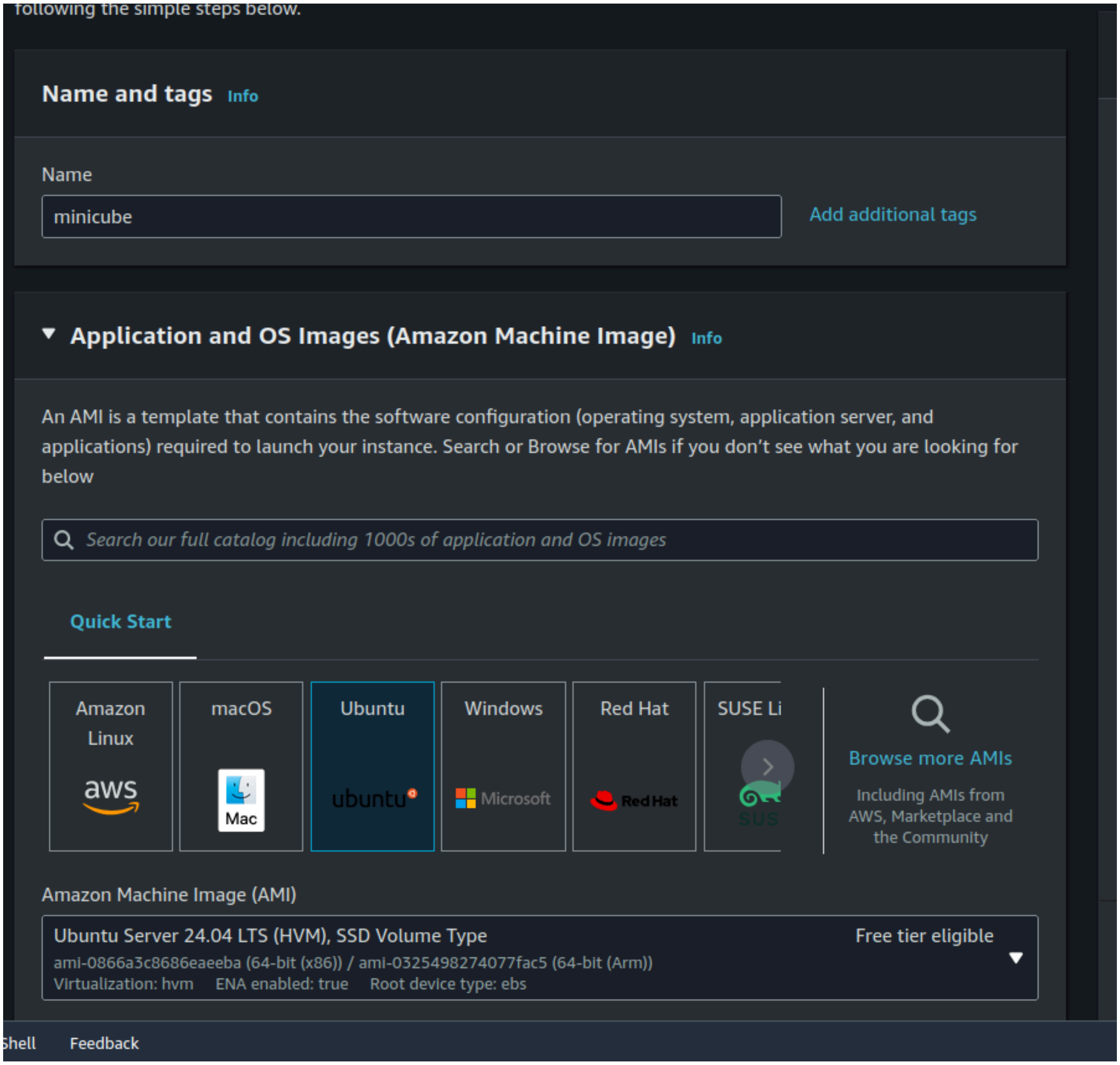
Task : Install and Setup Minikube on your system

What is Minikube ?

