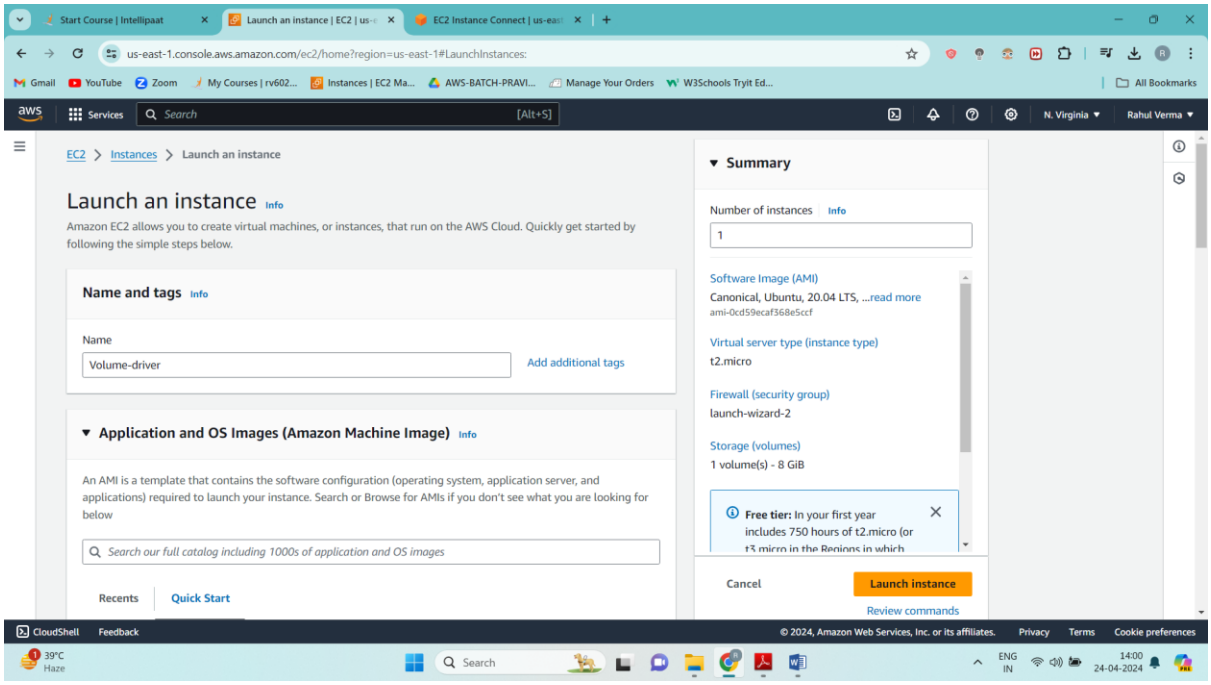
Volumes driver

1: Install the Driver Plugin.

sudo docker plugin install vieux/sshfs

```
root@ip-172-31-83-79:/home/ubuntu# sudo docker plugin install vieux/sshfs
Plugin "vieux/sshfs" is requesting the following privileges:
 - network: [host]
 - mount: [/var/lib/docker/plugins/]
 - mount: []
 - device: [/dev/fuse]
 - capabilities: [CAP_SYS_ADMIN]
Do you grant the above permissions? [y/N] y
latest: Pulling from vieux/sshfs
Digest: sha256:1d3c3e42c12138da5ef7873b97f7f32cf99fb6edde75fa4f0bcf9ed277855811
52d435ada6a4: Complete
Installed plugin vieux/sshfs
root@ip-172-31-83-79:/home/ubuntu#
```

