Advance DevOps Lab Experiment 03

Aim: To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud Platforms.

Roll No.	**22**
Name	**Sarthak Harade**
Class	**D15B**
Subject	**Advance DevOps Lab**
LO Mapped	**LO1: To understand the fundamentals of Cloud Computing and be fully proficient with Cloud based DevOps solution deployment options to meet your business requirements.** **LO2: To deploy single and multiple container applications and manage application deployments with rollouts in Kubernetes**
Grade:	

Aim: To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud Platforms.

Theory:

Container-based microservices architectures have profoundly changed the way development and operations teams test and deploy modern software. Containers help companies modernize by making it easier to scale and deploy applications, but containers have also introduced new challenges and more complexity by creating an entirely new infrastructure ecosystem.

Large and small software companies alike are now deploying thousands of container instances daily, and that's a complexity of scale they have to manage. So how do they do it?

Enter the age of Kubernetes.

Originally developed by Google, Kubernetes is an open-source container orchestration platform designed to automate the deployment, scaling, and management of containerized applications. In fact, Kubernetes has established itself as the defacto standard for container orchestration and is the flagship project of the Cloud Native Computing Foundation (CNCF), backed by key players like Google, AWS, Microsoft, IBM, Intel, Cisco, and Red Hat.

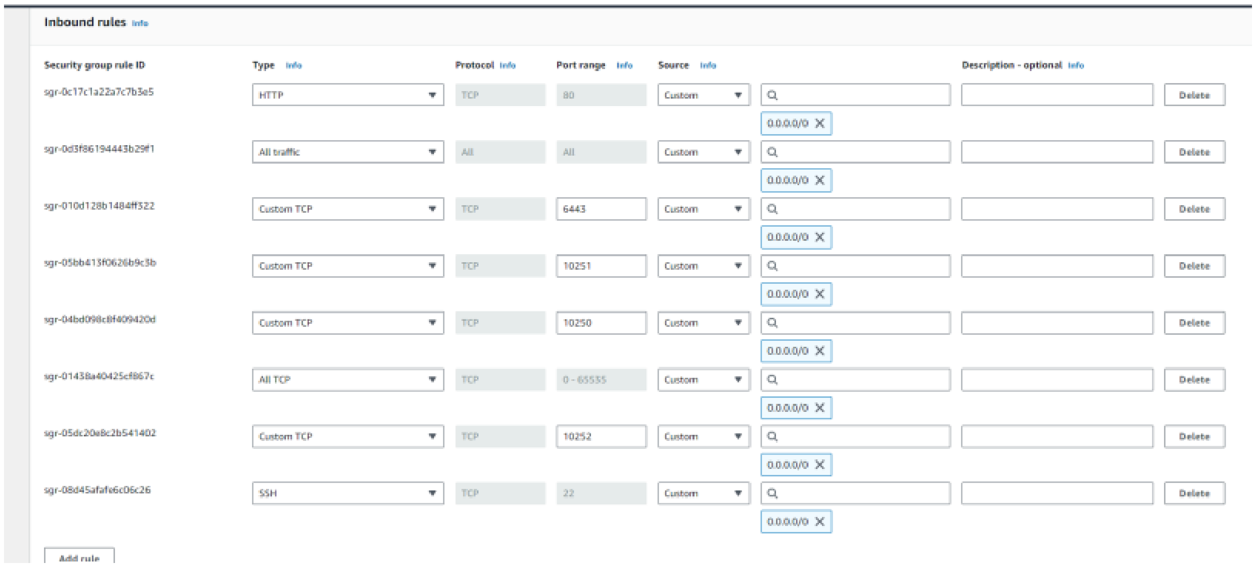
Kubernetes makes it easy to deploy and operate applications in a microservice architecture. It does so by creating an abstraction layer on top of a group of hosts so that development teams can deploy their applications and let Kubernetes manage the following activities:

- **Controlling resource consumption by application or team**
- **Evenly spreading application load across a hosting infrastructure**
- **Automatically load balancing requests across the different instances of an application**
- **Monitoring resource consumption and resource limits to automatically stop applications from consuming too many resources and restarting the applications again**
- **Moving an application instance from one host to another if there is a shortage of resources in a host, or if the host dies**
- **Automatically leveraging additional resources made available when a new host is added to the cluster**
- **Easily performing canary deployments and rollbacks**

