Kubernetes (K8S)

Mini Kube Cluster Setup

By Mr. Ashok

Mini Kube

-> As the name suggests, minikube is a single node Kubernetes (k8s) cluster.

-> Anyone who is new to the Kubernetes and wants to learn and try deploying application on it, then minikube is the solution.

-> Minikube provides a command line interface to manage Kubernetes (k8s) cluster and its component.

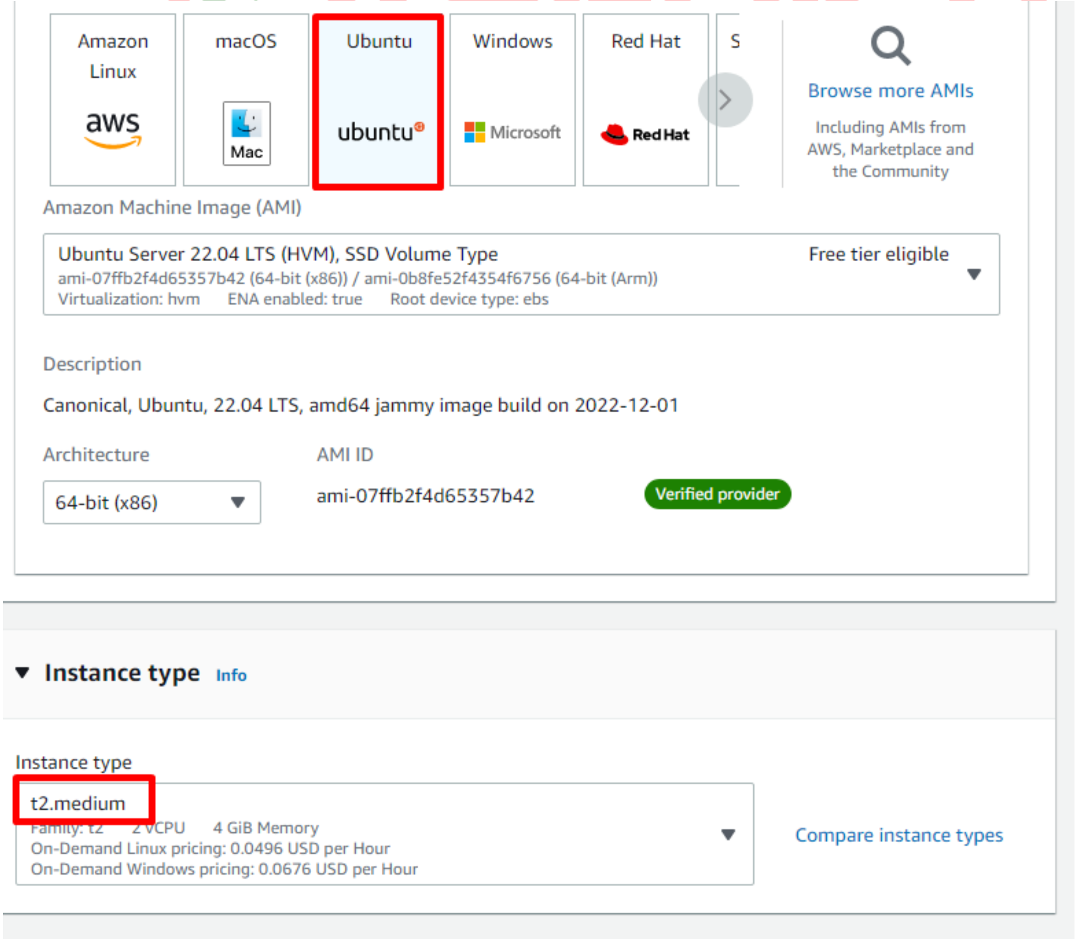
Minikube System Requirements

- 2 GB RAM or more
- 2 CPU / vCPU or more
- 20 GB free hard disk space or more
- Docker / Virtual Machine Manager – KVM & VirtualBox

Note: We will be using Docker container as a base for minikube.

Step -1) Launch AWS EC2 Medium instance with Ubuntu AMI (Ubuntu Server 20.04)

Note: t2.medium instance we are using here



Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-07ffb2f4d65357b42 (64-bit (x86)) / ami-0b8fe52f4354f6756 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

