virtual SATA drives added to Server-7

Verify New Disks on Server-7

After powering on Server-7, I verified the system detected the new drives

```
[root@Server-7 ~]# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda      8:0    0   15G  0 disk
├─sda1   8:1    0    1G  0 part /boot
├─sda2   8:2    0  1.5G  0 part [SWAP]
└─sda3   8:3    0 12.5G  0 part /
sdb      8:16   0    5G  0 disk
sdc      8:32   0    5G  0 disk
[root@Server-7 ~]#
```

Figure 35 New 5 GB unpartitioned disks detected on Server-7

Create Physical Volumes

I used **pvcreate** to initialize the two new disks as **LVM physical volumes**, which will be grouped together into a volume group.

```
[root@Server-7 ~]# pvcreate /dev/sdb /dev/sdc
  Physical volume "/dev/sdb" successfully created.
  Physical volume "/dev/sdc" successfully created.
  Creating devices file /etc/lvm/devices/system.devices
[root@Server-7 ~]#
```

Figure 36 Physical volumes created from /dev/sdb and /dev/sdc

To confirm the status of the newly created physical volumes, I used both pvs and pvdisplay. The output confirmed that /dev/sdb and /dev/sdc were successfully initialized as LVM physical volumes and are ready to be grouped.

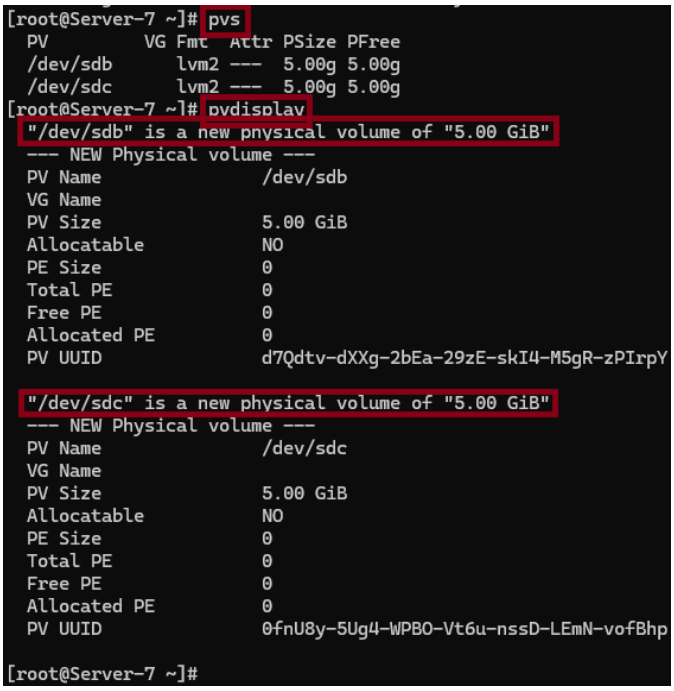


Figure 37 Confirmation of physical volumes using pvs and pvdisplay

Create Volume Group

I created a volume group named VM1_VG using the two 5 GB physical volumes. This volume group will be used to allocate space for a logical volume.

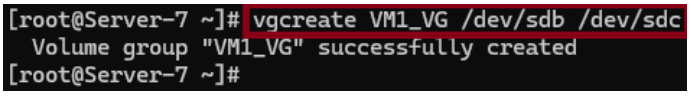


Figure 38 Volume group VM1_VG created using /dev/sdb and /dev/sdc

After creating the volume group, I used vgs and vgdisplay to confirm that both disks were included and the volume group was healthy and ready to use.

```
[root@Server-7 ~]# vgs
  VG     #PV #LV #SN Attr   VSize VFree
  VM1_VG   2   0   0 wz--n- 9.99g 9.99g
[root@Server-7 ~]# vgdisplay VM1_VG
  --- Volume group ---
  VG Name               VM1_VG
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               9.99 GiB
  PE Size               4.00 MiB
  Total PE              2558
  Alloc PE / Size       0 / 0
  Free  PE / Size       2558 / 9.99 GiB
  VG UUID               bJkIP8-iX1b-MB7E-9Ka1-iccy-9xNF-i8vxVu

[root@Server-7 ~]#
```

Figure 39 Volume group VM1_VG verified using vgs and vgdisplay

Create Logical Volume LV1 (8 GB)

I created a logical volume named LV1 within the VM1_VG volume group, allocating 8 GiB from the available space then I verified the logical volume using

```
[root@Server-7 ~]# lvcreate -L 8G -n LV1 VM1_VG
  Logical volume "LV1" created.
[root@Server-7 ~]# lvs
  LV   VG     Attr       LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  LV1  VM1_VG -wi-a----- 8.00g
[root@Server-7 ~]# lvdisplay
  --- Logical volume ---
  LV Path                /dev/VM1_VG/LV1
  LV Name                LV1
  VG Name                VM1_VG
  LV UUID                2z8bXl-acdq-DS1g-d23S-55sK-62Ed-ZWu8KZ
  LV Write Access        read/write
  LV Creation host, time Server-7, 2025-04-04 16:09:49 -0400
  LV Status              available
  # open                 0
  LV Size                8.00 GiB
  Current LE             2048
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0

[root@Server-7 ~]#
```

Figure 40 Logical volume LV1 (8 GiB) created and verified

Format Logical Volume as XFS

I formatted the newly created logical volume LV1 with the **XFS filesystem** to prepare it for mounting.

```
[root@Server-7 ~]# mkfs.xfs /dev/VM1_VG/LV1
meta-data=/dev/VM1_VG/LV1        isize=512    agcount=4, agsize=524288 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1 nrext64=0
data     =                       bsize=4096   blocks=2097152, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=16384, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[root@Server-7 ~]#
```

```
[root@Server-7 ~]# lsblk -f
NAME         FSTYPE      FSVER    LABEL UUID                                   FSAVAIL FSUSE% MOUNTPOINTS
sda
├─sda1       xfs                       c6bbfa8a-a661-4468-ae23-5afa0d0c694e    666.9M    31% /boot
├─sda2       swap        1             f7712360-0416-4e7a-83a5-c48eb5398353                  [SWAP]
└─sda3       xfs                       38f512ba-b5b9-4d93-8047-7dae099c66b4     11.1G    11% /
sdb          LVM2_member LVM2 001      d7Qdtv-dXXg-2bEa-29zE-skI4-M5gR-zPIrpY
└─VM1_VG-LV1 xfs                       4c9cecc7-7079-4c46-b9f4-d15249b32f18
sdc          LVM2_member LVM2 001      0fnU8y-5Ug4-WPBO-Vt6u-nssD-LEmN-vofBhp
└─VM1_VG-LV1 xfs                       4c9cecc7-7079-4c46-b9f4-d15249b32f18
[root@Server-7 ~]#
```