Steps:-

Security Groups:-

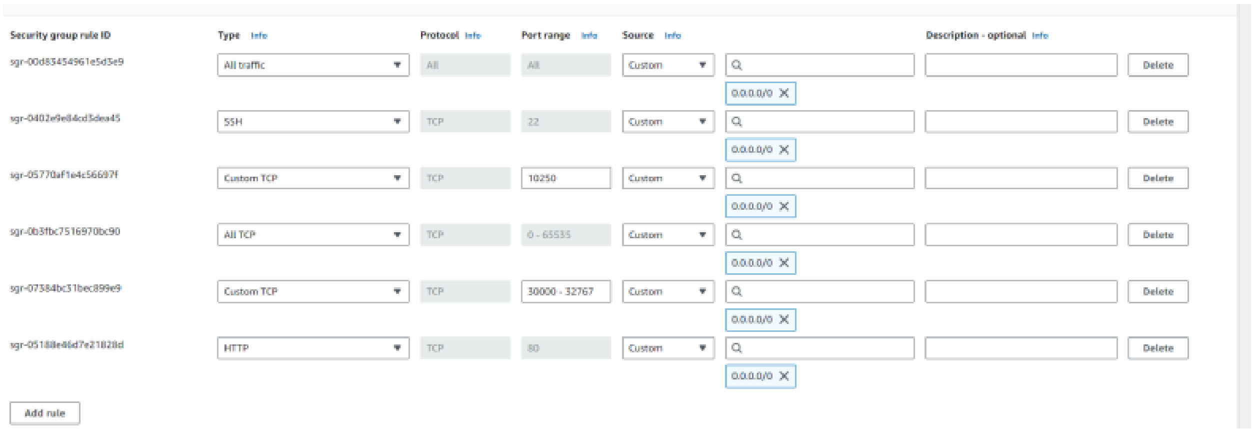
Master:-



Inbound rules

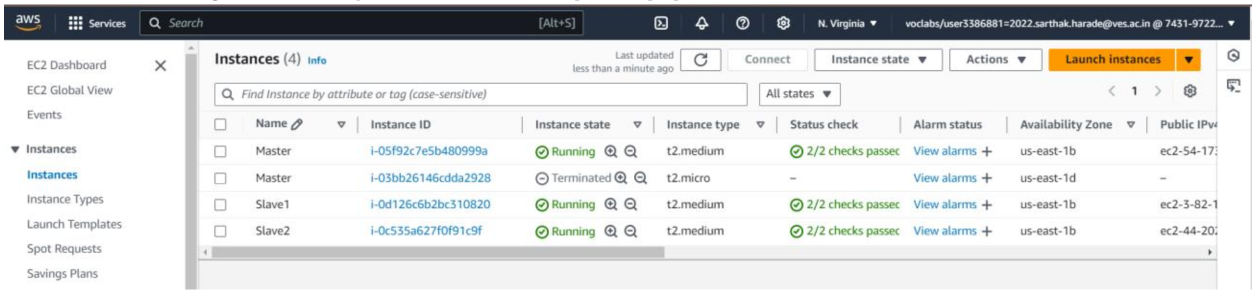
Security group rule ID	Type	Protocol	Port range	Source		Description - optional
sgr-0c17c1a22a7c7b3e5	HTTP	TCP	80	Custom	0.0.0.0/0	
sgr-0d3f86194443b29f1	All traffic	All	All	Custom	0.0.0.0/0	
sgr-010d128b1484ff322	Custom TCP	TCP	6443	Custom	0.0.0.0/0	
sgr-05bb413f0626b9c3b	Custom TCP	TCP	10251	Custom	0.0.0.0/0	
sgr-04bd098c8f409420d	Custom TCP	TCP	10250	Custom	0.0.0.0/0	
sgr-01438a40425cf867c	All TCP	TCP	0 - 65535	Custom	0.0.0.0/0	
sgr-05dc20e8c2b541402	Custom TCP	TCP	10252	Custom	0.0.0.0/0	
sgr-08d45afafe6c06c26	SSH	TCP	22	Custom	0.0.0.0/0	

Slave:-



Security group rule ID	Type	Protocol	Port range	Source		Description - optional
sgr-00d83454961e5d3e9	All traffic	All	All	Custom	0.0.0.0/0	
sgr-0402e9e84cd3dea45	SSH	TCP	22	Custom	0.0.0.0/0	
sgr-05770af1e4c56697f	Custom TCP	TCP	10250	Custom	0.0.0.0/0	
sgr-0b3fbc7516970bc90	All TCP	TCP	0 - 65535	Custom	0.0.0.0/0	
sgr-07384bc31bec899e9	Custom TCP	TCP	30000 - 32767	Custom	0.0.0.0/0	
sgr-05188e46d7e21828d	HTTP	TCP	80	Custom	0.0.0.0/0	

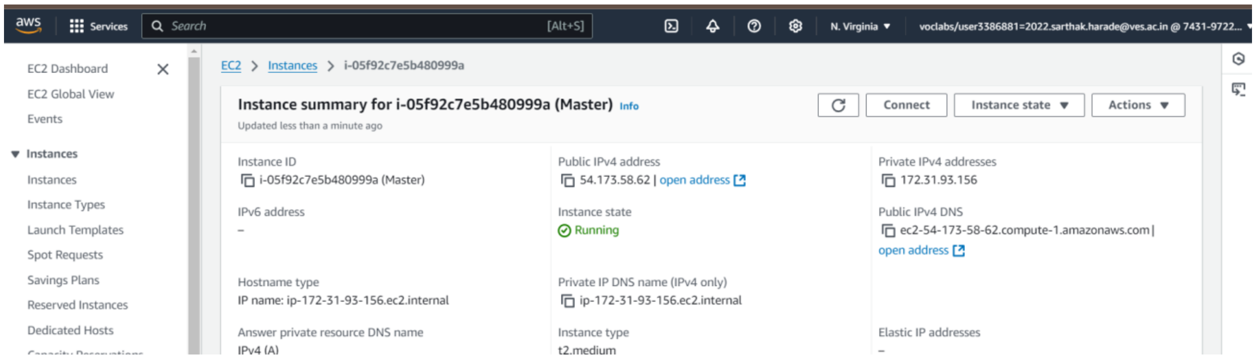
Creating 3 instances(1 Master, 2 Worker Nodes)(Use the same Key for connecting each instance through SSH to your command prompt)



voclabs/user3386881=2022.sarthak.harade@ves.ac.in @ 7431-9722...

Instances (4)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv
Master	i-05f92c7e5b480999a	Running	t2.medium	2/2 checks passec	View alarms	us-east-1b	ec2-54-17
Master	i-03bb26146cdda2928	Terminated	t2.micro	–	View alarms	us-east-1d	–
Slave1	i-0d126c6b2bc310820	Running	t2.medium	2/2 checks passec	View alarms	us-east-1b	ec2-3-82-1
Slave2	i-0c535a627f0f91c9f	Running	t2.medium	2/2 checks passec	View alarms	us-east-1b	ec2-44-20

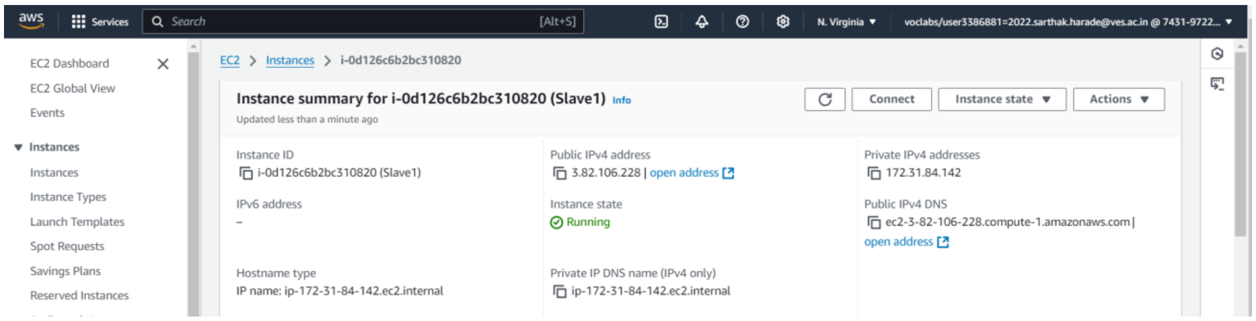


EC2 > Instances > i-05f92c7e5b480999a

Instance summary for i-05f92c7e5b480999a (Master) Info

Updated less than a minute ago

Instance ID i-05f92c7e5b480999a (Master)	Public IPv4 address 54.173.58.62 | open address	Private IPv4 addresses 172.31.93.156
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-54-173-58-62.compute-1.amazonaws.com | open address
Hostname type IP name: ip-172-31-93-156.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-93-156.ec2.internal	
Answer private resource DNS name IPv4 (A)	Instance type t2.medium	Elastic IP addresses –