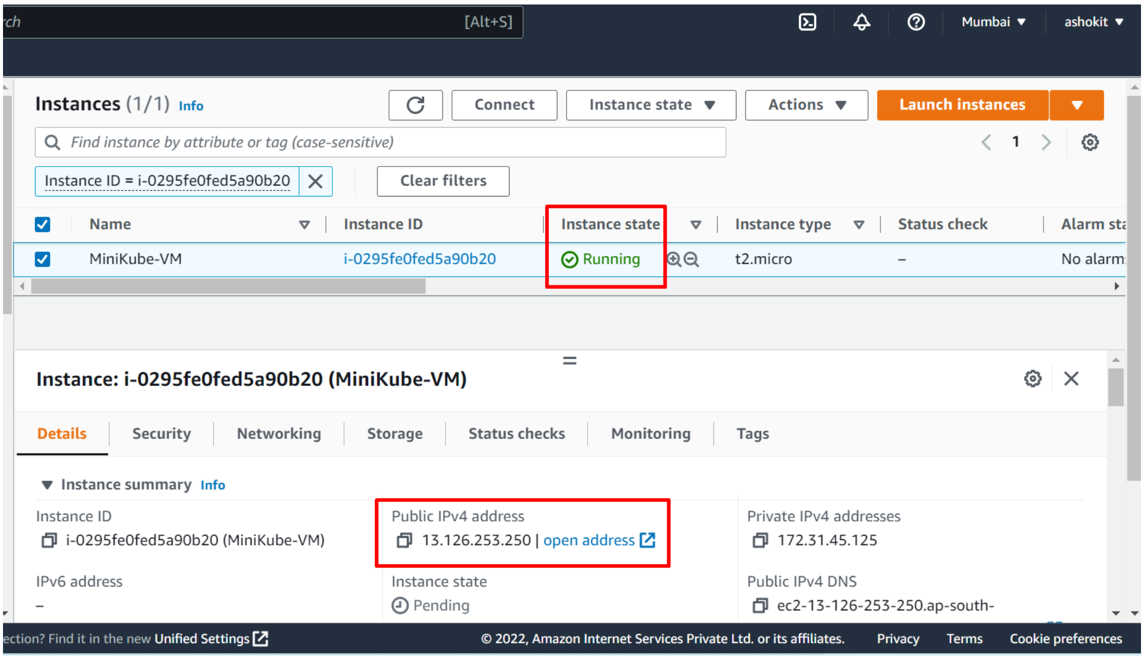
Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2022-12-01

Architecture: 64-bit (x86)

AMI ID: ami-07ffb2f4d65357b42

Instance type

Instance type: t2.medium
Family: t2 2 vCPU 4 GiB Memory
On-Demand Linux pricing: 0.0496 USD per Hour
On-Demand Windows pricing: 0.0676 USD per Hour



Instances (1/1) Info

Instance ID = i-0295fe0fed5a90b20

Name	Instance ID	Instance state	Instance type	Status check	Alarm sta
MiniKube-VM	i-0295fe0fed5a90b20	Running	t2.micro	–	No alarm

Instance: i-0295fe0fed5a90b20 (MiniKube-VM)

Instance summary Info

Instance ID: i-0295fe0fed5a90b20 (MiniKube-VM)

