Experiment:4

Name: Shivpratik Hande **Class:D15c Roll no.14**

Aim:ToinstallKubectlandexecuteKubectlcommandstomanagethe Kubernetes cluster and deploy Your First Kubernetes Application.

What is Kubernetes?

Kubernetes, often referred to as K8s, is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. Originally developed by Google, it has become the industry standard for managing container workloads due to its flexibility and robust features.

- Core Concepts of Kubernetes
  - Containers:Thesearelightweight,portablepackagesthatincludeeverything needed to run an application, ensuring consistency across different environments.
-
  - Pods:ThesmallestdeployableunitsinKubernetes,podscancontain one or more containers that share storage and network resources.
-
  - Nodes:AnodeisaworkermachineintheKubernetesclusterthatrunsat least one pod. Nodes can be either physical or virtual machines.
-
  - Clusters:Aclustercomprisesmultiplenodesthatruncontainerized applications. The control plane manages the cluster's state.
-
  - Services:Servicesprovidestableendpointsforaccessingpodsandfacilitate load balancing and service discovery.
-
  - Deployments:Adeploymentmanagesthelifecycleofpods,allowinguserstospecify the number of replicas and facilitating rolling updates and rollbacks.

Role of Kubernetes

What is Kubectl?

Kubectl is the command-line interface used to interact with the Kubernetes API server. It enables users to manage resources within a Kubernetes cluster effectively.

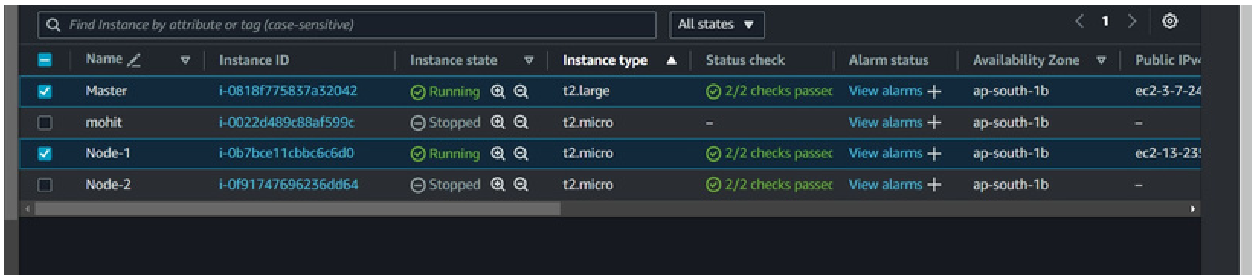
Configuration Files

Configuration files are essential for defining how resources should be created or modified within Kubernetes. Users can employ declarative configurations (using YAML/JSON files) or imperative commands directly in the terminal.

Application Deployment on Kubernetes

- DefineApplicationRequirements:Identifynecessaryresourcessuchas CPU, memory,storage,etc.
- CreateDeploymentConfigurations:Writedeploymentmanifests specifying container images, replicas for scaling, health checks, etc.
- DeployingwithKubectl:Usekubectlcommandslikekubectlapplytodeploy applications based on these configurations.
- MonitoringandScalingApplications:Monitorperformancemetricsand adjust deployments based on traffic demands.
- UpdatingApplications:Modifydeploymentconfigurationsforupdates; Kubernetes supports rolling updates by default.
- Rollback Capabilities: If an update causes issues, kubectl allows easy rollback to previous versions using commands like kubectl rollout undo.

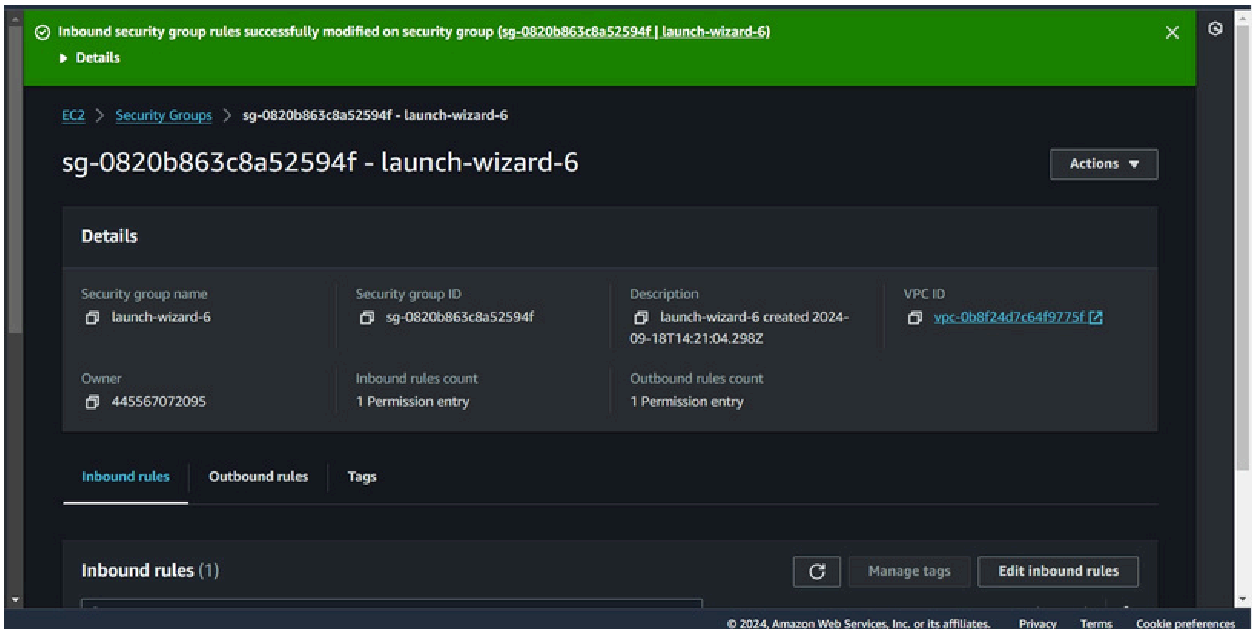
Step 1.Creation of 2 EC2 Ubuntu Instances on AWS.