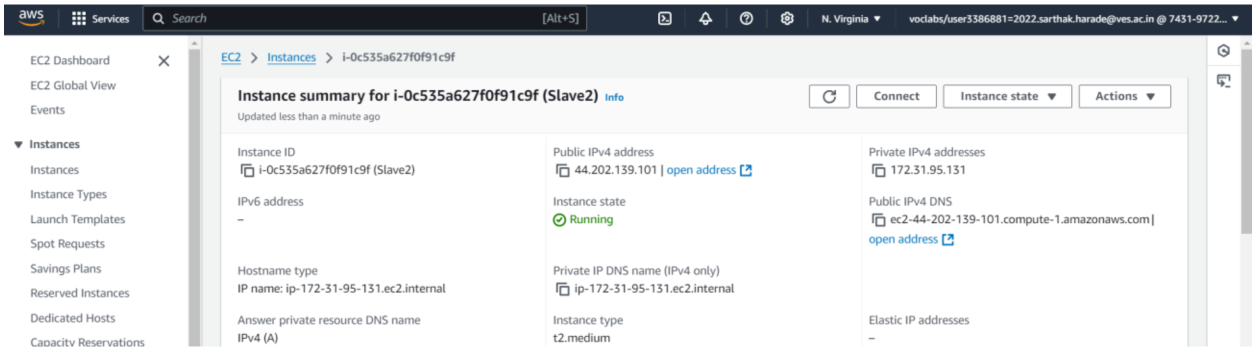


EC2 > Instances > i-0d126c6b2bc310820

Instance summary for i-0d126c6b2bc310820 (Slave1) Info

Updated less than a minute ago

Instance ID i-0d126c6b2bc310820 (Slave1)	Public IPv4 address 3.82.106.228 | open address	Private IPv4 addresses 172.31.84.142
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-3-82-106-228.compute-1.amazonaws.com | open address
Hostname type IP name: ip-172-31-84-142.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-84-142.ec2.internal	



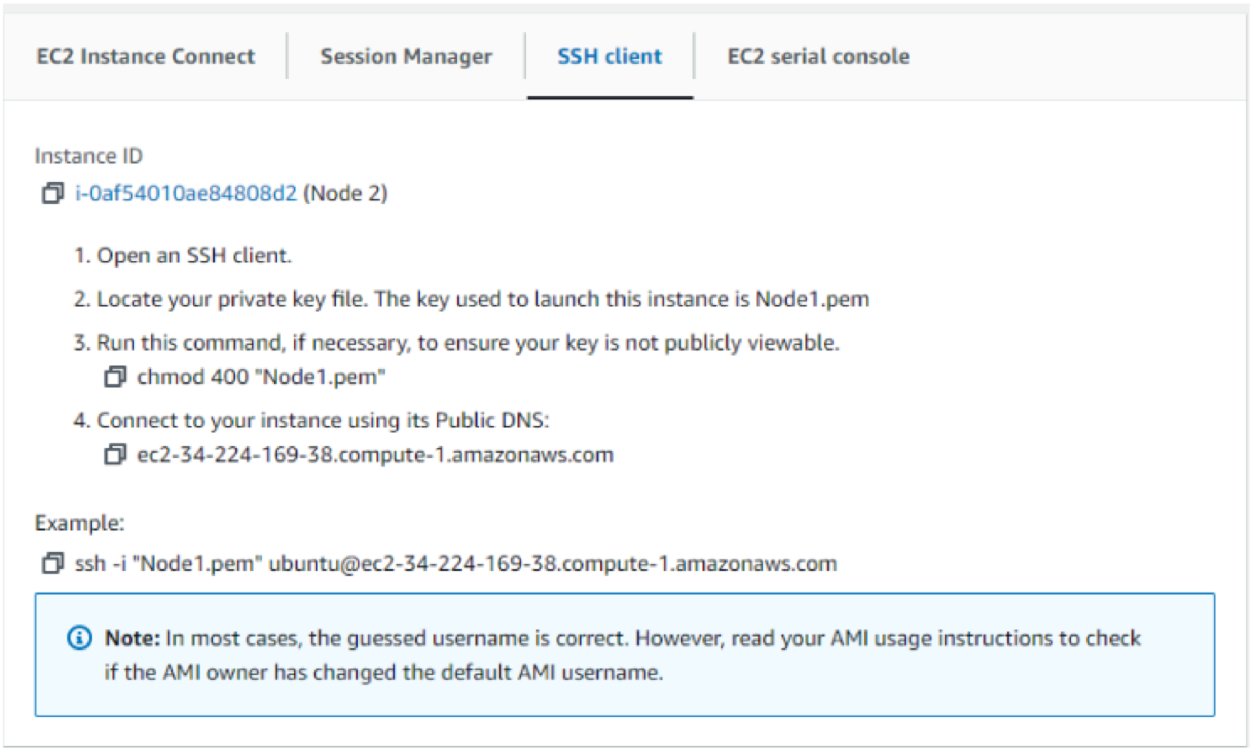
EC2 > Instances > i-0c535a627f0f91c9f

Instance summary for i-0c535a627f0f91c9f (Slave2) Info

Updated less than a minute ago

Instance ID i-0c535a627f0f91c9f (Slave2)	Public IPv4 address 44.202.139.101 | open address	Private IPv4 addresses 172.31.95.131
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-44-202-139-101.compute-1.amazonaws.com | open address
Hostname type IP name: ip-172-31-95-131.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-95-131.ec2.internal	
Answer private resource DNS name IPv4 (A)	Instance type t2.medium	Elastic IP addresses –

Connect to your EC2 instance through SSH by running the command having such format:-
ssh -i “<Your_saved_key>.pem” ubuntu@<your-instance-public-ip>.<the region of created instance>.compute.amazonaws.com
Showed when you go to the SSH Client section when you select your instance and press Connect like here:-