Figure 41 Logical volume LV1 formatted with XFS

Mount Logical Volume Temporarily

To complete the partitioning task on Server-7, I created the directory /Linux_Share and mounted the 8 GiB logical volume LV1 on it.

```
[root@Server-7 ~]# mkdir /Linux_Share
[root@Server-7 ~]# mount /dev/VM1_VG/LV1 /Linux_Share
[root@Server-7 ~]# df -Th
Filesystem              Type      Size  Used Avail Use% Mounted on
devtmpfs                devtmpfs  4.0M     0  4.0M   0% /dev
tmpfs                   tmpfs     867M     0  867M   0% /dev/shm
tmpfs                   tmpfs     347M  5.0M  342M   2% /run
/dev/sda3               xfs        13G  1.4G   12G  11% /
/dev/sda1               xfs       960M  294M  667M  31% /boot
tmpfs                   tmpfs     174M     0  174M   0% /run/user/0
/dev/mapper/VM1_VG-LV1  xfs       8.0G   89M  7.9G   2% /Linux_Share
[root@Server-7 ~]#
```

Figure 42 Logical volume LV1 mounted at /Linux_Share on Server-7

3- Users/Groups and Permissions

On the AlmaLinux server (Server-7)

Create Groups

I created two groups on Server-7 for organizing user access to shared directories.

```
[root@Server-7 ~]# groupadd -g 4000 admins
[root@Server-7 ~]# groupadd -g 5000 employees
[root@Server-7 ~]#
```

Figure 43 Groups 'admins' and 'employees' created with specific GIDs

Create Users and Set Passwords

Two users were created and assigned to their respective groups. Each user was given the UID specified in the assignment and the password alma.

```
[root@Server-7 ~]# groupadd -g 4000 admins
[root@Server-7 ~]# groupadd -g 5000 employees
[root@Server-7 ~]# useradd -u 2001 admin1
[root@Server-7 ~]# usermod -g admins admin1
[root@Server-7 ~]# useradd -u 2002 employee1
[root@Server-7 ~]# usermod -g employees employee1
[root@Server-7 ~]# passwd admin1
Changing password for user admin1.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@Server-7 ~]# passwd employee1
Changing password for user employee1.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@Server-7 ~]#
```

Figure 44 Users admin1 and employee1 created and assigned to groups with password set

Create and Assign Shared Directories

I created two directories under /Linux_Share one for each group and set their ownership accordingly. Permissions will be configured in Part 2 during NFS and Samba setup.

```
[root@Server-7 ~]# mkdir /Linux_Share/admins
[root@Server-7 ~]# chown -R admin1:admins /Linux_Share/admins
[root@Server-7 ~]# mkdir /Linux_Share/employees
[root@Server-7 ~]# chown -R employee1:employees /Linux_Share/employees
[root@Server-7 ~]# ls -ld /Linux_Share/admins /Linux_Share/employees
drwxr-xr-x. 2 admin1   admins    6 Apr  4 21:06 /Linux_Share/admins
drwxr-xr-x. 2 employee1 employees 6 Apr  4 21:08 /Linux_Share/employees
[root@Server-7 ~]#
```

Figure 45 /Linux_Share subdirectories created and ownership assigned

PART 2 – Linux Services

SSH Configuration

On Server-7 (AlmaLinux)

Verify SSH Service is Installed and Running

Before configuring SSH access rules, I verified that the **OpenSSH server** was already installed and running on Server-7.

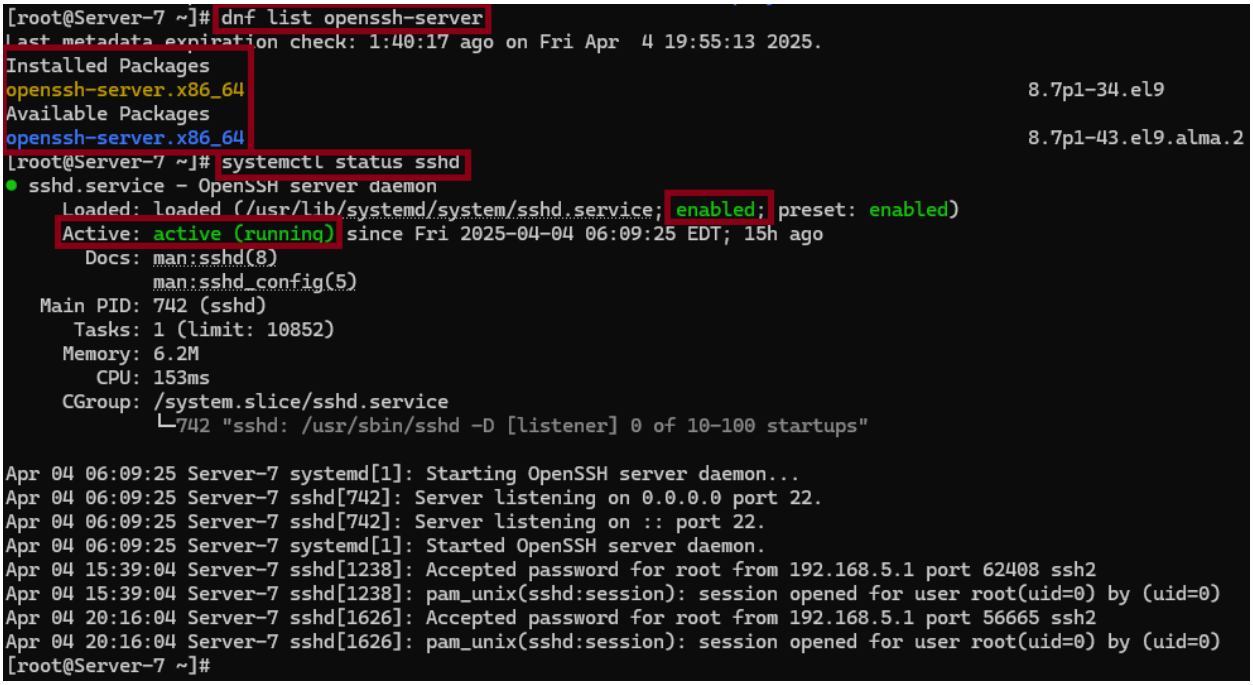


Figure 46 SSH service is installed and active on Server-7

Create SSH Login Banner

To display a login message when users connect via SSH, I created a simple banner file and configured the SSH daemon to reference it.