Minikube is a tool that lets you run a single-node Kubernetes cluster locally on your machine. It's designed for developers to easily test and develop Kubernetes applications without needing a full-scale production cluster.

Step 1: Create an ec2 t2.medium instance and launch it .



following the simple steps below.

Name and tags Info

Name

minicube

Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)

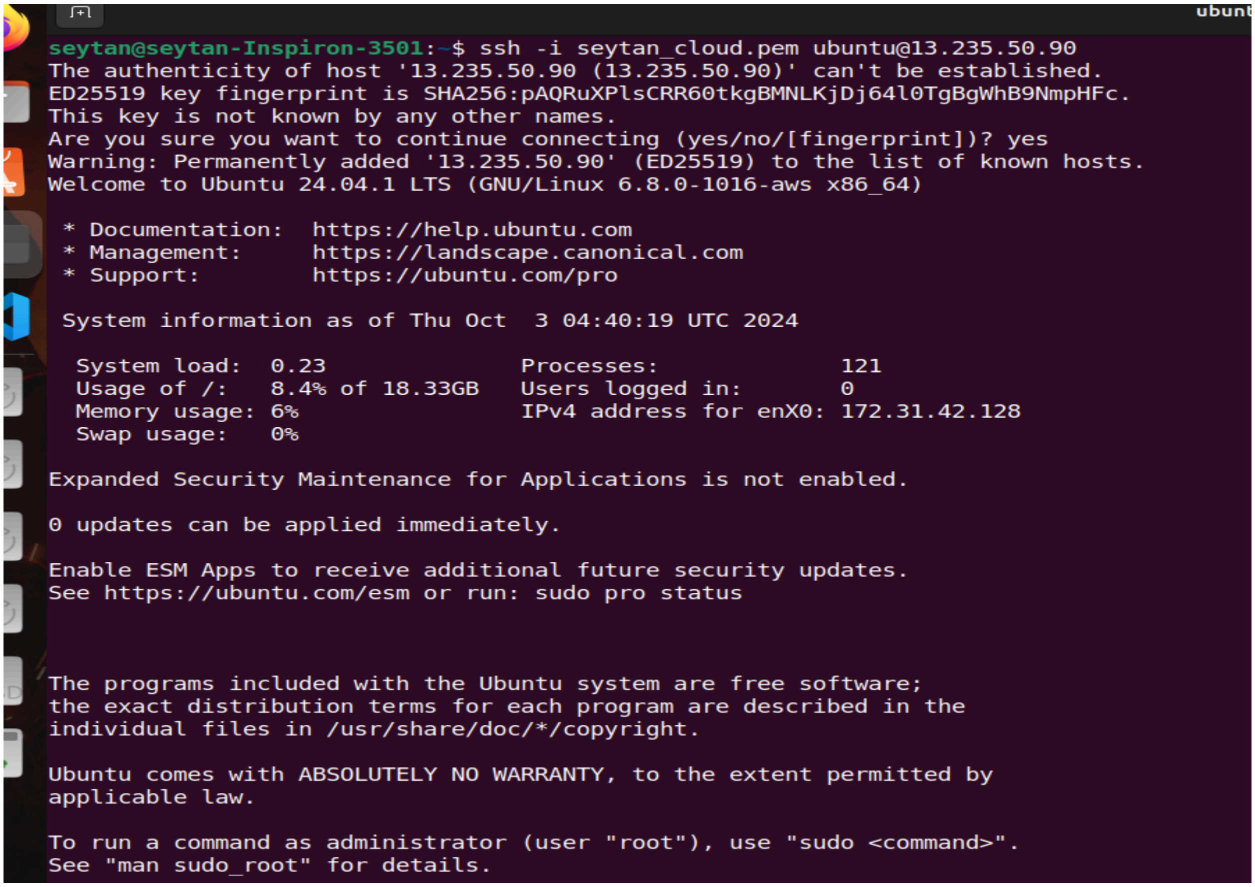
Ubuntu Server 24.04 LTS (HVM), SSD Volume Type Free tier eligible

ami-0866a3c8686eaeeba (64-bit (x86)) / ami-0325498274077fac5 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Step 2: Take ssh of instance.

