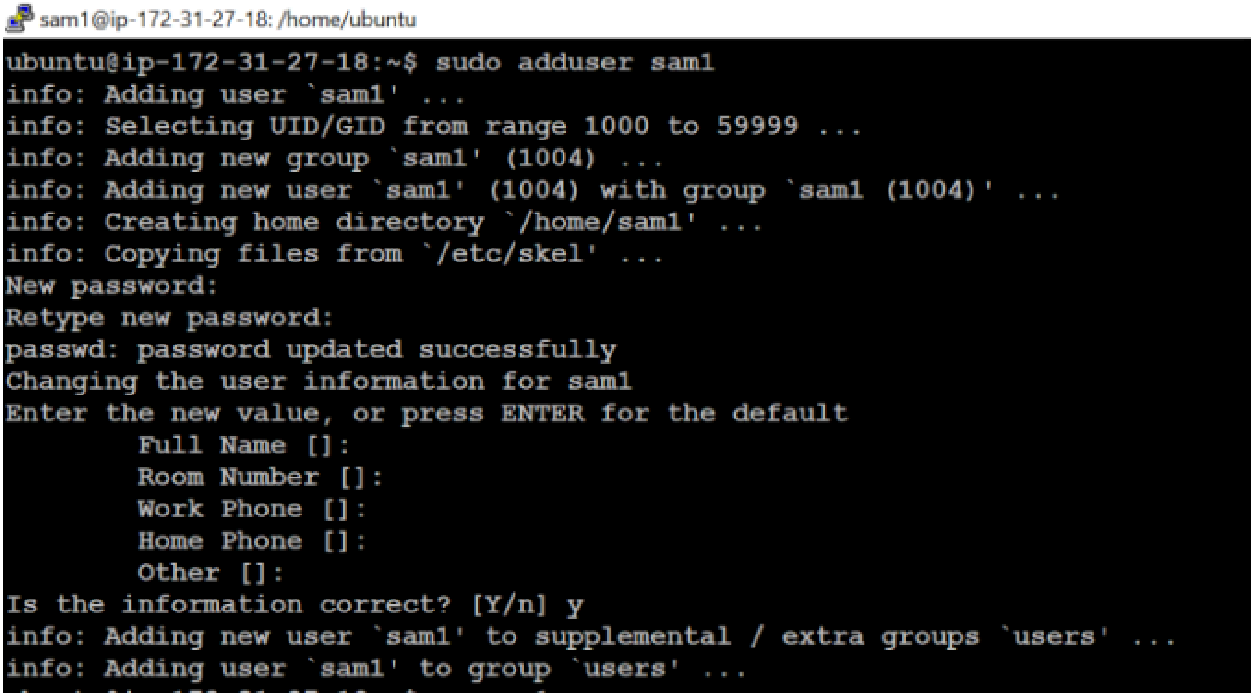
Linux Task Day-1:-

1. **Create a linux user and set a password for the user.**
2. **Switch the user and create a directory including its sub directory under the newly created user's home directory.**
3. **Create file using cat and echo command.**
4. **Change the executable file permission for the file.**
5. **Change the owner of the file.**

1. Create a linux user and set a password for the user.

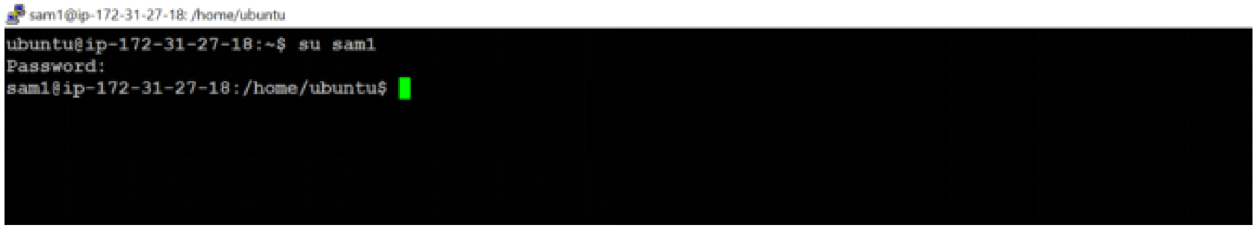
- Create the user with the **adduser** or **useradd** command and the **-m** option to create a home directory.
- Set the password for the user with the **passwd** command.



```
ubuntu@ip-172-31-27-18:~$ sudo adduser sam1
info: Adding user `sam1' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `sam1' (1004) ...
info: Adding new user `sam1' (1004) with group `sam1 (1004)' ...
info: Creating home directory `/home/sam1' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for sam1
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
info: Adding new user `sam1' to supplemental / extra groups `users' ...
info: Adding user `sam1' to group `users' ...
```

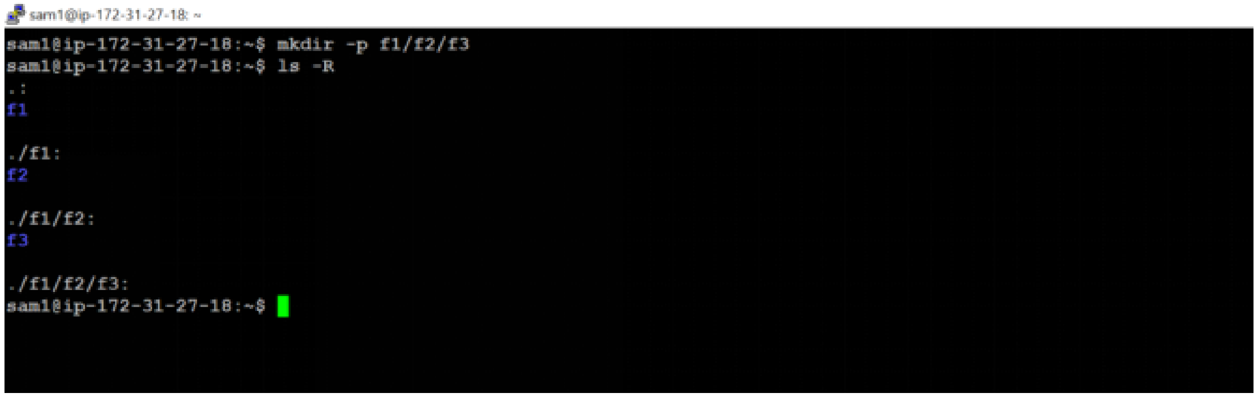
2. Switch the user and create a directory including its sub directory under the newly created user's home directory.