Instance ID

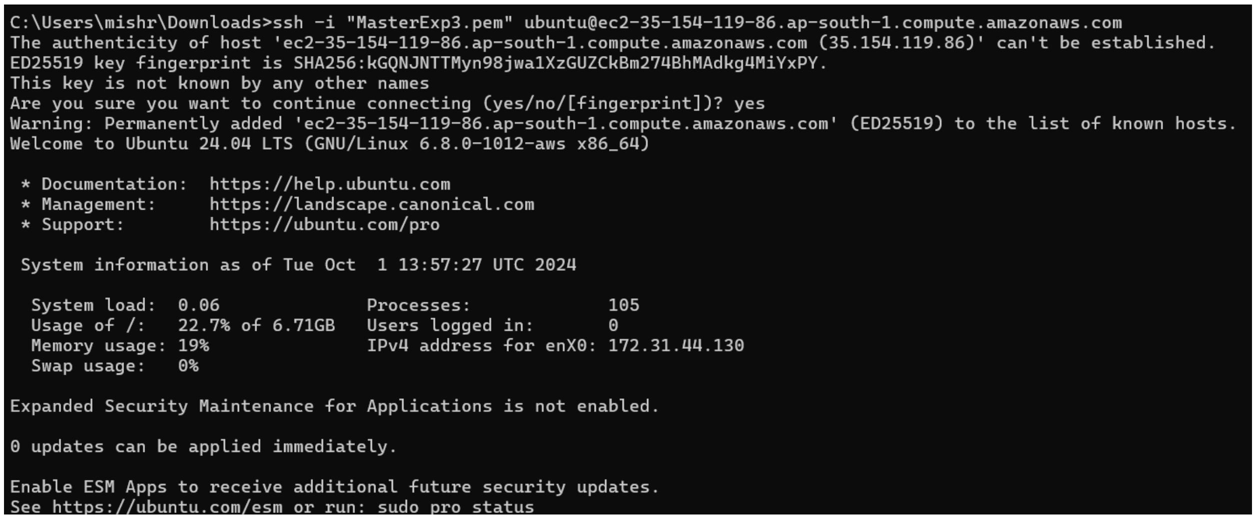
i-0af54010ae84808d2 (Node 2)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is Node1.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 "Node1.pem"
4. Connect to your instance using its Public DNS:
   ec2-34-224-169-38.compute-1.amazonaws.com

Example:

ssh -i "Node1.pem" ubuntu@ec2-34-224-169-38.compute-1.amazonaws.com

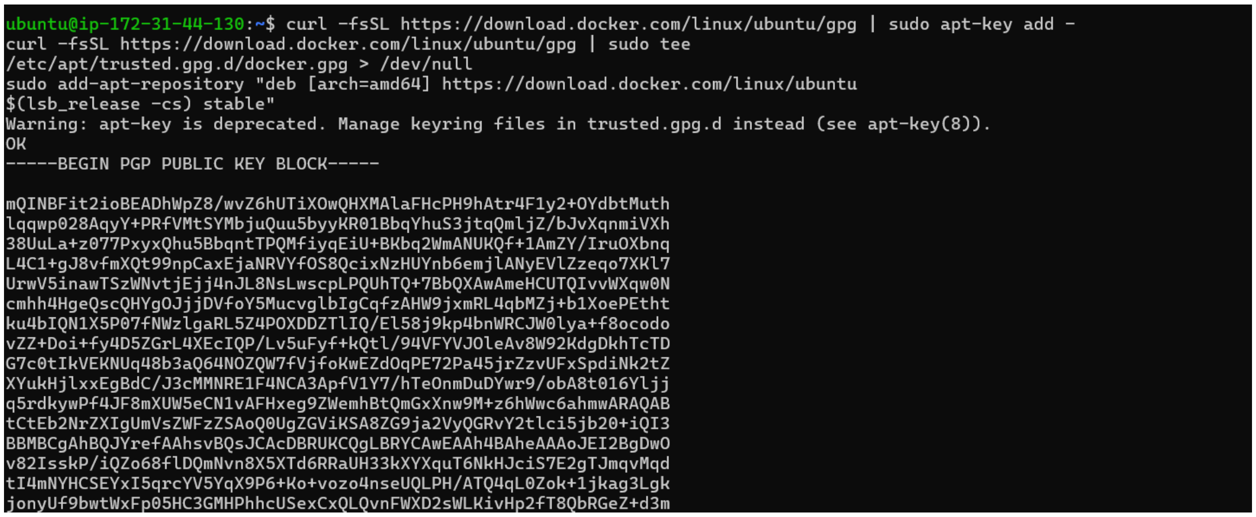
Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.



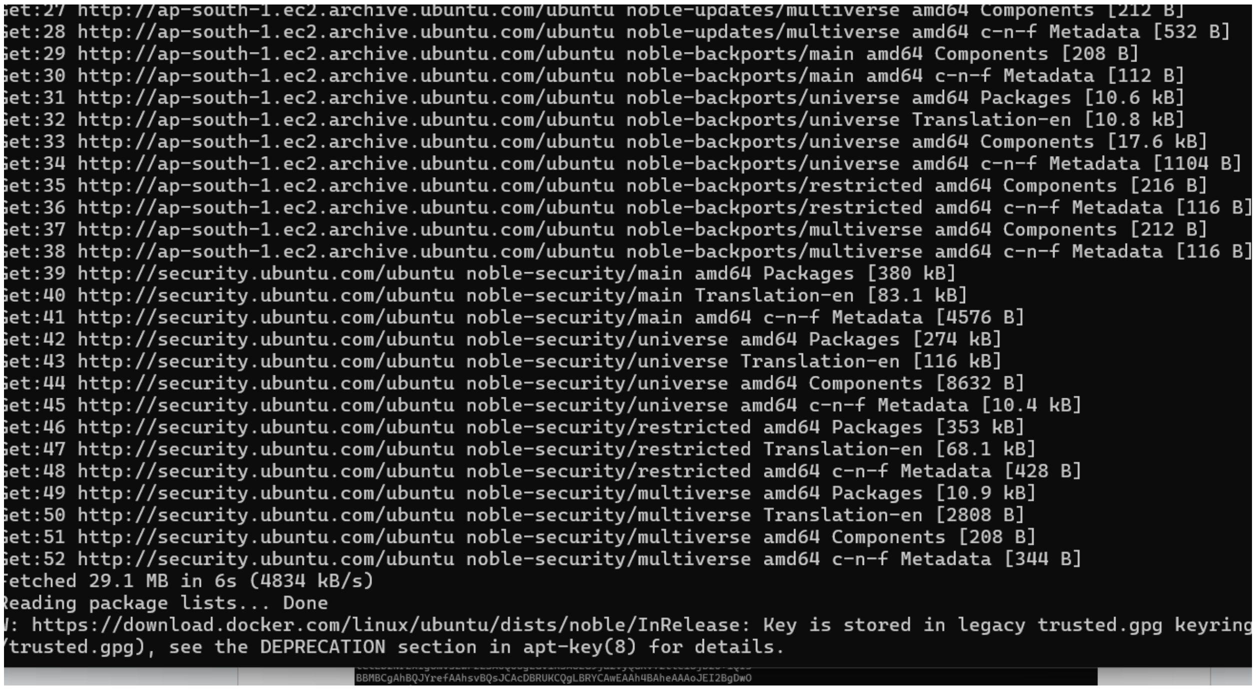
```
C:\Users\mishr\Downloads>ssh -i "MasterExp3.pem" ubuntu@ec2-35-154-119-86.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-35-154-119-86.ap-south-1.compute.amazonaws.com (35.154.119.86)' can't be established.
ED25519 key fingerprint is SHA256:kGQNJNTTMyn98jwa1XzGUZCkBm274BhMAdkg4MiYxPY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-35-154-119-86.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

 System information as of Tue Oct  1 13:57:27 UTC 2024

  System load:  0.06              Processes:             105
  Usage of /:   22.7% of 6.71GB   Users logged in:       0
  Memory usage: 19%               IPv4 address for enX0: 172.31.44.130
  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
```