Public IPv4 address: 13.126.253.250 | open address

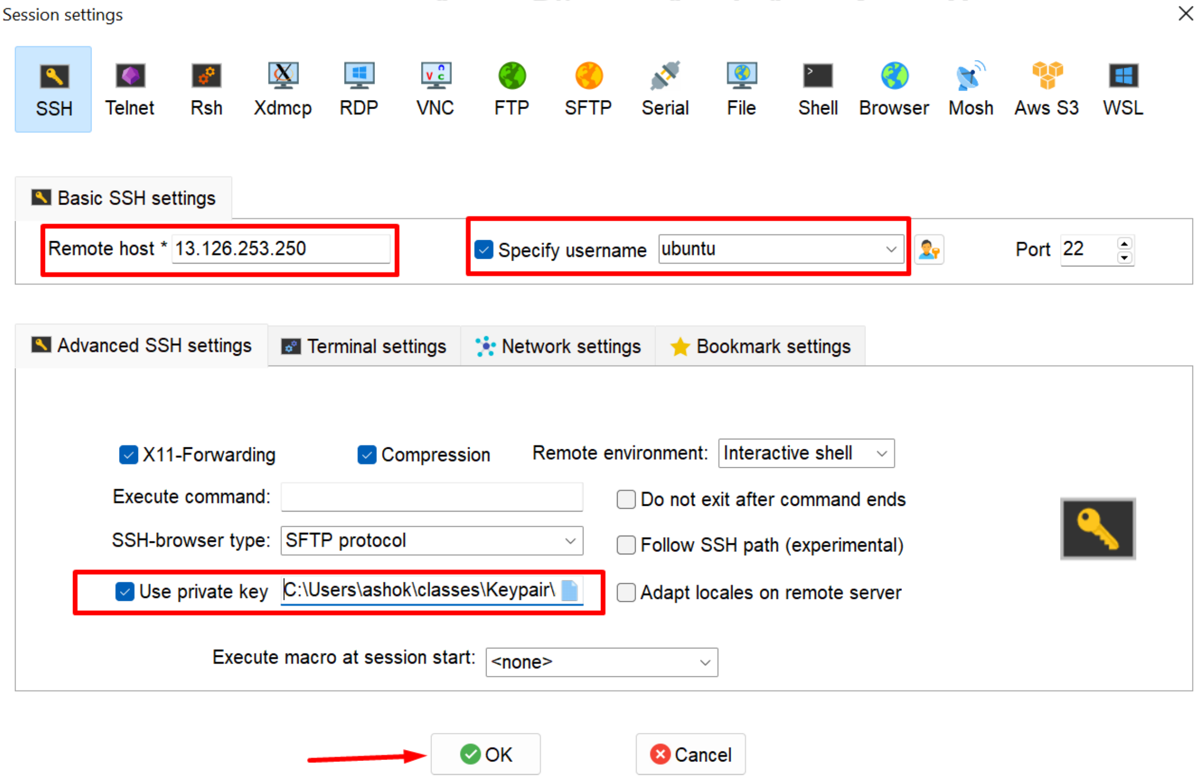
Private IPv4 addresses: 172.31.45.125

IPv6 address: –

Instance state: Pending

Public IPv4 DNS: ec2-13-126-253-250.ap-south-

Step – 2) Connect to Ubuntu VM using MobaXterm



Session settings

SSH selected

Basic SSH settings

Remote host * 13.126.253.250

☑ Specify username ubuntu

Port 22

Advanced SSH settings

☑ X11-Forwarding ☑ Compression Remote environment: Interactive shell

Execute command:

□ Do not exit after command ends

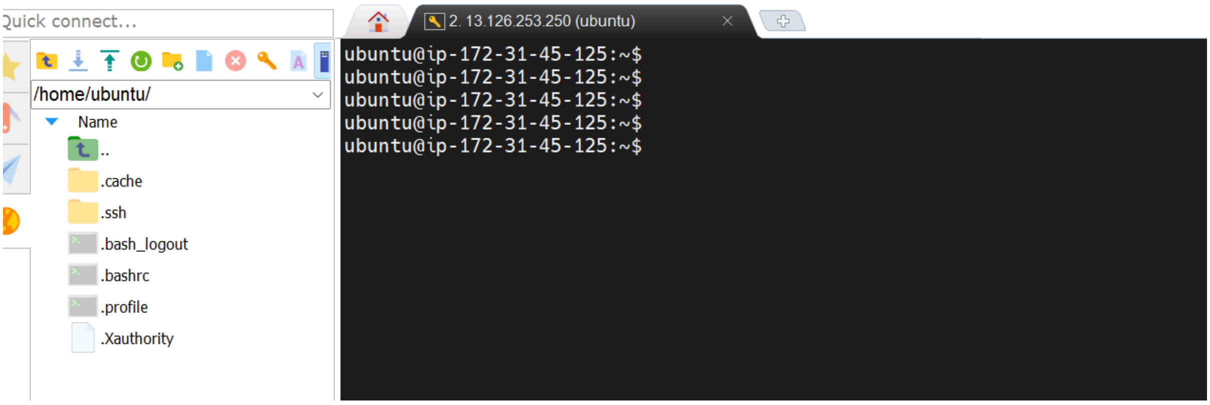
SSH-browser type: SFTP protocol

□ Follow SSH path (experimental)

☑ Use private key C:\Users\ashok\classes\Keypair\

□ Adapt locales on remote server

Execute macro at session start: <none>



```
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$
```

Note: once we connect to Ec2 instance, it will open terminal like above

Step – 3) Apply all updates of existing packages of your system by executing the following apt commands

$ sudo apt update -y

$ sudo apt upgrade -y

Once all the updates are installed then reboot your system once.

