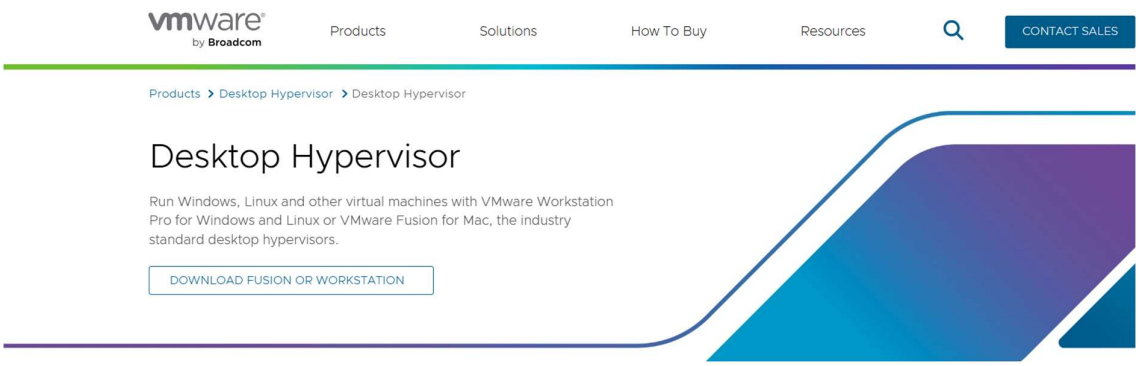
Task 1

Preparing Lab Environment

a) Installing VMware Workstation

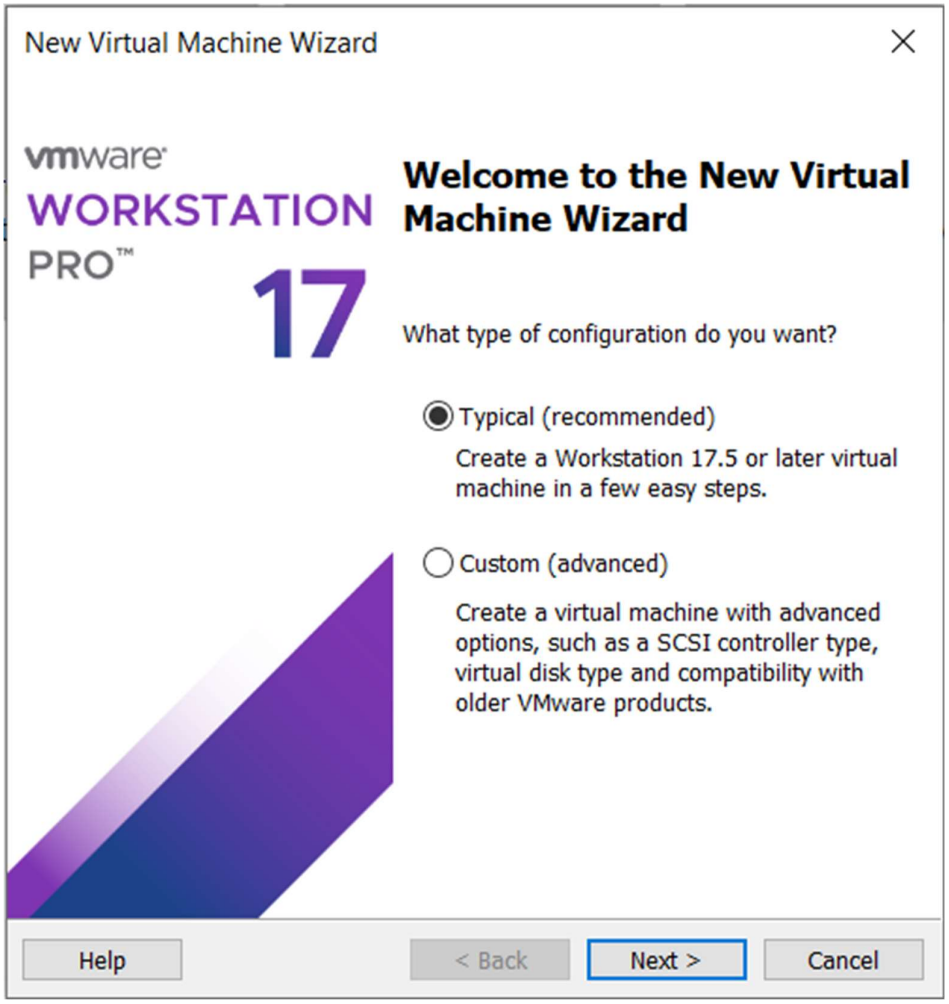
For the installation of VMware Workstation, I visited the official webpage of VMware and downloaded the .exe file that was used for installing VMware on a Windows machine.



b) Creating Virtual Machine of Linux (Alma Linux 9).

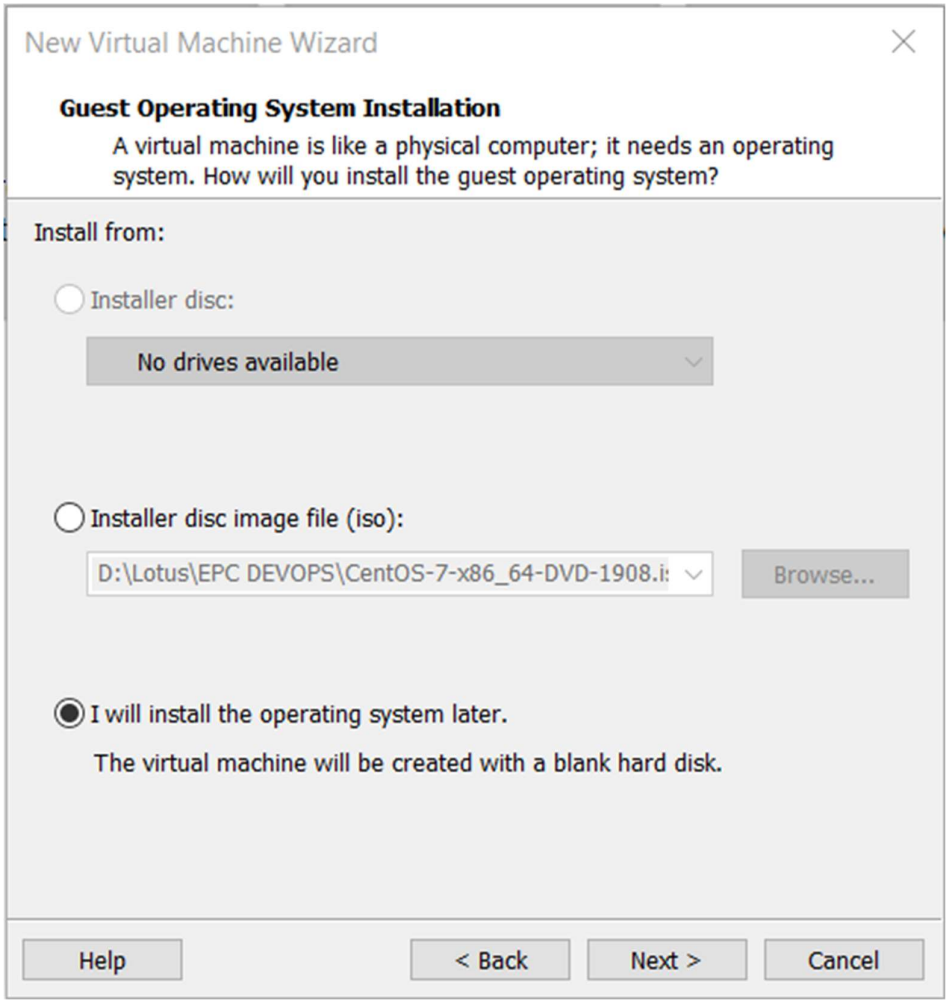
For the creation of a Virtual Machine of Alma Linux 9, the following steps were taken:

1. Firstly, the option for creation of a new virtual machine was selected, and a new Virtual Machine Wizard opened.

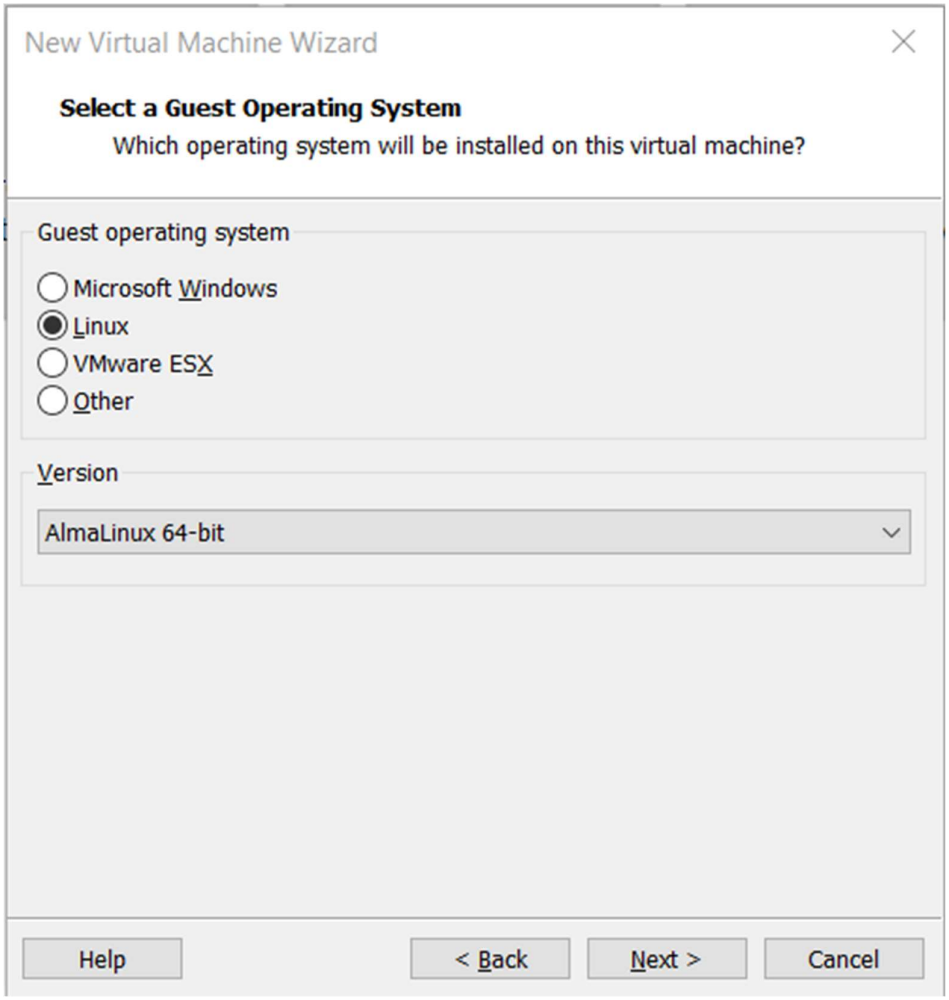


2. Then, the typical installation was selected, and the option stating I will install the

operating system later was also selected.