Figure 47 SSH banner file created and verified

To apply the banner, I edited the SSH configuration file at /etc/ssh/sshd_config.d/50-redhat.conf and added the following line:

```
# This system is following system-wide crypto policy. The changes to
# crypto properties (Ciphers, MACs, ...) will not have any effect in
# this or following included files. To override some configuration option,
# write it before this block or include it before this file.
# Please, see manual pages for update-crypto-policies(8) and sshd_config(5).
Include /etc/crypto-policies/back-ends/opensshserver.config

Banner /etc/ssh/banner.txt

SyslogFacility AUTHPRIV

ChallengeResponseAuthentication no

GSSAPIAuthentication yes
GSSAPICleanupCredentials no

UsePAM yes

X11Forwarding yes
PermitRootLogin yes
# It is recommended to use pam_motd in /etc/pam.d/sshd instead of PrintMotd,
# as it is more configurable and versatile than the built-in version.
PrintMotd no
```

Figure 48 SSH banner line added in 50-redhat.conf

SSH Access Permissions Configuration

To meet the SSH configuration requirements, I set up access control using Linux user groups. SSH access was **allowed** for members of the admins group and **denied** for members of the employees group using the following directives in 50-redhat.conf:

```
# This system is following system-wide crypto policy. The changes to
# crypto properties (Ciphers, MACs, ...) will not have any effect in
# this or following included files. To override some configuration option,
# write it before this block or include it before this file.
# Please, see manual pages for update-crypto-policies(8) and sshd_config(5).
Include /etc/crypto-policies/back-ends/opensshserver.config

Banner /etc/ssh/banner.txt

SyslogFacility AUTHPRIV

ChallengeResponseAuthentication no

GSSAPIAuthentication yes
GSSAPICleanupCredentials no

UsePAM yes

X11Forwarding yes
PermitRootLogin yes
# It is recommended to use pam_motd in /etc/pam.d/sshd instead of PrintMotd,
# as it is more configurable and versatile than the built-in version.
PrintMotd no

AllowGroups admins

DenyGroups employees
```

Figure 49 Group-based SSH access rules defined in 50-redhat.conf

After modifying the SSH configuration, I applied the changes by reloading the SSH service

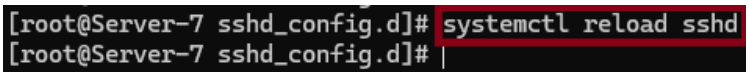


Figure 50 SSH service reloaded to apply new configuration

Note: These access restrictions also applied to the root user, who is not a member of the admins group by default. Since Part 3 – SSH Testing requires logging in as root, I added root to the admins group to permit access while still enforcing group-based restrictions.

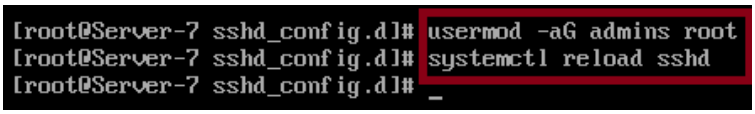


Figure 51 Root added to the admins group and SSH daemon reloaded

Verifying Root Login Over SSH

To confirm the configuration, I tested SSH login as root from my host machine using Windows PowerShell. The login was successful, and the welcome banner displayed as expected.

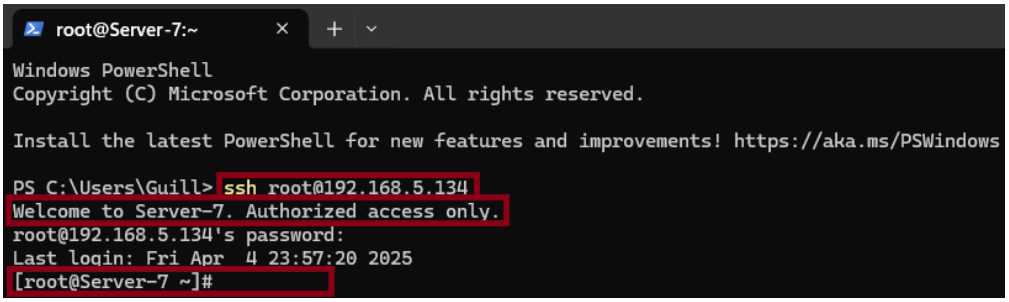


Figure 52 Successful SSH login as root via PowerShell with password

On Client-7 (Ubuntu)

Configure SSH Key-Based Authentication from Client-7

To enable root login on Server-7 from Client-7 without requiring a password, I configured SSH key-based authentication

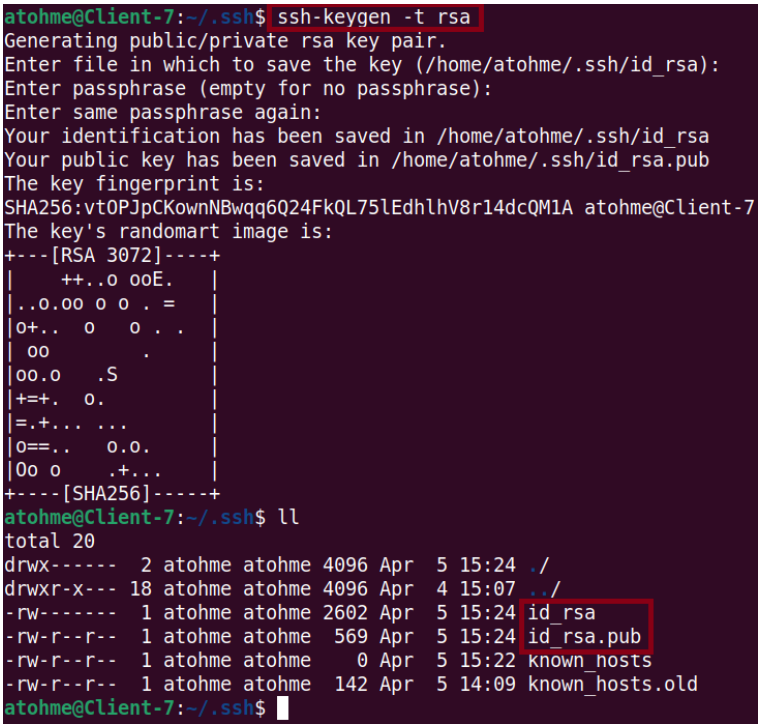


Figure 53 RSA SSH key pair generated and saved to ~/.ssh/id_rsa and ~/.ssh/id_rsa.pub