Step 4: Run on Master,Node 1,and Node 2 the below commands to install and setup Docker in Master, Node1, and Node2.
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/trusted.gpg.d/docker.gpg > /dev/null

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable

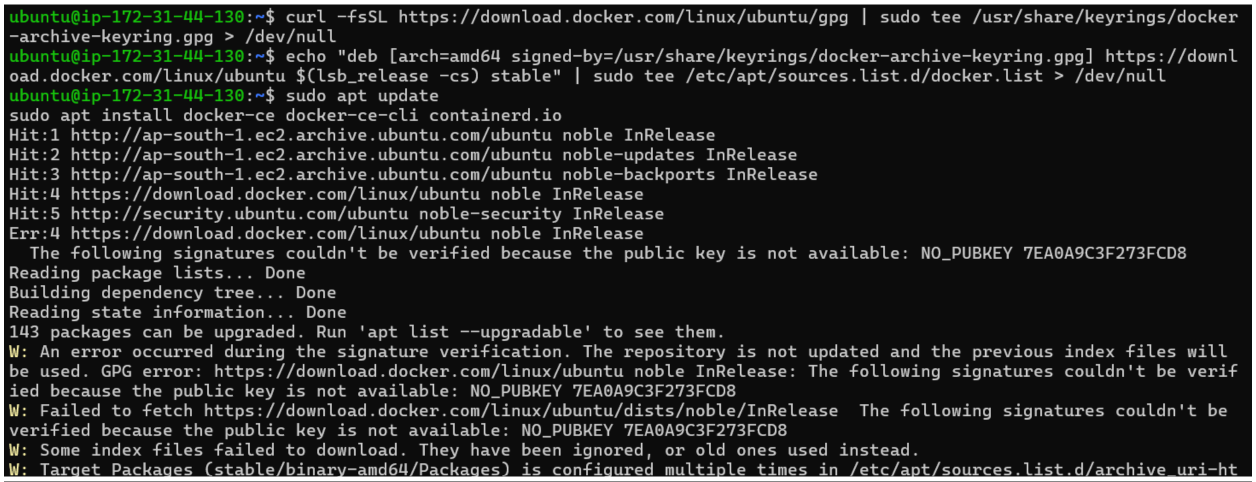


```
ubuntu@ip-172-31-44-130:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee
/etc/apt/trusted.gpg.d/docker.gpg > /dev/null
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBFit2ioBEADhWpZ8/wvZ6hUTiXOwQHXMAlaFHcPH9hAtr4F1y2+OYdbtMuth
lqqwp028AqyY+PRfVMtSYMbjuQuu5byyKR01BbqYhuS3jtqQmljZ/bJvXqnmiVXh
38UuLa+z077PxyxQhu5BbqntTPQMfiyqEiU+BKbq2WmANUKQf+1AmZY/IruOXbnq
L4C1+gJ8vfmXQt99npCaxEjaNRVYfOS8QcixNzHUYnb6emjlANyEVlZzeqo7XKl7
UrwV5inawTSzWNvtjEjj4nJL8NsLwscpLPQUhTQ+7BbQXAwAmeHCUTQIvvWXqw0N
cmhh4HgeQscQHYgOJjjDVfoY5MucvglbIgCqfzAHW9jxmRL4qbMZj+b1XoePEtht
ku4bIQN1X5P07fNWzlgaRL5Z4POXDDZTlIQ/El58j9kp4bnWRCJW0lya+f8ocodo
vZZ+Doi+fy4D5ZGrL4XEcIQP/Lv5uFyf+kQtl/94VFYVJOleAv8W92KdgDkhTcTD
G7c0tIkVEKNUq48b3aQ64NOZQW7fVjfoKwEZdOqPE72Pa45jrZzvUFxSpdiNk2tZ
XYukHjlxxEgBdC/J3cMMNRE1F4NCA3ApfV1Y7/hTeOnmDuDYwr9/obA8t016Yljj
q5rdkywPf4JF8mXUW5eCN1vAFHxeg9ZWemhBtQmGxXnw9M+z6hWwc6ahmwARAQAB
tCtEb2NrZXIgUmVsZWFzZSAoQ0UgZGViKSA8ZG9ja2VyQGRvY2tlci5jb20+iQI3
BBMBCgAhBQJYrefAAhsvBQsJCAcDBRUKCQgLBRYCAwEAAh4BAheAAAoJEI2BgDwO
v82IsskP/iQZo68flDQmNvn8X5XTd6RRaUH33kXYXquT6NkHJciS7E2gTJmqvMqd
tI4mNYHCSEYxI5qrcYV5YqX9P6+Ko+vozo4nseUQLPH/ATQ4qL0Zok+1jkag3Lgk
jonyUf9bwtWxFp05HC3GMHPhhcUSexCxQLQvnFWXD2sWLKivHp2fT8QbRGeZ+d3m
```