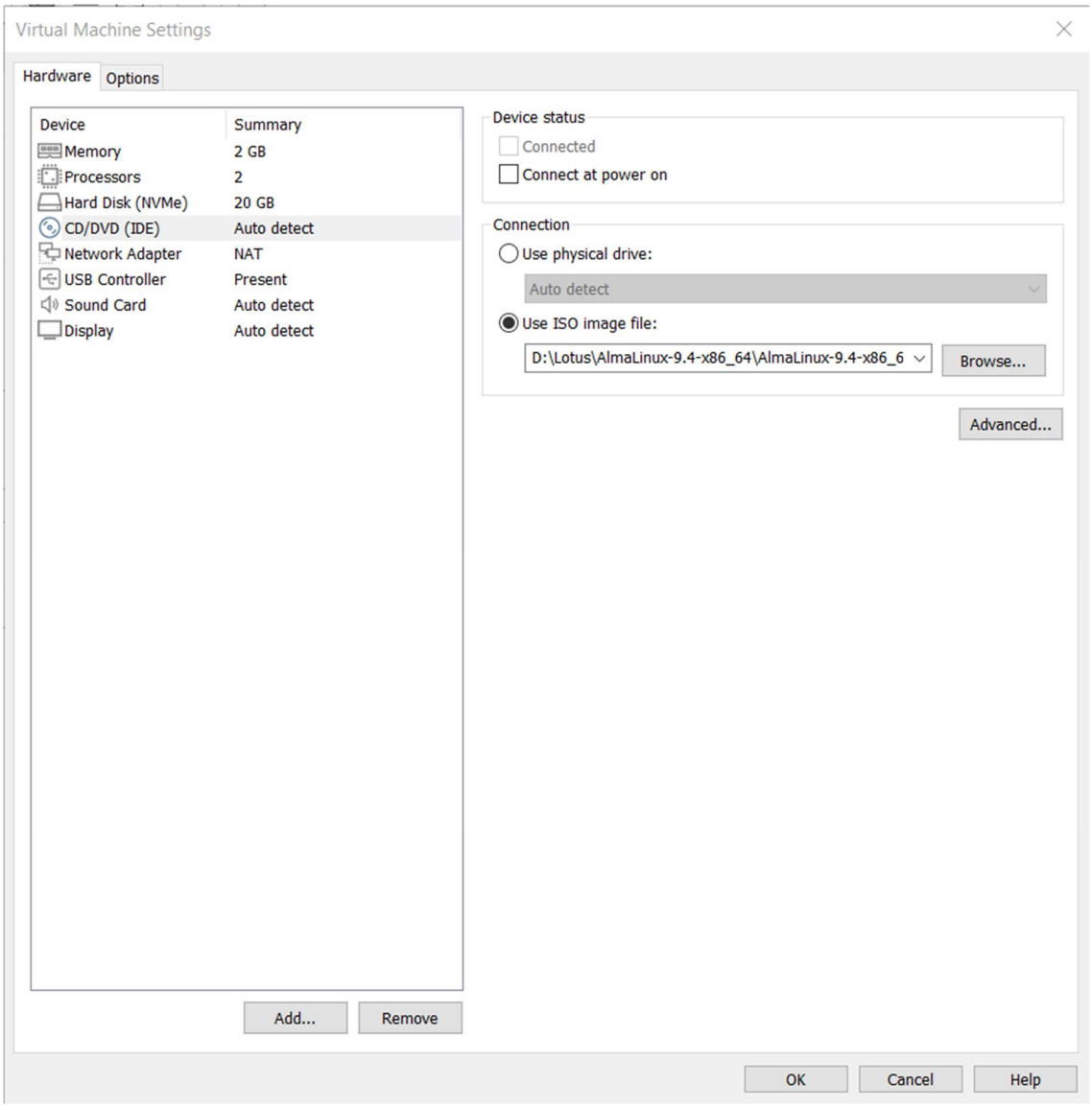


3. Alma Linux 64-bit was chosen as the guest operating system and then the location for virtual machine was selected.

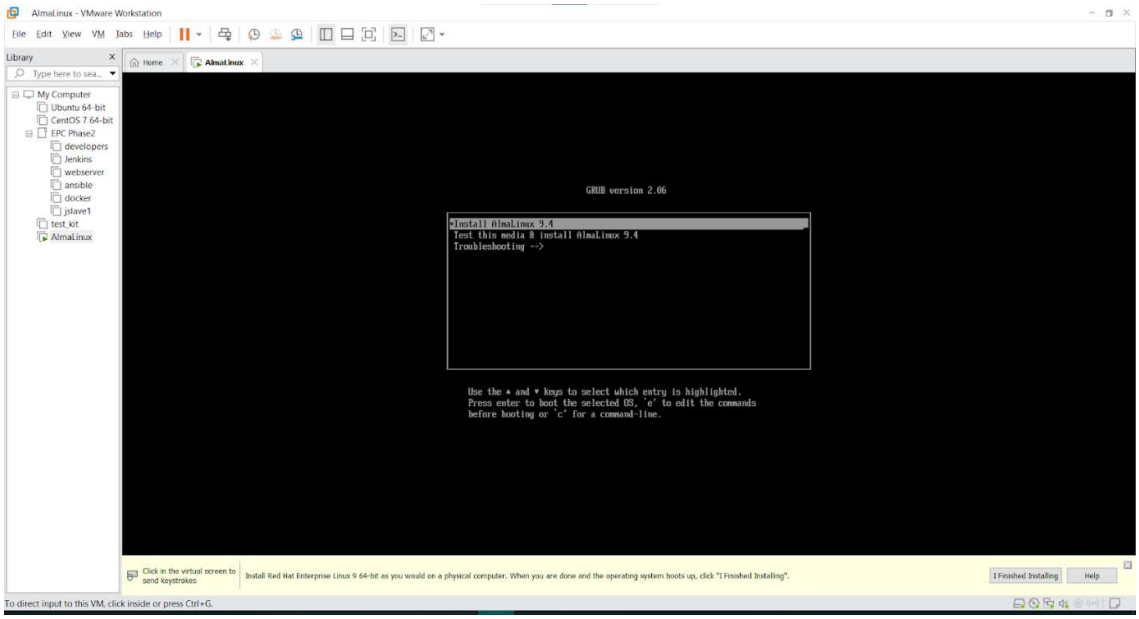


4.After that, the virtual machine storage type was chosen as multiple files and after verifying the installation details, the wizard was closed.

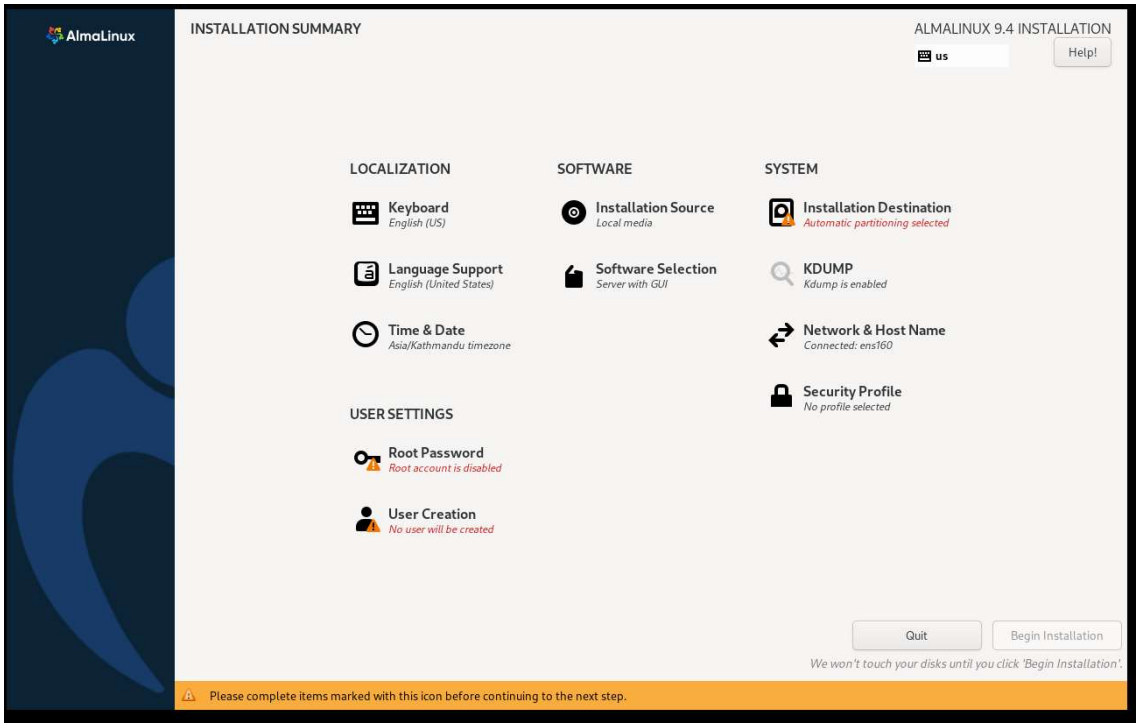
5. After that, the virtual machine was edited to use the ISO of Alma Linux for the boot.

