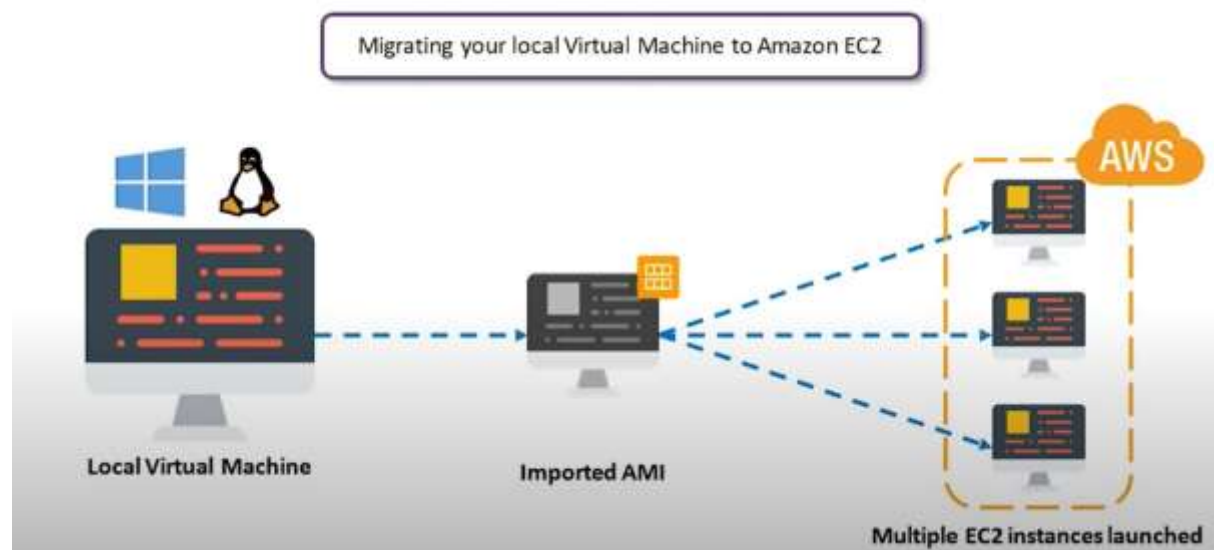


Virtual machine migration from on prem to cloud

Migrating your local Virtual machine to cloud (ec2)



How to migrate?

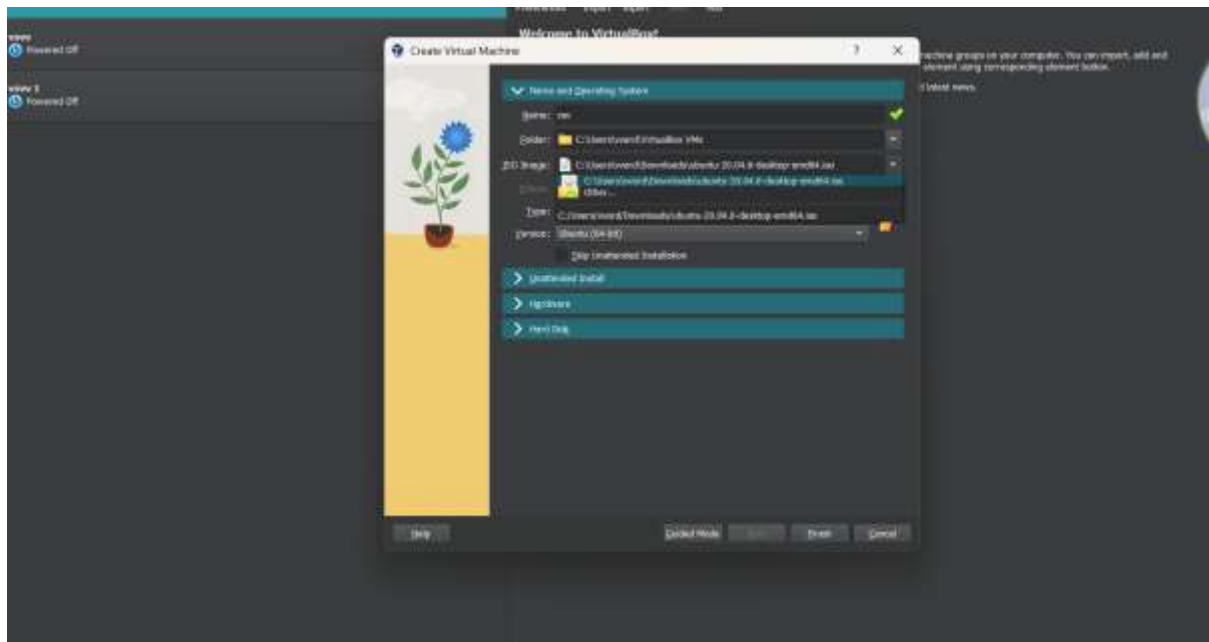
- ✓ Download and Install AWS CLI
- ✓ Export your local VM as a VMDK file and upload it to a S3 bucket
- ✓ Import the VM using commands
- ✓ Monitor your Import task until you see an AMI
- ✓ Launch Virtual Machines with AMI