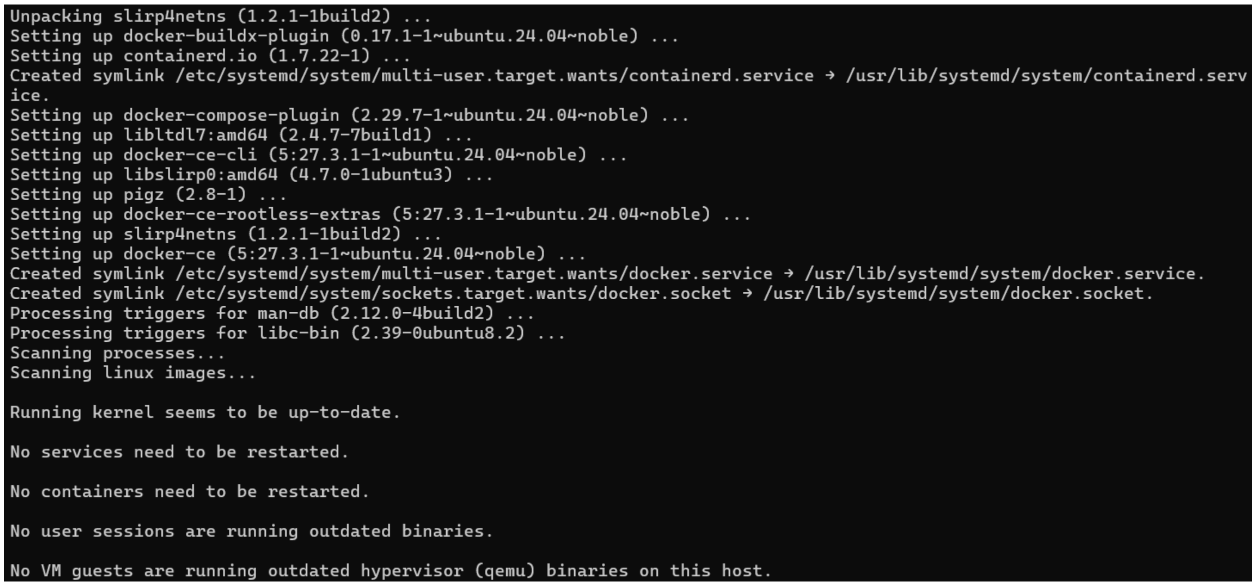


```
et:27 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B]
et:28 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
et:29 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
et:30 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
et:31 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.6 kB]
et:32 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.8 kB]
et:33 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
et:34 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
et:35 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
et:36 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
et:37 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
et:38 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
et:39 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [380 kB]
et:40 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [83.1 kB]
et:41 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4576 B]
et:42 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [274 kB]
et:43 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [116 kB]
et:44 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
et:45 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.4 kB]
et:46 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [353 kB]
et:47 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [68.1 kB]
et:48 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [428 B]
et:49 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
et:50 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
et:51 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
et:52 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
etched 29.1 MB in 6s (4834 kB/s)
eading package lists... Done
: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring
trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
BBMBCgAhBQJYrefAAhsvBQsJCAcDBRUKCQgLBRYCAwEAAh4BAheAAAoJEI2BgDwO
```

sudo apt-get update
sudo apt-get install -y docker-ce-cli-containerd.io



```
ubuntu@ip-172-31-44-130:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /usr/share/keyrings/docker
-archive-keyring.gpg > /dev/null
ubuntu@ip-172-31-44-130:~$ echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://downl
oad.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
ubuntu@ip-172-31-44-130:~$ sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io
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 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Err:4 https://download.docker.com/linux/ubuntu noble InRelease
  The following signatures couldn't be verified because the public key is not available: NO_PUBKEY 7EA0A9C3F273FCD8
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
143 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: An error occurred during the signature verification. The repository is not updated and the previous index files will
be used. GPG error: https://download.docker.com/linux/ubuntu noble InRelease: The following signatures couldn't be verif
ied because the public key is not available: NO_PUBKEY 7EA0A9C3F273FCD8
W: Failed to fetch https://download.docker.com/linux/ubuntu/dists/noble/InRelease  The following signatures couldn't be
verified because the public key is not available: NO_PUBKEY 7EA0A9C3F273FCD8
W: Some index files failed to download. They have been ignored, or old ones used instead.
W: Target Packages (stable/binary-amd64/Packages) is configured multiple times in /etc/apt/sources.list.d/archive_uri-ht
```