Copy Public Key to Server-7 (Root)

To enable SSH key-based authentication, I copied the public key (id_rsa.pub) from the Ubuntu client to the root user on Server-7 using ssh-copy-id. This allows the root user on Server-7 to authenticate using the private key from Client-7 without requiring a password.

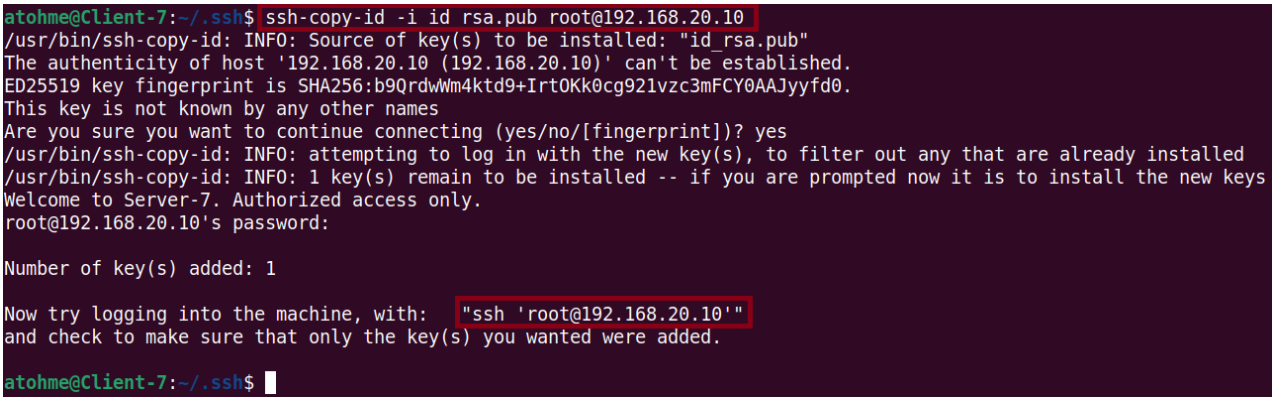


Figure 54 Public key successfully copied to root@192.168.20.10 via ssh-copy-id

NFS Configuration

On Server-7 (AlmaLinux)

Install the NFS utilities

To begin configuring NFS, I installed the necessary NFS utilities on Server-7 using the following command

```
[root@Server-7 ~]# dnf install -y nfs-utils
Last metadata expiration check: 0:36:11 ago on Sat Apr  5 02:50:12 2025.
Dependencies resolved.
================================================================================
 Package                                                 Architecture
================================================================================
Installing:
 nfs-utils                                               x86_64
Upgrading:
 device-mapper                                           x86_64
 device-mapper-event                                     x86_64
 device-mapper-event-libs                                x86_64
 device-mapper-libs                                      x86_64
 libsss_certmap                                          x86_64
 libsss_idmap                                            x86_64
 libsss_nss_idmap                                        x86_64
 libsss_sudo                                             x86_64
 libtevent                                               x86_64
 lvm2                                                    x86_64
 lvm2-libs                                               x86_64
 sssd-client                                             x86_64
 sssd-common                                             x86_64
 sssd-kcm                                                x86_64
Installing dependencies:
 gssproxy                                                x86_64
 keyutils                                                x86_64
 libev                                                   x86_64
 libnfsidmap                                             x86_64
 libtirpc                                                x86_64
 libverto-libev                                          x86_64
 python3-pyyaml                                          x86_64
 quota                                                   x86_64
 quota-nls                                               noarch
 rpcbind                                                 x86_64
 sssd-nfs-idmap                                          x86_64
```

Figure 55 Installation of NFS utilities and related dependencies on Server-7

Enable and Start NFS Services on Server-7

To prepare Server-7 for NFS sharing, I enabled and started the required services. I used the systemctl commands to enable and verify the status of each service

```
[root@Server-7 ~]# systemctl enable --now nfs-server
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service → /usr/lib/systemd/system/nfs-server.service.
[root@Server-7 ~]# systemctl status nfs-server
● nfs-server.service - NFS server and services
     Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; preset: disabled)
     Active: active (exited) since Sat 2025-04-05 03:29:53 EDT; 21s ago
       Docs: man:rpc.nfsd(8)
             man:exportfs(8)
    Process: 12989 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
    Process: 12990 ExecStart=/usr/sbin/rpc.nfsd (code=exited, status=0/SUCCESS)
    Process: 13009 ExecStart=/bin/sh -c if systemctl -q is-active gssproxy; then systemctl reload gssproxy ; fi (code=exited, status=0/SUCCESS)
   Main PID: 13009 (code=exited, status=0/SUCCESS)
        CPU: 27ms

Apr 05 03:29:53 Server-7 systemd[1]: Starting NFS server and services...
Apr 05 03:29:53 Server-7 systemd[1]: Finished NFS server and services.
[root@Server-7 ~]# systemctl status nfs-idmapd.service
● nfs-idmapd.service - NFSv4 ID-name mapping service
     Loaded: loaded (/usr/lib/systemd/system/nfs-idmapd.service; static)
     Active: active (running) since Sat 2025-04-05 03:29:53 EDT; 48s ago
       Docs: man:idmapd(8)
    Process: 12978 ExecStart=/usr/sbin/rpc.idmapd (code=exited, status=0/SUCCESS)
   Main PID: 12981 (rpc.idmapd)
      Tasks: 1 (limit: 10852)
     Memory: 588.0K
        CPU: 4ms
     CGroup: /system.slice/nfs-idmapd.service
             └─12981 /usr/sbin/rpc.idmapd

Apr 05 03:29:42 Server-7 systemd[1]: Starting NFSv4 ID-name mapping service...
Apr 05 03:29:42 Server-7 rpc.idmapd[12981]: Setting log level to 0
Apr 05 03:29:53 Server-7 systemd[1]: Started NFSv4 ID-name mapping service.
[root@Server-7 ~]# systemctl status rpcbind
● rpcbind.service - RPC Bind
     Loaded: loaded (/usr/lib/systemd/system/rpcbind.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-04-05 03:29:42 EDT; 1min 8s ago
TriggeredBy: ● rpcbind.socket
       Docs: man:rpcbind(8)
   Main PID: 12982 (rpcbind)
      Tasks: 1 (limit: 10852)
     Memory: 1.5M
        CPU: 19ms
     CGroup: /system.slice/rpcbind.service
             └─12982 /usr/bin/rpcbind -w -f

Apr 05 03:29:42 Server-7 systemd[1]: Starting RPC Bind...
Apr 05 03:29:42 Server-7 systemd[1]: Started RPC Bind.
[root@Server-7 ~]#
```