- The **su** command allows you to switch to another user account.



```
ubuntu@ip-172-31-27-18:~$ su sam1
Password:
sam1@ip-172-31-27-18:/home/ubuntu$
```

- ❖ create a directory and subdirectory under the new user's home directory. Use the **-p** option with **mkdir** to create the full path, including any intermediate directories that do not exist.

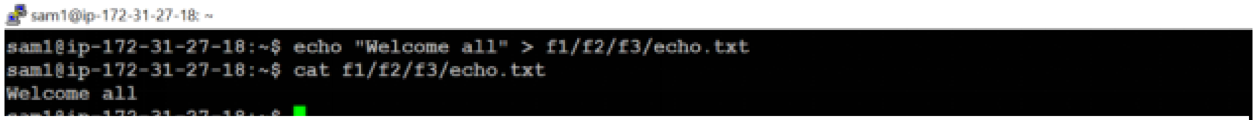


```
sam1@ip-172-31-27-18:~$ mkdir -p f1/f2/f3
sam1@ip-172-31-27-18:~$ ls -R
.:
f1

./f1:
f2

./f1/f2:
f3

./f1/f2/f3:
sam1@ip-172-31-27-18:~$
```

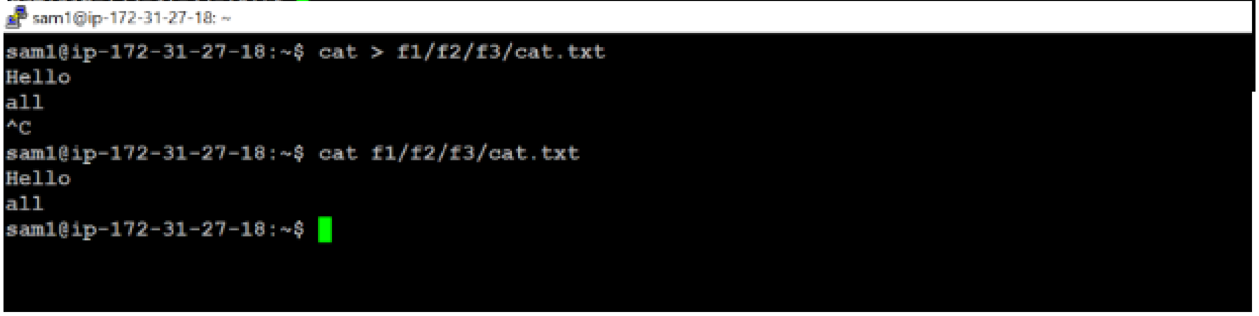
3. Create file using cat and echo command

- ❖ The cat command is commonly used to display file contents, but it can also be used to create files and input text interactively.

- ❖ The echo command outputs the text you provide to the terminal, but it can also be used to write text to files.



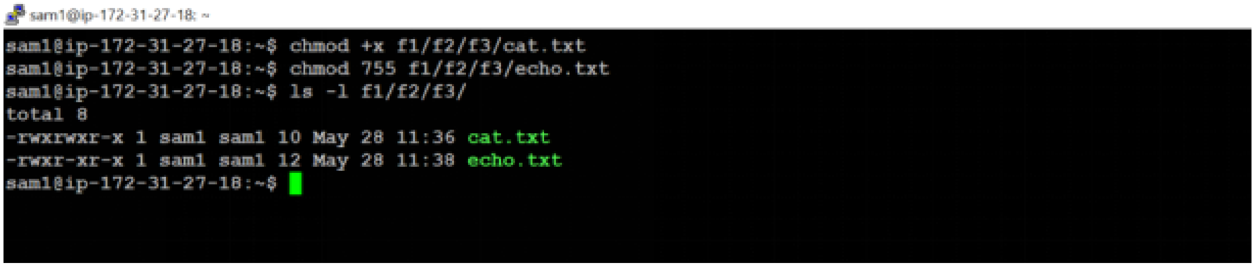
```
sam1@ip-172-31-27-18:~$ echo "Welcome all" > f1/f2/f3/echo.txt
sam1@ip-172-31-27-18:~$ cat f1/f2/f3/echo.txt
Welcome all
sam1@ip-172-31-27-18:~$
```



```
sam1@ip-172-31-27-18:~$ cat > f1/f2/f3/cat.txt
Hello
all
^C
sam1@ip-172-31-27-18:~$ cat f1/f2/f3/cat.txt
Hello
all
sam1@ip-172-31-27-18:~$
```

4. Change the executable file permission for the file

- The **chmod** command allows you to modify the permissions of a file. To make a file executable, you need to add execute permissions. You can do this using symbolic mode or numeric mode.