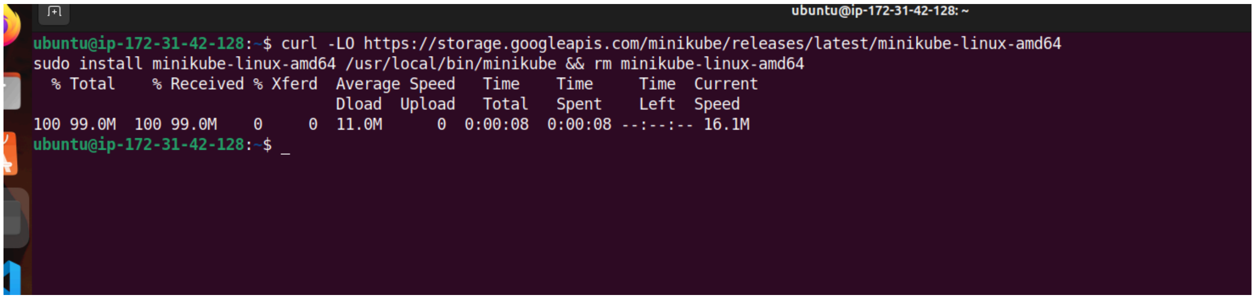
Ssh -i private_key.pem ubuntu@public_ip



```
seytan@seytan-Inspiron-3501:~$ ssh -i seytan_cloud.pem ubuntu@13.235.50.90
The authenticity of host '13.235.50.90 (13.235.50.90)' can't be established.
ED25519 key fingerprint is SHA256:pAQRuXPlsCRR60tkgBMNLKjDj64l0TgBgWhB9NmpHFc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.235.50.90' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

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

 System information as of Thu Oct  3 04:40:19 UTC 2024

  System load:  0.23              Processes:             121
  Usage of /:   8.4% of 18.33GB   Users logged in:       0
  Memory usage: 6%                IPv4 address for enX0: 172.31.42.128
  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 programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Step 3:Install the Minikube by using following commands.

curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64

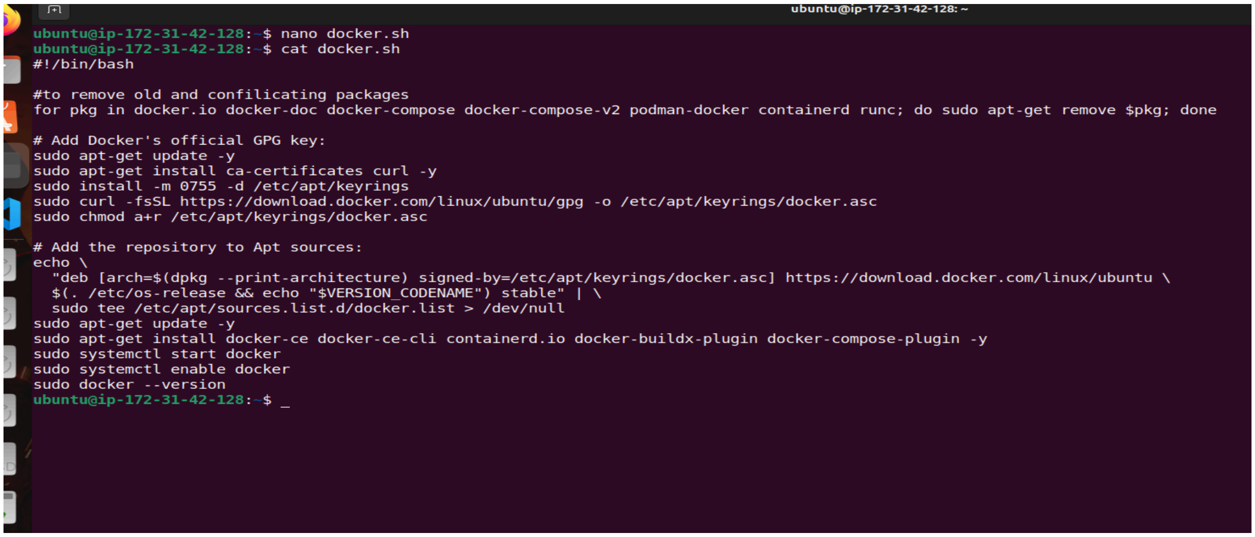