$ sudo reboot

Step – 4) Install Docker by executing below commands

$ curl -fsSL https://get.docker.com -o get-docker.sh

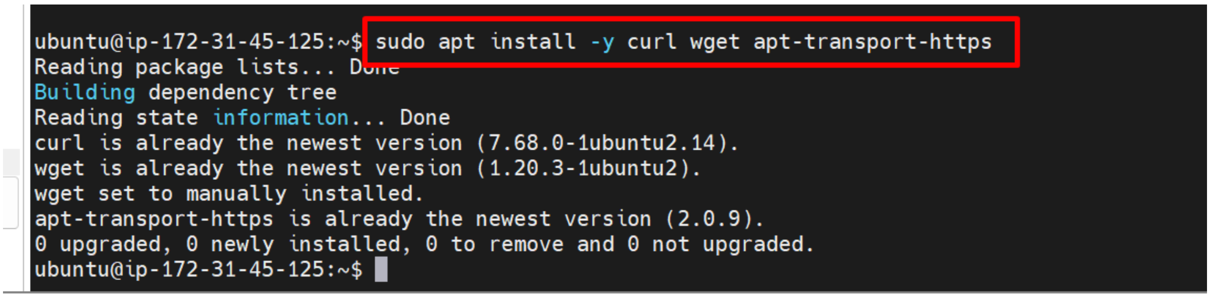
$ sudo sh get-docker.sh

$ sudo usermod -aG docker $USER && newgrp docker

Step – 5) Install Minikube dependencies

Install the following minikube dependencies by running below command

$ sudo apt install -y curl wget apt-transport-https

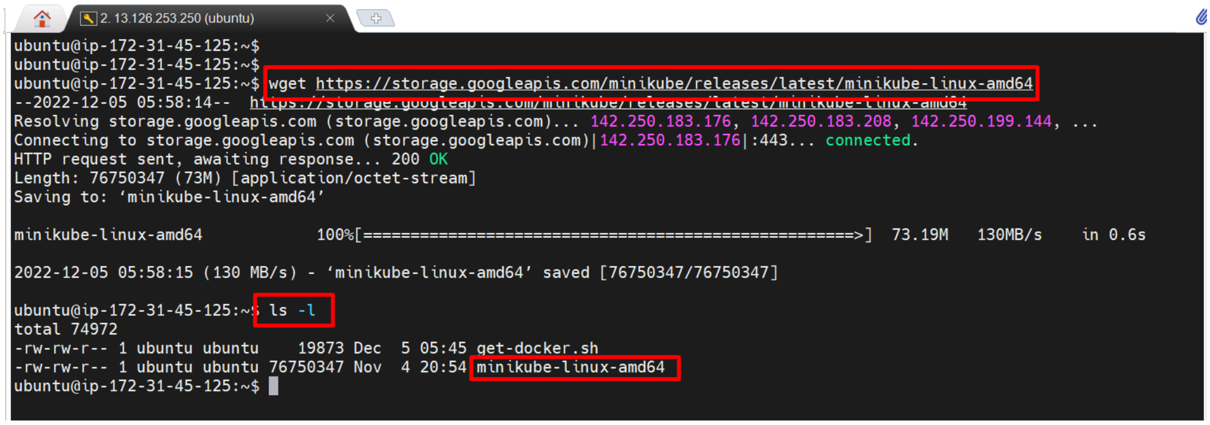


```
ubuntu@ip-172-31-45-125:~$ sudo apt install -y curl wget apt-transport-https
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.68.0-1ubuntu2.14).
wget is already the newest version (1.20.3-1ubuntu2).
wget set to manually installed.
apt-transport-https is already the newest version (2.0.9).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-45-125:~$
```

Step - 6) Download Minikube Binary

Use the following wget command to download latest minikube binary

$ wget https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64



```
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$ wget https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
--2022-12-05 05:58:14--  https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
Resolving storage.googleapis.com (storage.googleapis.com)... 142.250.183.176, 142.250.183.208, 142.250.199.144, ...
Connecting to storage.googleapis.com (storage.googleapis.com)|142.250.183.176|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 76750347 (73M) [application/octet-stream]
Saving to: ‘minikube-linux-amd64’

minikube-linux-amd64            100%[==================================================>]  73.19M   130MB/s    in 0.6s

2022-12-05 05:58:15 (130 MB/s) - ‘minikube-linux-amd64’ saved [76750347/76750347]

ubuntu@ip-172-31-45-125:~$ ls -l
total 74972
-rw-rw-r-- 1 ubuntu ubuntu    19873 Dec  5 05:45 get-docker.sh
-rw-rw-r-- 1 ubuntu ubuntu 76750347 Nov  4 20:54 minikube-linux-amd64
ubuntu@ip-172-31-45-125:~$
```

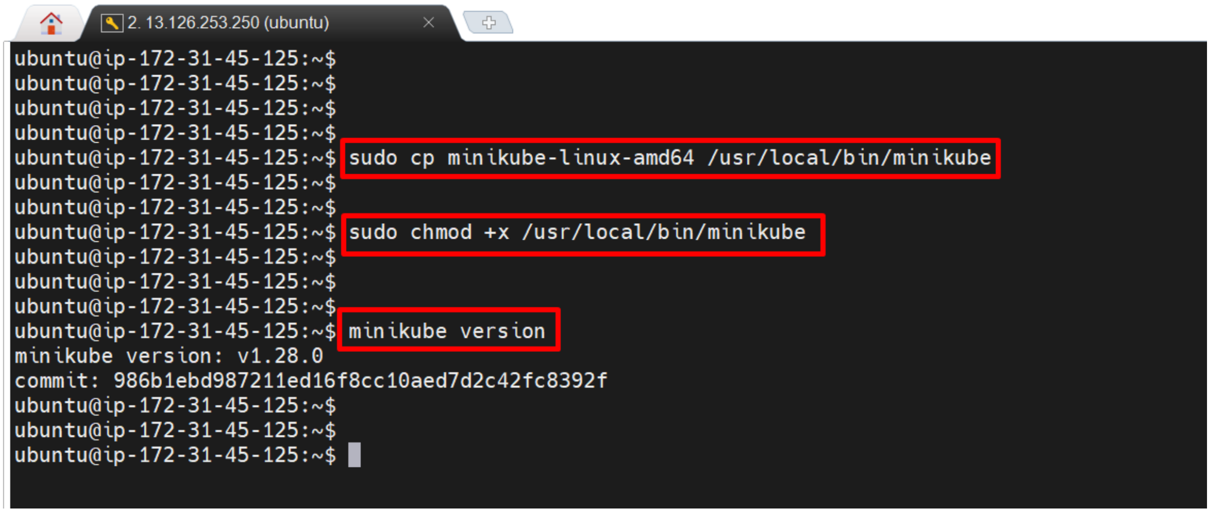
Once the binary is downloaded, copy it to the path /usr/local/bin and set the executable permissions on it by executing below commands