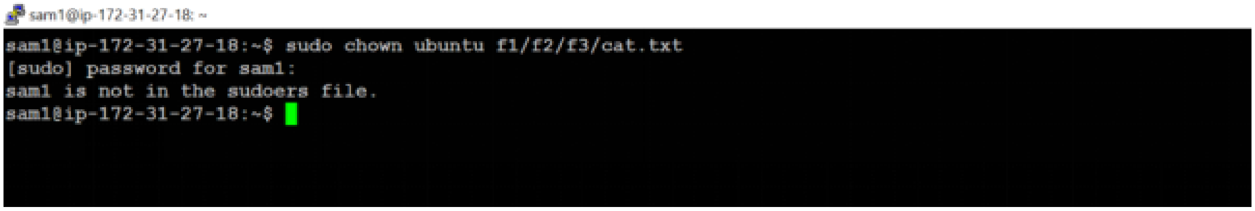


```
sam1@ip-172-31-27-18:~$ chmod +x f1/f2/f3/cat.txt
sam1@ip-172-31-27-18:~$ chmod 755 f1/f2/f3/echo.txt
sam1@ip-172-31-27-18:~$ ls -l f1/f2/f3/
total 8
-rwxrwxr-x 1 sam1 sam1 10 May 28 11:36 cat.txt
-rwxr-xr-x 1 sam1 sam1 12 May 28 11:38 echo.txt
sam1@ip-172-31-27-18:~$
```

5. Change the owner of the file

- The **chown** (change owner) command allows you to specify a new owner for the file.

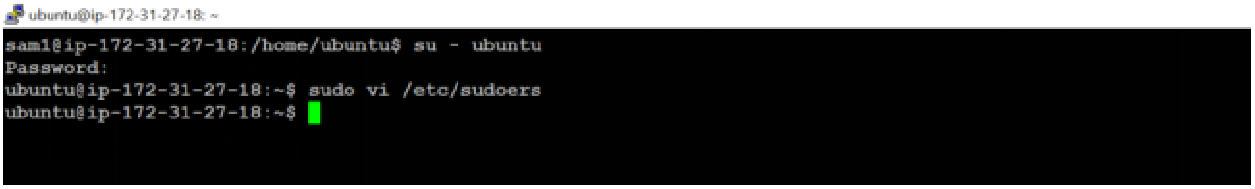
Error:



```
sam1@ip-172-31-27-18:~$ sudo chown ubuntu f1/f2/f3/cat.txt
[sudo] password for sam1:
sam1 is not in the sudoers file.
sam1@ip-172-31-27-18:~$
```

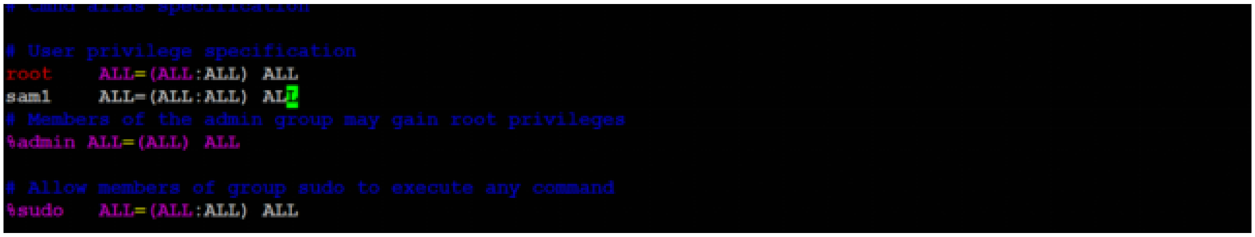
Solution:

- Switch to a User with **sudo** Privileges
- Edit the **sudoers** File



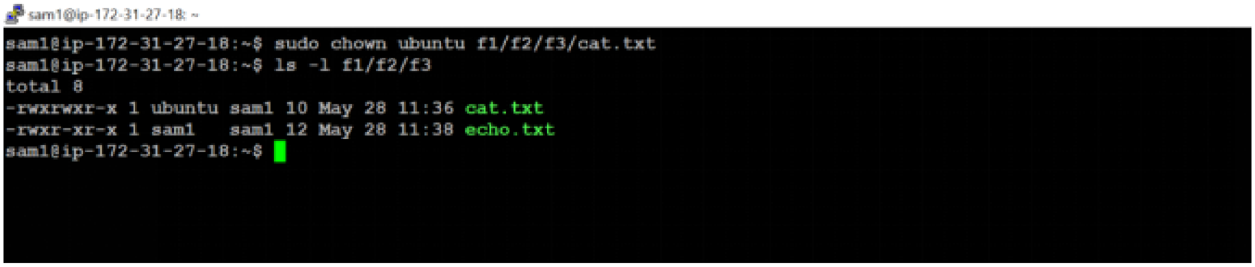
```
sam1@ip-172-31-27-18:/home/ubuntu$ su - ubuntu
Password:
ubuntu@ip-172-31-27-18:~$ sudo vi /etc/sudoers
ubuntu@ip-172-31-27-18:~$
```

- ❖ Add sam1 to the **sudoers** File
- ❖ Save and Exit



```
# Cmnd alias specification

# User privilege specification
root     ALL=(ALL:ALL) ALL
sam1     ALL=(ALL:ALL) ALL
# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo    ALL=(ALL:ALL) ALL
```

- ❖ Now we can see the file owner changed from **sam1** to **ubuntu** for cat.txt

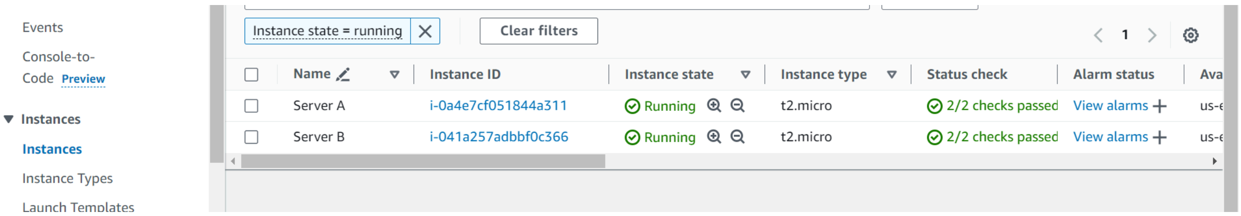


```
sam1@ip-172-31-27-18:~$ sudo chown ubuntu f1/f2/f3/cat.txt
sam1@ip-172-31-27-18:~$ ls -l f1/f2/f3
total 8
-rwxrwxr-x 1 ubuntu sam1 10 May 28 11:36 cat.txt
-rwxr-xr-x 1 sam1   sam1 12 May 28 11:38 echo.txt
sam1@ip-172-31-27-18:~$
```