```
ubuntu@ip-172-31-42-128:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 99.0M  100 99.0M    0     0  11.0M      0  0:00:08  0:00:08 --:--:-- 16.1M
ubuntu@ip-172-31-42-128:~$ _
```

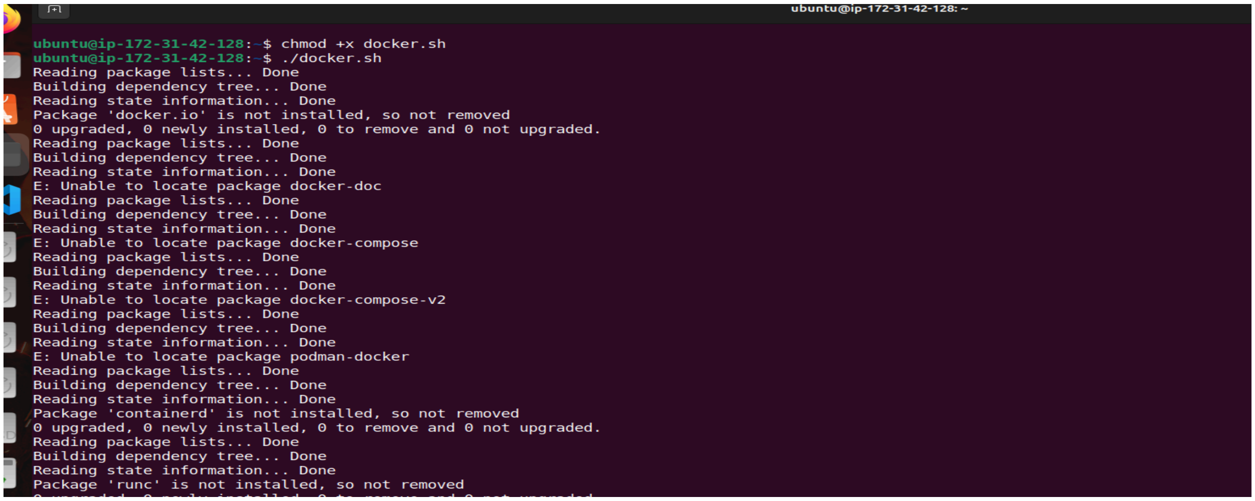
Step 4:Now we need to install docker .

Why we need to install docker ?
→ Minikube needs Docker to run containers in the local Kubernetes cluster, as Docker acts as the container runtime. It also enables Minikube to create and manage the node and workloads efficiently.

Create a script to install docker and give appropriate permissions and run it.



```
ubuntu@ip-172-31-42-128:~$ nano docker.sh
ubuntu@ip-172-31-42-128:~$ cat docker.sh
#!/bin/bash

#to remove old and confilicating packages
for pkg in docker.io docker-doc docker-compose docker-compose-v2 podman-docker containerd runc; do sudo apt-get remove $pkg; done