Find Instance by attribute or tag (case-sensitive) | All states | 1

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv
✓	Master	i-0818f775837a32042	Running	t2.large	2/2 checks passec	View alarms +	ap-south-1b	ec2-3-7-24
	mohit	i-0022d489c88af599c	Stopped	t2.micro	–	View alarms +	ap-south-1b	–
✓	Node-1	i-0b7bce11cbbc6c6d0	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-13-23
	Node-2	i-0f91747696236dd64	Stopped	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	–

Step 2.Edit inbound rules of security group ‘launch-wizard-1’ and set ‘All Traffic’



Inbound security group rules successfully modified on security group (sg-0820b863c8a52594f | launch-wizard-6)

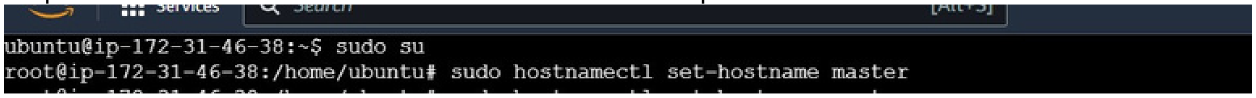
EC2 > Security Groups > sg-0820b863c8a52594f - launch-wizard-6

sg-0820b863c8a52594f - launch-wizard-6

Details

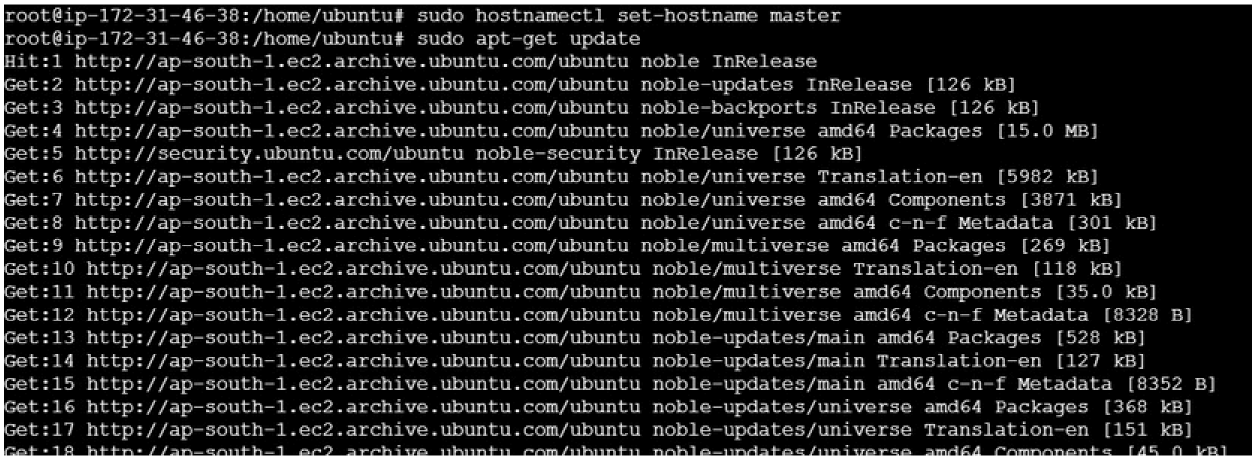
Security group name	Security group ID	Description	VPC ID
launch-wizard-6	sg-0820b863c8a52594f	launch-wizard-6 created 2024-09-18T14:21:04.298Z	vpc-0b8f24d7c64f9775f
Owner	Inbound rules count	Outbound rules count	
445567072095	1 Permission entry	1 Permission entry	

Step 3. Set master and worker as hostname on respective servers

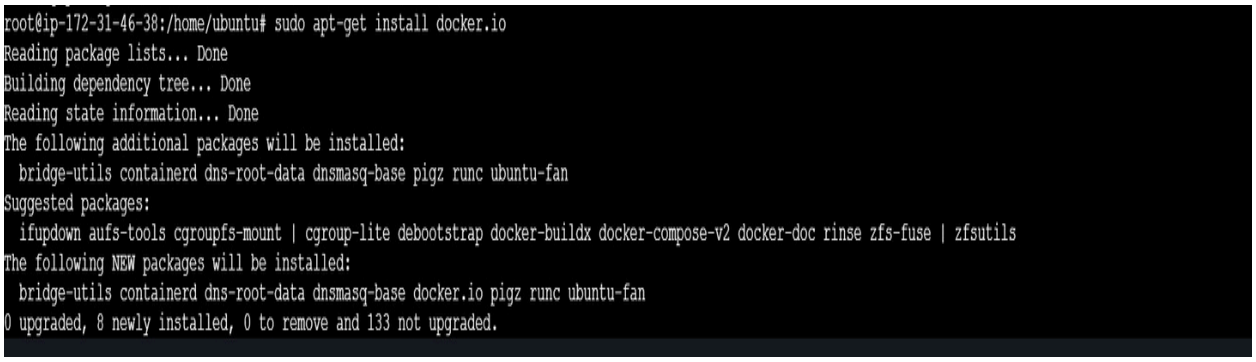


```
ubuntu@ip-172-31-46-38:~$ sudo su
root@ip-172-31-46-38:/home/ubuntu# sudo hostnamectl set-hostname master
```