Linux Task Day-2:-

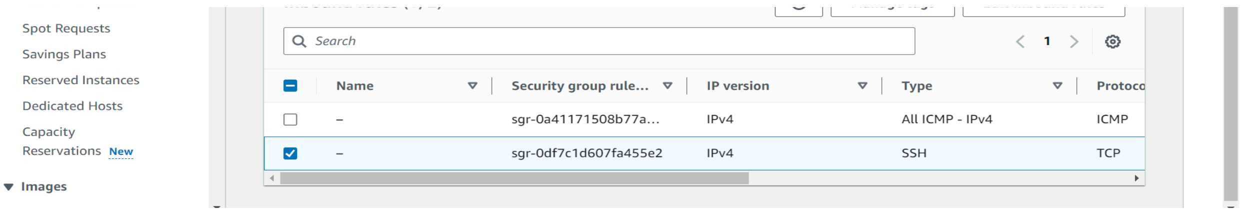
1. **Create two servers in aws and make sure both servers are pinged.**
2. **Create a file in server A and copy that file into Server B.**
3. **Create a shared directory in server A and Create a file under shared directory in server B and that should be reflected in server A under the shared directory.**

1. Create two servers in aws and make sure both servers are pinged.



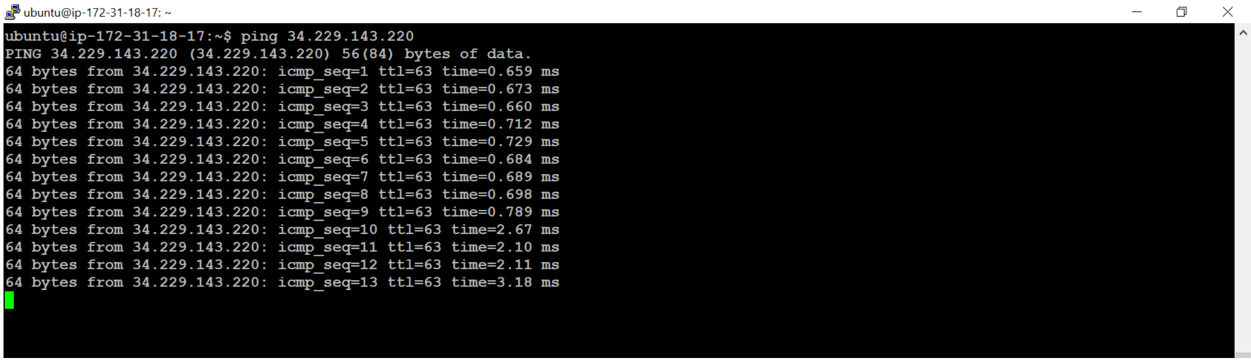
☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
☐	Server A	i-0a4e7cf051844a311	Running	t2.micro	2/2 checks passed	View alarms +	us-e
☐	Server B	i-041a257adbbf0c366	Running	t2.micro	2/2 checks passed	View alarms +	us-e

- ❖ Before pinging of two servers, need to ensure that ICMP port allowed or not



☐	Name	Security group rule...	IP version	Type	Protoco
☐	–	sgr-0a41171508b77a...	IPv4	All ICMP - IPv4	ICMP
☑	–	sgr-0df7c1d607fa455e2	IPv4	SSH	TCP

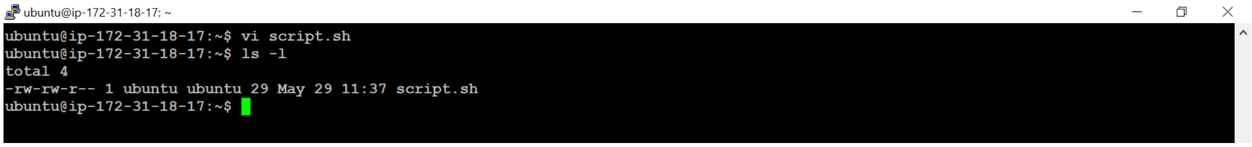
- ❖ Now, ping both servers using the ping command.



```
ubuntu@ip-172-31-18-17:~$ ping 34.229.143.220
PING 34.229.143.220 (34.229.143.220) 56(84) bytes of data.
64 bytes from 34.229.143.220: icmp_seq=1 ttl=63 time=0.659 ms
64 bytes from 34.229.143.220: icmp_seq=2 ttl=63 time=0.673 ms
64 bytes from 34.229.143.220: icmp_seq=3 ttl=63 time=0.660 ms
64 bytes from 34.229.143.220: icmp_seq=4 ttl=63 time=0.712 ms
64 bytes from 34.229.143.220: icmp_seq=5 ttl=63 time=0.729 ms
64 bytes from 34.229.143.220: icmp_seq=6 ttl=63 time=0.684 ms
64 bytes from 34.229.143.220: icmp_seq=7 ttl=63 time=0.689 ms
64 bytes from 34.229.143.220: icmp_seq=8 ttl=63 time=0.698 ms
64 bytes from 34.229.143.220: icmp_seq=9 ttl=63 time=0.789 ms
64 bytes from 34.229.143.220: icmp_seq=10 ttl=63 time=2.67 ms
64 bytes from 34.229.143.220: icmp_seq=11 ttl=63 time=2.10 ms
64 bytes from 34.229.143.220: icmp_seq=12 ttl=63 time=2.11 ms
64 bytes from 34.229.143.220: icmp_seq=13 ttl=63 time=3.18 ms
```