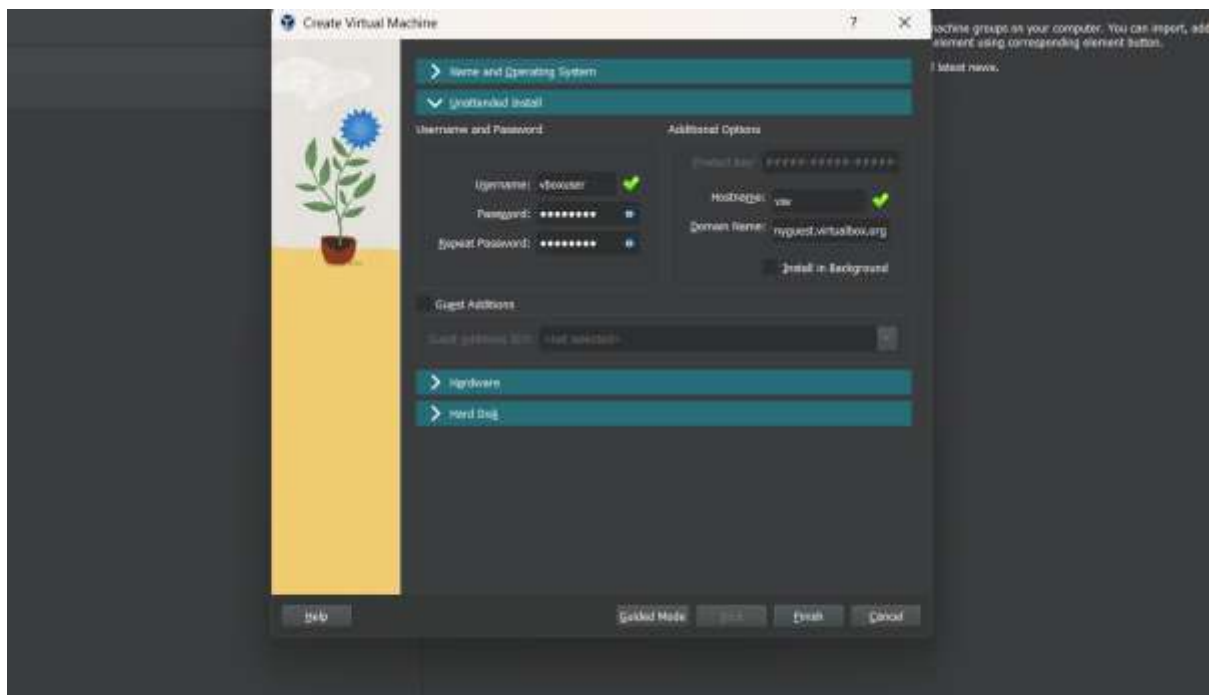
Click on new

<https://www.releases.ubuntu.com/20.04/ubuntu-20.04.6-desktop-amd64.iso>

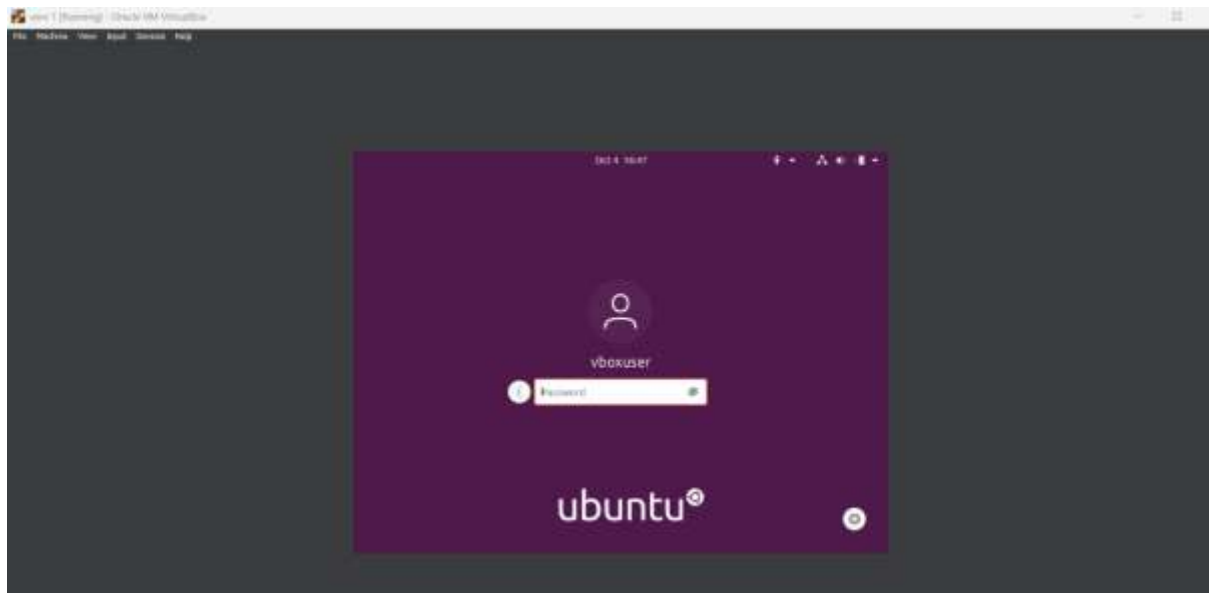
ubuntu desktop images download url



Select your ubuntu image file path



Here change your password

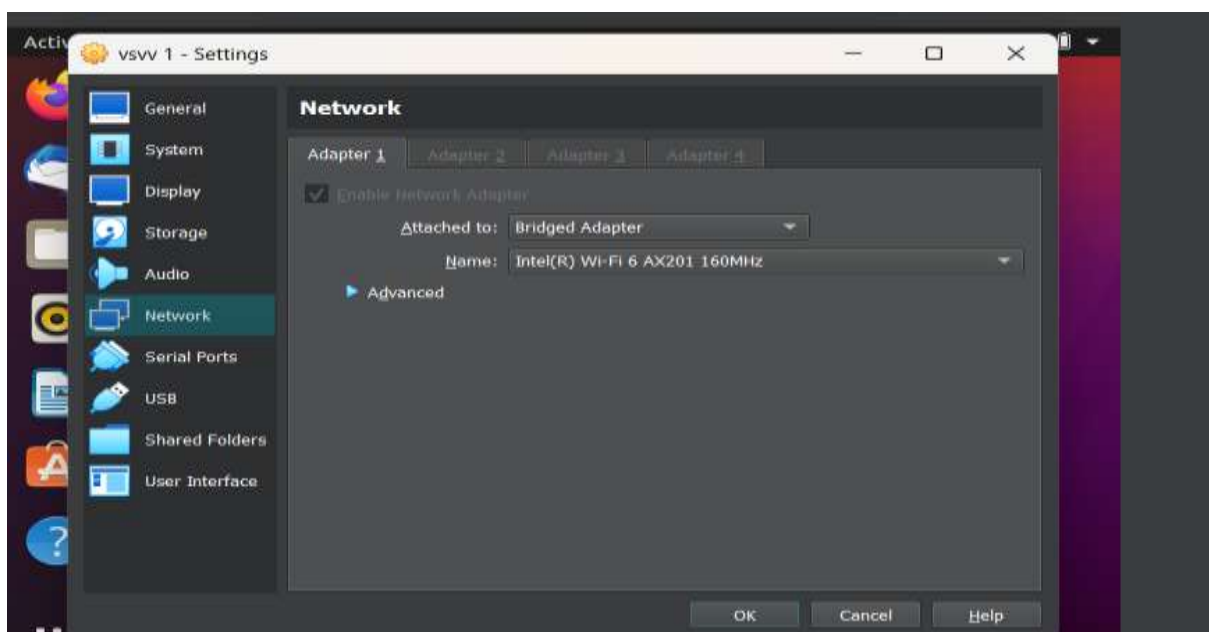


Login through your password

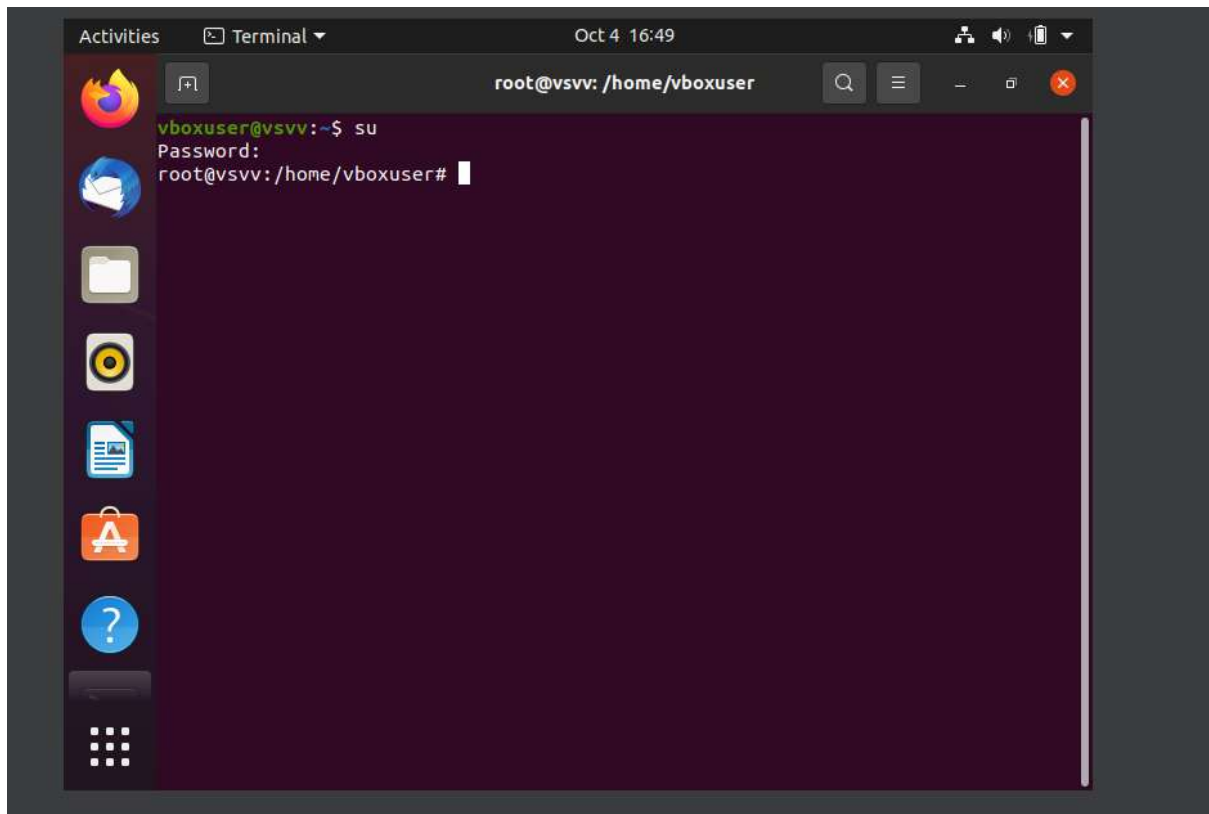


Change internet conectivity

Right click on network settings



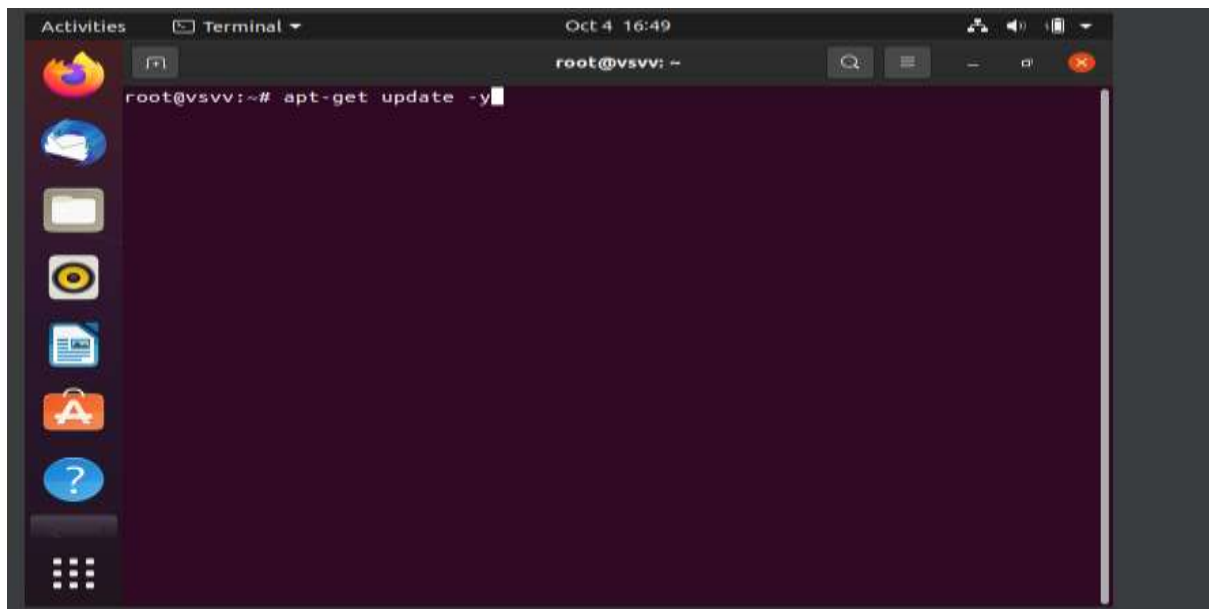
Select bridge adapter like above



Open the terminal in vm

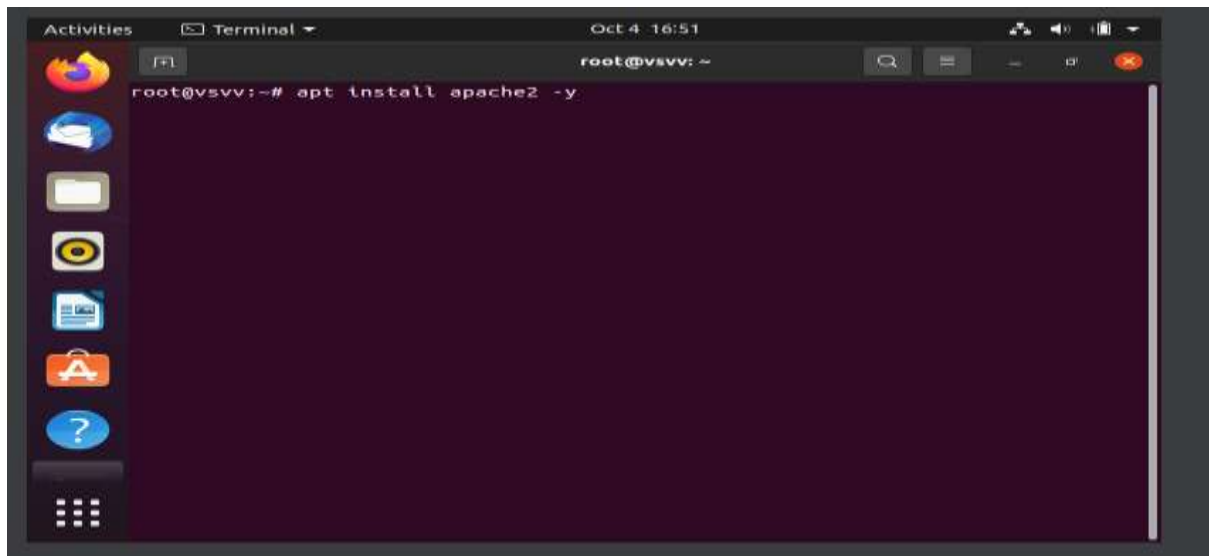
Switch to root using su

Password is your password



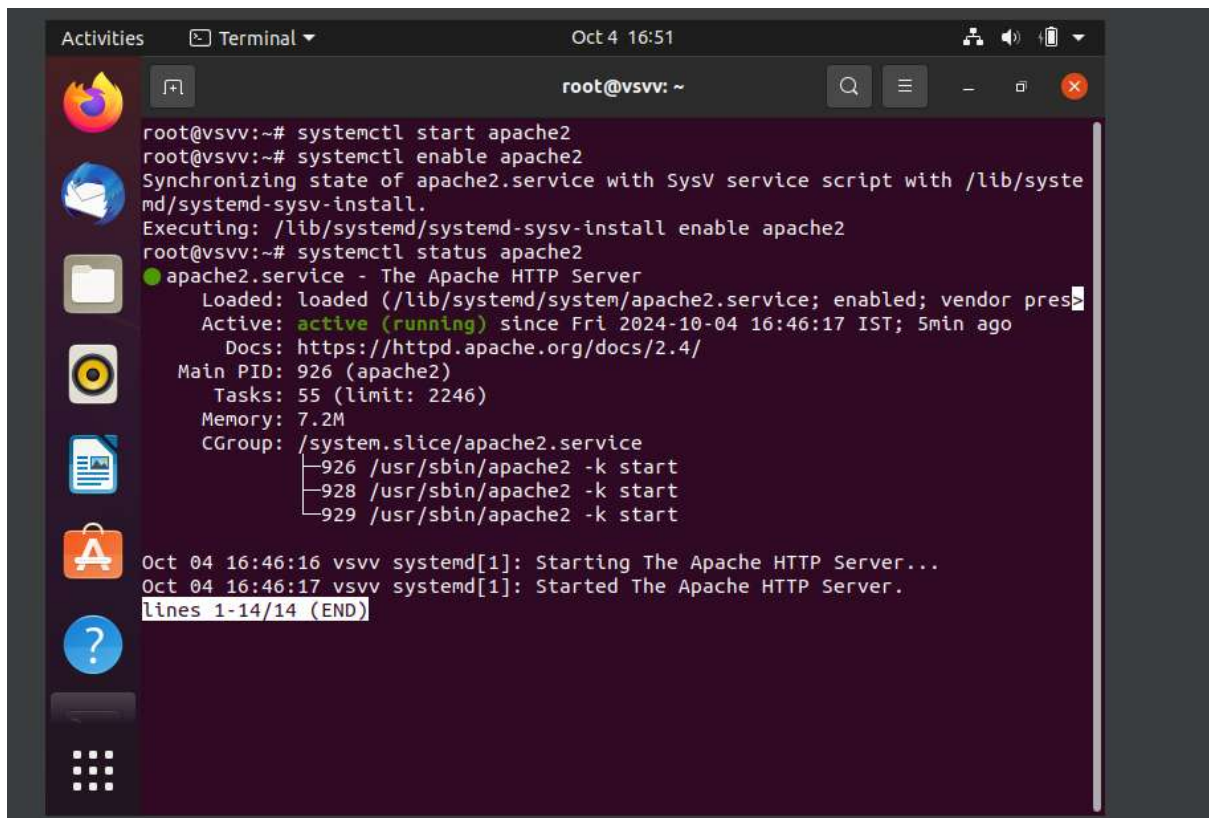
Update the system by using

apt-get update -y



Install apache2 in vm

apt install apache2 -y