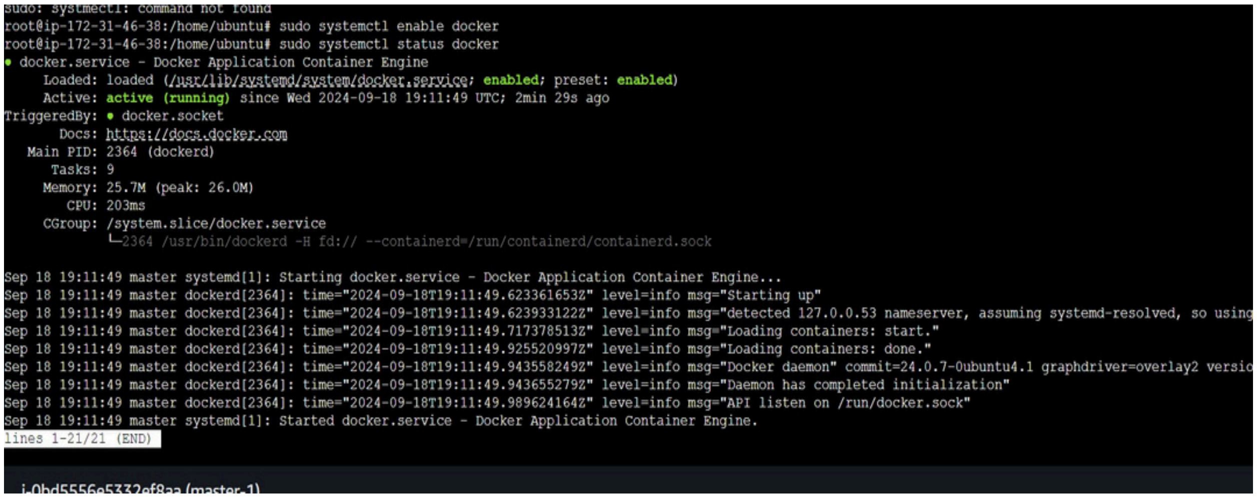
Step 4.Installation of docker



```
root@ip-172-31-46-38:/home/ubuntu# sudo hostnamectl set-hostname master
root@ip-172-31-46-38:/home/ubuntu# sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [528 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [127 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8352 B]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [368 kB]
Get:17 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [151 kB]
Get:18 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
```

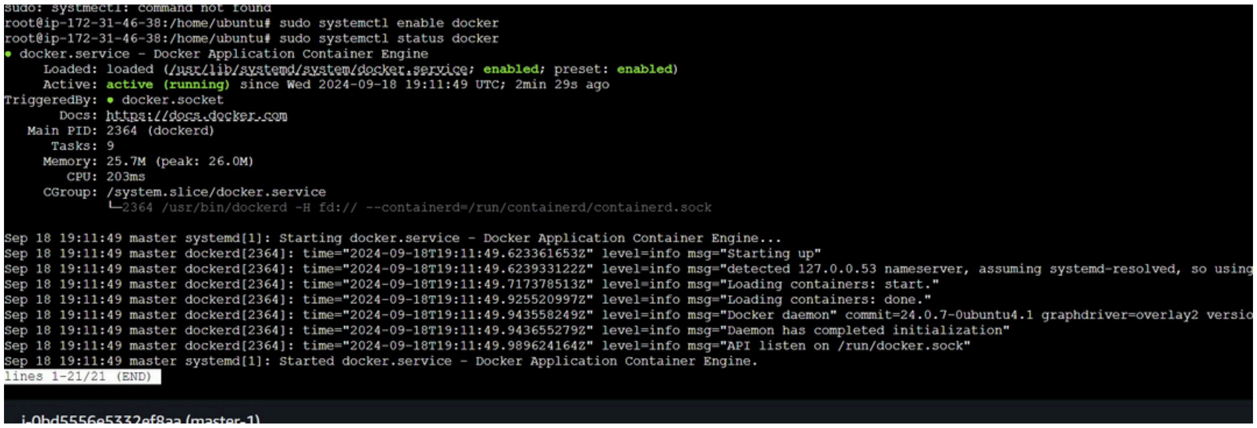


```
root@ip-172-31-46-38:/home/ubuntu# sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 133 not upgraded.
```