2. Create a file in server A and copy that file into Server B.

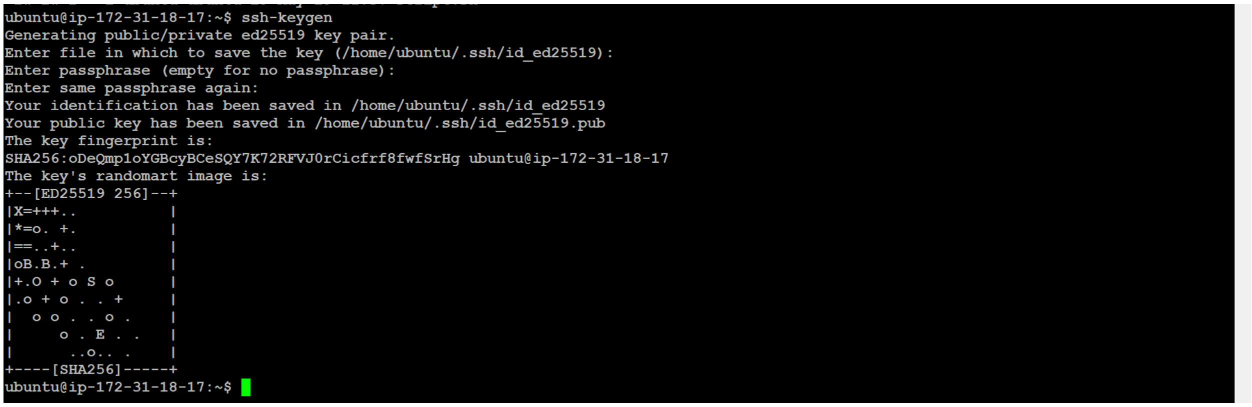
- ❖ File created in server A.



```
ubuntu@ip-172-31-18-17:~$ vi script.sh
ubuntu@ip-172-31-18-17:~$ ls -l
total 4
-rw-rw-r-- 1 ubuntu ubuntu 29 May 29 11:37 script.sh
ubuntu@ip-172-31-18-17:~$
```

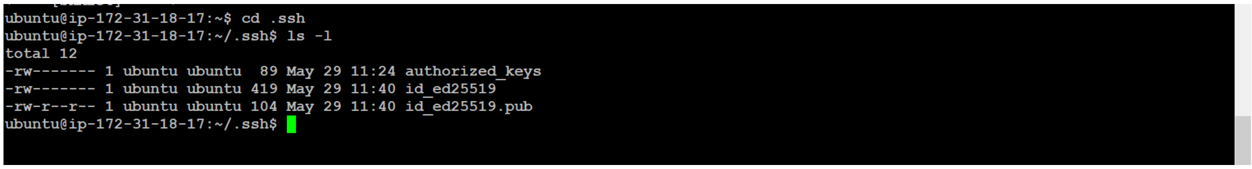
- ❖ Before that I need to set up an ssh connection for both servers.

Step 1: On each server run **ssh-keygen**



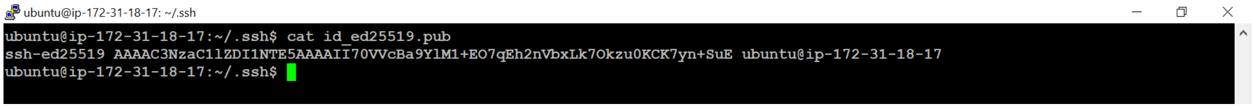
```
ubuntu@ip-172-31-18-17:~$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_ed25519
Your public key has been saved in /home/ubuntu/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:oDeQmp1oYGBcyBCeSQY7K72RFVJ0rCicfrf8fwfSrHg ubuntu@ip-172-31-18-17
The key's randomart image is:
+--[ED25519 256]--+
|X=+++..          |
|*=o. +.          |
|==..+..          |
|oB.B.+ .         |
|+.O + o S o      |
|.o + o . . +     |
|  o o . . o .    |
|     o . E . .   |
|      ..o.. .    |
+----[SHA256]-----+
ubuntu@ip-172-31-18-17:~$
```

Step 2: Created two files under **/home/ubuntu/.ssh**



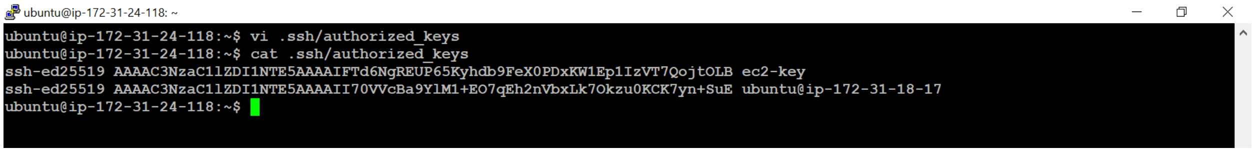
```
ubuntu@ip-172-31-18-17:~$ cd .ssh
ubuntu@ip-172-31-18-17:~/.ssh$ ls -l
total 12
-rw------- 1 ubuntu ubuntu  89 May 29 11:24 authorized_keys
-rw------- 1 ubuntu ubuntu 419 May 29 11:40 id_ed25519
-rw-r--r-- 1 ubuntu ubuntu 104 May 29 11:40 id_ed25519.pub
ubuntu@ip-172-31-18-17:~/.ssh$
```