Figure 56 NFS server, ID mapping, and RPC Bind services enabled and running on Server-7

Configure Firewall for NFS

Before exporting directories, I configured the firewall on Server-7 to allow necessary NFS services. This ensures that remote clients (like Client-7) can access the shared directories.

```
[root@Server-7 ~]# firewall-cmd --permanent --add-service=nfs --zone=nm-shared
success
[root@Server-7 ~]# firewall-cmd --permanent --add-service=mountd --zone=nm-shared
success
[root@Server-7 ~]# firewall-cmd --permanent --add-service=rpc-bind --zone=nm-shared
success
[root@Server-7 ~]# firewall-cmd --reload
success
[root@Server-7 ~]# firewall-cmd --list-services --zone=nm-shared
dhcp dns mountd nfs rpc-bind ssh
[root@Server-7 ~]#
```

Figure 57 NFS related services allowed through the firewall in the nm-shared zone on Server-7

Set Directory Permissions for NFS Shares

To enforce the correct access rules for each NFS share, I applied specific permissions to the shared directories:

- /Linux_Share/admins was configured with **read and write** permissions for the admins group only.
- /Linux_Share/employees was configured with **read-only** permissions for the employees group.

```
[root@Server-7 ~]# chmod -R 770 /Linux_Share/admins
[root@Server-7 ~]# chmod -R 750 /Linux_Share/employees
[root@Server-7 ~]#
```

Figure 58 Directory permissions set for NFS shares on Server-7

NFS Export Configuration

To share the appropriate directories over the network, I configured the NFS exports on Server-7. The /Linux_Share/admins directory was configured with **read-write access** for the admins group, and the /Linux_Share/employees directory was configured with **read-only access** for the employees group

```
[root@Server-7 ~]# echo "/Linux_Share/admins 192.168.20.0/24(rw,sync,no_all_squash)" >> /etc/exports
[root@Server-7 ~]# echo "/Linux_Share/employees 192.168.20.0/24(ro,sync,no_all_squash)" >> /etc/exports
[root@Server-7 ~]# exportfs -arv
exporting 192.168.20.0/24:/Linux_Share/employees
exporting 192.168.20.0/24:/Linux_Share/admins
[root@Server-7 ~]# exportfs -s
/Linux_Share/admins  192.168.20.0/24(sync,wdelay,hide,no_subtree_check,sec=sys,rw,secure,root_squash,no_all_squash)
/Linux_Share/employees  192.168.20.0/24(sync,wdelay,hide,no_subtree_check,sec=sys,ro,secure,root_squash,no_all_squash)
[root@Server-7 ~]#
```

Figure 59 NFS shares configured and verified using exportfs commands

SAMBA Configuration

To create a public file share accessible without authentication, I configured **SAMBA**.

On Server-7 (AlmaLinux)

Install SAMBA Package

I installed the required samba package, which includes the necessary components to set up and manage SAMBA shares.

```
[root@Server-7 ~]# dnf install -y samba
Last metadata expiration check: 1:22:33 ago on Sat Apr  5 02:50:12 2025.
Dependencies resolved.
================================================================================
 Package                    Architecture          Version
================================================================================
Installing:
 samba                      x86_64                4.20.2-2.el9_5.alma.1
Upgrading:
 libldb                     x86_64                2.9.1-2.el9
 libtalloc                  x86_64                2.4.2-1.el9
 libtdb                     x86_64                1.4.10-1.el9
Installing dependencies:
 avahi-libs                 x86_64                0.8-21.el9
 cups-libs                  x86_64                1:2.3.3op2-31.el9_5
 libicu                     x86_64                67.1-9.el9
 libnetapi                  x86_64                4.20.2-2.el9_5.alma.1
 libwbclient                x86_64                4.20.2-2.el9_5.alma.1
 samba-client-libs          x86_64                4.20.2-2.el9_5.alma.1
 samba-common               noarch                4.20.2-2.el9_5.alma.1
 samba-common-libs          x86_64                4.20.2-2.el9_5.alma.1
 samba-common-tools         x86_64                4.20.2-2.el9_5.alma.1
 samba-dcerpc               x86_64                4.20.2-2.el9_5.alma.1
 samba-ldb-ldap-modules     x86_64                4.20.2-2.el9_5.alma.1
 samba-libs                 x86_64                4.20.2-2.el9_5.alma.1
```