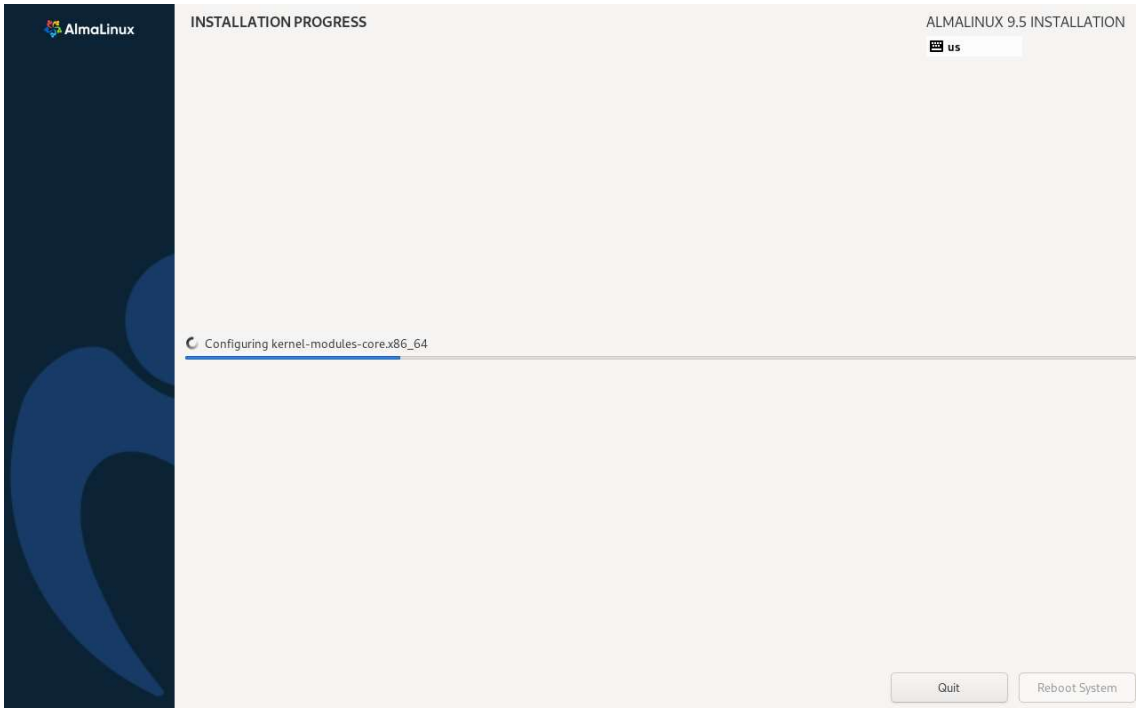


6. Then the machine was powered on and the option to install AlmaLinux 9.4 was selected.



7.After that, the installation summary appears, after checking and verifying all the details, the installation was started.

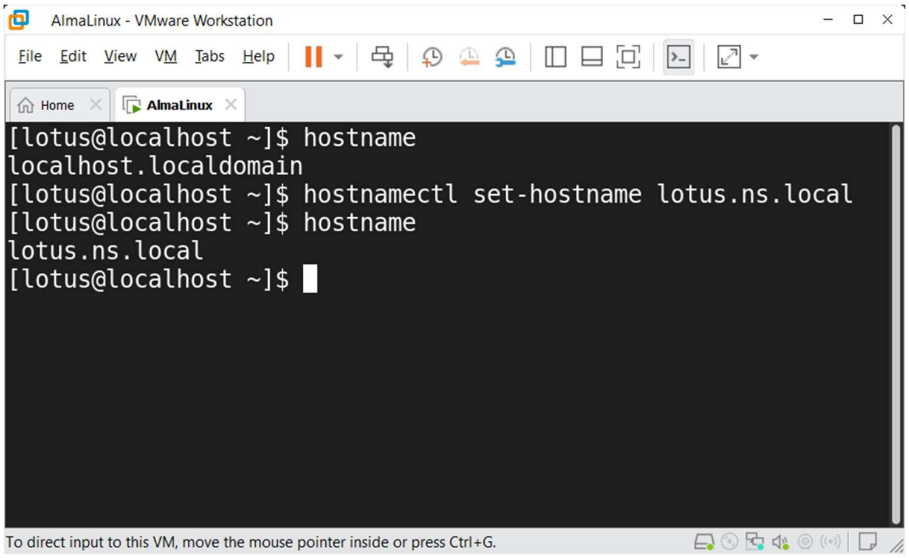




8. Then, on reboot, we are prompted to create a user for the machine, after which the installation is completed.

c) Assign hostname of the virtual machine to lotus.ns.local.

To assign the hostname in Linux to a user defined one a simple command is used: hostnamectl set-hostname <new_hostname>



The change will be visible properly after a system reboot.

```
[lotus@lotus ~]$ hostname
lotus.ns.local
[lotus@lotus ~]$
```

d) Configure your network with static IP address.

A static IP address is a fixed, unchanging IP assigned to a device or machine in a network, as opposed to a dynamic IP that can change over time. It is often used in scenarios where consistent access is required, such as servers, printers, or virtual machines, as it ensures a device remains reachable at the same address. Configuring a

static IP helps streamline communication, avoid conflicts, and improve network management, especially in environments that rely on stable, long-term connections.

To configure static IP address, multiple methods can be used such as changing the network interface file, using the nmcli command line and nmuti as well. Here, nmcli method is used.

To configure the static IP, three main things must be observed, network gateway, network mask and network interface used for the virtual machine.

These can be done easily using the commands, 'route -n' to get the gateway and mask

for the network connection, 'nmcli connection show' to get the network interface device and finally 'hostname -I' to get the IP address of the machine initially.

```
[lotus@lotus ~]$ nmcli connection show
NAME    UUID                                  TYPE      DEVICE
ens160  bbd0f63d-0585-31e8-a8d3-fc9a4630bc7a  ethernet  ens160
lo      1cfcc2f7-56b8-4fe0-9af3-34015cff72f1  loopback  lo
[lotus@lotus ~]$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.62.2    0.0.0.0         UG    100    0        0 ens160
192.168.62.0    0.0.0.0         255.255.255.0   U     100    0        0 ens160
[lotus@lotus ~]$ hostname -I
192.168.62.129
[lotus@lotus ~]$ S
```

After this, we can use the following command to set the static IP address:

nmcli connection modify <connection_name> ipv4.method manual

ipv4.addresses <static_ip_address>/<prefix> ipv4.gateway <gateway>

ipv4.dns <dns1>,<dns2> ipv6.method ignore

```
[lotus@lotus ~]$ nmcli connection modify ens160 ipv4.method manual ipv4.addresses 192.168.62
.200/24 ipv4.gateway 192.168.62.2 ipv4.dns 192.168.62.2 ipv6.method ignore
[lotus@lotus ~]$ nmcli connection down
apath     ens160    filename  help      id        lo        path      uuid
[lotus@lotus ~]$ nmcli connection down ens160
Connection 'ens160' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkMan
ager/ActiveConnection/2)
[lotus@lotus ~]$ nmcli connection up ens160
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/Active
Connection/3)
[lotus@lotus ~]$ ping.google.com -c 2
bash: ping.google.com: command not found...
[lotus@lotus ~]$ ping google.com -c 2
PING google.com (142.250.207.238) 56(84) bytes of data.
64 bytes from del12s11-in-f14.1e100.net (142.250.207.238): icmp_seq=1 ttl=128 time=41.3 ms
64 bytes from del12s11-in-f14.1e100.net (142.250.207.238): icmp_seq=2 ttl=128 time=42.7 ms

--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 41.302/41.999/42.697/0.697 ms
[lotus@lotus ~]$
```

After setting the static IP the connection was refreshed by powering the device on and

off, and the connection was checked using ping command.

e) Map your static IP address to your hostname in configuration file at /etc/hosts.

Initially, the hosts file contains mapping to the internal network only as:

```
[root@lotus lotus]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
[root@lotus lotus]#
```

To map the IP address to the hostname, the file was edited, and the new host was added.

```
[root@lotus lotus]# vi /etc/hosts
[root@lotus lotus]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.62.200 lotus.ns.local

[root@lotus lotus]# ping lotus.ns.local
PING lotus.ns.local (192.168.62.200) 56(84) bytes of data.
64 bytes from lotus.ns.local (192.168.62.200): icmp_seq=1 ttl=64 time=0.033 ms
64 bytes from lotus.ns.local (192.168.62.200): icmp_seq=2 ttl=64 time=0.049 ms
64 bytes from lotus.ns.local (192.168.62.200): icmp_seq=3 ttl=64 time=0.148 ms
64 bytes from lotus.ns.local (192.168.62.200): icmp_seq=4 ttl=64 time=0.084 ms
64 bytes from lotus.ns.local (192.168.62.200): icmp_seq=5 ttl=64 time=0.050 ms
64 bytes from lotus.ns.local (192.168.62.200): icmp_seq=6 ttl=64 time=0.048 ms
^C
--- lotus.ns.local ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5103ms
rtt min/avg/max/mdev = 0.033/0.068/0.148/0.038 ms
[root@lotus lotus]# S
```

Then to verify the added host, the new hostname was pinged, and we can verify that our static IP was the address pinged.