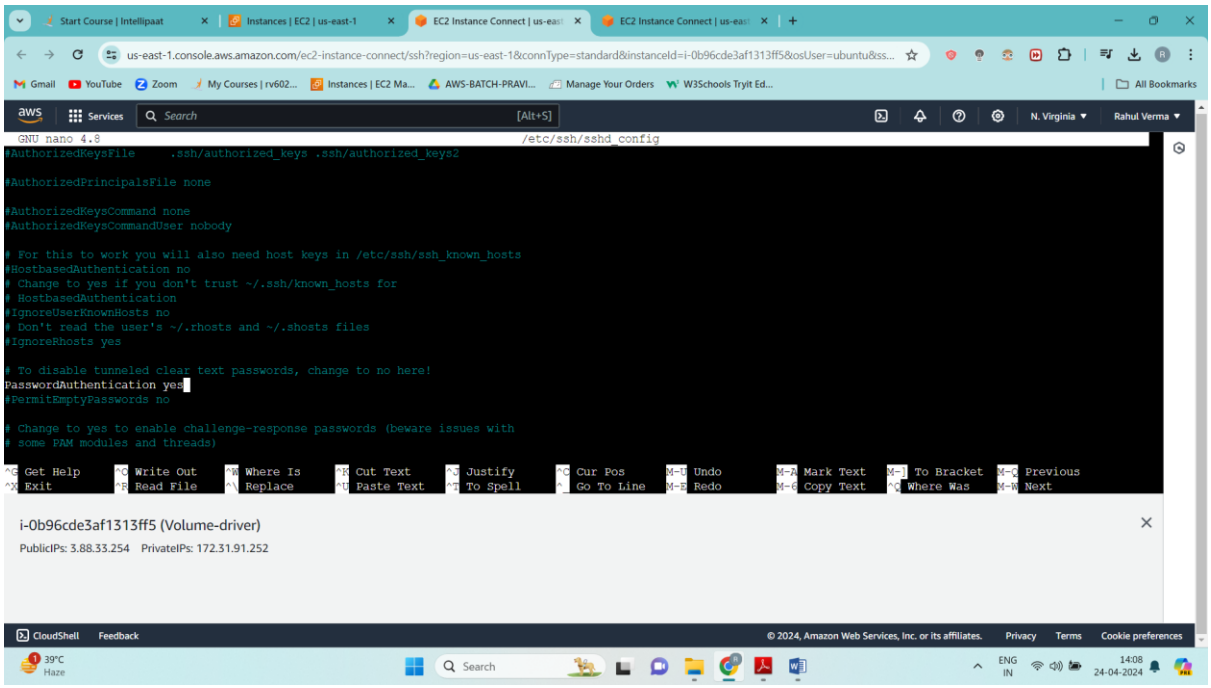
2: Create a second Ubuntu instance.



Do this step on your second instance

3: Now you have to enable a password for the new instance, for doing that there are a few steps to follow, first allow PasswordAuthentication as yes in the sshd_config file.

sudo nano /etc/ssh/sshd_config



Press ctrl+s and ctrl+x to save and exit from nano editor

4: Then you need to type in the following command in the new instance. This will prompt you for the new password to be enabled.

sudo passwd <username>

Note: username will be Ubuntu only

Now enter password

Retype new password to confirm

```
ubuntu@ip-172-31-91-252:~$ sudo passwd ubuntu
New password:
Retype new password:
passwd: password updated successfully
ubuntu@ip-172-31-91-252:~$
```

5: Finally you just need to restart the sshd service using the below command to ensure that the changes take place.

sudo service sshd restart

```
ubuntu@ip-172-31-91-252:~$ sudo service sshd restart
ubuntu@ip-172-31-91-252:~$
```

6: Go back to the Original instance. Create a volume "sshvol" using the volume driver "sshfs" using the name below. Mention the same password here as the one you had enabled previously. The path you mention here is the one that will be used as volume storage.

sudo docker volume create –driver <driver_name> -o
sshcmd=<username@hostIP:path>
-o password= <password><volume_name>