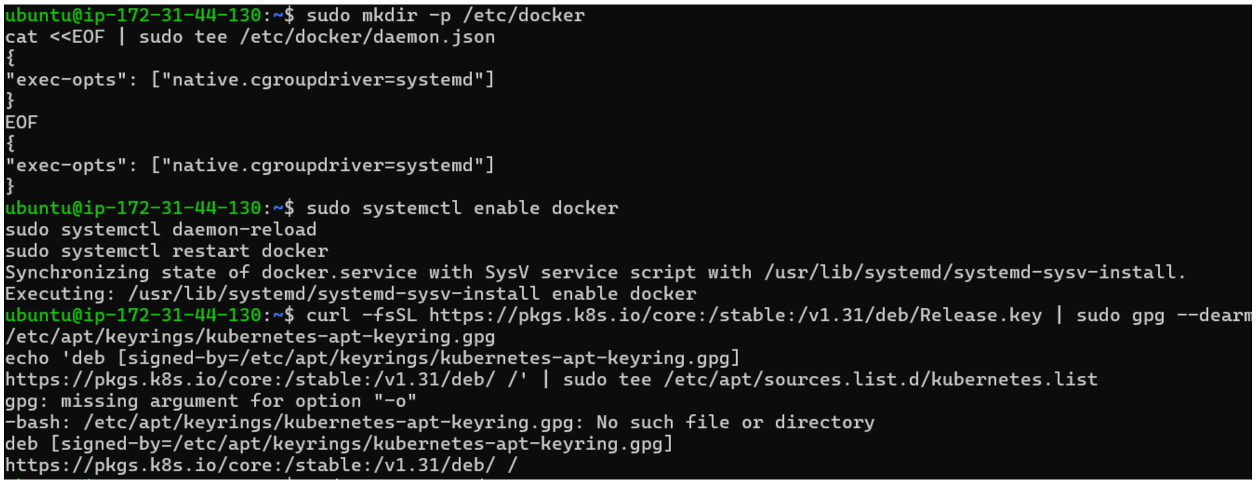
```
Unpacking slirp4netns (1.2.1-1build2) ...
Setting up docker-buildx-plugin (0.17.1-1~ubuntu.24.04~noble) ...
Setting up containerd.io (1.7.22-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.serv
ice.
Setting up docker-compose-plugin (2.29.7-1~ubuntu.24.04~noble) ...
Setting up libltdl7:amd64 (2.4.7-7build1) ...
Setting up docker-ce-cli (5:27.3.1-1~ubuntu.24.04~noble) ...
Setting up libslirp0:amd64 (4.7.0-1ubuntu3) ...
Setting up pigz (2.8-1) ...
Setting up docker-ce-rootless-extras (5:27.3.1-1~ubuntu.24.04~noble) ...
Setting up slirp4netns (1.2.1-1build2) ...
Setting up docker-ce (5:27.3.1-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
EOF

And

sudo systemctl enable docker
sudo systemctl daemon-reload

sudo systemctl restart docker

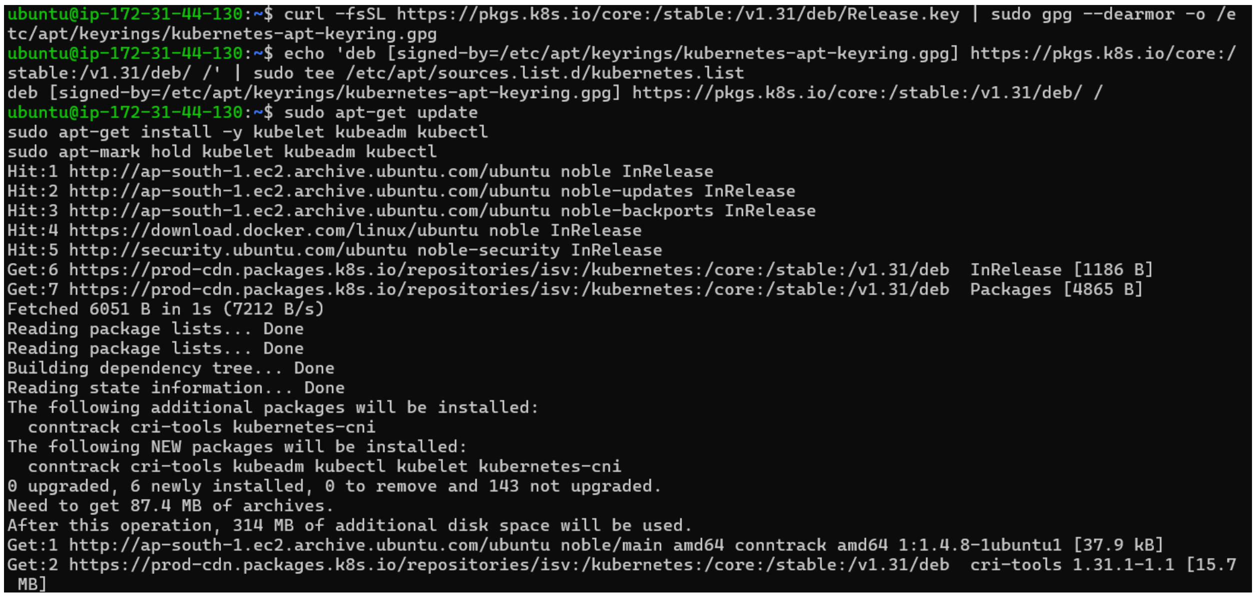


```
ubuntu@ip-172-31-44-130:~$ sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
ubuntu@ip-172-31-44-130:~$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
ubuntu@ip-172-31-44-130:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearm
/etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
gpg: missing argument for option "-o"
-bash: /etc/apt/keyrings/kubernetes-apt-keyring.gpg: No such file or directory
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
```

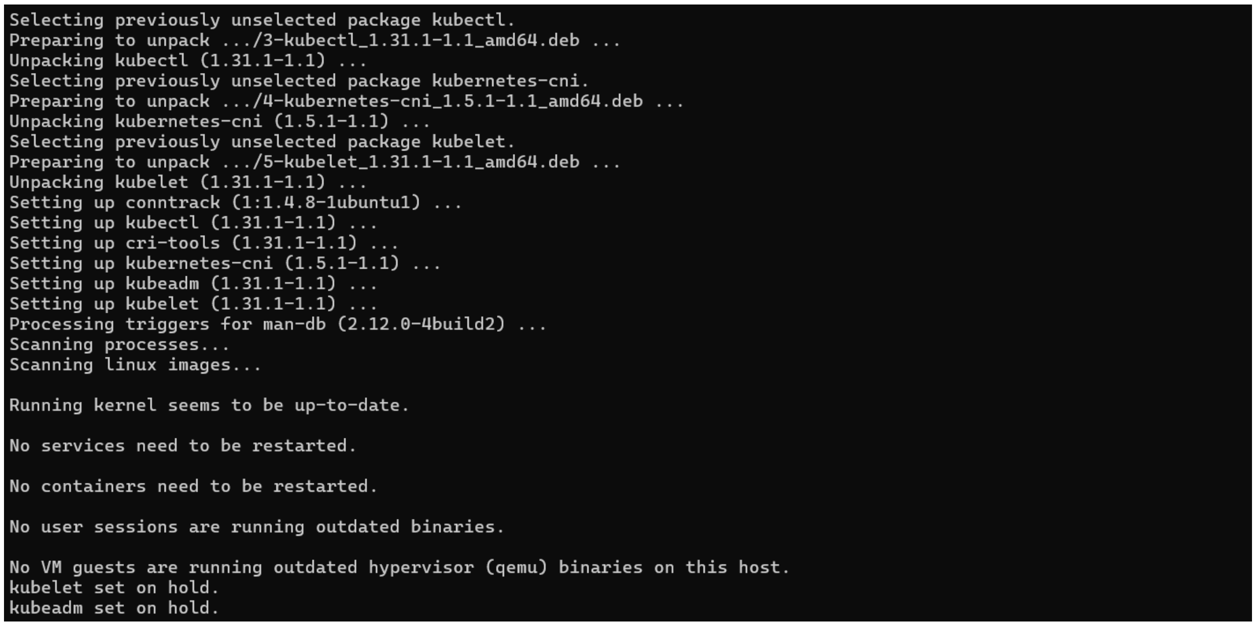
Step 5: Run the below command to install Kubernets. curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

And
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl

If any errors faced here please refer to:-
https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/
And
https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/

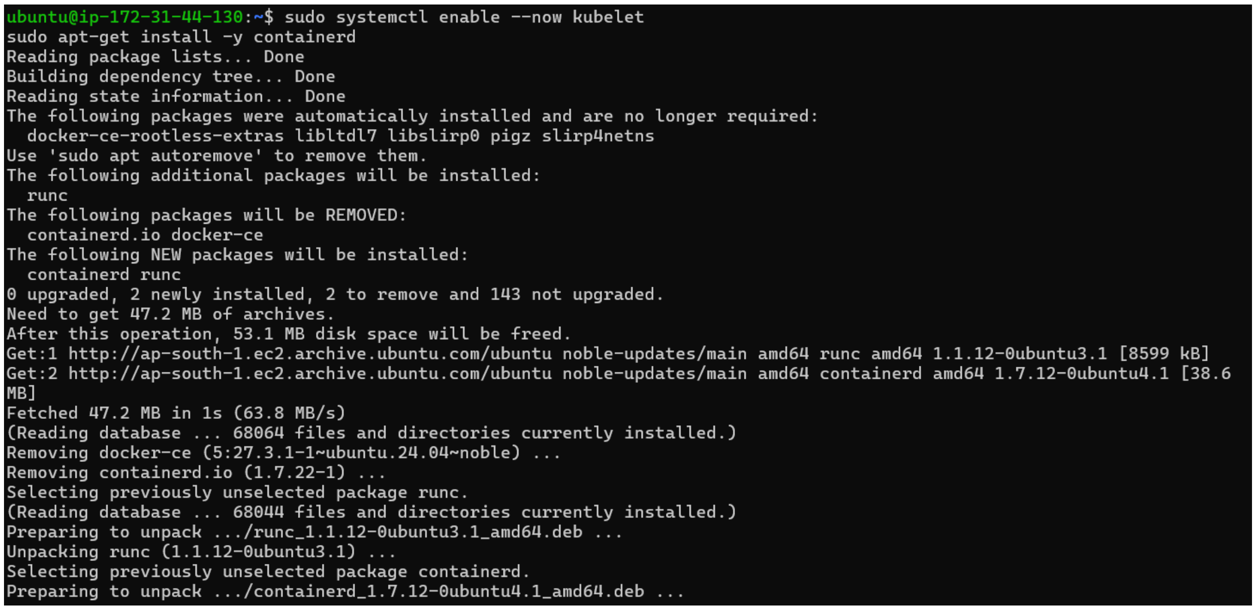