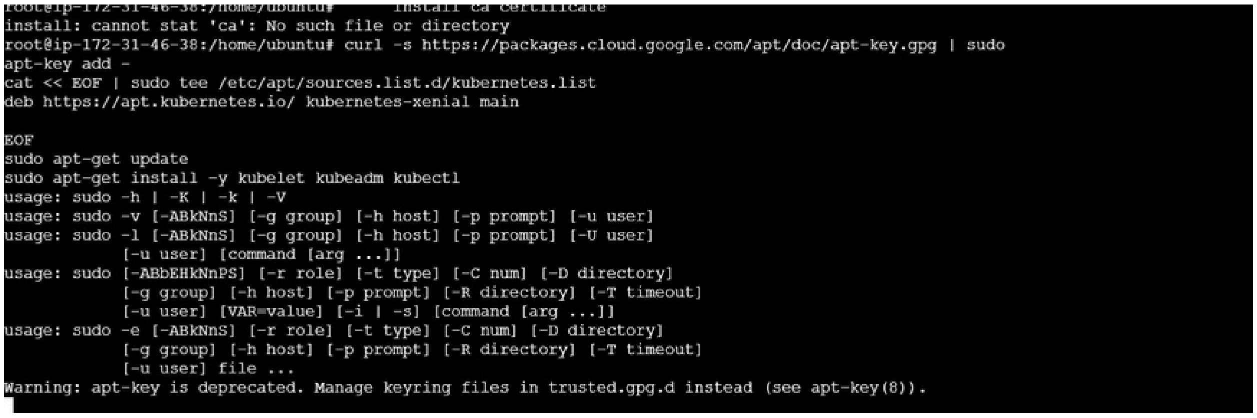


```
sudo: systmectl: command not found
root@ip-172-31-46-38:/home/ubuntu# sudo systemctl enable docker
root@ip-172-31-46-38:/home/ubuntu# sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: active (running) since Wed 2024-09-18 19:11:49 UTC; 2min 29s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 2364 (dockerd)
      Tasks: 9
     Memory: 25.7M (peak: 26.0M)
        CPU: 203ms
     CGroup: /system.slice/docker.service
             └─2364 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 18 19:11:49 master systemd[1]: Starting docker.service - Docker Application Container Engine...
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.623361653Z" level=info msg="Starting up"
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.623933122Z" level=info msg="detected 127.0.0.53 nameserver, assuming systemd-resolved, so using
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.717378513Z" level=info msg="Loading containers: start."
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.925520997Z" level=info msg="Loading containers: done."
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.943558249Z" level=info msg="Docker daemon" commit=24.0.7-0ubuntu4.1 graphdriver=overlay2 versio
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.943655279Z" level=info msg="Daemon has completed initialization"
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.989624164Z" level=info msg="API listen on /run/docker.sock"
Sep 18 19:11:49 master systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-21/21 (END)

i-0bd5556e5332ef8aa (master-1)
```

Step 5.Installation of Kubernetes-

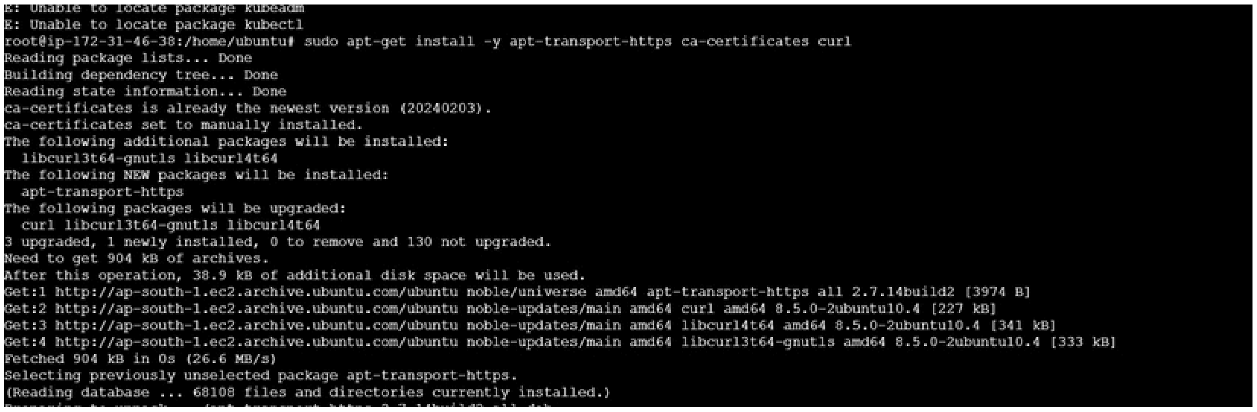


```
sudo: systmectl: command not found
root@ip-172-31-46-38:/home/ubuntu# sudo systemctl enable docker
root@ip-172-31-46-38:/home/ubuntu# sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: active (running) since Wed 2024-09-18 19:11:49 UTC; 2min 29s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 2364 (dockerd)
      Tasks: 9
     Memory: 25.7M (peak: 26.0M)
        CPU: 203ms
     CGroup: /system.slice/docker.service
             └─2364 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 18 19:11:49 master systemd[1]: Starting docker.service - Docker Application Container Engine...
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.623361653Z" level=info msg="Starting up"
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.623933122Z" level=info msg="detected 127.0.0.53 nameserver, assuming systemd-resolved, so using
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.717378513Z" level=info msg="Loading containers: start."
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.925520997Z" level=info msg="Loading containers: done."
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.943558249Z" level=info msg="Docker daemon" commit=24.0.7-0ubuntu4.1 graphdriver=overlay2 versio
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.943655279Z" level=info msg="Daemon has completed initialization"
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.989624164Z" level=info msg="API listen on /run/docker.sock"
Sep 18 19:11:49 master systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-21/21 (END)

i-0bd5556e5332ef8aa (master-1)
```

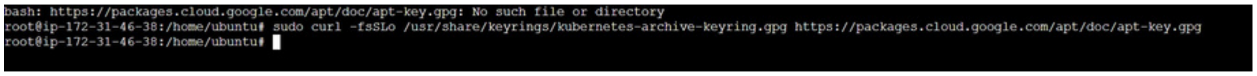


```
root@ip-172-31-46-38:/home/ubuntu#      install ca certificate
install: cannot stat 'ca': No such file or directory
root@ip-172-31-46-38:/home/ubuntu# curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo
apt-key add -
cat << EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main

EOF
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
usage: sudo -h | -K | -k | -V
usage: sudo -v [-ABkNnS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-ABkNnS] [-g group] [-h host] [-p prompt] [-U user]
            [-u user] [command [arg ...]]
usage: sudo [-ABbEHkNnPS] [-r role] [-t type] [-C num] [-D directory]
            [-g group] [-h host] [-p prompt] [-R directory] [-T timeout]
            [-u user] [VAR=value] [-i | -s] [command [arg ...]]
usage: sudo -e [-ABkNnS] [-r role] [-t type] [-C num] [-D directory]
            [-g group] [-h host] [-p prompt] [-R directory] [-T timeout]
            [-u user] file ...
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
```