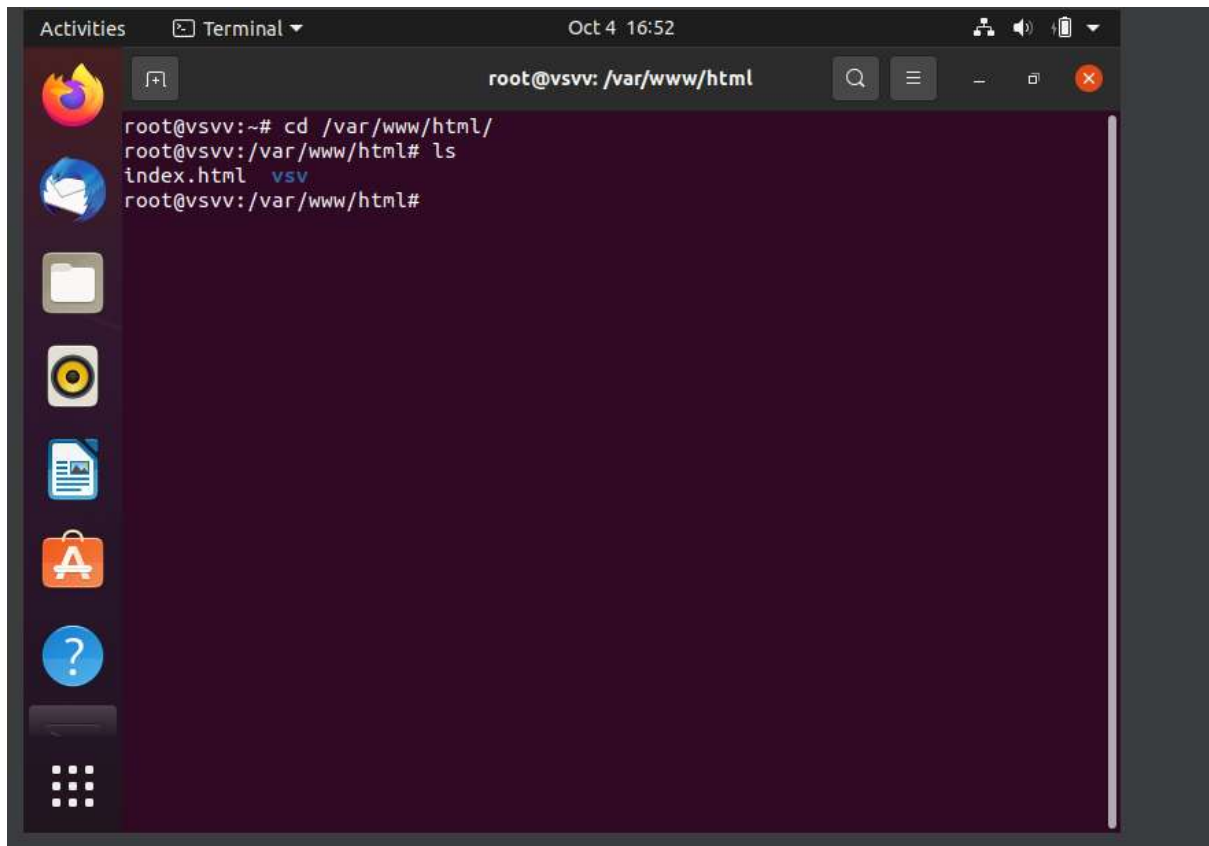


Start and enable the server

systemctl start apache2

systemctl enable apache2

systemctl status apache2

A screenshot of a Linux terminal window. The window title bar shows 'Activities', 'Terminal', and the date/time 'Oct 4 16:52'. The terminal prompt is 'root@vsvv: /var/www/html'. The user has entered the command 'cd /var/www/html/' and the prompt has changed to 'root@vsvv:/var/www/html#'. The user then enters 'ls' and the output shows 'index.html' and 'vsv'. The user then enters 'vsv' and the prompt changes to 'root@vsvv:/var/www/html#'. The terminal window has a dark purple background and a sidebar on the left with various application icons.

```
root@vsvv:~# cd /var/www/html/
root@vsvv:/var/www/html# ls
index.html  vsv
root@vsvv:/var/www/html#
```