Figure 60 SAMBA and dependencies installed on Server-7

Enable and Start the SAMBA Service

After installing the necessary packages, I enabled and started the SAMBA service (smb) to ensure it runs immediately and on every system boot.

```
[root@Server-7 ~]# systemctl enable --now smb
Created symlink /etc/systemd/system/multi-user.target.wants/smb.service → /usr/lib/systemd/system/smb.service.
[root@Server-7 ~]# systemctl status smb
● smb.service - Samba SMB Daemon
     Loaded: loaded (/usr/lib/systemd/system/smb.service; enabled; preset: disabled)
     Active: active (running) since Sat 2025-04-05 04:24:23 EDT; 8s ago
       Docs: man:smbd(8)
             man:samba(7)
             man:smb.conf(5)
   Main PID: 13767 (smbd)
     Status: "smbd: ready to serve connections..."
      Tasks: 3 (limit: 10852)
     Memory: 8.4M
        CPU: 39ms
     CGroup: /system.slice/smb.service
             ├─13767 /usr/sbin/smbd --foreground --no-process-group
             ├─13769 /usr/sbin/smbd --foreground --no-process-group
             └─13770 /usr/sbin/smbd --foreground --no-process-group

Apr 05 04:24:23 Server-7 systemd[1]: Starting Samba SMB Daemon...
Apr 05 04:24:23 Server-7 smbd[13767]: [2025/04/05 04:24:23.631179,  0] ../../source3/smbd/server.c:1746(main)
Apr 05 04:24:23 Server-7 smbd[13767]:   smbd version 4.20.2 started.
Apr 05 04:24:23 Server-7 smbd[13767]:   Copyright Andrew Tridgell and the Samba Team 1992-2024
Apr 05 04:24:23 Server-7 systemd[1]: Started Samba SMB Daemon.
[root@Server-7 ~]#
```

Figure 61 SAMBA SMB Daemon enabled and actively running on Server-7

Configure Firewall for SAMBA Access

To ensure SAMBA traffic is allowed through the firewall, I added the samba service to the nm-shared zone, then reloaded the firewall to apply the changes. The final service list confirms SAMBA is now allowed.

```
[root@Server-7 ~]# firewall-cmd --permanent --add-service=samba --zone=nm-shared
success
[root@Server-7 ~]# firewall-cmd --reload
success
[root@Server-7 ~]# firewall-cmd --list-services --zone=nm-shared
dhcp dns mountd nfs rpc-bind samba ssh
[root@Server-7 ~]#
```

Figure 62 SAMBA service successfully added to firewall and verified in the nm-shared zone

Configure Public Anonymous SAMBA Share

To configure an anonymous SAMBA share, I created a new directory /Public and set the appropriate permissions and security context to allow access by guest users.

```
[root@Server-7 ~]# mkdir /Public
[root@Server-7 ~]# chmod -R 777 /Public
[root@Server-7 ~]# chown -R nobody:nobody /Public
[root@Server-7 ~]# chcon -t samba_share_t /Public
[root@Server-7 ~]#
```

Figure 63 Public directory created and configured for SAMBA access

Configure the SAMBA Service

I edited the /etc/samba/smb.conf configuration file to include the global setting to enable guest access for unknown users, and added a new share definition for Public_Sharing. This configuration ensures the share is writable, browsable, and accessible without authentication

```
[root@Server-7 ~]# vim /etc/samba/smb.conf
[root@Server-7 ~]#
```

Figure 64 Opening the Samba configuration file /etc/samba/smb.conf for editing

```
[global]
        workgroup = SAMBA
        security = user

        passdb backend = tdbsam
        map to guest = Bad User

        printing = cups
        printcap name = cups
        load printers = yes
        cups options = raw

[homes]
        comment = Home Directories
        valid users = %S, %D%w%S
        browseable = No
        read only = No
        inherit acls = Yes

[printers]
        comment = All Printers
        path = /var/tmp
        printable = Yes
        create mask = 0600
        browseable = No

[print$]
        comment = Printer Drivers
        path = /var/lib/samba/drivers
        write list = @printadmin root
        force group = @printadmin
        create mask = 0664
        directory mask = 0775

[Public_Sharing]
        comment = Public Share
        path = /Public
        browsable = yes
        writable = yes
        guest ok = yes
        read only = no
        force user = nobody
```

Key Samba Configuration Directives:

- path: Directory being shared.
- browsable: Makes share visible.
- writable: Allows write access.
- guest ok: Enables guest access.
- read only: Set to "no" for write access.
- force user: Files are created as this user.
- map to guest: Unknown users treated as guests.

Figure 65 Samba configuration with guest access enabled and the Public_Sharing anonymous share defined.

Verifying and Restarting Samba Configuration

I used testparm to validate the updated Samba configuration. The output confirmed that the configuration file was correct and the server is running in standalone mode. To apply the changes, I restarted the Samba service.

```
[root@Server-7 ~]# testparm
Load smb config files from /etc/samba/smb.conf
Loaded services file OK.
Weak crypto is allowed by GnuTLS (e.g. NTLM as a compatibility fallback)

Server role: ROLE_STANDALONE
```

Figure 66 Testparm valid Samba configuration

```
[root@Server-7 ~]# systemctl restart smb
[root@Server-7 ~]#
```

Figure 67 Samba service successfully restarted

PART 3 – Validation Test

SSH Testing

On Client-7 (Ubuntu)

SSH Testing – Root Access

From the Ubuntu client, I initiated an SSH connection to the Server-7 root account.

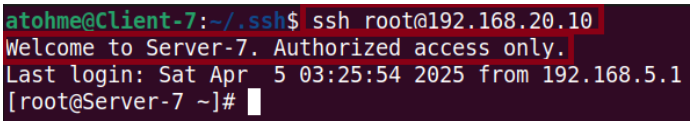


Figure 68 Successful SSH login as root from the Ubuntu client with banner displayed and no password prompt.

SSH Group-Based Access Test (Admin User)

To validate the group-based SSH permissions, I attempted to log in from the Ubuntu client using the admin1 user. As expected, access was granted since admin1 is a member of the admins group, which is allowed via the AllowGroups directive in the SSH configuration.

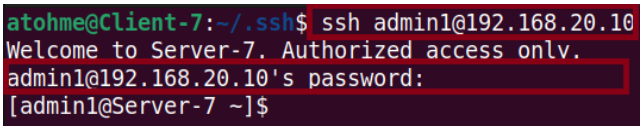


Figure 69 Successful SSH login to Server-7 as admin1 from Client-7

SSH Group-Based Access Test (Employee User)

To verify that SSH access is denied for users in the employees group, I attempted to log in to Server-7 from Client-7 using the employee1 account. As configured with the DenyGroups employees directive, the connection was correctly refused.

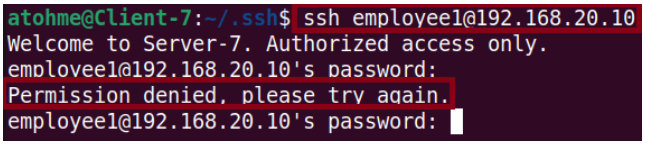


Figure 70 SSH login denied for employee1 from Client-7 due to group-based restriction.