$ sudo cp minikube-linux-amd64 /usr/local/bin/minikube

$ sudo chmod +x /usr/local/bin/minikube

Verify the minikube version by executing below command

$ minikube version



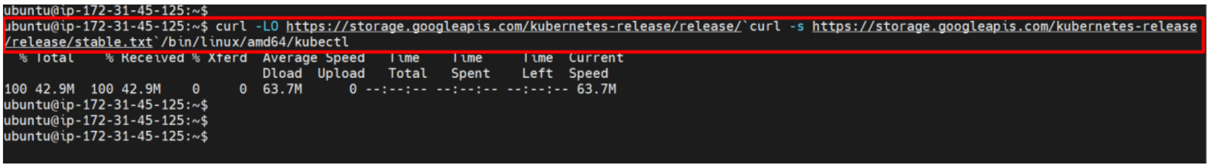
```
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$ sudo cp minikube-linux-amd64 /usr/local/bin/minikube
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$ sudo chmod +x /usr/local/bin/minikube
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$ minikube version
minikube version: v1.28.0
commit: 986b1ebd987211ed16f8cc10aed7d2c42fc8392f
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$
```

Step - 7) Install Kubectl utility

-> Kubectl is a command line utility which is used to interact with Kubernetes cluster.

-> It is used for managing deployments, service and pods etc. Use below curl command to download latest version of kubectl.

$ curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl



```
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$ curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release
/release/stable.txt`/bin/linux/amd64/kubectl
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 42.9M  100 42.9M    0     0  63.7M      0 --:--:-- --:--:-- --:--:-- 63.7M
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$
```

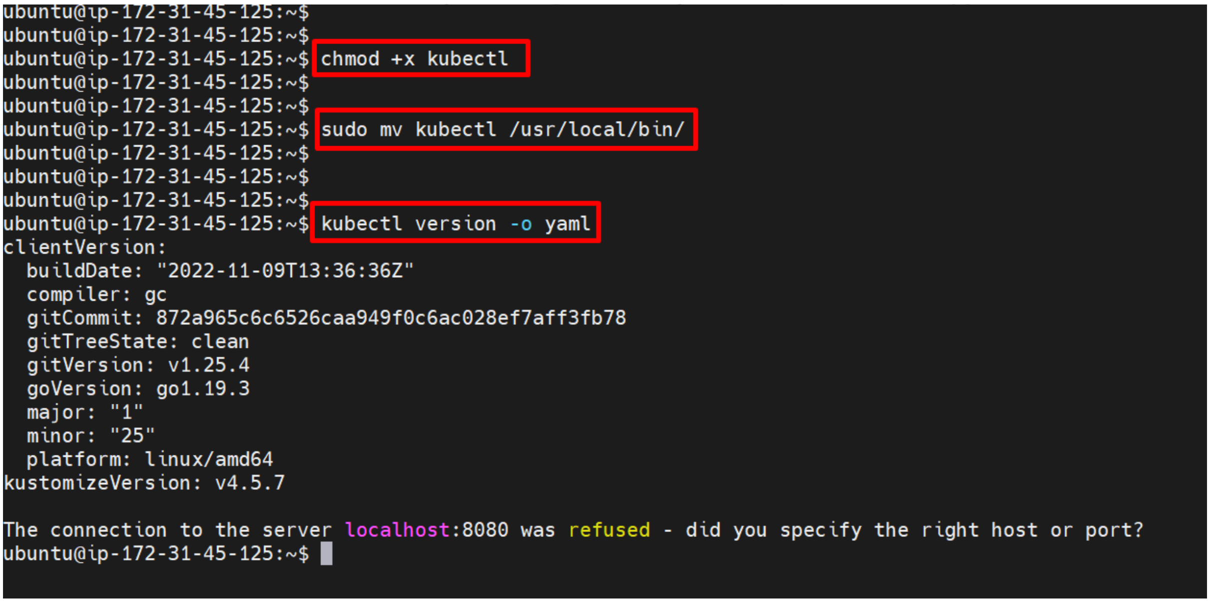
Once kubectl is downloaded then set the executable permissions on kubectl binary and move it to the path /usr/local/bin by executing below commands