Go to apache default directory

```
cd /var/www/html
```

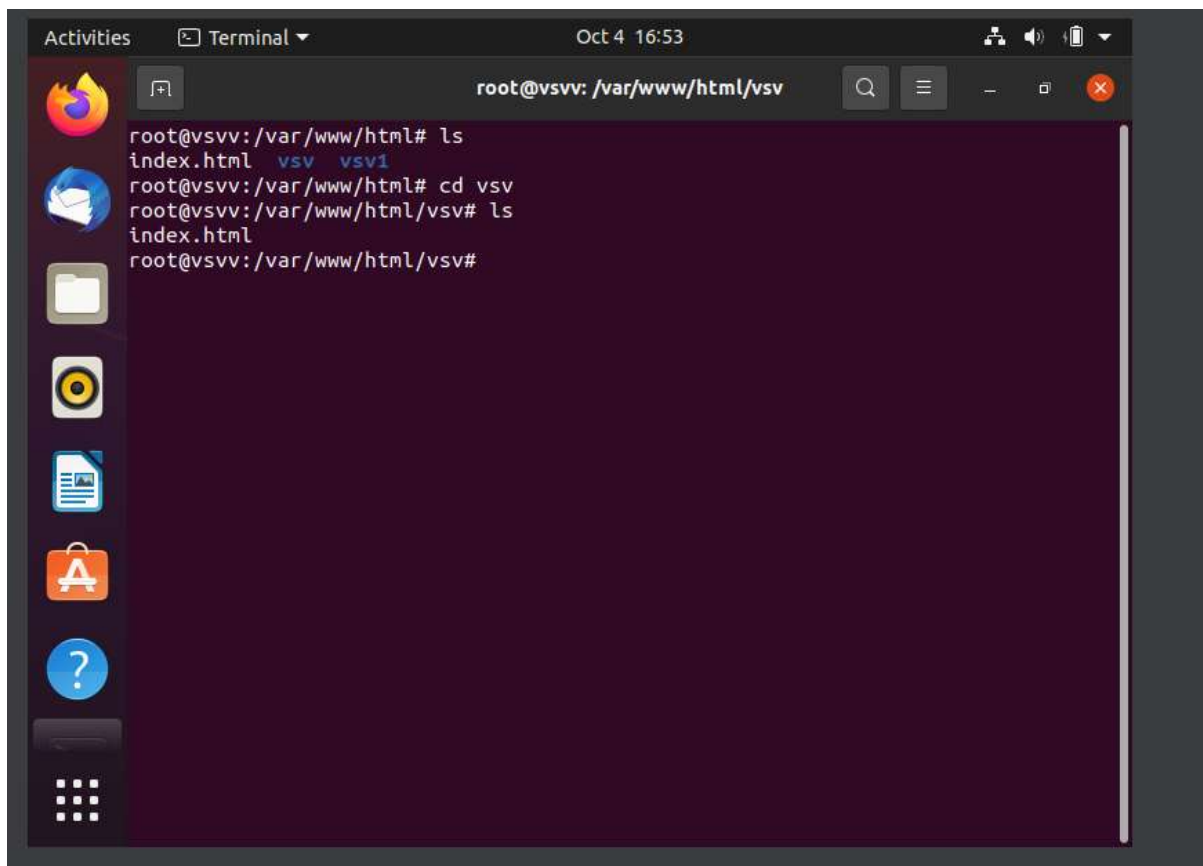
create new path

```
mkdir vsv
```

inside directory create new index file

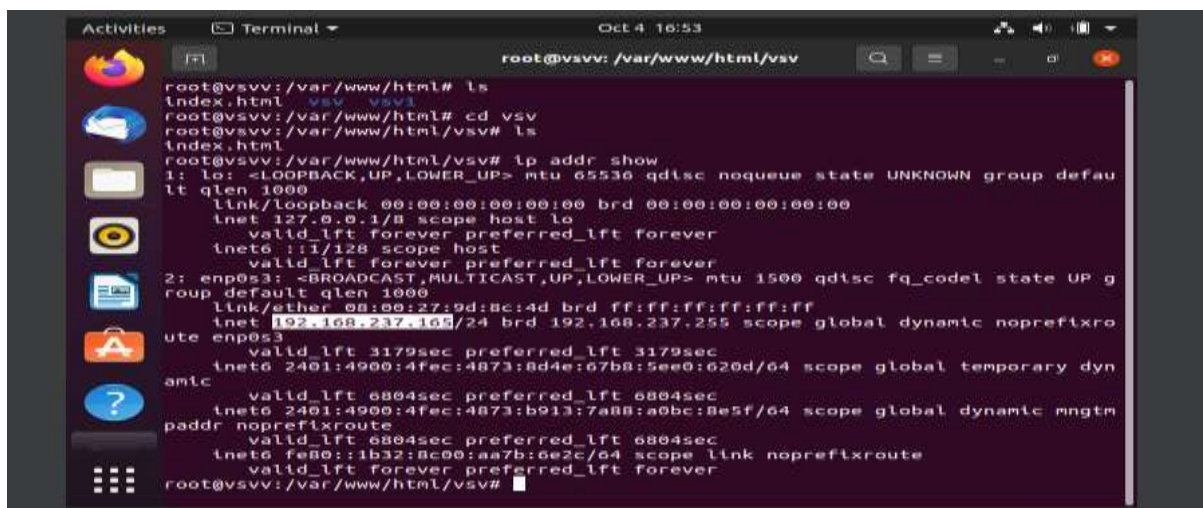
```
vi index.html
```

---- ADD Sample Content -----

A terminal window titled 'Terminal' with a date and time of 'Oct 4 16:53'. The prompt is 'root@vsvv: /var/www/html/vsv'. The user has navigated from /var/www/html to /var/www/html/vsv and listed the contents, which include 'index.html'.

```
root@vsvv: /var/www/html# ls
index.html  vsv  vsv1
root@vsvv: /var/www/html# cd vsv
root@vsvv: /var/www/html/vsv# ls
index.html
root@vsvv: /var/www/html/vsv#
```

Here apache is completed

A terminal window showing the output of the 'ip addr show' command. It displays details for the loopback interface 'lo' and the ethernet interface 'enp0s3', including their IP addresses and other network parameters.

```
root@vsvv: /var/www/html/vsv# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
lt qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:9d:8c:4d brd ff:ff:ff:ff:ff:ff
    inet 192.168.237.169/24 brd 192.168.237.255 scope global dynamic noprefixro
ute enp0s3
        valid_lft 3179sec preferred_lft 3179sec
    inet6 2401:4900:4fec:4873:8d4e:67b8:5ee0:620d/64 scope global temporary dyn
amic
        valid_lft 6804sec preferred_lft 6804sec
    inet6 2401:4900:4fec:4873:b913:7a88:a0bc:8e5f/64 scope global dynamic mngtm
paddr noprefixroute
        valid_lft 6804sec preferred_lft 6804sec
    inet6 fe80::1b32:8c00:aa7b:6e2c/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
root@vsvv: /var/www/html/vsv#
```