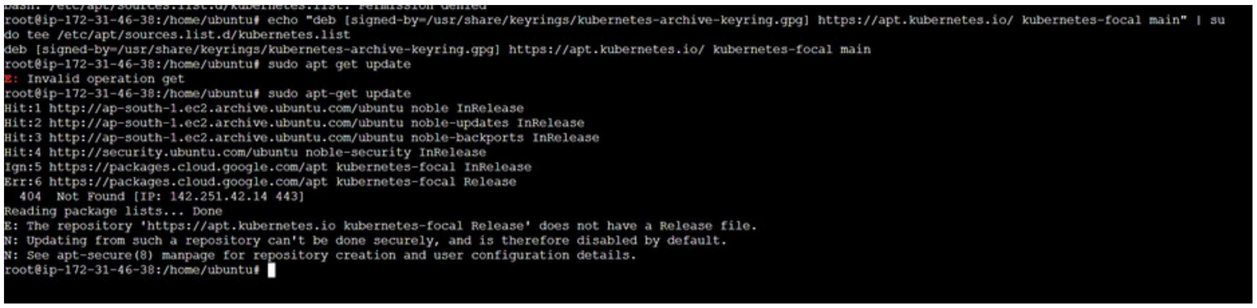
Step.6 Kubernetes Deployment



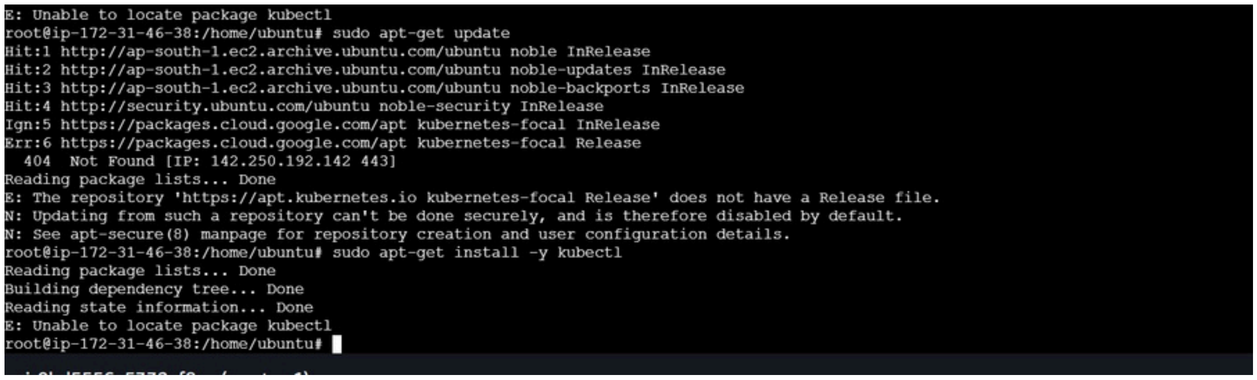
```
E: Unable to locate package kubeadm
E: Unable to locate package kubectl
root@ip-172-31-46-38:/home/ubuntu# sudo apt-get install -y apt-transport-https ca-certificates curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
The following additional packages will be installed:
  libcurl3t64-gnutls libcurl4t64
The following NEW packages will be installed:
  apt-transport-https
The following packages will be upgraded:
  curl libcurl3t64-gnutls libcurl4t64
3 upgraded, 1 newly installed, 0 to remove and 130 not upgraded.
Need to get 904 kB of archives.
After this operation, 38.9 kB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 apt-transport-https all 2.7.14build2 [3974 B]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 curl amd64 8.5.0-2ubuntu10.4 [227 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libcurl4t64 amd64 8.5.0-2ubuntu10.4 [341 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libcurl3t64-gnutls amd64 8.5.0-2ubuntu10.4 [333 kB]
Fetched 904 kB in 0s (26.6 MB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 68108 files and directories currently installed.)
```



```
bash: https://packages.cloud.google.com/apt/doc/apt-key.gpg: No such file or directory
root@ip-172-31-46-38:/home/ubuntu# sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg
root@ip-172-31-46-38:/home/ubuntu#
```

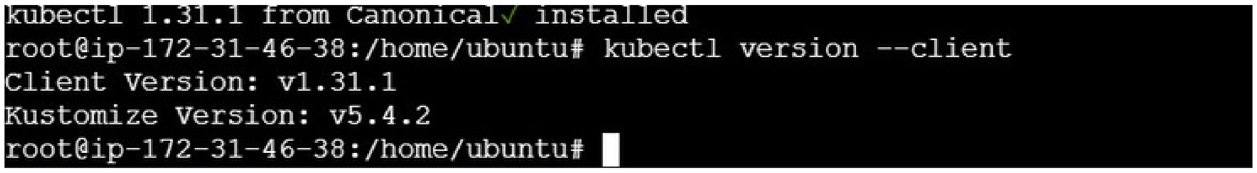


```
bash: /etc/apt/sources.list.d/kubernetes.list: Permission denied
root@ip-172-31-46-38:/home/ubuntu# echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-focal main" | su
do tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-focal main
root@ip-172-31-46-38:/home/ubuntu# sudo apt get update
E: Invalid operation get
root@ip-172-31-46-38:/home/ubuntu# sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:5 https://packages.cloud.google.com/apt kubernetes-focal InRelease
Err:6 https://packages.cloud.google.com/apt kubernetes-focal Release
  404  Not Found [IP: 142.251.42.14 443]
Reading package lists... Done
E: The repository 'https://apt.kubernetes.io kubernetes-focal Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
root@ip-172-31-46-38:/home/ubuntu#
```