Task 2

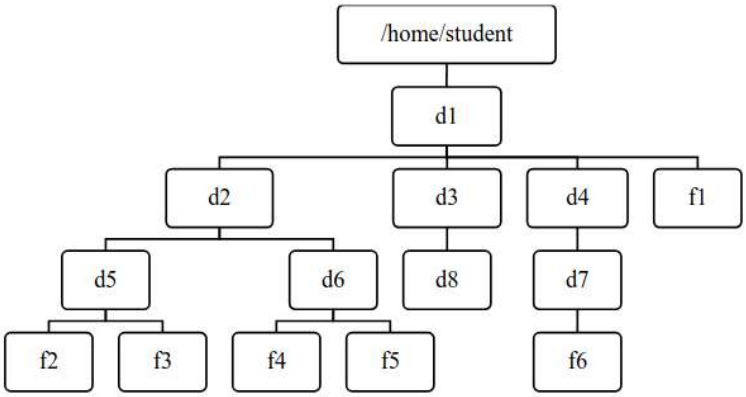
Users, Groups and Permission

a) Create a user named student.

To create a new user in Linux, a simple command ‘adduser <name>’ is used. After the creation of new user, they are listed in the ‘/etc/passwd’ file. After creation of user the password token must also be set for the user, which is done by the command ‘passwd <name>’, after which the p which the hash of the entered password and the aging information for the user is stored in the file ‘/etc/shadow’.

```
[root@lotus lotus]# adduser student
[root@lotus lotus]# cat /etc/passwd | grep student
student:x:1001:1001::/home/student:/bin/bash
[root@lotus lotus]# passwd student
Changing password for user student.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@lotus lotus]# cat /etc/passwd | grep student
student:x:1001:1001::/home/student:/bin/bash
[root@lotus lotus]# cat /etc/shadow | grep student
student:$6$rounds=100000$XBoKUZJnUBIB4mhQ$DTc9oF9noFJA7Yb5qc38daMs/P2sUkoWQuJfcriS/ZhB3WpsVA
bqCrr9/OGfEBEyW4fKKrAwDbI/2OxY4QnAz1:20118:0:99999:7:::
[root@lotus lotus]# s
```

b) Login from the student user and then create files and folders according to the following structure. (where d refers to directory and f refers to file)



To login from the user student a simple command ‘su student’ is used, since we were previously working from the root user, we were not asked to enter the password for the user student.

```
[root@lotus lotus]# whoami
root
[root@lotus lotus]# su student
[student@lotus lotus]$ whoami
student
[student@lotus lotus]$
```

After that for creation of the folder a command ‘mkdir <name>’ is used and to create a new empty file a command ‘touch <name>’ is used. The use of ‘-p’ flag with the ‘mkdir’ command adds the parent directory if not present as well.

For the creation of the specified structure we can perform the following operations:

mkdir -p /home/student/d1/{d2/{d5,d6},d3/d8,d4/d7}

touch /home/student/d1/{f1,d2/d5/{f2,f3},d2/d6/{f4,f5},d4/d7/f6}

```
[student@lotus lotus]$ mkdir -p /home/student/d1{d2/{d5,d6},d3/d8,d4/d7}
[student@lotus lotus]$ touch /home/student/d1{f1,d2/d5/{f2,f3},d2/d6/{f4,f5},d4/d7/f6}
[student@lotus lotus]$ SS
```

After which to verify the creation of the structure, we used the ’tree’ command as:

```
[student@lotus lotus]$ mkdir -p /home/student/d1/{d2/{d5,d6},d3/d8,d4/d7}
[student@lotus lotus]$ touch /home/student/d1/{f1,d2/d5/{f2,f3},d2/d6/{f4,f5},d4/d7/f6}
[student@lotus lotus]$ tree /home/student/d1/
/home/student/d1/
├── d2
│   ├── d5
│   │   ├── f2
│   │   └── f3
│   └── d6
│       ├── f4
│       └── f5
├── d3
│   └── d8
├── d4
│   └── d7
│       └── f6
└── f1

7 directories, 6 files
[student@lotus lotus]$ S
```

c) Change the permission of the file f1 so that the owner will get full permission, group member will get read and execute permission and others will get read-only permissions.

To change the permission of any file a simple command ‘chmod <new permission> <filename>’ is used. To view the permission for any file a flag ‘-l’ is used with ‘ls’ as ‘ls -l’. The permission is listed in the form of:

t uuu ggg ooo

Where, the first bit t refers to the type of file, which may be a file denoted by ‘-’, a directory denoted as ‘d’, links as ‘l’ and so on. The next three bits refer to the permission for the user

who owns the file, the next three bit for the users in the group of the file owner, and finally the last three bit refers to the permission for other users.

Here, for the d1

```
[student@lotus ~]$ ls -l
total 0
drwxr-xr-x. 5 student student 46 Jan 30 07:28 d1
```

The item is a directory, owned by the user student and belonging to group student. The owner has permission to read, write and execute the directory, the group has permission to read and execute the directory and others also have permission to read and execute.

Now to change the permission of file f1:

```
[student@lotus ~]$ ls -l d1/f1
-rw-r--r--. 1 student student 0 Jan 30 07:28 d1/f1
[student@lotus ~]$ chmod u=rwx,g=rx,o=r d1/f1
[student@lotus ~]$ ls -l d1/f1
-rwxr-xr--. 1 student student 0 Jan 30 07:28 d1/f1
[student@lotus ~]$
```

d) Change permission of the file f2 such that the owner's and group members will get read and write permission, but others will get no permission.

```
[student@lotus ~]$ ls -l d1/d2/d5/f2
-rw-r--r--. 1 student student 0 Jan 30 07:28 d1/d2/d5/f2
[student@lotus ~]$ chmod u=rw,g=rw,o= d1/d2/d5/f2
[student@lotus ~]$ ls -l d1/d2/d5/f2
-rw-rw----. 1 student student 0 Jan 30 07:28 d1/d2/d5/f2
```

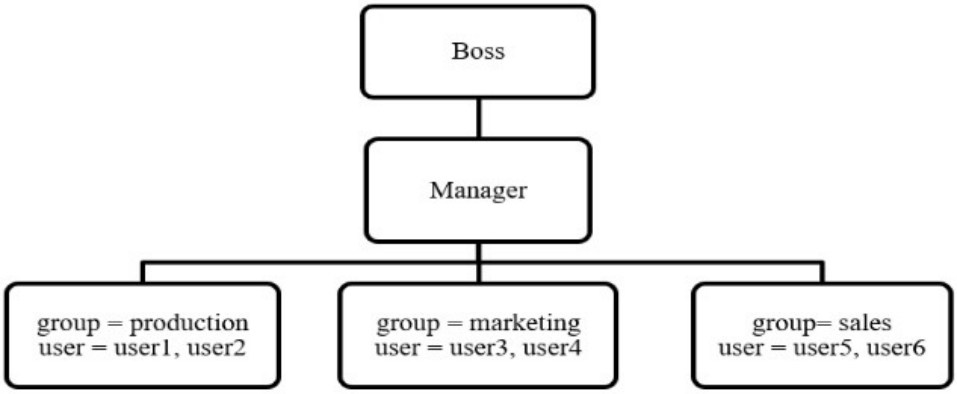
e) Change permission of directory d3 such that all categories of users will get full permissions.

```
[student@lotus ~]$ ls -l d1 | grep d3
drwxr-xr-x. 3 student student 16 Jan 30 07:27 d3
[student@lotus ~]$ chmod 77 d1/d3
[student@lotus ~]$ ls -l d1 | grep d3
d---rwxrwx. 3 student student 16 Jan 30 07:27 d3
```

Task 3

User and Group Administration

All tasks below are based on the following structure.



a) Create group for each department (production, marketing, sales)

First, the command to add a group is 'groupadd <name>', which can be verified by viewing the file '/etc/groups'.

```
[root@lotus ~]# groupadd production
[root@lotus ~]# groupadd marketing
[root@lotus ~]# groupadd sales
[root@lotus ~]# cat /etc/group | grep sales
sales:x:1004:
[root@lotus ~]# cat /etc/group | grep marketing
marketing:x:1003:
[root@lotus ~]# cat /etc/group | grep production
production:x:1002:
[root@lotus ~]# S
```

b) Create user account (user1, user2, user3, user4, user5, user6, manager, boss) for each employee assigning them respective group.

To add a user to a defined group, we can use flag '-G' while creating a user as 'useradd -G <group_name> <name>'.

```
[root@lotus ~]# useradd -m -G production user1
[root@lotus ~]# useradd -m -G production user2
[root@lotus ~]# useradd -m -G marketing user3
[root@lotus ~]# useradd -m -G marketing user4
[root@lotus ~]# useradd -m -G sales user5
[root@lotus ~]# useradd -m -G sales user6
[root@lotus ~]# useradd manager
[root@lotus ~]# useradd boss
```

To check the defined groups that a specific user belongs to, we can use the command 'groups <user_name>'.

```
[root@lotus ~]# groups user1
user1 : user1 production
[root@lotus ~]# groups user2
user2 : user2 production
[root@lotus ~]# groups user3
user3 : user3 marketing
[root@lotus ~]# groups user4
user4 : user4 marketing
[root@lotus ~]# groups user5
user5 : user5 sales
[root@lotus ~]# groups user6
user6 : user6 sales
[root@lotus ~]# groups manager
manager : manager
[root@lotus ~]# groups boss
boss : boss
```

c) Create common directory (/root/production, /root/marketing and /root/sales) for each department.

```
[root@lotus ~]# mkdir /root/production
[root@lotus ~]# mkdir /root/marketing
[root@lotus ~]# mkdir /root/sales
[root@lotus ~]# ls /root
anaconda-ks.cfg  Documents  marketing  Pictures    Public  Templates
Desktop          Downloads  Music      production  sales   Videos
```

d) Change ownership of group directories such that boss will become the owner and the respective groups will be group owner.