# Add Docker's official GPG key:
sudo apt-get update -y
sudo apt-get install ca-certificates curl -y
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update -y
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y
sudo systemctl start docker
sudo systemctl enable docker
sudo docker --version
ubuntu@ip-172-31-42-128:~$ _
```

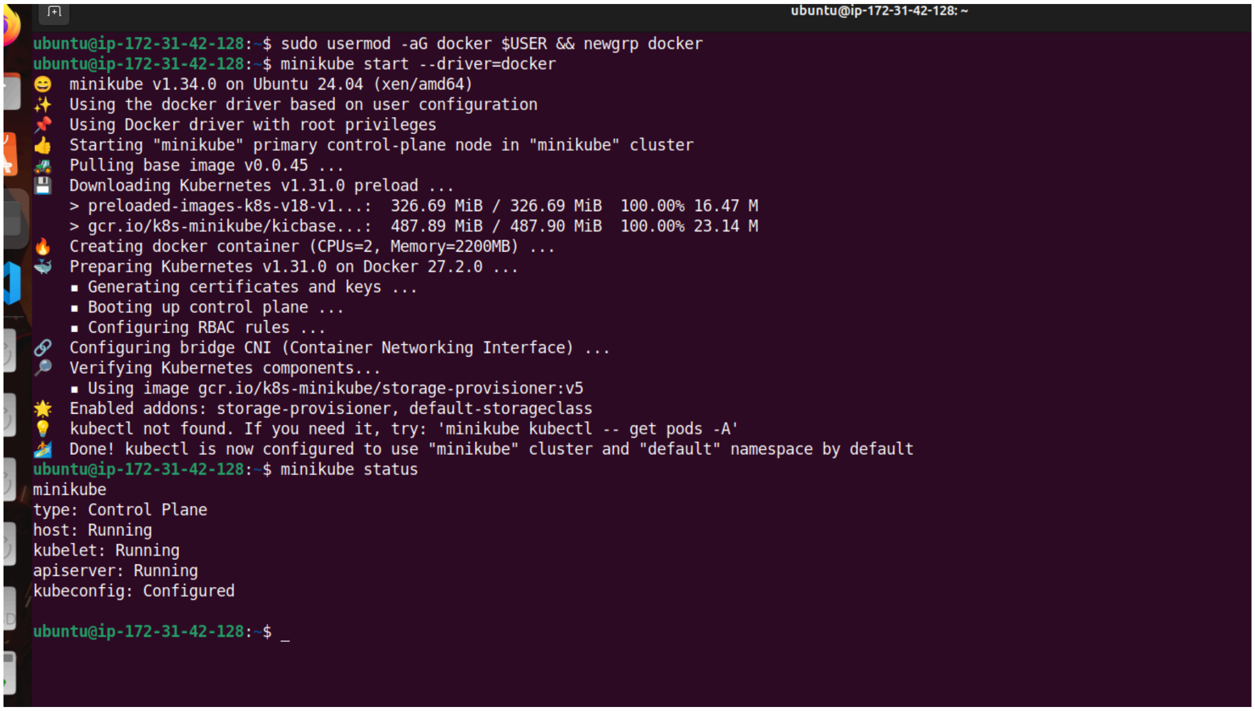


```
ubuntu@ip-172-31-42-128:~$ chmod +x docker.sh
ubuntu@ip-172-31-42-128:~$ ./docker.sh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package 'docker.io' is not installed, so not removed
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package docker-doc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package docker-compose
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package docker-compose-v2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package podman-docker
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package 'containerd' is not installed, so not removed
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package 'runc' is not installed, so not removed
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Step 5: Now add the local user to the docker group .
Command: sudo usermod -aG docker $USER && newgrp docker.

After that run the following command to start minikube
→ <u>minikube start –driver=docker</u>

Check the status of minikube by <u>minikube status</u>



```
ubuntu@ip-172-31-42-128:~$ sudo usermod -aG docker $USER && newgrp docker
ubuntu@ip-172-31-42-128:~$ minikube start --driver=docker
😄  minikube v1.34.0 on Ubuntu 24.04 (xen/amd64)
✨  Using the docker driver based on user configuration
📌  Using Docker driver with root privileges
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.45 ...
💾  Downloading Kubernetes v1.31.0 preload ...
    > preloaded-images-k8s-v18-v1...:  326.69 MiB / 326.69 MiB  100.00% 16.47 M
    > gcr.io/k8s-minikube/kicbase...:  487.89 MiB / 487.90 MiB  100.00% 23.14 M
🔥  Creating docker container (CPUs=2, Memory=2200MB) ...
🐳  Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
💡  kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
ubuntu@ip-172-31-42-128:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

ubuntu@ip-172-31-42-128:~$ _
```

Note : Setup alias for minikube kubectl – :
<u>alias kubectl="minikube kubectl --"</u>