NFS Testing

On Client-7 (Ubuntu)

NFS Client Setup on Client-7

Before mounting NFS shares from Server-7, the Ubuntu client must have the required NFS utilities installed. I updated the package index and installed the necessary tools

```
atohme@Client-7:~$ sudo apt update
[sudo] password for atohme:
Get:1 https://dl.google.com/linux/chrome/deb stable InRelease [1,825 B]
Hit:2 http://ca.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 https://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1,217 B]
Get:4 http://ca.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:6 http://ca.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:7 http://ca.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2,461 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [43.1 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 DEP-11 Metadata [208 B]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [125 kB]
Get:11 http://ca.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [103 kB]
Get:12 http://ca.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 DEP-11 Metadata [212 B]
Get:13 http://ca.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [359 kB]
Get:14 http://ca.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 DEP-11 Metadata [208 B]
Get:16 http://ca.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [7,096 B]
Get:17 http://ca.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 DEP-11 Metadata [212 B]
Get:18 http://ca.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [22.9 kB]
Get:19 http://ca.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 DEP-11 Metadata [212 B]
Fetched 3,511 kB in 2s (1,752 kB/s)
```

Figure 71 Ubuntu client system packages updated

```
atohme@Client-7:~$ sudo apt -y install nfs-common nfs4-acl-tools vim
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  keyutils libevent-core-2.1-7 libnfsidmap1 rpcbind vim-runtime
Suggested packages:
  open-iscsi watchdog ctags vim-doc vim-scripts
The following NEW packages will be installed:
  keyutils libevent-core-2.1-7 libnfsidmap1 nfs-common nfs4-acl-tools rpcbind vim vim-runtime
0 upgraded, 8 newly installed, 0 to remove and 34 not upgraded.
Need to get 9,067 kB of archives.
After this operation, 39.4 MB of additional disk space will be used.
```

Figure 72 NFS and required utilities installed

Verifying NFS Exports from Server-7

After installing the NFS client utilities, I used the showmount command to confirm that the NFS shares on Server-7 were correctly exported and visible from Client-7

```
atohme@Client-7:~$ showmount -e 192.168.20.10
Export list for 192.168.20.10:
/Linux_Share/employees 192.168.20.0/24
/Linux_Share/admins    192.168.20.0/24
atohme@Client-7:~$
```

Figure 73 NFS exports from Server-7

Recreate User Accounts and Groups on Client-7

To ensure proper permission mapping when accessing the NFS shares, I recreated the same users and groups on the Ubuntu client using the same UIDs and GIDs as configured on the AlmaLinux server.

```
atohme@Client-7:~$ sudo addgroup --gid 4000 admins
[sudo] password for atohme:
Adding group `admins' (GID 4000) ...
Done.
atohme@Client-7:~$ sudo addgroup --gid 5000 employees
Adding group `employees' (GID 5000) ...
Done.
```

Figure 74 Groups admins and employees created with matching GIDs.

```
atohme@Client-7:~$ sudo adduser --uid 2001 --ingroup admins admin1
Adding user `admin1' ...
Adding new user `admin1' (2001) with group `admins' ...
Creating home directory `/home/admin1' ...
Copying files from `/etc/skel' ...
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
Changing the user information for admin1
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
atohme@Client-7:~$ sudo adduser --uid 2002 --ingroup employees employee1
Adding user `employee1' ...
Adding new user `employee1' (2002) with group `employees' ...
Creating home directory `/home/employee1' ...
Copying files from `/etc/skel' ...
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
Changing the user information for employee1
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
atohme@Client-7:~$
```

/L

Figure 75 Users admin1 and employee1 added with correct UIDs and group assignments.

Mounting NFS Shares

On Client-7, I created the local directories and mounted both the admins and employees NFS shares from Server-7. I then confirmed the mounts using the mount command.

```
atohme@Client-7:~$ sudo mkdir -p /Share/admins /Share/employees
[sudo] password for atohme:
atohme@Client-7:~$ sudo mount -t nfs 192.168.20.10:/Linux_Share/admins /Share/admins
atohme@Client-7:~$ sudo mount -t nfs 192.168.20.10:/Linux_Share/employees /Share/employees
atohme@Client-7:~$ sudo mount | grep nfs
192.168.20.10:/Linux_Share/admins on /Share/admins type nfs4 (rw,relatime,vers=4.2,rsize=262144,wsize=262144,namlen=255,hard,proto
=tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.20.20,local_lock=none,addr=192.168.20.10)
192.168.20.10:/Linux_Share/employees on /Share/employees type nfs4 (rw,relatime,vers=4.2,rsize=262144,wsize=262144,namlen=255,hard
,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.20.20,local_lock=none,addr=192.168.20.10)
atohme@Client-7:~$
```

Figure 76 NFS shares mounted and verified on Client-7

Testing Permissions for Both Groups

Admin1 User

To verify write access for the **admins** group, I switched to the admin1 user on the Client-7 and successfully created a test file inside the mounted NFS share /Share/admins.

```
atohme@Client-7:~$ su - admin1
Password:
admin1@Client-7:~$ touch /Share/admins/test-admin.txt
admin1@Client-7:~$
```

Figure 77 Creating a test file as admin1 in /Share/admins on the Ubuntu client

```
[root@Server-7 ~]# ls -la /Linux_Share/admins
total 0
drwxrwx---. 2 admin1 admins 28 Apr  5 06:56 .
drwxr-xr-x. 4 root   root   37 Apr  4 21:08 ..
-rw-r--r--. 1 admin1 admins  0 Apr  5 06:56 test-admin.txt
[root@Server-7 ~]#
```

Figure 78 File test-admin.txt created by admin1 visible in /Linux_Share/admins on the NFS server

To confirm read-only restrictions for the **employees** share, I attempted to create a file inside /Share/employees while logged in as admin1. As expected, the operation failed with a "Permission denied" message, verifying that write access is correctly restricted.

```
admin1@Client-7:~$ touch /Share/employees/test-admin-employee.txt
touch: cannot touch '/Share/employees/test-admin-employee.txt': Permission denied
admin1@Client-7:~$
```

Figure 79 Write attempt by admin1 denied in the read-only /Share/employees NFS share

Employee1 User

To verify the read-only permissions on the /Share/employees NFS mount, I attempted to create a file while logged in as employee1. The operation failed with a "Read-only file system" error, confirming correct access control.

```
employee1@Client-7:~$ touch /Share/employees/test-employee.txt
touch: cannot touch '/Share/employees/test-employee.txt': Read-only file system
employee1@Client-7:~$
```

Figure 80 File creation denied for employee1 in /Share/employees due to read-only access.