Step 3: On Server A, cat and copy the public key



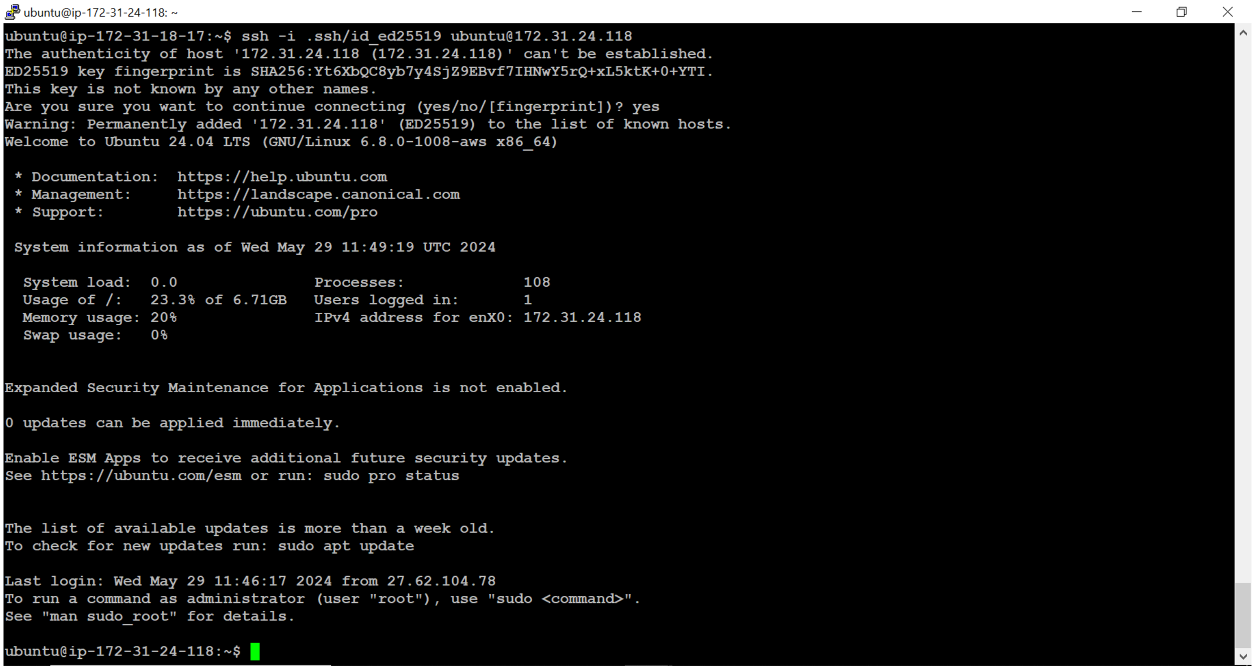
```
ubuntu@ip-172-31-18-17:~/.ssh$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAII70VVcBa9YlM1+EO7qEh2nVbxLk7Okzu0KCK7yn+SuE ubuntu@ip-172-31-18-17
ubuntu@ip-172-31-18-17:~/.ssh$
```

Step 4: select and copy to the key and append the key into /home/ubuntu/.ssh/authorized_keys file of Server B.



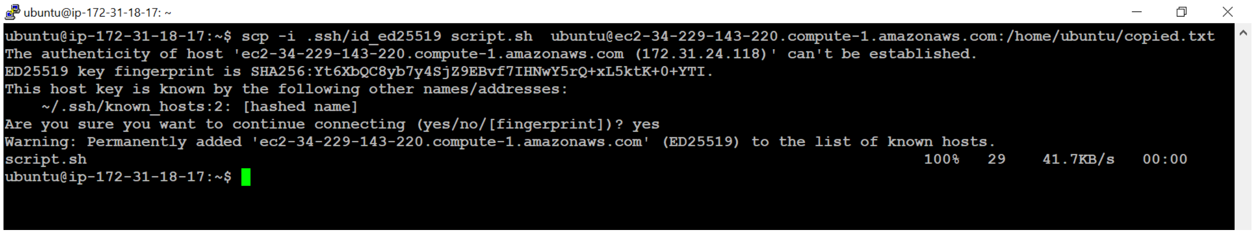
```
ubuntu@ip-172-31-24-118:~$ vi .ssh/authorized_keys
ubuntu@ip-172-31-24-118:~$ cat .ssh/authorized_keys
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIFTd6NgREUP65Kyhdb9FeX0PDxKW1Ep1IzVT7QojtOLB ec2-key
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAII70VVcBa9YlM1+EO7qEh2nVbxLk7Okzu0KCK7yn+SuE ubuntu@ip-172-31-18-17
ubuntu@ip-172-31-24-118:~$
```

Step 5: Now ssh from server A



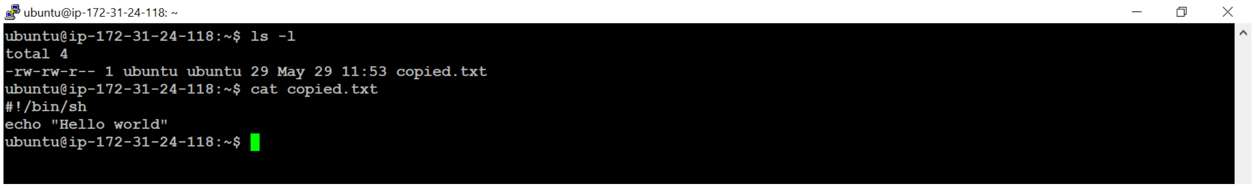
```
ubuntu@ip-172-31-18-17:~$ ssh -i .ssh/id_ed25519 ubuntu@172.31.24.118
The authenticity of host '172.31.24.118 (172.31.24.118)' can't be established.
ED25519 key fingerprint is SHA256:Yt6XbQC8yb7y4SjZ9EBvf7IHNwY5rQ+xL5ktK+0+YTI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.24.118' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1008-aws x86_64)

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

 System information as of Wed May 29 11:49:19 UTC 2024

  System load:  0.0               Processes:             108
  Usage of /:   23.3% of 6.71GB   Users logged in:       1
  Memory usage: 20%               IPv4 address for enX0: 172.31.24.118
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Wed May 29 11:46:17 2024 from 27.62.104.78
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-24-118:~$
```

- ❖ Now paste the copied file from Server A into Server B using secure copy command.



```
ubuntu@ip-172-31-18-17:~$ scp -i .ssh/id_ed25519 script.sh  ubuntu@ec2-34-229-143-220.compute-1.amazonaws.com:/home/ubuntu/copied.txt
The authenticity of host 'ec2-34-229-143-220.compute-1.amazonaws.com (172.31.24.118)' can't be established.
ED25519 key fingerprint is SHA256:Yt6XbQC8yb7y4SjZ9EBvf7IHNwY5rQ+xL5ktK+0+YTI.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:2: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-229-143-220.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
script.sh                                                                                              100%   29    41.7KB/s   00:00
ubuntu@ip-172-31-18-17:~$
```

- ❖ Now do ls -l in Server B we can see the cooped file from Server A.