Ip addr show

This is your public ip of the vm

Check with ip you are getting response or not from web apache2

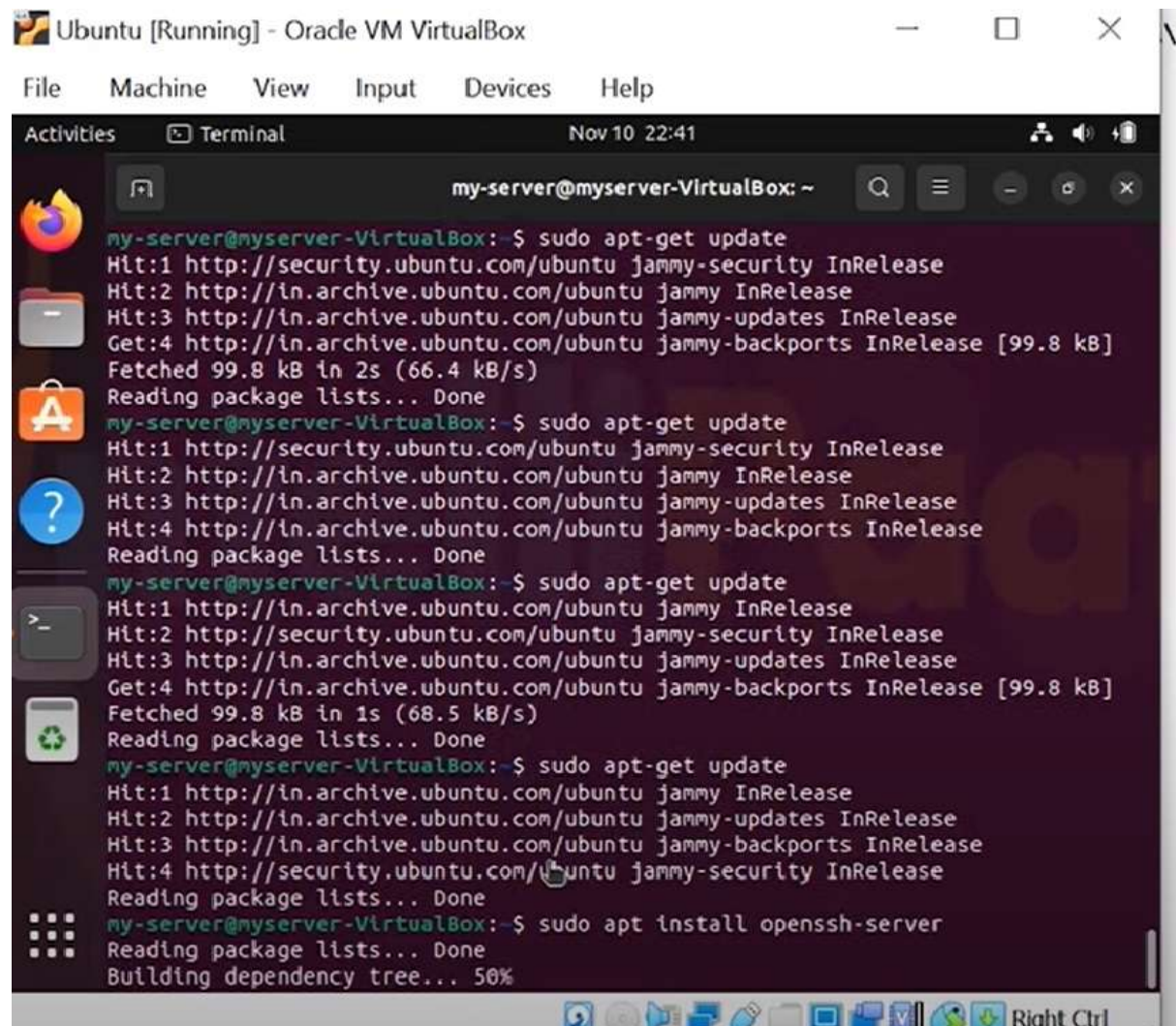
Very Important ***

Next install open ssh server for port enable to connect after migrate

\$sudo apt update

\$sudo apt -get update

\$sudo apt install openssh -server



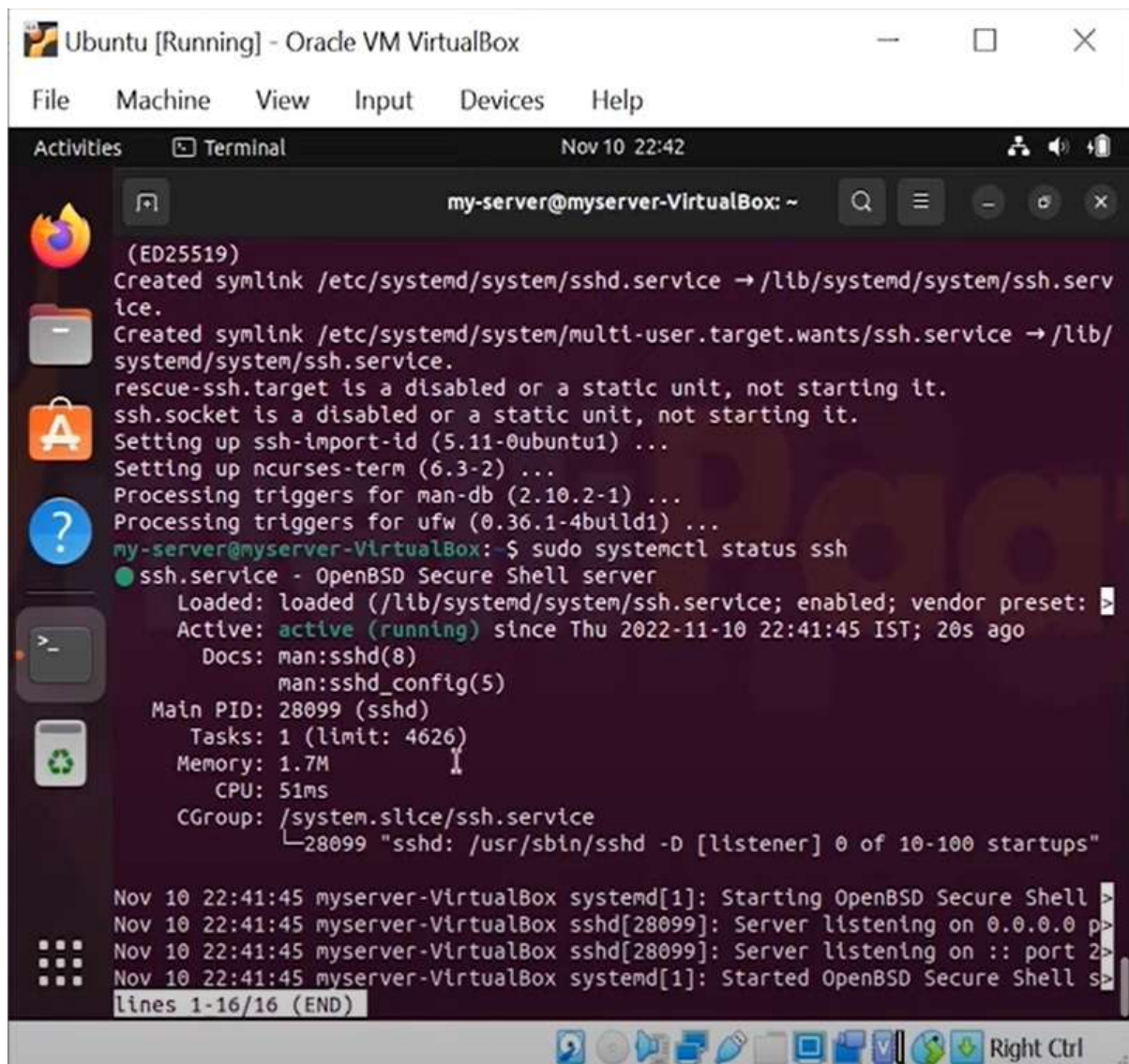
The screenshot shows a terminal window titled "my-server@myserver-VirtualBox: ~" with a search bar and window controls. The terminal output shows the execution of several commands to update the system and install the SSH server. The first three attempts to run "sudo apt-get update" show the system fetching updates from various sources (security.ubuntu.com, in.archive.ubuntu.com) and reading package lists. The final command is "sudo apt install openssh-server", which shows the system building a dependency tree for the installation.

```
my-server@myserver-VirtualBox: ~  
my-server@myserver-VirtualBox: $ sudo apt-get update  
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]  
Fetched 99.8 kB in 2s (66.4 kB/s)  
Reading package lists... Done  
my-server@myserver-VirtualBox: $ sudo apt-get update  
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Reading package lists... Done  
my-server@myserver-VirtualBox: $ sudo apt-get update  
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]  
Fetched 99.8 kB in 1s (68.5 kB/s)  
Reading package lists... Done  
my-server@myserver-VirtualBox: $ sudo apt-get update  
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Reading package lists... Done  
my-server@myserver-VirtualBox: $ sudo apt install openssh-server  
Reading package lists... Done  
Building dependency tree... 50%
```

sudo systemctl status ssh

sudo systemctl status ssh

sudo systemctl start ssh



```
Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Nov 10 22:42
my-server@myserver-VirtualBox: ~
(E025519)
Created symlink /etc/systemd/system/ssh.service → /lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/systemd/system/ssh.service.
rescue-ssh.target is a disabled or a static unit, not starting it.
ssh.socket is a disabled or a static unit, not starting it.
Setting up ssh-import-id (5.11-0ubuntu1) ...
Setting up ncurses-term (6.3-2) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for ufw (0.36.1-4build1) ...
my-server@myserver-VirtualBox: $ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-11-10 22:41:45 IST; 20s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 28099 (sshd)
    Tasks: 1 (limit: 4626)
   Memory: 1.7M
      CPU: 51ms
   CGroup: /system.slice/ssh.service
           └─28099 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Nov 10 22:41:45 myserver-VirtualBox systemd[1]: Starting OpenBSD Secure Shell server:
Nov 10 22:41:45 myserver-VirtualBox sshd[28099]: Server listening on 0.0.0.0 port 22.
Nov 10 22:41:45 myserver-VirtualBox sshd[28099]: Server listening on :: port 22.
Nov 10 22:41:45 myserver-VirtualBox systemd[1]: Started OpenBSD Secure Shell server:
lines 1-16/16 (END)
```

Now press ctrl +z

sudo ufw allow ssh

sudo ufw allow 22


```
Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Nov 10 22:42
my-server@myserver-VirtualBox: ~
ssh.socket is a disabled or a static unit, not starting it.
Setting up ssh-import-id (5.11-0ubuntu1) ...
Setting up ncurses-term (6.3-2) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for ufw (0.36.1-4build1) ...
my-server@myserver-VirtualBox:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset:
   Active: active (running) since Thu 2022-11-10 22:41:45 IST; 20s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 28099 (sshd)
      Tasks: 1 (limit: 4626)
     Memory: 1.7M
        CPU: 51ms
    CGroup: /system.slice/ssh.service
            └─28099 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Nov 10 22:41:45 myserver-VirtualBox systemd[1]: Starting OpenBSD Secure Shell
Nov 10 22:41:45 myserver-VirtualBox sshd[28099]: Server listening on 0.0.0.0 p
Nov 10 22:41:45 myserver-VirtualBox sshd[28099]: Server listening on :: port 2
Nov 10 22:41:45 myserver-VirtualBox systemd[1]: Started OpenBSD Secure Shell s
my-server@myserver-VirtualBox:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
my-server@myserver-VirtualBox:~$ sudo ufw allow 22
Rules updated
Rules updated (v6)
my-server@myserver-VirtualBox:~$
```

```
Activities Terminal Oct 4 16:53
root@vsvv: /var/www/html/vsv
root@vsvv:/var/www/html# ls
index.html vsv vsv1
root@vsvv:/var/www/html# cd vsv
root@vsvv:/var/www/html/vsv# ls
index.html
root@vsvv:/var/www/html/vsv# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
    link/ether 08:00:27:9d:8c:4d brd ff:ff:ff:ff:ff:ff
    inet 192.168.237.165/24 brd 192.168.237.255 scope global dynamic noprefixro
        valid_lft 3179sec preferred_lft 3179sec
    inet6 2401:4900:4fec:4873:b913:7a88:a0bc:8e5f/64 scope global dynamic dyn
        valid_lft 6804sec preferred_lft 6804sec
    paddr noprefixroute
        valid_lft 6804sec preferred_lft 6804sec
    inet6 fe80::1b32:8c00:aa7b:6e2c/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
root@vsvv:/var/www/html/vsv#
```

After that try to connect your onprem vm into mobaxtreem

note

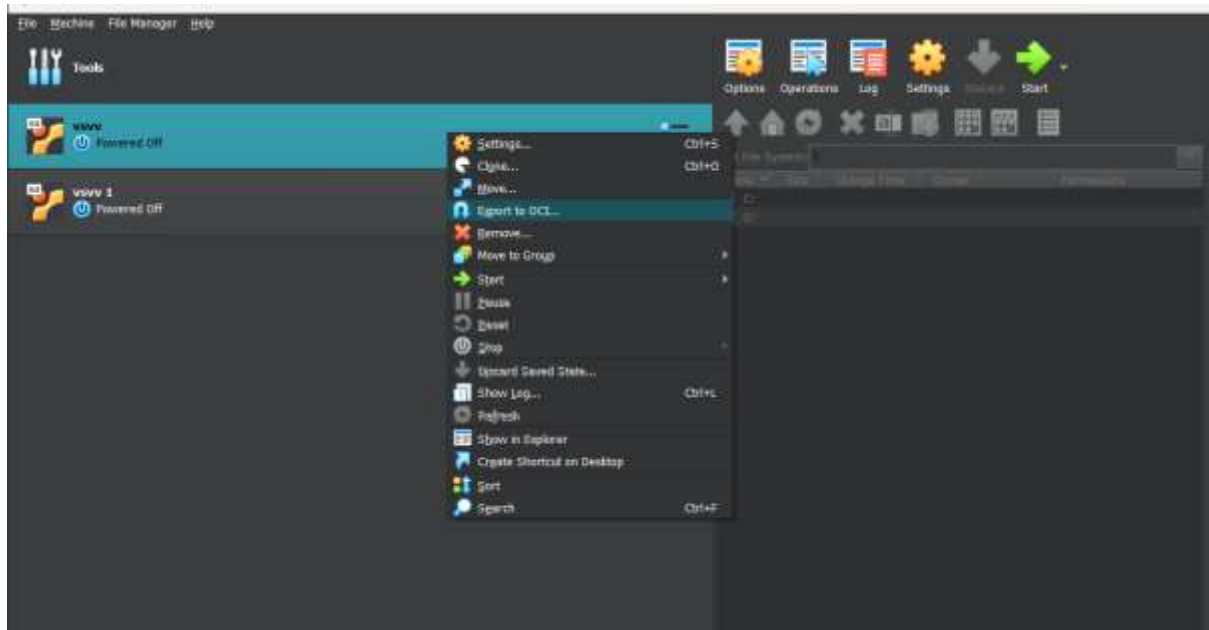
if u want to connect onprem vm we are able to connect from mobaxtreem or third party also