Still logged in as employee1, I attempted to write to the /Share/admins mount, which is intended only for the admins group. The attempt failed with a permission denied error, confirming that non-admin users cannot write to the admins share.

```
employee1@Client-7:~$ touch /Share/admins/test-employee-admin.txt
touch: cannot touch '/Share/admins/test-employee-admin.txt': Permission denied
employee1@Client-7:~$
```

Figure 81 Access denied for employee1 when attempting to create a file in the /Share/admins directory.

SAMBA Testing

On Client-7 (Ubuntu)

SAMBA Client Setup on Client-7

To prepare the Ubuntu client for SAMBA testing, I updated the package list and installed the smbclient and related utilities using APT. These tools allow access to SMB shared directories from the terminal.

```
atohme@Client-7:~$ sudo apt update
[sudo] password for atohme:
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:2 https://dl.google.com/linux/chrome/deb stable InRelease [1,825 B]
Hit:3 http://ca.archive.ubuntu.com/ubuntu jammy InRelease
Get:4 http://ca.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:5 http://ca.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:6 http://ca.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [103 kB]
Get:7 http://ca.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 DEP-11 Metadata [212 B]
Get:8 http://ca.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [359 kB]
Get:9 http://ca.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [43.1 kB]
Get:11 http://ca.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [7,080 B]
Get:12 http://ca.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 DEP-11 Metadata [212 B]
Get:13 http://ca.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [22.9 kB]
Get:14 http://ca.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 DEP-11 Metadata [212 B]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 DEP-11 Metadata [208 B]
Get:16 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [125 kB]
Get:17 https://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1,216 B]
Get:18 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 DEP-11 Metadata [208 B]
Fetched 1,049 kB in 1s (791 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
34 packages can be upgraded. Run 'apt list --upgradable' to see them.
atohme@Client-7:~$ sudo apt install samba-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'smbclient' instead of 'samba-client'
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  python3-gpg python3-samba python3-tdb samba-common samba-common-bin samba-dsdb-modules
Suggested packages:
  heimdal-clients python3-markdown python3-dnspython cifs-utils
The following NEW packages will be installed:
  python3-gpg python3-samba python3-tdb samba-common samba-common-bin samba-dsdb-modules smbclient
0 upgraded, 7 newly installed, 0 to remove and 34 not upgraded.
Need to get 4,827 kB of archives.
After this operation, 29.0 MB of additional disk space will be used.
```

Figure 82 Installing smbclient and dependencies on the Ubuntu client for SAMBA share access.

SAMBA Public Share

To verify the anonymous access to the public SAMBA share, I used the smbclient utility from Client-7 to connect to the Public_Sharing directory on Server-7. Once connected, I created a test directory to confirm write permissions.

```
atohme@Client-7:~$ smbclient //192.168.20.10/Public Sharing
Password for [WORKGROUP\atohme]:
Try "help" to get a list of possible commands.
smb: \> mkdir test-public-share
smb: \>
```

Figure 83 Connecting to the Public SAMBA share and creating a test directory from Client-7.

Back on Server-7, I verified that the directory was successfully created inside /Public with the appropriate ownership set to nobody:nobody, confirming that anonymous write access was functioning correctly.

```
[root@Server-7 ~]# ls -la /Public
total 0
drwxrwxrwx.  3 nobody nobody  31 Apr  5 07:25 .
dr-xr-xr-x. 20 root   root   268 Apr  5 04:32 ..
drwxr-xr-x.  2 nobody nobody   6 Apr  5 07:25 test-public-share
[root@Server-7 ~]#
```

Figure 84 Verifying the newly created folder on Server-7's /Public directory.

Additional Verification – Samba Listening Ports

To confirm the Samba service is actively listening and communicating over the expected ports, I installed the net-tools package and used netstat to check network activity.

```
[root@Server-7 ~]# dnf install net-tools -y
Last metadata expiration check: 1:29:02 ago on Sat Apr  5 06:04:12 2025.
Dependencies resolved.
================================================================================
 Package            Architecture     Version                        Repository        Size
================================================================================
Installing:
 net-tools          x86_64           2.0-0.64.20160912git.el9       baseos           294 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 294 k
Installed size: 906 k
Downloading Packages:
net-tools-2.0-0.64.20160912git.el9.x86_64.rpm                  5.0 MB/s | 294 kB     00:00
--------------------------------------------------------------------------------
Total                                                          639 kB/s | 294 kB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : net-tools-2.0-0.64.20160912git.el9.x86_64              1/1
  Running scriptlet: net-tools-2.0-0.64.20160912git.el9.x86_64              1/1
  Verifying        : net-tools-2.0-0.64.20160912git.el9.x86_64              1/1

Installed:
  net-tools-2.0-0.64.20160912git.el9.x86_64
```

Figure 85 Installing net-tools for access to the netstat command.

```
[root@Server-7 ~]# netstat -tunap | grep smb
tcp        0      0 0.0.0.0:139             0.0.0.0:*               LISTEN      13904/smbd
tcp        0      0 0.0.0.0:445             0.0.0.0:*               LISTEN      13904/smbd
tcp        0      0 192.168.20.10:445       192.168.20.20:41072     ESTABLISHED 14169/smbd
tcp6       0      0 :::139                  :::*                    LISTEN      13904/smbd
tcp6       0      0 :::445                  :::*                    LISTEN      13904/smbd
[root@Server-7 ~]#
```

Figure 86 Using netstat to confirm Samba is listening on ports 139 and 445

Conclusion

This assignment helped me gain hands-on experience with configuring and testing SSH, NFS, and Samba services on a Linux server. I applied group-based access controls, set up key-based authentication, and verified permissions through practical tests. Each service worked as expected, and I was able to meet all the requirements outlined in the instructions.