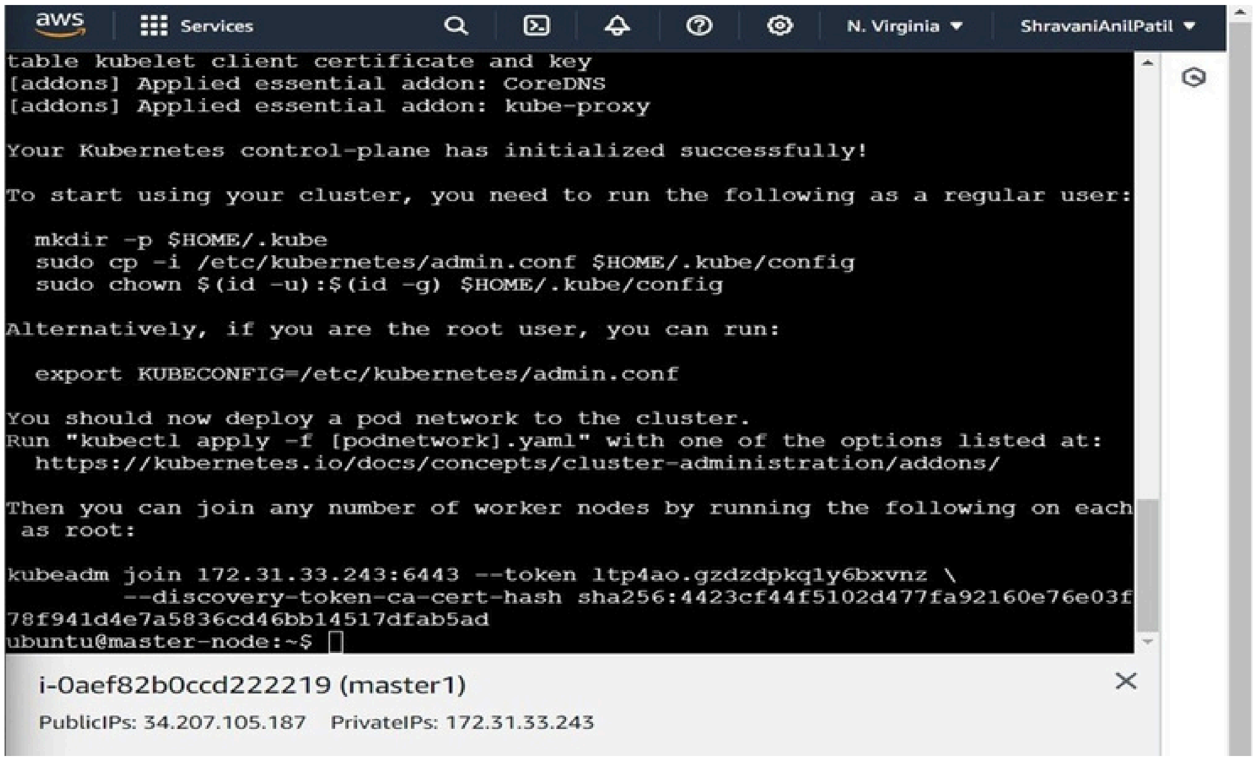


```
E: Unable to locate package kubectl
root@ip-172-31-46-38:/home/ubuntu# sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:5 https://packages.cloud.google.com/apt kubernetes-focal InRelease
Err:6 https://packages.cloud.google.com/apt kubernetes-focal Release
  404  Not Found [IP: 142.250.192.142 443]
Reading package lists... Done
E: The repository 'https://apt.kubernetes.io kubernetes-focal Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
root@ip-172-31-46-38:/home/ubuntu# sudo apt-get install -y kubectl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package kubectl
root@ip-172-31-46-38:/home/ubuntu#
```

Extra:



```
kubectl 1.31.1 from Canonical✓ installed
root@ip-172-31-46-38:/home/ubuntu# kubectl version --client
Client Version: v1.31.1
Kustomize Version: v5.4.2
root@ip-172-31-46-38:/home/ubuntu#
```