public ip

login as :vboxuser

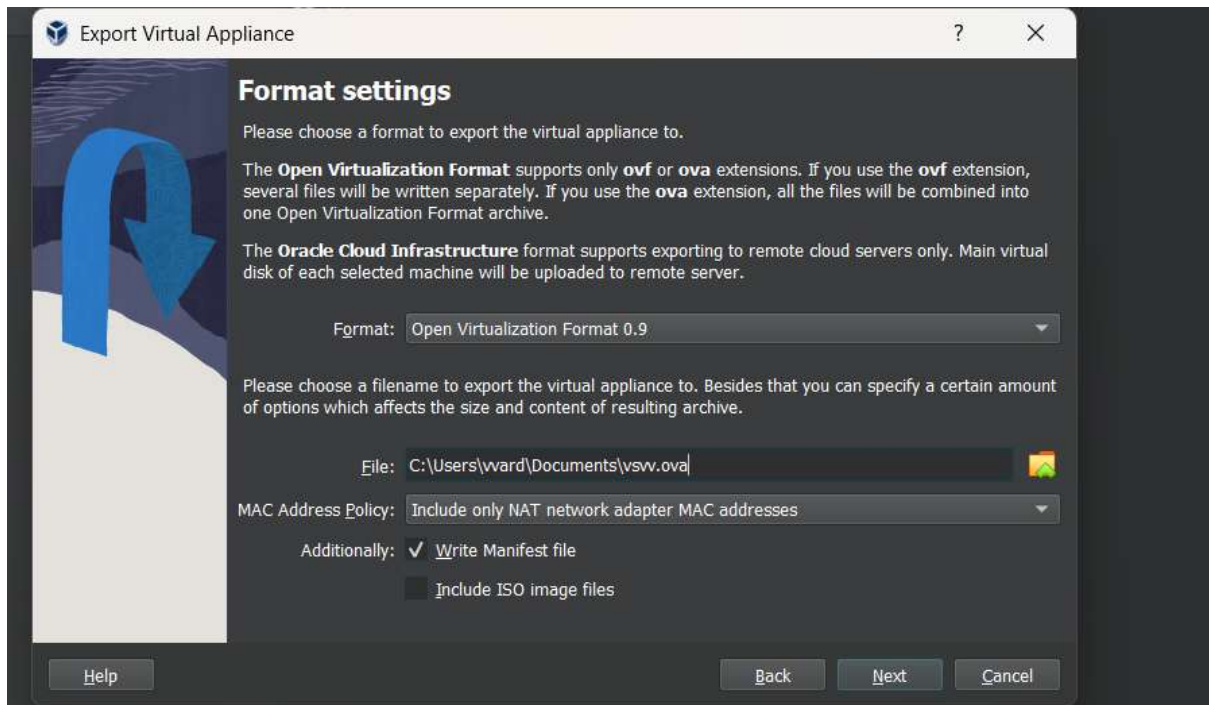
password:changeme or your custom onprem password