$ chmod +x kubectl

$ sudo mv kubectl /usr/local/bin/

Now verify the kubectl version

$ kubectl version -o yaml

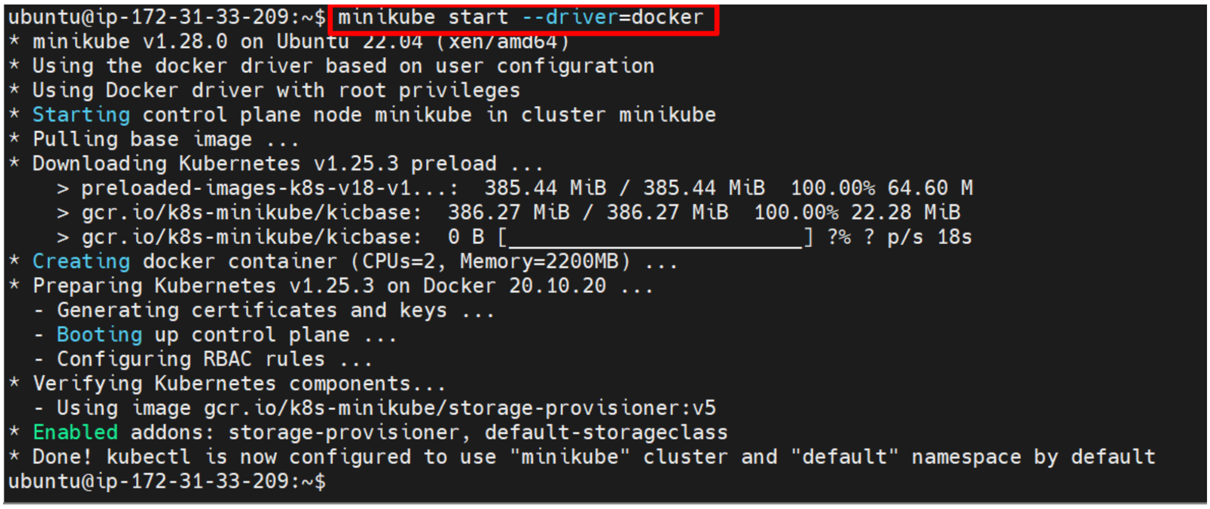


```
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$ chmod +x kubectl
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$ sudo mv kubectl /usr/local/bin/
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$
ubuntu@ip-172-31-45-125:~$ kubectl version -o yaml
clientVersion:
  buildDate: "2022-11-09T13:36:36Z"
  compiler: gc
  gitCommit: 872a965c6c6526caa949f0c6ac028ef7aff3fb78
  gitTreeState: clean
  gitVersion: v1.25.4
  goVersion: go1.19.3
  major: "1"
  minor: "25"
  platform: linux/amd64
kustomizeVersion: v4.5.7

The connection to the server localhost:8080 was refused - did you specify the right host or port?
ubuntu@ip-172-31-45-125:~$
```

Step – 8) Start minikube

As we are already stated in the beginning that we would be using docker as base for minikube, so start the minikube with the docker driver, run

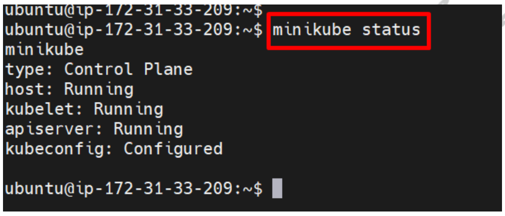
$ minikube start --driver=docker



```
ubuntu@ip-172-31-33-209:~$ minikube start --driver=docker
* minikube v1.28.0 on Ubuntu 22.04 (xen/amd64)
* Using the docker driver based on user configuration
* Using Docker driver with root privileges
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Downloading Kubernetes v1.25.3 preload ...
    > preloaded-images-k8s-v18-v1...:  385.44 MiB / 385.44 MiB  100.00% 64.60 M
    > gcr.io/k8s-minikube/kicbase:  386.27 MiB / 386.27 MiB  100.00% 22.28 MiB
    > gcr.io/k8s-minikube/kicbase:  0 B [________________________] ?% ? p/s 18s
* Creating docker container (CPUs=2, Memory=2200MB) ...
* Preparing Kubernetes v1.25.3 on Docker 20.10.20 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
ubuntu@ip-172-31-33-209:~$
```

Perfect, above confirms that minikube cluster has been configured and started successfully.

Run below minikube command to check status,

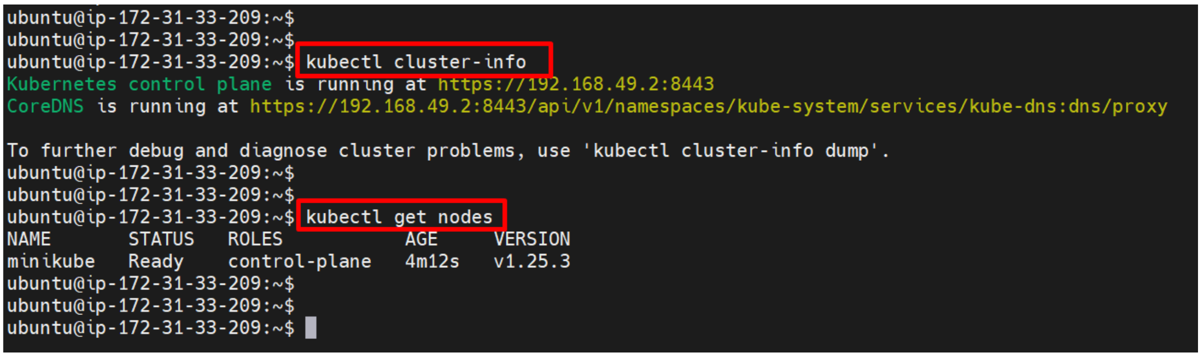


```
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

ubuntu@ip-172-31-33-209:~$
```

Run following kubectl command to verify the Kubernetes version, node status and cluster info.