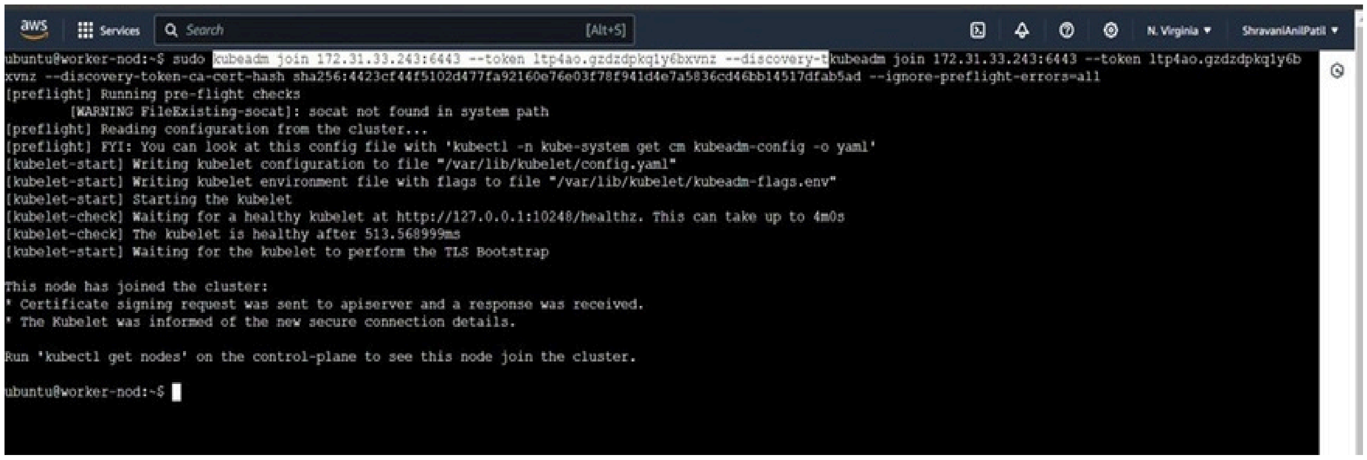


```
table kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each
 as root:

kubeadm join 172.31.33.243:6443 --token ltp4ao.gzdzdpkq1y6bxvnz \
        --discovery-token-ca-cert-hash sha256:4423cf44f5102d477fa92160e76e03f
78f941d4e7a5836cd46bb14517dfab5ad
ubuntu@master-node:~$
```

i-0aef82b0ccd222219 (master1)

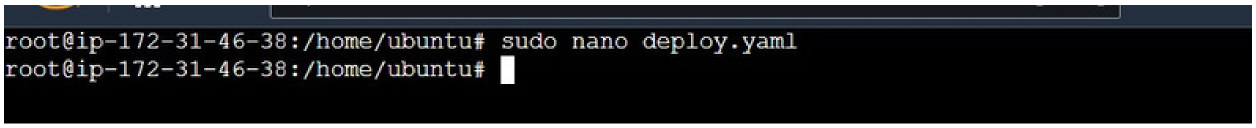
PublicIPs: 34.207.105.187   PrivateIPs: 172.31.33.243

Step 7. Deploy Pod Network to Cluster and Join Worker Node to Cluster

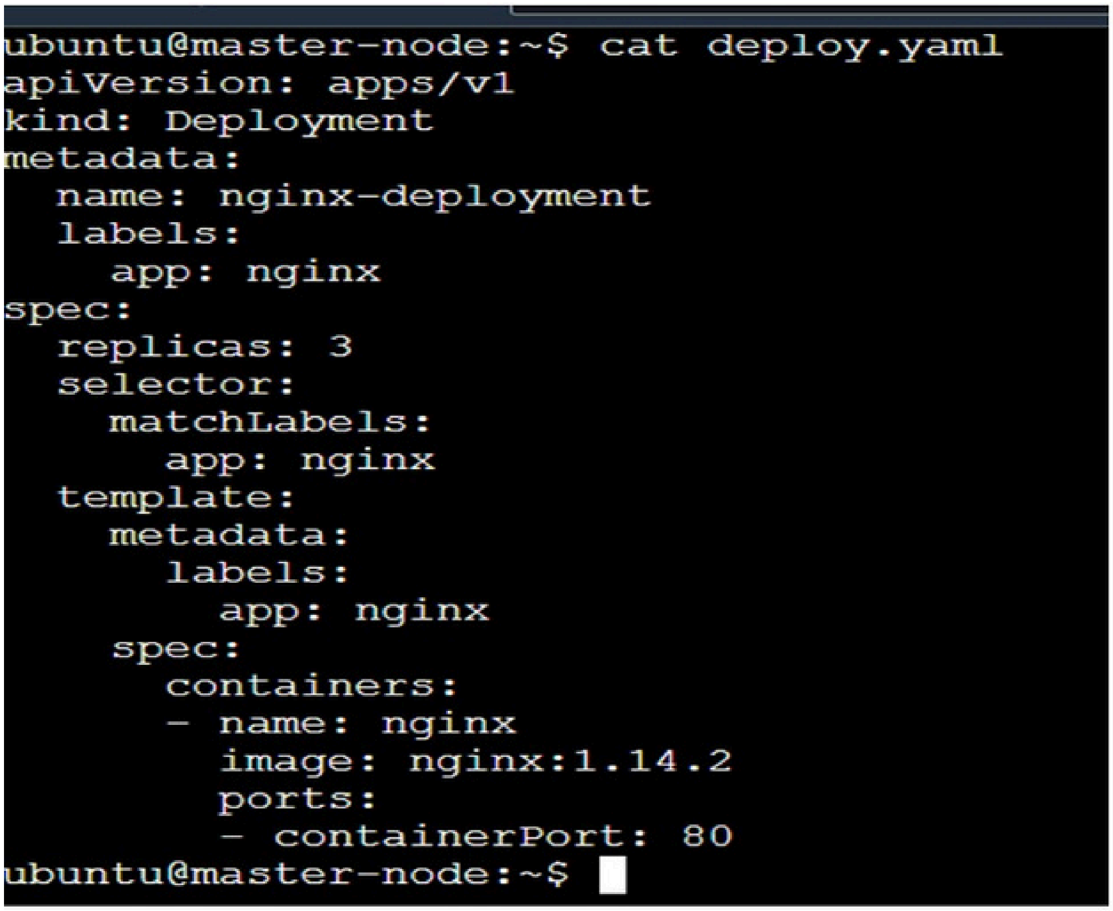


```
ubuntu@worker-nod:~$ sudo kubeadm join 172.31.33.243:6443 --token ltp4ao.gzdzdpkqly6bxvnz --discovery-tkubeadm join 172.31.33.243:6443 --token ltp4ao.gzdzdpkqly6b
xvnz --discovery-token-ca-cert-hash sha256:4423cf44f5102d477fa92160e76e03f78f941d4e7a5836cd46bb14517dfab5ad --ignore-preflight-errors=all
[preflight] Running pre-flight checks
        [WARNING FileExisting-socat]: socat not found in system path
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 513.568999ms
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

ubuntu@worker-nod:~$
```