Before that let's make a directory in your second instance

mkdir docker
ls
cd docker

```
ubuntu@ip-172-31-91-252:~$ mkdir docker
ubuntu@ip-172-31-91-252:~$ ls
docker
ubuntu@ip-172-31-91-252:~$ cd docker
ubuntu@ip-172-31-91-252:~/docker$
```

i-0b96cde3af1313ff5 (Volume-driver)

Come back to your main instance and write these commands mentioned above also

sudo docker volume create –driver <driver_name> -o sshcmd=<username@hostIP:path>
-o password= <password><volume_name>

```
root@ip-172-31-83-79:/home/ubuntu# sudo docker volume create -d vieux/sshfs -o sshcmd=ubuntu@3.88.33.254:/home/ubuntu/docker -o password=rahul Demo-vol
Demo-vol
root@ip-172-31-83-79:/home/ubuntu#
```

let's check the volumes

```
root@ip-172-31-83-79:/home/ubuntu# docker volume ls
DRIVER              VOLUME NAME
vieux/sshfs:latest  Demo-vol
local               box-vol
local               new-vol
local               new-vol2
local               rovol
local               volume
root@ip-172-31-83-79:/home/ubuntu#
```

7: Then run a Container with the Volume you just created.

sudo docker run –d –name <name_of_container> --mount source=<volume_name>,target=<Destination><image-name>

```
root@ip-172-31-83-79:/home/ubuntu# docker run -d --name demo-container --mount source=Demo-vol,target=/app nginx:latest
59a0662299f7cfdea6a6e64791979fb79748cf30d2f507589b5f2ae4d6d27404
root@ip-172-31-83-79:/home/ubuntu# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATUS              PORTS     NAMES
59a0662299f7   nginx:latest   "/docker-entrypoint.…"   About a minute ago   Up About a minute   80/tcp    demo-container
```

Now if we create any file inside /docker on second Ubuntu instance then that same file should be there in our main instance as well and vise-versa.

8: create some file in your second instance

```
ubuntu@ip-172-31-91-252:~$ cd docker
ubuntu@ip-172-31-91-252:~/docker$ touch hello.txt
ubuntu@ip-172-31-91-252:~/docker$ ls
hello.txt
ubuntu@ip-172-31-91-252:~/docker$
```

i-0b96cde3af1313ff5 (Volume-driver)

PublicIPs: 3.88.33.254 PrivateIPs: 172.31.91.252

And now check in our main instance if such file exists or not

```
root@ip-172-31-83-79:/home/ubuntu# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATUS              PORTS     NAMES
59a0662299f7   nginx:latest   "/docker-entrypoint.…"   About a minute ago   Up About a minute   80/tcp    demo-container
root@ip-172-31-83-79:/home/ubuntu#
root@ip-172-31-83-79:/home/ubuntu#
root@ip-172-31-83-79:/home/ubuntu#
root@ip-172-31-83-79:/home/ubuntu# docker exec -it 59a0662299f7 bash
root@59a0662299f7:/# ls
app  bin  boot  dev  docker-entrypoint.d  docker-entrypoint.sh  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@59a0662299f7:/# cd app
root@59a0662299f7:/app# ls
hello.txt
root@59a0662299f7:/app#
```

9: Now create some file in main instance inside /app

```
root@59a0662299f7:/app# touch AI.txt
root@59a0662299f7:/app# ls
AI.txt  hello.txt
root@59a0662299f7:/app#
```

And will check our second instance now

```
ubuntu@ip-172-31-91-252:~$ cd docker
ubuntu@ip-172-31-91-252:~/docker$ touch hello.txt
ubuntu@ip-172-31-91-252:~/docker$ ls
hello.txt
ubuntu@ip-172-31-91-252:~/docker$ ls
AI.txt  hello.txt
ubuntu@ip-172-31-91-252:~/docker$
```

i-0b96cde3af1313ff5 (Volume-driver)

PublicIPs: 3.88.33.254 PrivateIPs: 172.31.91.252