$ kubectl cluster-info

$ kubectl get nodes



```
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$ kubectl get nodes
NAME       STATUS   ROLES           AGE     VERSION
minikube   Ready    control-plane   4m12s   v1.25.3
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$
```

Step - 9) Verify Minikube Installation By Deploying Nginx Server

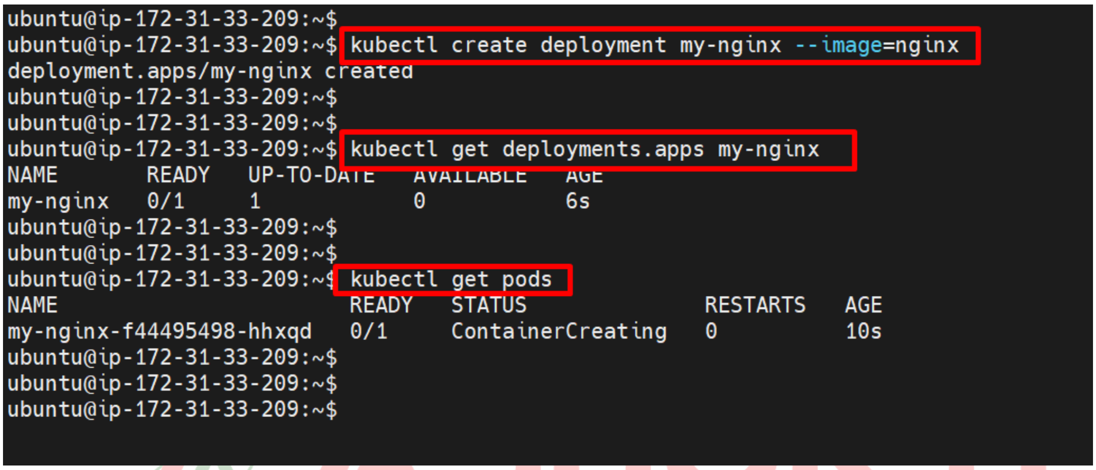
# Run below kubectl command to install nginx based deployment.

$ kubectl create deployment my-nginx --image=nginx

# Run following kubectl command to verify deployment status

$ kubectl get deployments.apps my-nginx

$ kubectl get pods

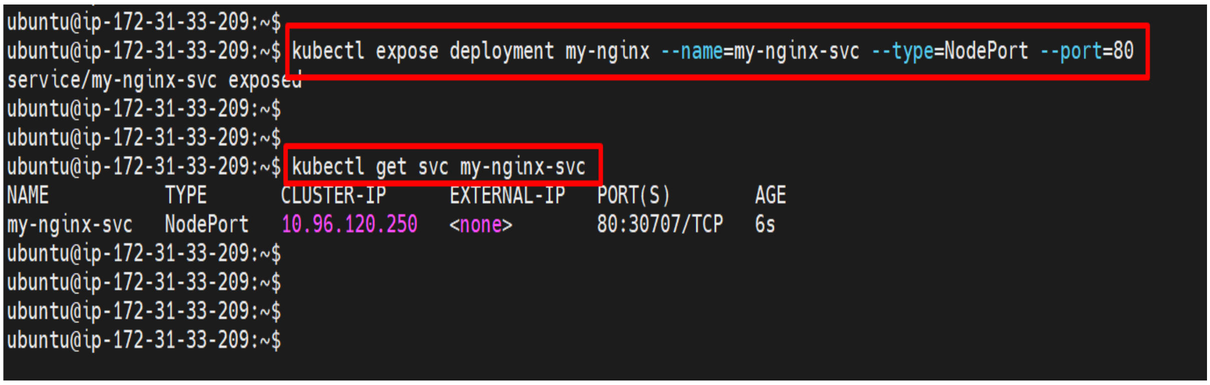


```
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$ kubectl create deployment my-nginx --image=nginx
deployment.apps/my-nginx created
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$ kubectl get deployments.apps my-nginx
NAME       READY   UP-TO-DATE   AVAILABLE   AGE
my-nginx   0/1     1            0           6s
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$ kubectl get pods
NAME                        READY   STATUS              RESTARTS   AGE
my-nginx-f44495498-hhxqd    0/1     ContainerCreating   0          10s
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$
```