To change the ownership of a file and directory, we use the simple command 'chown <owner>:<group> <item_name>', which can be visible by using the command 'ls-l'.

```
[root@lotus ~]# chown boss:marketing /root/marketing/
[root@lotus ~]# chown boss:production /root/production/
[root@lotus ~]# chown boss:sales /root/sales/
[root@lotus ~]# ls -l /root/
total 4
-rw-------. 1 root root       1012 Jan 30 06:41 anaconda-ks.cfg
drwxr-xr-x. 2 root root          6 Jan 30 08:00 Desktop
drwxr-xr-x. 2 root root          6 Jan 30 08:00 Documents
drwxr-xr-x. 2 root root          6 Jan 30 08:00 Downloads
drwxr-xr-x. 2 boss marketing     6 Jan 30 08:06 marketing
drwxr-xr-x. 2 root root          6 Jan 30 08:00 Music
drwxr-xr-x. 2 root root          6 Jan 30 08:00 Pictures
drwxr-xr-x. 2 boss production    6 Jan 30 08:06 production
drwxr-xr-x. 2 root root          6 Jan 30 08:00 Public
drwxr-xr-x. 2 boss sales         6 Jan 30 08:06 sales
```

e) Change the permission of the group directories such that only the owner and group member will get full permission and other will not get any permission.

Only the owner (boss) and group members have full permissions (read, write, execute), and others have no permissions.

```
[root@lotus ~]# chmod 770 /root/production/
[root@lotus ~]# chmod 770 /root/sales/
[root@lotus ~]# chmod 770 /root/marketing/
[root@lotus ~]# ls -l /root/
total 4
-rw-------. 1 root root       1012 Jan 30 06:41 anaconda-ks.cfg
drwxr-xr-x. 2 root root          6 Jan 30 08:00 Desktop
drwxr-xr-x. 2 root root          6 Jan 30 08:00 Documents
drwxr-xr-x. 2 root root          6 Jan 30 08:00 Downloads
drwxrwx---. 2 boss marketing     6 Jan 30 08:06 marketing
drwxr-xr-x. 2 root root          6 Jan 30 08:00 Music
drwxr-xr-x. 2 root root          6 Jan 30 08:00 Pictures
drwxrwx---. 2 boss production    6 Jan 30 08:06 production
drwxr-xr-x. 2 root root          6 Jan 30 08:00 Public
drwxrwx---. 2 boss sales         6 Jan 30 08:06 sales
```

f) Set SGID and sticky bits on the departmental directories. SGID ensures that files created in the directory inherit the group of the directory, and the sticky bit ensures that only the owner of a file can delete it.

```
[root@lotus ~]# chmod g+s /root/production/
[root@lotus ~]# chmod g+s /root/sales/
[root@lotus ~]# chmod g+s /root/marketing/
[root@lotus ~]# chmod +t /root/marketing/
[root@lotus ~]# chmod +t /root/sales/
[root@lotus ~]# chmod +t /root/production/
[root@lotus ~]# ld -ld /root/
ld: cannot find -ld
ld: read in flex scanner failed
[root@lotus ~]# ls -ld /root/
dr-xr-x---. 17 root root 4096 Jan 30 08:15 /root/
[root@lotus ~]# ls -ld /root/*
-rw-------. 1 root root       1012 Jan 30 06:41 /root/anaconda-ks.cfg
drwxr-xr-x. 2 root root          6 Jan 30 08:00 /root/Desktop
drwxr-xr-x. 2 root root          6 Jan 30 08:00 /root/Documents
drwxr-xr-x. 2 root root          6 Jan 30 08:00 /root/Downloads
drwxrws--T. 2 boss marketing     6 Jan 30 08:06 /root/marketing
drwxr-xr-x. 2 root root          6 Jan 30 08:00 /root/Music
drwxr-xr-x. 2 root root          6 Jan 30 08:00 /root/Pictures
drwxrws--T. 2 boss production    6 Jan 30 08:06 /root/production
drwxr-xr-x. 2 root root          6 Jan 30 08:00 /root/Public
drwxrws--T. 2 boss sales         6 Jan 30 08:06 /root/sales
```

g) Assign special permission (ACL) to anonymous called david such that it can see

what's inside the common directory for sales group i.e., /root/sales.

```
[root@lotus ~]# useradd -m david
[root@lotus ~]# setfacl -m u:david:rx /root/sales/
[root@lotus ~]# getfacl /root/sales/
getfacl: Removing leading '/' from absolute path names
# file: root/sales/
# owner: boss
# group: sales
# flags: -st
user::rwx
user:david:r-x
group::rwx
mask::rwx
other::---
```

Task 4

Firewall configuration in Linux

a) Install firewalld package as well as start and enable firewall services.

To install any package in Alma Linux, simple command 'yum install <package_name>' is used.

```
[root@lotus ~]# sudo yum install firewalld -y
Last metadata expiration check: 1:20:07 ago on Thu 30 Jan 2025 07:01:44 AM
 +0545.
Package firewalld-1.3.4-7.el9.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

To enable, and start the package, 'systemctl' command is used in conjunction with the service name and status is observed through systemctl status <serive_name>:

```
[root@lotus ~]# systemctl enable firewalld.service
[root@lotus ~]# systemctl start firewalld.service
[root@lotus ~]# systemctl status firewalld.service
● firewalld.service - firewalld - dynamic firewall daemon
     Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; >
     Active: active (running) since Thu 2025-01-30 07:59:43 +0545; 24min >
       Docs: man:firewalld(1)
   Main PID: 812 (firewalld)
      Tasks: 2 (limit: 10740)
     Memory: 36.8M
        CPU: 870ms
     CGroup: /system.slice/firewalld.service
             └─812 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nop>
```

b) Add the following services and ports to allow packets through the firewall. [Service =http, smtp port = 25 /tcp, 25/udp, 110/tcp]

To add anything to the firewall, we use the 'firewall-cmd' command interface. To add a service, we use the command 'sudo firewall-cmd --permanent --add-service=<name>' and similary for ports we use 'sudo firewall-cmd --permanent --add- port=<port>/<protocol>'.

```
[root@lotus ~]# sudo firewall-cmd --permanent --add-service=http
success
[root@lotus ~]# sudo firewall-cmd --permanent --add-port=25/tcp
success
[root@lotus ~]# sudo firewall-cmd --permanent --add-port=25/udp
success
[root@lotus ~]# sudo firewall-cmd --permanent --add-port=110/tcp
success
```

For the changes to take place, first we need to reload the firewall, then we can view the permissions in firewall by listing all using 'firewall-cmd --list-all', which lists all the permissions in the default, and active zone.

```
[root@lotus ~]# firewall-cmd --reload
success
[root@lotus ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens160
  sources:
  services: cockpit dhcpv6-client http ssh
  ports: 25/tcp 25/udp 110/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

c) Remove the following services and ports to block packets through the firewall.[Service = smtp port = 25 /tcp, 25/udp]

To remove any service or port the flag '--remove-service=<name>' or '--remove-port=<port>/<protocol>' is used in conjunction with 'firewall-cmd'.

```
[root@lotus ~]# firewall-cmd --permanent --remove-port=25/tcp
success
[root@lotus ~]# firewall-cmd --permanent --remove-port=25/udp
success
```

```
[root@lotus ~]# firewall-cmd --reload
success
[root@lotus ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens160
  sources:
  services: cockpit dhcpv6-client http ssh
  ports: 110/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Task 5

Configuring SSH Server to allow/deny root login and allow/deny users login

a) Install required package for OpenSSH server

To install any package in Alma Linux, simple command 'yum install <package_name>' is used. For OpenSSH Server, we simply install the package named 'openssh-server'.

```
[root@lotus ~]# yum install -y openssh-server
Last metadata expiration check: 2:19:47 ago on Thu 30 Jan 2025 07:01:44
AM +0545.
Package openssh-server-8.7p1-43.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

b) Allow ssh packets to enter through the firewall

To add anything to the firewall, we use the 'firewall-cmd' command interface. To allow ssh packets to enter through the firewall, we can simply use the command, 'firewall-cmd --permanent --add-service=ssh', and then by reloading the firewall

```
[root@lotus ~]# firewall-cmd --add-service=ssh --permanent
Warning: ALREADY_ENABLED: ssh
success
[root@lotus ~]# firewall -cmd --reload
bash: firewall: command not found...
[root@lotus ~]# firewall-cmd --reload
success
[root@lotus ~]#
```