Next export vm to s3



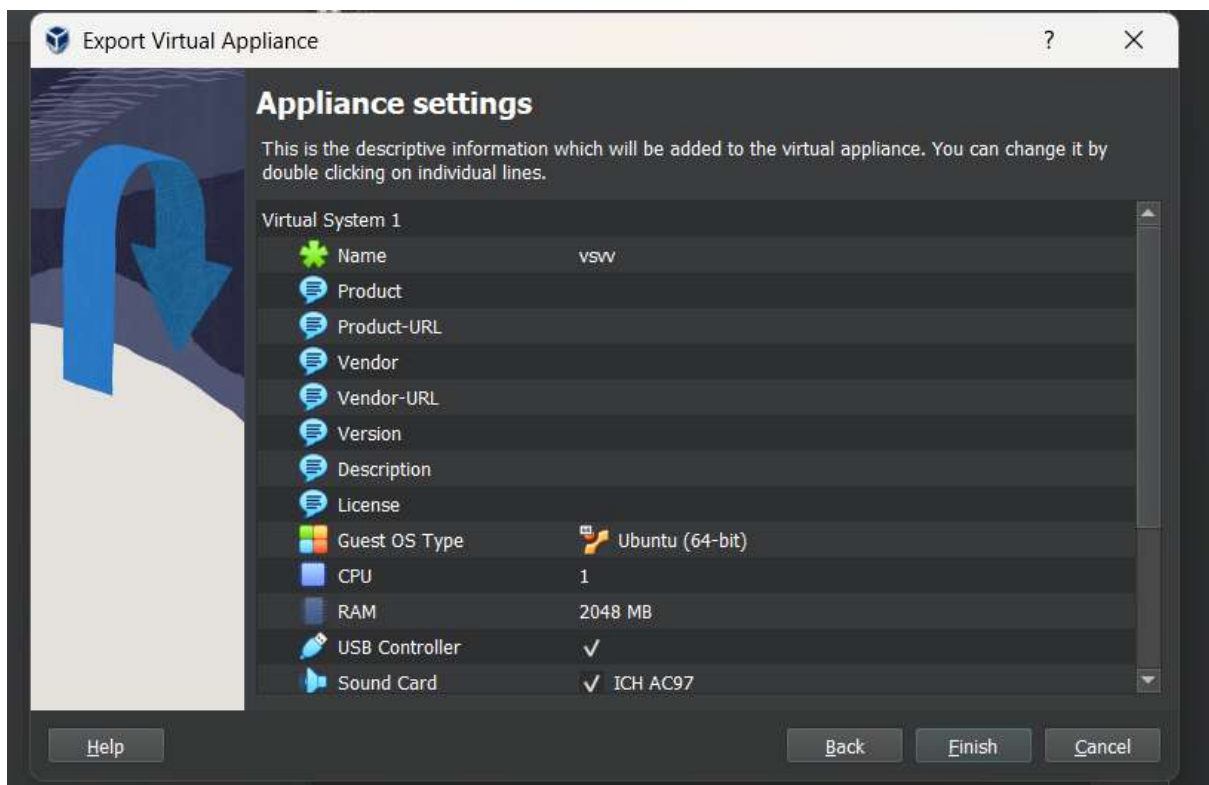
Right click on your vm

Click on export to oci



Select Window and Next

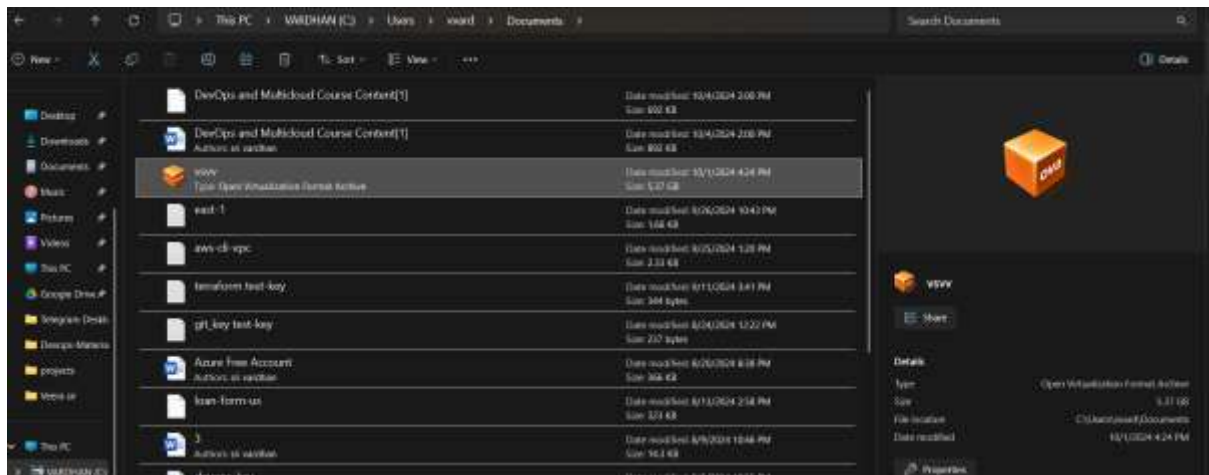
The above path this file is saved



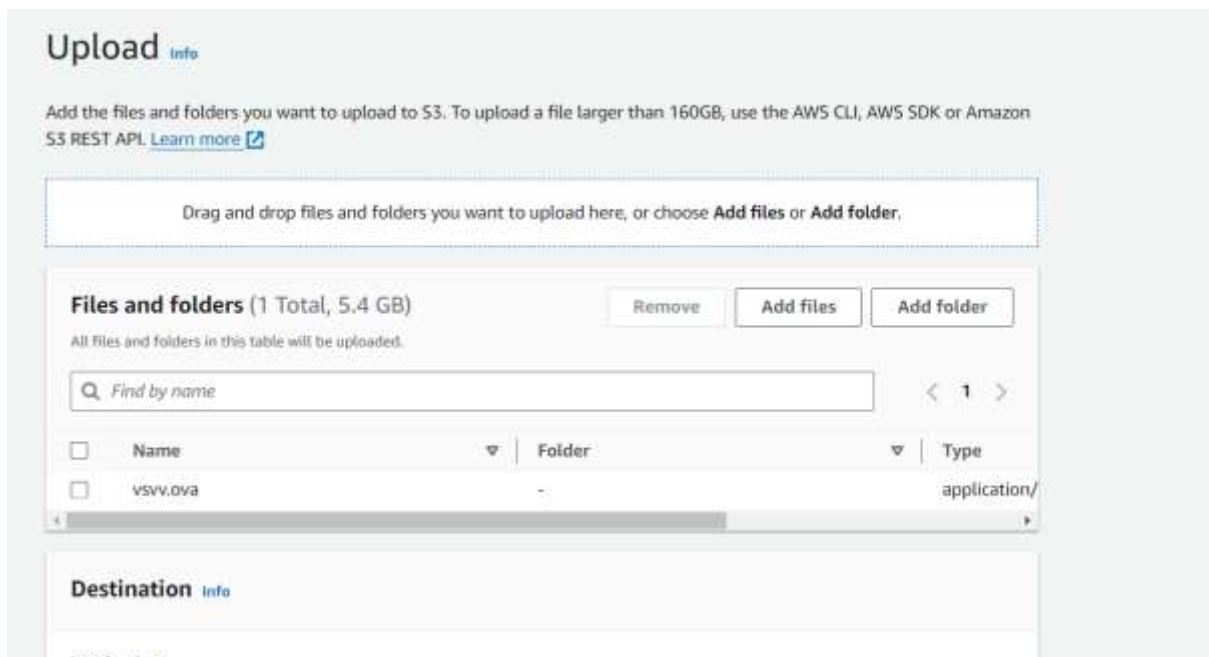
Click on finish

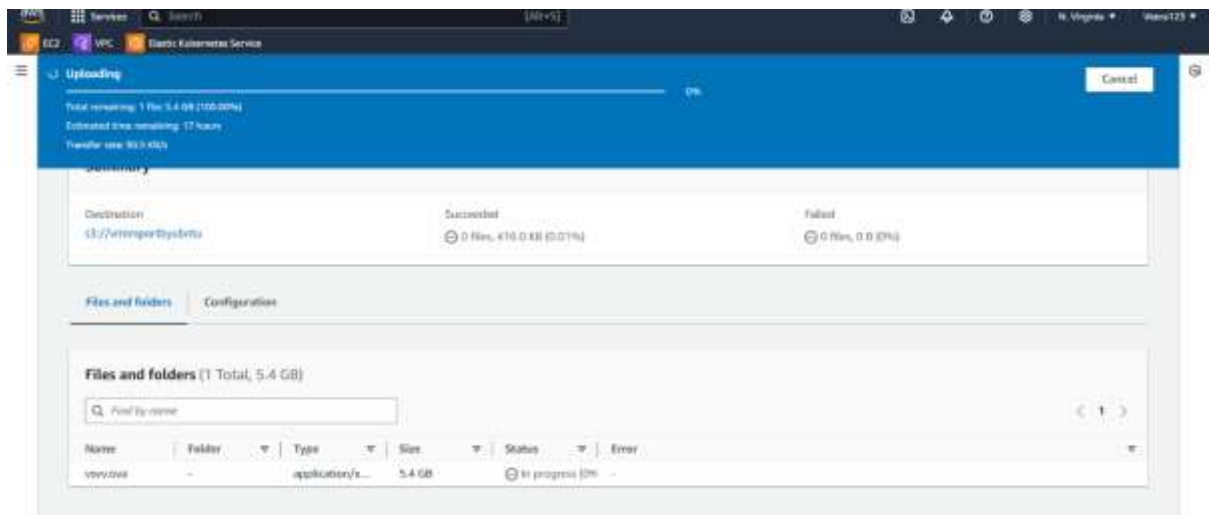
Go to the folder previously were created just check windows folder displayed over and save as type : .ova

After completing export appliance .Ova was created in our folder .we see image as below.



Open AWS console management and create s3 bucket and upload .ova file into your s3 bucket take nearly 40 mins .

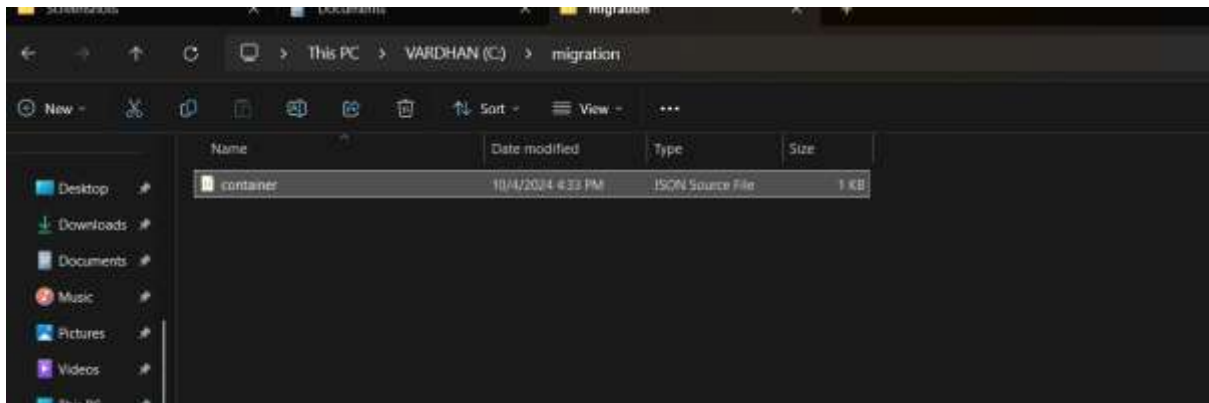




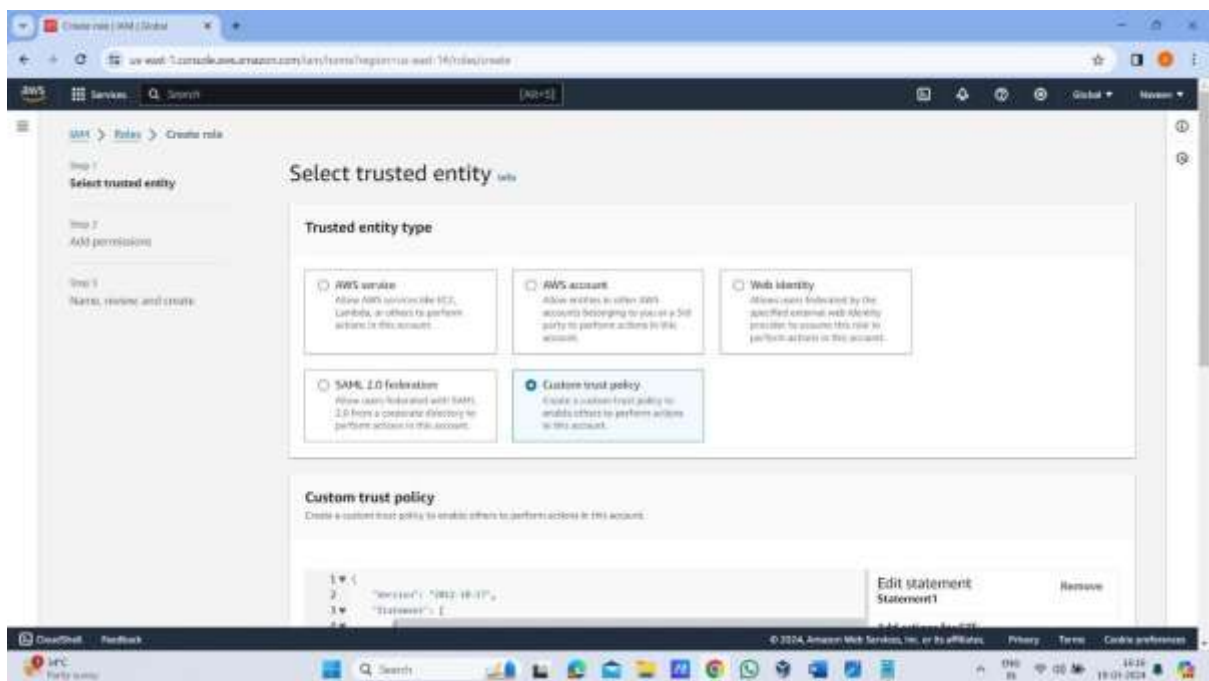
Meanwhile we can Create another folder (migration) in local laptop.

In that make template container.json file format by using vs code.

```
[
{
  "Description": "ubuntu os",
  "Format": "ova",
  "UserBucket":
  {
    "S3Bucket": "vmimportbyubuntu",
    "S3Key": "server-1.ova"
  }
}
```

After that go to console .Create IAM role select custom trust policy change principal as below mention:



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Principal": {
        "Service": "vmie.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
}
```

```
]
```

```
}
```

- After add permission EC2 , S3 full access to this role and save it .

Go Command prompt in local laptop.

- Configure aws using access and secret key.
- Run this command :

aws ec2 import-image --description "ubuntu" --disk-containers "file:///C:\migration\container.json" --role ami-policy

```
C:\Users\vvard>aws ec2 import-image --description 'linux' --disk-containers 'file:///C:\migration\container.json' --role ami-policy
{
  "Description": "linux",
  "ImportTaskId": "import-ami-019c07b289084f2af",
  "Progress": "1",
  "SnapshotDetails": [
    {
      "Description": "Window os",
      "DiskImageSize": 0.0,
      "Format": "OVA",
      "UserBucket": {
        "S3Bucket": "vmimportbyubntu",
        "S3Key": "server-1.ova"
      }
    }
  ],
  "Status": "active",
  "StatusMessage": "pending"
}
```

aws ec2 describe-import-image-tasks --import-task-ids "(copy above image import-ami-id)"

aws ec2 describe-import-image-tasks --import-task-ids " import-ami-0a958649a4d59db16"

```
C:\Users\vvard>aws ec2 describe-import-image-tasks --import-task-ids "import-ami-019c07b289084f2af"
{
  "ImportImageTasks": [
    {
      "Description": "linux",
      "ImportTaskId": "import-ami-019c07b289084f2af",
      "Progress": "19",
      "SnapshotDetails": [
        {
          "DiskImageSize": 6571246592.0,
          "Format": "VMDK",
          "Status": "active",
          "UserBucket": {
            "S3Bucket": "vmimportbyubuntu",
            "S3Key": "server-1.ova"
          }
        }
      ],
      "Status": "active",
      "StatusMessage": "converting",
      "Tags": []
    }
  ]
}
```

aws ec2 describe-import-image-tasks --import-task-ids "import-ami-019c07b289084f2af"

■ converting