Step 8. Create one file deploy.yaml

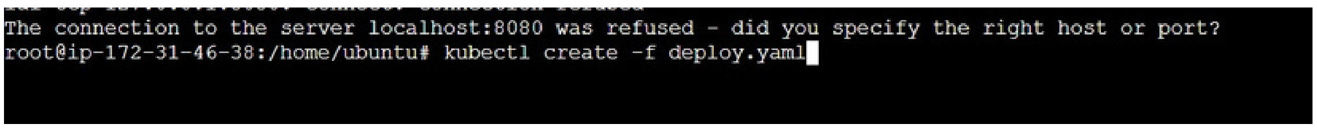


```
root@ip-172-31-46-38:/home/ubuntu# sudo nano deploy.yaml
root@ip-172-31-46-38:/home/ubuntu#
```

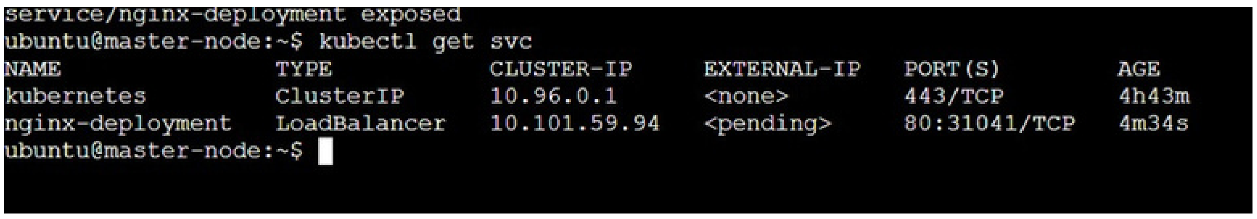


```
ubuntu@master-node:~$ cat deploy.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
ubuntu@master-node:~$
```

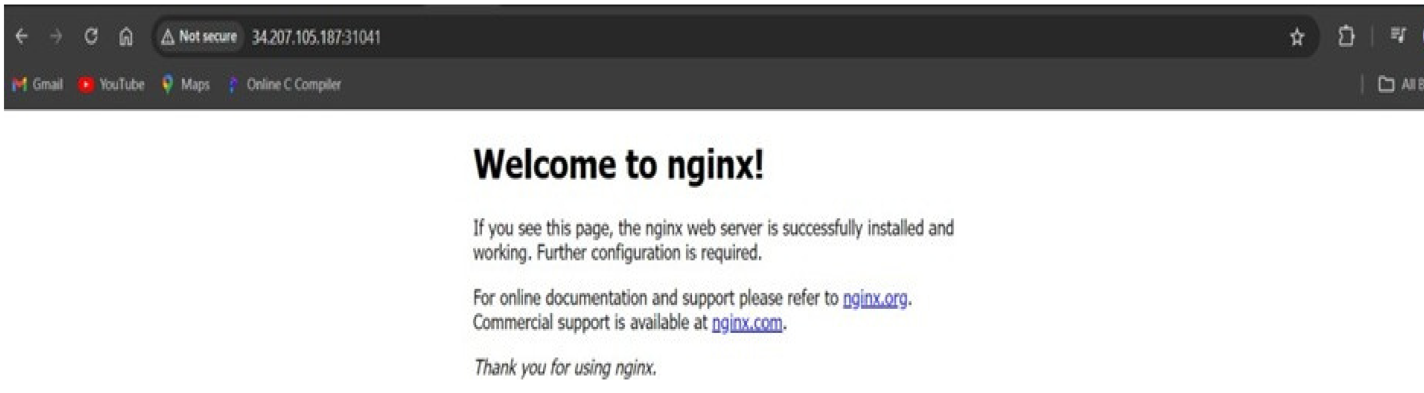
Step 9 : Create Deployment



```
The connection to the server localhost:8080 was refused - did you specify the right host or port?
root@ip-172-31-46-38:/home/ubuntu# kubectl create -f deploy.yaml
```



```
service/nginx-deployment exposed
ubuntu@master-node:~$ kubectl get svc
NAME               TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubernetes         ClusterIP      10.96.0.1      <none>        443/TCP        4h43m
nginx-deployment   LoadBalancer   10.101.59.94   <pending>     80:31041/TCP   4m34s
ubuntu@master-node:~$
```


Not secure 34.207.105.187:31041

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Conclusion:
That sounds like a great project! Setting up Kubernetes and Docker on an AWS EC2 instance is a fantastic way to gain hands-on experience with container orchestration.Using Flannel for networking ensures that your pods can communicate effectively,
which is crucial for a well-functioning cluster.
Deploying Nginx via a Kubernetes Deployment is a solid choice, as it showcases key Kubernetes features like scaling and rolling updates. The fact that you successfully accessed the Nginx server using port forwarding and received a 200 OK response confirms that your deployment was set up correctly.
This project not only demonstrates your ability to configure and manage Kubernetes but also highlights its effectiveness in orchestrating containerized applications. If you have any specific questions or areas you'd like to explore further, feel free to ask!