# Expose the deployment using following command

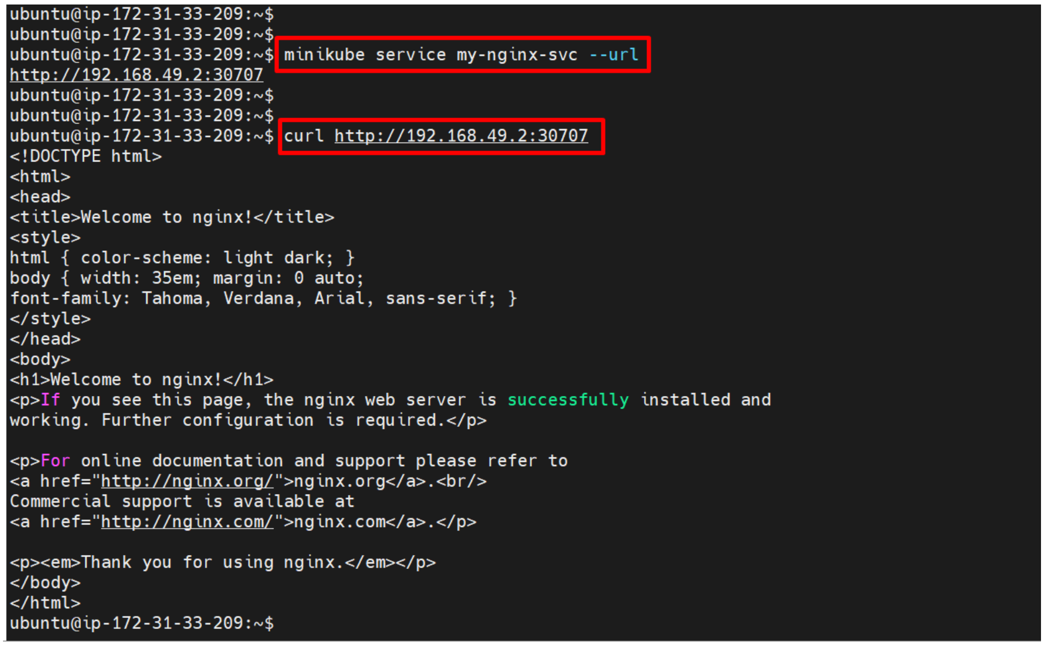
$ kubectl expose deployment my-nginx --name=my-nginx-svc --type=NodePort --port=80

$ kubectl get svc my-nginx-svc



```
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$ kubectl expose deployment my-nginx --name=my-nginx-svc --type=NodePort --port=80
service/my-nginx-svc exposed
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$ kubectl get svc my-nginx-svc
NAME           TYPE       CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
my-nginx-svc   NodePort   10.96.120.250   <none>        80:30707/TCP   6s
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$
```

# Use below command to get your service url, after getting URL we can access that URL using curl command like below



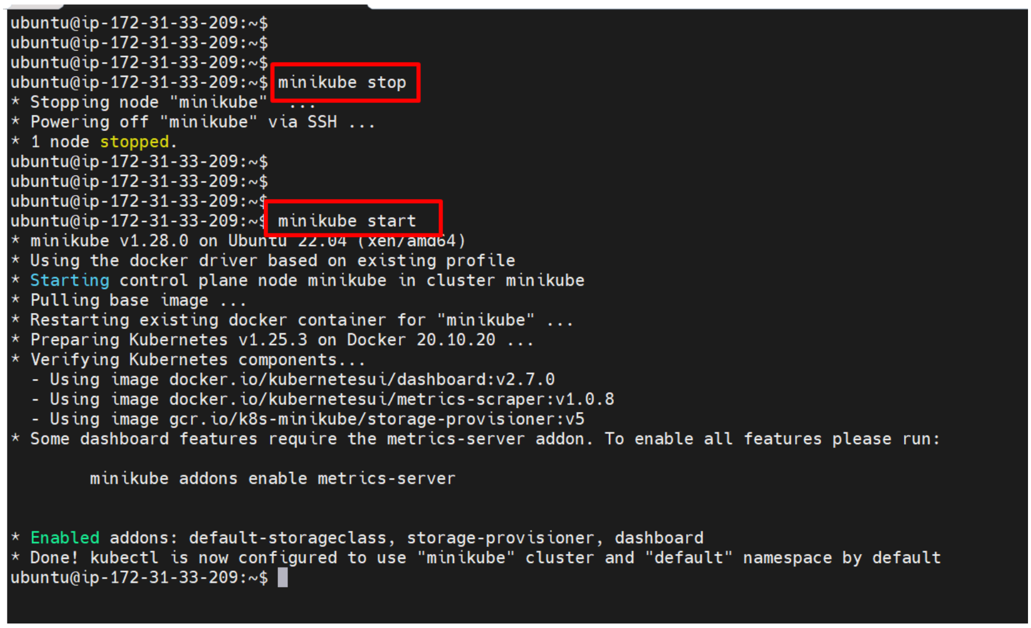
```
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$ minikube service my-nginx-svc --url
http://192.168.49.2:30707
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$ curl http://192.168.49.2:30707
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
ubuntu@ip-172-31-33-209:~$
```

Note: If we are able to get above output, that means Nginx server deployed successfully in Kubernetes.

Step - 10) Managing Minikube Cluster

$ minikube stop

$ minikube start



```
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$ minikube stop
* Stopping node "minikube"  ...
* Powering off "minikube" via SSH ...
* 1 node stopped.
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$
ubuntu@ip-172-31-33-209:~$ minikube start
* minikube v1.28.0 on Ubuntu 22.04 (xen/amd64)
* Using the docker driver based on existing profile
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Restarting existing docker container for "minikube" ...
* Preparing Kubernetes v1.25.3 on Docker 20.10.20 ...
* Verifying Kubernetes components...
  - Using image docker.io/kubernetesui/dashboard:v2.7.0
  - Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Some dashboard features require the metrics-server addon. To enable all features please run:

        minikube addons enable metrics-server


* Enabled addons: default-storageclass, storage-provisioner, dashboard
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
ubuntu@ip-172-31-33-209:~$
```

=== Learn Here.. Lead Anywhere ...!! ===