c) Start and enable ssh service

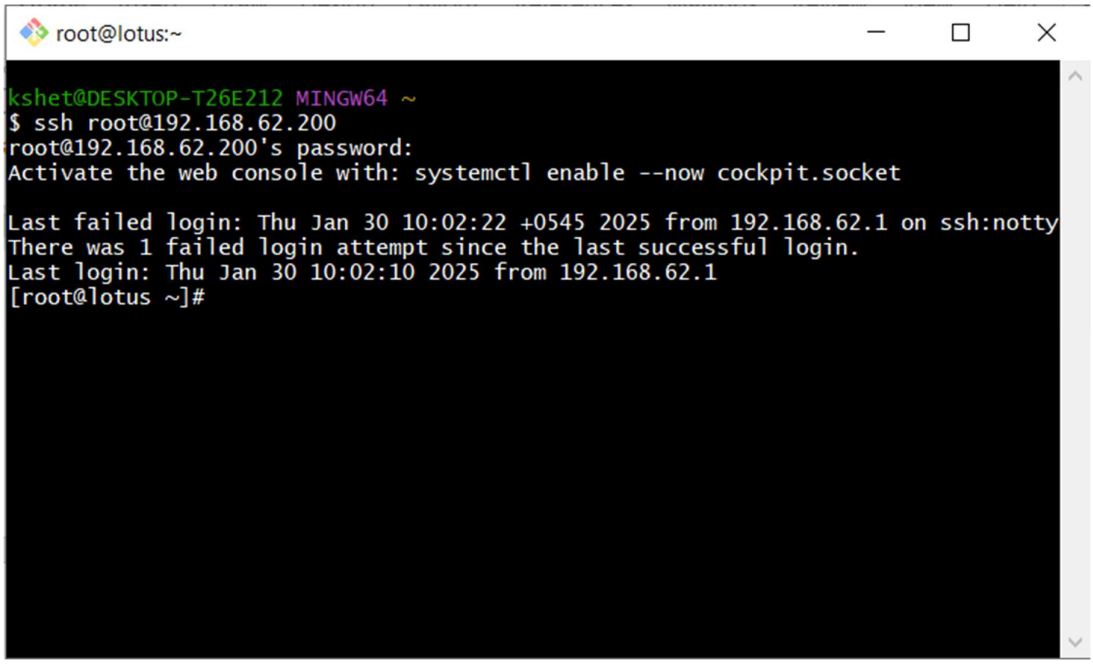
OpenSSH Server works as a service named 'sshd', whose status can be viewed by using the command 'systemctl status sshd'. To enable it, we can simply use the command as 'systemctl enable sshd --now', which starts the sshd service at the same time as well as creates a symlink such that the sshd service starts on every boot.

```
[root@lotus ~]# systemctl enable sshd --now
[root@lotus ~]# systemctl status sshd
● sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; prese>
     Active: active (running) since Thu 2025-01-30 09:31:44 +0545; 2min 1>
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 962 (sshd)
      Tasks: 1 (limit: 22800)
     Memory: 2.6M
        CPU: 22ms
     CGroup: /system.slice/sshd.service
             └─962 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startu>

Jan 30 09:31:44 lotus.ns.local systemd[1]: Starting OpenSSH server daemon>
Jan 30 09:31:44 lotus.ns.local sshd[962]: Server listening on 0.0.0.0 por>
Jan 30 09:31:44 lotus.ns.local sshd[962]: Server listening on :: port 22.
Jan 30 09:31:44 lotus.ns.local systemd[1]: Started OpenSSH server daemon.
```

d) Configure OpenSSH server to deny direct root login

To configure OpenSSH server to deny direct root login, on Alma Linux 9, we can go to the path /etc/ssh/ and then modify the file present there named sshd_config, and set the flag named ‘PermitRootLogin’ to ‘no’ so we can deny direct root login.

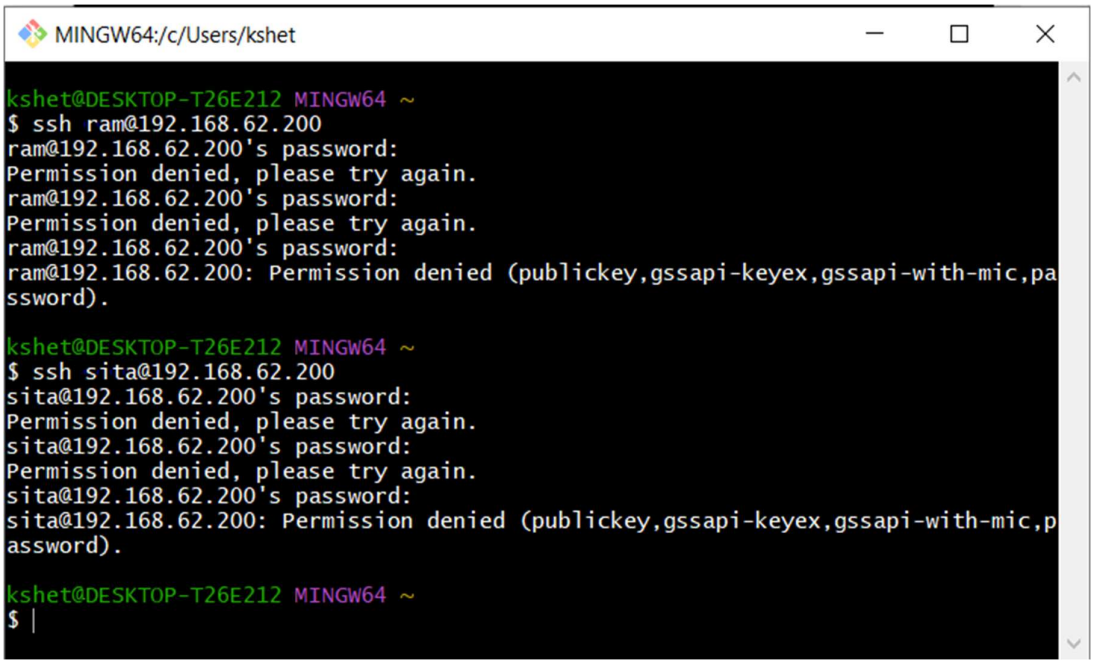
```
[root@lotus ssh]# vi sshd_config
[root@lotus ssh]# sudo systemctl restart sshd
[root@lotus ssh]# service sshd restart
Redirecting to /bin/systemctl restart sshd.service
[root@lotus ssh]# sudo grep -i "PermitRootLogin" /etc/ssh/sshd_config
#PermitRootLogin prohibitpassword
#PermitRootLogin no
# the setting of "PermitRootLogin without-password".
```



e) Configure OpenSSH Server to block login from users named ram and sita.

To deny logins from specific user into the server, we can again modify the same 'sshd_config' file to add a block of 'DenyUsers' as follows:

```
[root@lotus ssh]# vi sshd_config
[root@lotus ssh]# tail -n 3 sshd_config

#DenyUsers Block
DenyUsers ram sita
[root@lotus ssh]#
```


MINGW64:/c/Users/kshet
kshet@DESKTOP-T26E212 MINGW64 ~
$ ssh ram@192.168.62.200
ram@192.168.62.200's password:
Permission denied, please try again.
ram@192.168.62.200's password:
Permission denied, please try again.
ram@192.168.62.200's password:
ram@192.168.62.200: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,pa
ssword).
kshet@DESKTOP-T26E212 MINGW64 ~
$ ssh sita@192.168.62.200
sita@192.168.62.200's password:
Permission denied, please try again.
sita@192.168.62.200's password:
Permission denied, please try again.
sita@192.168.62.200's password:
sita@192.168.62.200: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,p
assword).
kshet@DESKTOP-T26E212 MINGW64 ~
$

Task 6

Configuring SSH Server to allow/deny SSH login from selected hosts only

a) Configure OpenSSH server to deny all hosts except the host machine.

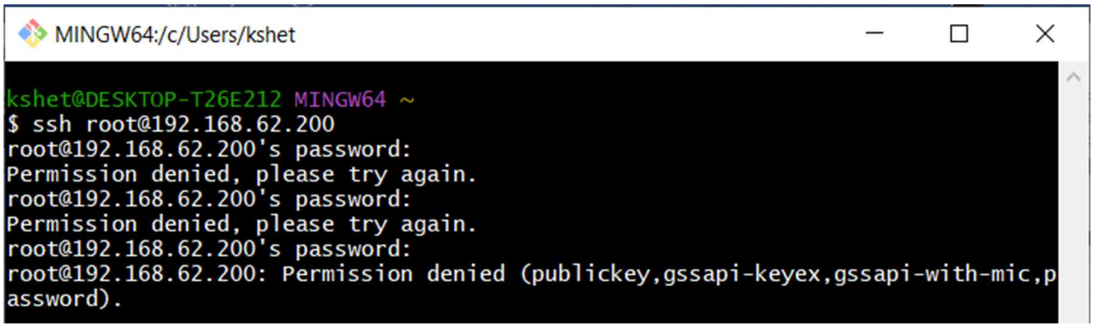
To configure OpenSSH server to deny all hosts except the host machine itself, we can simply add a entry named 'AllowUsers' in the /etc/ssh/sshd_config file. This entry overrides any and all other configuration and allows ssh into the machine from the user and address defined in the entry only.

Here, I have not defined any user and limited the address for AllowUser to 127.0.0.1, so all users trying to ssh with that address were allowed, but all other tried were stopped.