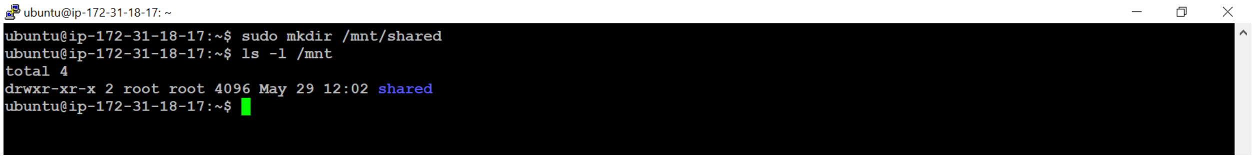
```
ubuntu@ip-172-31-24-118:~$ ls -l
total 4
-rw-rw-r-- 1 ubuntu ubuntu 29 May 29 11:53 copied.txt
ubuntu@ip-172-31-24-118:~$ cat copied.txt
#!/bin/sh
echo "Hello world"
ubuntu@ip-172-31-24-118:~$
```

3. Create a shared directory in server A and Create a file under shared directory in server B and that should be reflected in server A under the shared directory.

- ❖ Install NFS Server Software on Server A

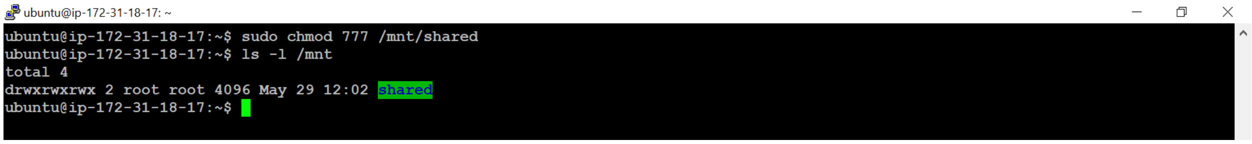
    sudo apt update && sudo apt install -y nfs-kernel-server

- ❖ Create a directory in Server A under **/mnt** that will serve as the shared directory.



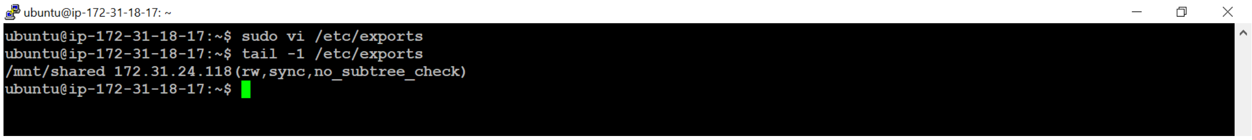
```
ubuntu@ip-172-31-18-17:~$ sudo mkdir /mnt/shared
ubuntu@ip-172-31-18-17:~$ ls -l /mnt
total 4
drwxr-xr-x 2 root root 4096 May 29 12:02 shared
ubuntu@ip-172-31-18-17:~$
```

❖ Adjust the permissions of the shared directory to allow access to the users or groups that need it.



```
ubuntu@ip-172-31-18-17:~$ sudo chmod 777 /mnt/shared
ubuntu@ip-172-31-18-17:~$ ls -l /mnt
total 4
drwxrwxrwx 2 root root 4096 May 29 12:02 shared
ubuntu@ip-172-31-18-17:~$
```

❖ Edit **/etc/exports** to Configure NFS Exports.This file contains the configuration for which directories are shared and with which clients they are shared.

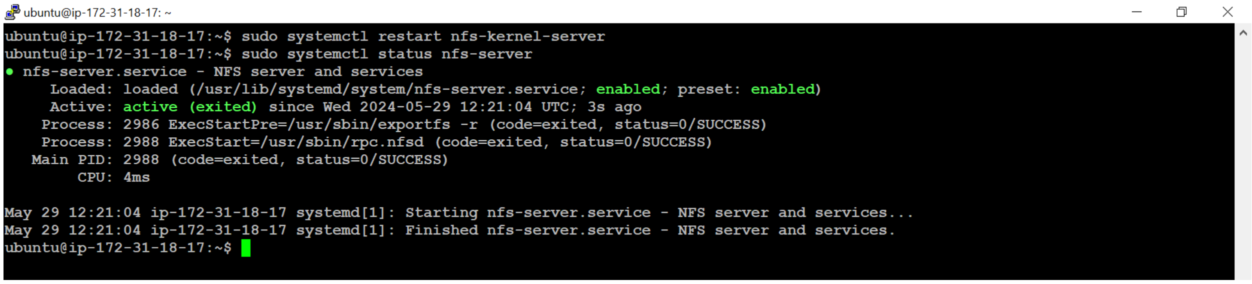


```
ubuntu@ip-172-31-18-17:~$ sudo vi /etc/exports
ubuntu@ip-172-31-18-17:~$ tail -1 /etc/exports
/mnt/shared 172.31.24.118(rw,sync,no_subtree_check)
ubuntu@ip-172-31-18-17:~$
```