```
C:\Users\vvard>aws ec2 describe-import-image-tasks --import-task-ids "import-ami-019c07b289084f2af"
{
  "ImportImageTasks": [
    {
      "Description": "linux",
      "ImportTaskId": "import-ami-019c07b289084f2af",
      "Progress": "20",
      "SnapshotDetails": [
        {
          "DiskImageSize": 6571246592.0,
          "Format": "VMDK",
          "Status": "completed",
          "UserBucket": {
            "S3Bucket": "vmimportbyubuntu",
            "S3Key": "server-1.ova"
          }
        }
      ],
      "Status": "active",
      "StatusMessage": "updating",
      "Tags": []
    }
  ]
}
```

aws ec2 describe-import-image-tasks --import-task-ids "import-ami-019c07b289084f2af"

■ updating

```
C:\Users\vvard>aws ec2 describe-import-image-tasks --import-task-ids "import-ami-019c07b289084f2af"
{
  "ImportImageTasks": [
    {
      "Architecture": "x86_64",
      "Description": "linux",
      "ImportTaskId": "import-ami-019c07b289084f2af",
      "LicenseType": "BYOL",
      "Platform": "Linux",
      "Progress": "39",
      "SnapshotDetails": [
        {
          "DeviceName": "/dev/sda1",
          "DiskImageSize": 6571246592.0,
          "Format": "VMDK",
          "Status": "completed",
          "UserBucket": {
            "S3Bucket": "vmimportbyubntu",
            "S3Key": "server-1.ova"
          }
        }
      ],
      "Status": "active",
      "StatusMessage": "booting",
      "Tags": []
    }
  ]
}
```

aws ec2 describe-import-image-tasks --import-task-ids "import-ami-019c07b289084f2af"

■ booting

```
C:\Users\vvard>aws ec2 describe-import-image-tasks --import-task-ids "import-ami-019c07b289084f2af"
{
  "ImportImageTasks": [
    {
      "Architecture": "x86_64",
      "Description": "linux",
      "ImportTaskId": "import-ami-019c07b289084f2af",
      "LicenseType": "BYOL",
      "Platform": "Linux",
      "Progress": "47",
      "SnapshotDetails": [
        {
          "DeviceName": "/dev/sda1",
          "DiskImageSize": 6571246592.0,
          "Format": "VMDK",
          "Status": "completed",
          "UserBucket": {
            "S3Bucket": "vmimportbyubntu",
            "S3Key": "server-1.ova"
          }
        }
      ],
      "Status": "active",
      "StatusMessage": "booted",
      "Tags": []
    }
  ]
}
```

aws ec2 describe-import-image-tasks --import-task-ids "import-ami-019c07b289084f2af"

■ booted

```
C:\Users\vvard>aws ec2 describe-import-image-tasks --import-task-ids "import-ami-019c07b289084f2af"
{
  "ImportImageTasks": [
    {
      "Architecture": "x86_64",
      "Description": "linux",
      "ImportTaskId": "import-ami-019c07b289084f2af",
      "LicenseType": "BYOL",
      "Platform": "Linux",
      "Progress": "55",
      "SnapshotDetails": [
        {
          "DeviceName": "/dev/sda1",
          "DiskImageSize": 6571246592.0,
          "Format": "VMDK",
          "Status": "completed",
          "UserBucket": {
            "S3Bucket": "vmimportbyubntu",
            "S3Key": "server-1.ova"
          }
        }
      ],
      "Status": "active",
      "StatusMessage": "preparing ami",
      "Tags": []
    }
  ]
}
```

aws ec2 describe-import-image-tasks --import-task-ids "import-ami-019c07b289084f2af"

- preparing ami

```
C:\Users\vvard>aws ec2 describe-import-image-tasks --import-task-ids "import-ami-019c07b289084f2af"
{
  "ImportImageTasks": [
    {
      "Architecture": "x86_64",
      "Description": "linux",
      "ImageId": "ami-0aec6929e7aceab06",
      "ImportTaskId": "import-ami-019c07b289084f2af",
      "LicenseType": "BYOL",
      "Platform": "Linux",
      "SnapshotDetails": [
        {
          "DeviceName": "/dev/sda1",
          "DiskImageSize": 6571246592.0,
          "Format": "VMDK",
          "SnapshotId": "snap-03da934a07e15124c",
          "Status": "completed",
          "UserBucket": {
            "S3Bucket": "vmimportbyubntu",
            "S3Key": "server-1.ova"
          }
        }
      ],
      "Status": "completed",
      "Tags": []
    }
  ]
}
```

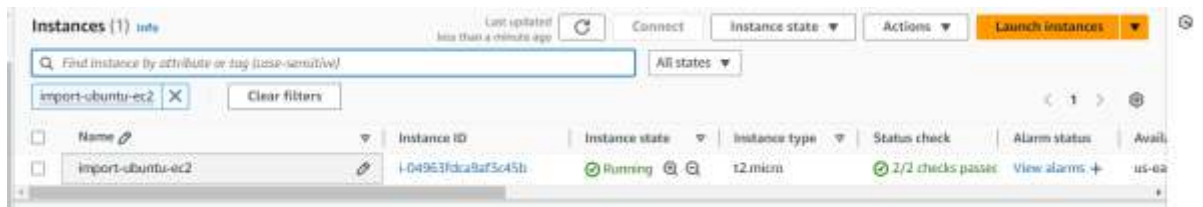
aws ec2 describe-import-image-tasks --import-task-ids "import-ami-019c07b289084f2af"

- completed
- After completed above process just go and check aws console over ec2 ami there you find imported image created by you.
- By using ami image launch ec2 instance.
- After launching instance take public ip

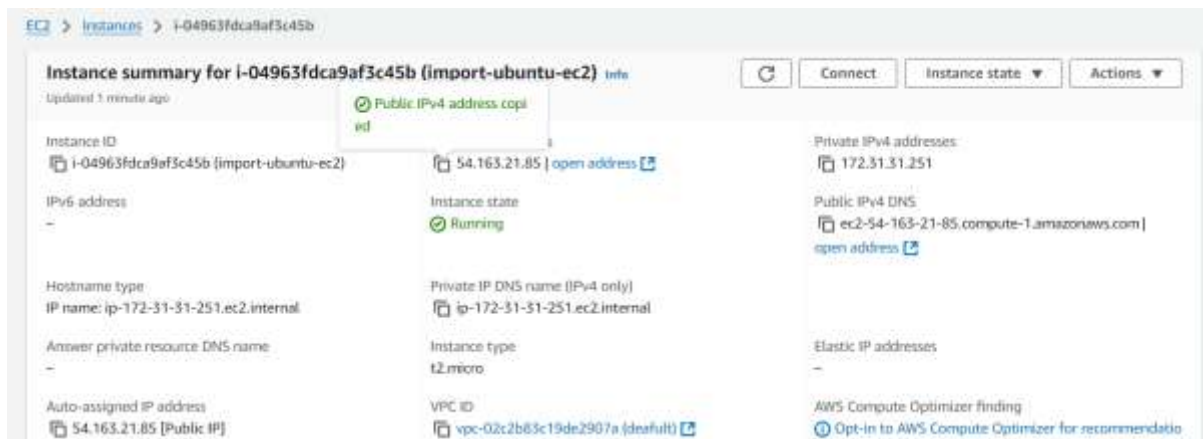


Above one is your ec2-

Select your ami and launch the ec2



Open your ec2



The the public ip and search In web



Enetr your path after the public ip



Yeah we are successfully migrated on premises vm into aws cloud

Before Migrating checks#####

Before migrating your **Ubuntu Linux VM** into AWS, it's essential to configure the VM to ensure that you can connect and access your applications after the migration. Here's what you need to install and configure to make the migration smooth and ensure post-migration connectivity and application access.

1. Enable SSH for Remote Access

Ensure **SSH** (Secure Shell) is installed, enabled, and configured to allow access to your Ubuntu VM once it is migrated to AWS.

- **Install and enable SSH:** If SSH is not installed, you can install it by running:

```
sudo apt update
```

```
sudo apt install openssh-server
```

- **Ensure SSH starts on boot:**

```
sudo systemctl enable ssh
```

```
sudo systemctl start ssh
```

- **Configure firewall (if applicable):** If you have ufw (Uncomplicated Firewall) enabled, ensure port **22** (the default SSH port) is allowed:

```
sudo ufw allow ssh
```

```
sudo ufw enable
```



```
sudo ufw status
```

2. Install Cloud-Init

Cloud-init is a widely used package on Ubuntu for initializing cloud instances. AWS relies on it to perform instance setup tasks (such as network configuration, SSH key injection, etc.) after launching the instance.

- **Install cloud-init:**

```
sudo apt update
```

```
sudo apt install cloud-init
```

- Ensure cloud-init is enabled and correctly configured for your environment. This helps with EC2 instance setup upon launch (setting up SSH access, hostname, and networking).

3. Install Necessary Network Drivers and Dependencies

AWS instances use specific networking drivers (e.g., **ENA** for enhanced networking) and other storage drivers. However, modern Linux distributions like Ubuntu come with these drivers built-in.

```
sudo apt update && sudo apt upgrade -y
```

4. Ensure the Application and Services Start on Boot

After the migration, you'll want to ensure that your application starts automatically when the VM boots up on AWS EC2.

- **Configure your application to start on boot** using systemd. For example, if your application is a web server, you would ensure services like Apache or Nginx start automatically:

```
sudo systemctl enable apache2 # For Apache
```

```
add application into path /var/www/html
```

```
sudo systemctl start apache2
```

note

if u want to connect ec2 we are able to connect from mobaxtreem or third party only

public ip

login as :vboxuser

password:changeme or your custom onprem password