First, I updated the sshd_config file by adding the following entry at the bottom of the file as:

```
[root@lotus ssh]# pwd
/etc/ssh
[root@lotus ssh]# vi sshd_config
[root@lotus ssh]# tail -n 3 sshd_config

#Allow Virtual machine only
AllowUsers *@127.0.0.1
[root@lotus ssh]# service sshd restart
Redirecting to /bin/systemctl restart sshd.service
[root@lotus ssh]# sudo systemctl restart sshd
[root@lotus ssh]# S
```

Then, we try to ssh into the machine from the host device itself as:



But from inside the virtual machine itself:

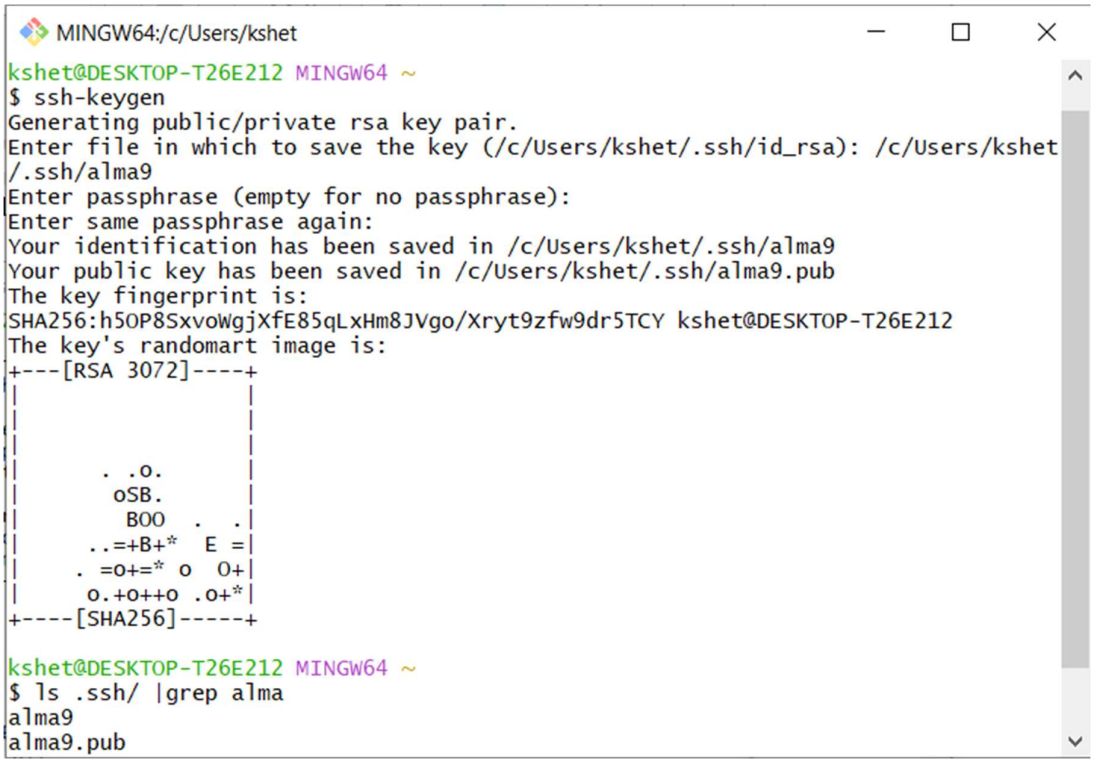
```
[root@lotus ssh]# ssh root@127.0.0.1
The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:Qns2M0ww0AOVqSykh+aVsERay/tS434rZh61QDGUVGs.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '127.0.0.1' (ED25519) to the list of known hosts.
root@127.0.0.1's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last failed login: Thu Jan 30 10:19:18 +0545 2025 from 192.168.62.1 on ssh:notty
There were 4 failed login attempts since the last successful login.
Last login: Thu Jan 30 10:17:21 2025 from 192.168.62.1
```

Task 7

Configuring SSH Server for direct SSH login by generating and publishing private and public key

a) Generate SSH key pair (public and private) in local host.

To generate SSH key pair in localhost, we can simply use the command, 'ssh-keygen', and then define the file where we want to save the keys as well as provide passphrases if needed.



b) Send a copy of the public key to the ssh server in which you want to direct login.

Now, to send the public key to the Alma Linux instance to start direct login, we can simply use the command 'ssh-copy-id -i public_key user@server'.

```
kshet@DESKTOP-T26E212 MINGW64 ~
$ ssh-copy-id -i .ssh/alma9.pub root@192.168.62.200
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: ".ssh/alma9.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
root@192.168.62.200's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@192.168.62.200'"
and check to make sure that only the key(s) you wanted were added.
```

Then, we try to login via ssh, we get

```
kshet@DESKTOP-T26E212 MINGW64 ~
$ ssh 'root@192.168.62.200'
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Thu Jan 30 10:39:18 2025 from 192.168.62.1
[root@lotus ~]#
```

In this way, we can configure our server for direct SSH using private and public keys.

Task 8

Secure Network Copy using “SCP”

a) Copy remote file into the local system

To copy any file to-and from a server using ssh, we can simply use the command line tool ‘scp’, its syntax is as ‘scp source destination’, and copies the file and resources found at source to destination.

Here, I copied the file named sshd_config from my remote server into local path as:

```
kshet@DESKTOP-T26E212 MINGW64 /d/Lotus/Study Materials/7th Sem/NS/NS
$ scp root@192.168.62.200:/etc/ssh/sshd_config "/d/Lotus/Study Materials/7th Sem/NS/server"
root@192.168.62.200's password:
sshd_config                                   100% 3649     1.9MB/s   00:00
```

```
kshet@DESKTOP-T26E212 MINGW64 /d/Lotus/Study Materials/7th Sem/NS/NS
$ ls "/d/Lotus/Study Materials/7th Sem/NS/server"
sshd_config
```

b) Copy local files to the remote host

To copy the file into the remote host, similar command is used, but we must make sure that the user we are logging into the server during scp has permission to work in that specific directory.

```
kshet@DESKTOP-T26E212 MINGW64 /d/Lotus/Study Materials/7th Sem/NS/NS
$ scp "NS.txt" root@192.168.62.200:/root/
root@192.168.62.200's password:
NS.txt                                        100%    0     0.0KB/s   00:00
```

So, we can use ‘scp’ command to copy files to and from a remote server using ssh as base.

Task 9

Security Enhanced Linux (SE Linux)

a) Check the current status of SE Linux

To check the current status of SE Linux, we can simply use the command 'getenforce', which then gives the status of SE Linux.

```
[root@lotus ~]# getenforce
Enforcing
[root@lotus ~]#
```

We can also use the command 'sestatus', to check the status of SE Linux as:

```
[root@lotus ~]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
```

b) Configure the server to enable (enforcing) SE Linux

If the server is not enforcing SE Linux, we can enforce it by editing the file /etc/sysconfig/selinux, and then changing the context of SE Linux to enforcing as:

```
[root@lotus ~]# grep =enforcing /etc/sysconfig/selinux
SELINUX=enforcing
```

c) Configure SELinux for a custom HTTP port 8090 and custom SSH port 4455.

First, to add a custom HTTP port into SE Linux, we can use the 'semanage' command to add a new port into the SE Linux state. For this, we need the tag which denotes which action the port enabled via SE Linux is enabled to take, for example, 'ssh_port_t' for SSH and 'http_port_t' for HTTP. We also need to add the new custom port as TCP port into the SE Linux status.

To add custom HTTP port 8090 into SE Linux context, we can simply use the command, 'sudo semanage port -a -t http_port_t -p tcp 8090', which adds a custom port 8090 for HTTP into SE Linux.

```
[root@lotus ~]# sudo semanage port -a -t http_port_t -p tcp 8090
[root@lotus ~]# sudo semanage port -l | grep 8090
http_port_t                    tcp      8090, 80, 81, 443, 488, 8008, 8009, 8443, 9000
```

Then, to add custom SSH port 4455 into SE Linux, we can simply use the command, 'sudo semanage port -a -t ssh_port_t -p tcp 4455', which adds a custom port 4455 as ssh_port_t (which is SE Linux tag for SSH).

```
[root@lotus ~]# sudo semanage port -l | grep ssh_port_t
ssh_port_t                     tcp      22
[root@lotus ~]# sudo semanage port -a -t ssh_port_t -p tcp 4455
[root@lotus ~]# sudo semanage port -l | grep ssh_port_t
ssh_port_t                     tcp      4455, 22
```

d) Change SELinux context of /mysite to httpd_sys_content_t using semanage and chcon respectively.

We can use the commands 'semanage', and 'chcon' to change the SE Linux context of any file or directory. For the purpose of this lab, we create a folder named 'mysite' at / and add a sample index.html file in it.

Then, we add the context of 'httpd_sys_content_t' in it, which is the default context of /var/www/html, the root directory of Apache web server. We can do this as: Initailly, we have the directory and its content as:

```
[root@lotus mysite]# ls
index.html
[root@lotus mysite]# ll -ldZ
drwxr-xr-x. 2 root root unconfined_u:object_r:default_t:s0 24 Jan 30 11:10 .
```

Then, we first use the command, 'semanage fcontext -a -t httpd_sys_content_t "/mysite(/.*)?"' which assignsthe context of the folder /mysite and all of its content.

After that, running the 'restorecon -Rv /mysite' command applies the new context to the directory.

```
[root@lotus mysite]# semanage fcontext -a -t httpd_sys_content_t "/mysite(/.*)?"
[root@lotus mysite]# sudo restorecon -Rv /mysite
Relabeled /mysite from unconfined_u:object_r:default_t:s0 to unconfined_u:object
_r:httpd_sys_content_t:s0
Relabeled /mysite/index.html from unconfined_u:object_r:default_t:s0 to unconfir
ed_u:object_r:httpd_sys_content_t:s0
```

We can also use the chcon command to directly change the SE Linux context of the directory as well, which can be performed by using the command, 'sudo chcon -R -t httpd_sys_content_t /mysite', which would directly assign the SE Linux context to the content of that path.

```
[root@lotus mysite]# chcon -R -t httpd_sys_content_t /mysite/
```

We can again test this by running the command 'ls -ldZ' to check the SE Linux context as:

```
[root@lotus mysite]# ls -ldZ /mysite/
drwxr-xr-x. 2 root root unconfined_u:object_r:httpd_sys_content_t:s0 24 Jan 30
 11:10 /mysite/
```

e) Change the SELinux context of /mysite2 using reference context of /var/www/html.