❖ Export the shared directory using this command.

```
sudo /etc/exportfs -a
```

❖ Restart the NFS service

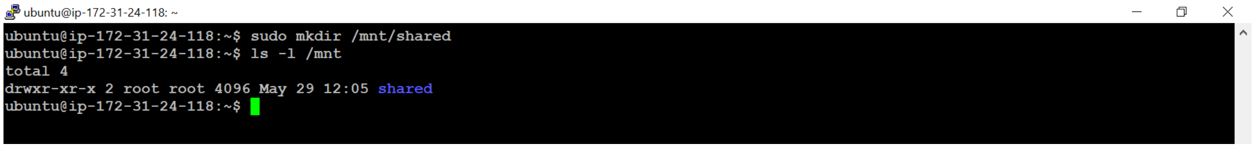


```
ubuntu@ip-172-31-18-17:~$ sudo systemctl restart nfs-kernel-server
ubuntu@ip-172-31-18-17:~$ sudo systemctl status nfs-server
● nfs-server.service - NFS server and services
     Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; preset: enabled)
     Active: active (exited) since Wed 2024-05-29 12:21:04 UTC; 3s ago
    Process: 2986 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
    Process: 2988 ExecStart=/usr/sbin/rpc.nfsd (code=exited, status=0/SUCCESS)
   Main PID: 2988 (code=exited, status=0/SUCCESS)
        CPU: 4ms

May 29 12:21:04 ip-172-31-18-17 systemd[1]: Starting nfs-server.service - NFS server and services...
May 29 12:21:04 ip-172-31-18-17 systemd[1]: Finished nfs-server.service - NFS server and services.
ubuntu@ip-172-31-18-17:~$
```

❖ Install NFS client in Server B using this command

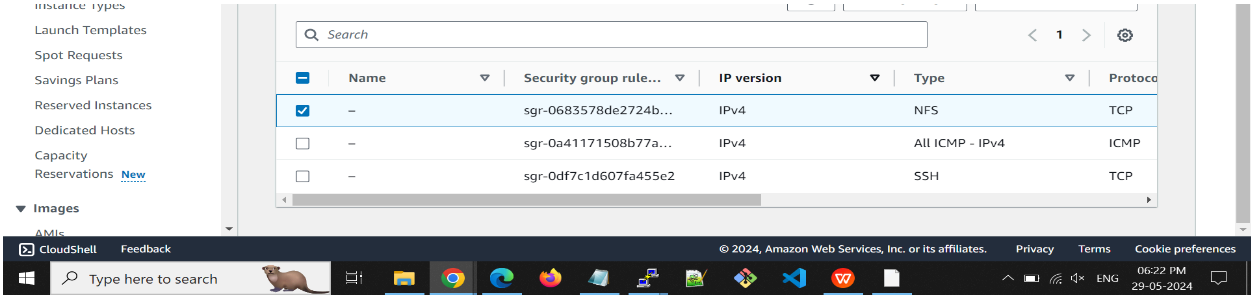
```
sudo apt update && sudo apt install -y nfs-common
```

❖ Create a directory in Server B under **/mnt** for mount to Server A shared directory.



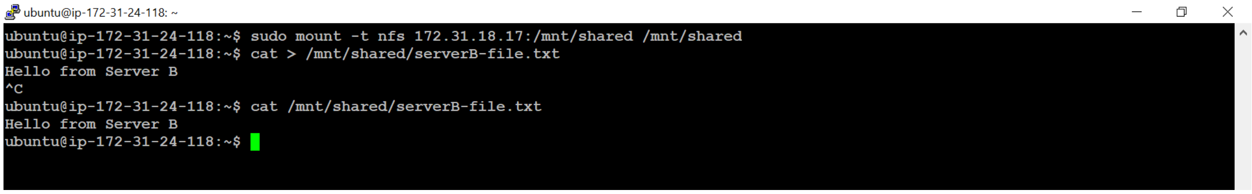
```
ubuntu@ip-172-31-24-118:~$ sudo mkdir /mnt/shared
ubuntu@ip-172-31-24-118:~$ ls -l /mnt
total 4
drwxr-xr-x 2 root root 4096 May 29 12:05 shared
ubuntu@ip-172-31-24-118:~$
```

❖ Ensure that NFS ports are allowed in the security group. If now allowed, add the rule for NFS ports.



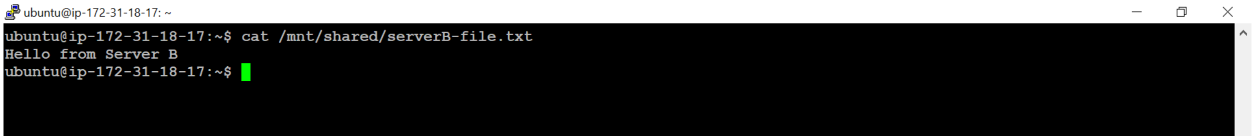
	Name	Security group rule...	IP version	Type	Protoco
☑	–	sgr-0683578de2724b...	IPv4	NFS	TCP
□	–	sgr-0a41171508b77a...	IPv4	All ICMP - IPv4	ICMP
□	–	sgr-0df7c1d607fa455e2	IPv4	SSH	TCP

- ❖ Now, mount the Server A and Server B directories. Created file under /mnt/shared/serverB-file.



```
ubuntu@ip-172-31-24-118:~$ sudo mount -t nfs 172.31.18.17:/mnt/shared /mnt/shared
ubuntu@ip-172-31-24-118:~$ cat > /mnt/shared/serverB-file.txt
Hello from Server B
^C
ubuntu@ip-172-31-24-118:~$ cat /mnt/shared/serverB-file.txt
Hello from Server B
ubuntu@ip-172-31-24-118:~$
```

- ❖ Now we can see the Server B file in Server A.



```
ubuntu@ip-172-31-18-17:~$ cat /mnt/shared/serverB-file.txt
Hello from Server B
ubuntu@ip-172-31-18-17:~$
```