We can also change context of one directory directly by using another path or directory by using it as a reference using the command 'chcon' as 'sudo chcon --reference=/var/www/html -R /mysite2'

```
[root@lotus mysite2]# pwd
/mysite2
[root@lotus mysite2]# ls -ldZ
drwxr-xr-x. 2 root root unconfined_u:object_r:default_t:s0 24 Jan 30 11:15 .
[root@lotus mysite2]# sudo chcon --reference=/var/www/html -R /mysite2
[root@lotus mysite2]# ls -ldZ
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 24 Jan 30 11:
15 .
```

Task 9

Configuring SSL-Enabled Apache (HTTPS) Server (self-signed)

a) Install required package for HTTPS server (httpd, mod_ssl). Also, start and enable web service.

To install all the required package for HTTPS server, i.e httpd, mod_ssl, and openssl, we can simply use the command ‘yum install mod_ssl openssl httpd’.

```
[root@lotus mysite2]# yum install mod_ssl openssl httpd
Last metadata expiration check: 0:56:49 ago on Thu 30 Jan 2025 10:30:06 AM +05
45.
Package mod_ssl-1:2.4.62-1.el9_5.2.x86_64 is already installed.
Package openssl-1:3.2.2-6.el9_5.x86_64 is already installed.
Package httpd-2.4.62-1.el9_5.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

b) Allow https (port 443) packets to enter through the firewall.

We can simply use the firewall-cmd command to allow https packets through the firewall, rather than https as a service, we need to add http as service and then 44/tcp as port through the firewall.

```
[root@lotus mysite2]# firewall-cmd --permanent --add-port=443/tcp
success
[root@lotus mysite2]# firewall-cmd --permanent --add-service=http
Warning: ALREADY_ENABLED: http
success
[root@lotus mysite2]# firewall-cmd --reload
success
```

c) Generate self-signed key and cert files using openssl.

We can simply use openssl to generate a self-signed key and certificate file that we can use as ssl certificate, for that, we can use the command as follows:

```
[root@lotus mysite2]# openssl req -newkey rsa:2048 -nodes -keyout /etc/pki/tls/private/demo.localhost.com.key -x509 -days 365 -out /etc/pki/tls/certs/demo.local.com.crt
.....+.....+..+...+....+.+.+..+.+..+.......+....+.................+..+...++++++++++++++++++++++++++++++++++++++++*.+.............++++++++++++++++++++++++++
...+.............+..........+.........+.....+.+..........+....+...+...+.....................................+.......+...+..+.....................................+...+..+...+...
.+....+.....+......+.+..++++++
.......++++++++++++++++++++++++++++++++++++++++*..............+.......+...+...+...........+......+...+......+.+........+....+..+.+...+...+..+.+...
.....+......++++++++++++++++++++++++++++++++++++++++*...+......+........+....+......+..+.......+..+...+...+.......+..............+...........+......+.+..+.+..........+.
..........+...+...+.....+.......+..............+...+........+....+...+..+.+.....+...........+..+.............+...+...+..........+..+..+......+......+.....
....+.....................+.+.....+..........+......+.+........+......+....+..+.+........+.....+...........+...+.+..+....+...........+...+....+......+.....................
.............+..+...+......+...+..+.................+.....+....+....+...+.+.....+.+........+.......+..............+.+.................+.....+.......+.........+......+.+
+....+.....+.......+.........+...+..+..+...+...+......+....+..+.....................+.......+......+.......+..+...+...+....+.....+.......+........+.+...........+......
........+.+..+.......+..+.............+.....+...+..........+.........+.....+.+....++++++
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:
State or Province Name (full name) []:
Locality Name (eg, city) [Default City]:
Organization Name (eg, company) [Default Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:
Email Address []:
```

d) Configure web server to listen from port 443 and set DocumentRoot to "/cab/ns/mystie", locate the required key and cert files. Include the necessary SELinux configuration.

First, we need to define the domain as host entry in our /etc/hosts file, which we have previously done in the lab itself. Then, we need to define a new apache configuration file at the path /etc/httpd/conf.d, as:

```
[root@localhost demo1]# vi index.html
[root@localhost demo1]# cat index.html
<h1>Hello! I am Lotus <h1>
[root@localhost demo1]# cd /etc/httpd/
[root@localhost httpd]# ls
conf  conf.d  conf.modules.d  logs  modules  run  state
[root@localhost httpd]# cd conf.d
[root@localhost conf.d]# ls
autoindex.conf  README  ssl.conf  userdir.conf  welcome.conf
[root@localhost conf.d]# vi localhost.com.conf
[root@localhost conf.d]# cat localhost.com.conf
<VirtualHost *:443>
        DocumenRoot "/var/demo1"
        ServerName localhost.com
        <Directory "/var/demo11">
                AllowOverride All
                Require all granted
        </Directory>
</VirtualHost>
[root@localhost conf.d]#
```

The configuration file contains the ssl certificate we generated previously so that we can use https as well.

After writing the configuration, we can check the syntax of the configuration and the restart the webserver as:

```
[root@localhost /]# httpd -t
AH00558: httpd: Could not reliably determine the server's fully qualified
domain name, using localhost.localdomain. Set the 'ServerName' directive g
lobally to suppress this message
Syntax OK
[root@localhost /]# systemctl restart httpd
```

Then, we need to add the necessary SE Linux configuration to the files at /cab/ns/demo1.

e) Host a web page called index.html on web server named localhost.ns.local.

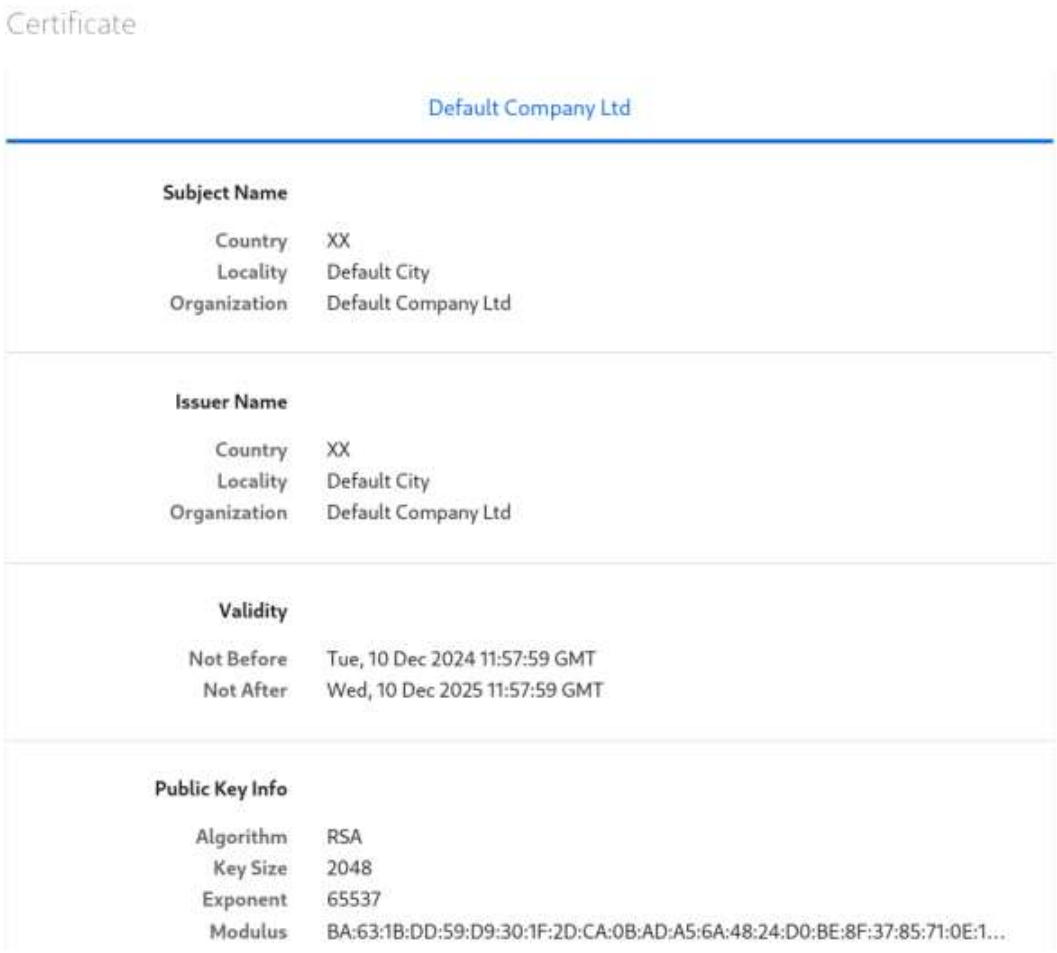
Now, to first host the website at domain, localhost.ns.local, we can add an index.html file at the path /cab/ns/mysite named index.html, and the give the directory required permission so that, the apache user can acess that location as:

The index.html file contains:

```
[root@localhost demo1]# vi index.html
[root@localhost demo1]# cat index.html
<h1>Hello! I am Lotus <h1>
[root@localhost demo1]#
```

Then, we can reload httpd server and then can visit the site from our internal browser as:

We can further inspect the certificate as:



Miscellaneous

Serial Number	51:5C:D3:1C:88:D7:38:7C:78:AD:86:40:4E:83:F1:44:63:71:90:D6
Signature Algorithm	SHA-256 with RSA Encryption
Version	3
Download	

Fingerprints

SHA-256	63:DB:D8:8C:7B:E7:BE:22:7D:90:7A:DE:8E:9D:02:C8:E7:01:A7:54:2C:9F:23:0…
SHA-1	E8:6C:75:63:E2:67:CB:1C:FF:33:85:CA:B3:83:71:26:5A:BD:A5:30

Basic Constraints

Certificate Authority	Yes

Subject Key ID

Key ID	2F:EE:0D:B1:A6:44:63:D0:DE:7C:DD:6C:14:28:99:10:08:D1:04:A8

Authority Key ID

Key ID	2F:EE:0D:B1:A6:44:63:D0:DE:7C:DD:6C:14:28:99:10:08:D1:04:A8