```
ubuntu@ip-172-31-44-130:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /e
tc/apt/keyrings/kubernetes-apt-keyring.gpg
ubuntu@ip-172-31-44-130:~$ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/
stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
ubuntu@ip-172-31-44-130:~$ sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
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 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  InRelease [1186 B]
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  Packages [4865 B]
Fetched 6051 B in 1s (7212 B/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools kubernetes-cni
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubectl kubelet kubernetes-cni
0 upgraded, 6 newly installed, 0 to remove and 143 not upgraded.
Need to get 87.4 MB of archives.
After this operation, 314 MB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 conntrack amd64 1:1.4.8-1ubuntu1 [37.9 kB]
Get:2 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  cri-tools 1.31.1-1.1 [15.7
 MB]
```



```
Selecting previously unselected package kubectl.
Preparing to unpack .../3-kubectl_1.31.1-1.1_amd64.deb ...
Unpacking kubectl (1.31.1-1.1) ...
Selecting previously unselected package kubernetes-cni.
Preparing to unpack .../4-kubernetes-cni_1.5.1-1.1_amd64.deb ...
Unpacking kubernetes-cni (1.5.1-1.1) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.31.1-1.1_amd64.deb ...
Unpacking kubelet (1.31.1-1.1) ...
Setting up conntrack (1:1.4.8-1ubuntu1) ...
Setting up kubectl (1.31.1-1.1) ...
Setting up cri-tools (1.31.1-1.1) ...
Setting up kubernetes-cni (1.5.1-1.1) ...
Setting up kubeadm (1.31.1-1.1) ...
Setting up kubelet (1.31.1-1.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
kubelet set on hold.
kubeadm set on hold.
```

sudo systemctl enable --now kubelet
sudo apt-get install -y containerd

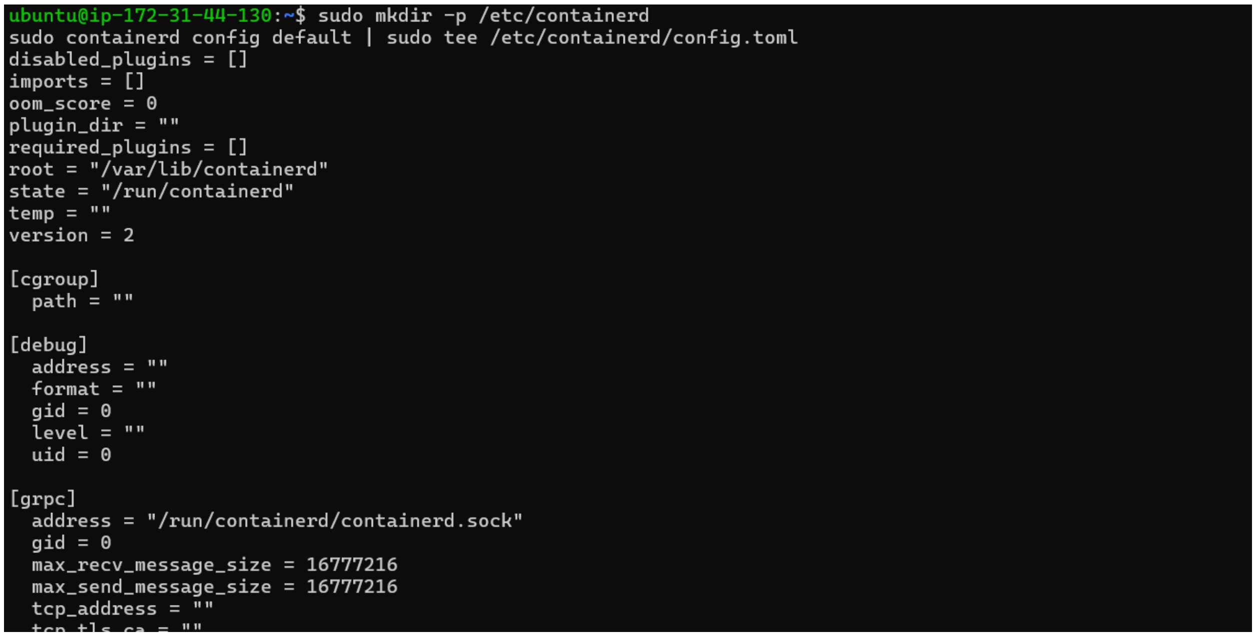


```
ubuntu@ip-172-31-44-130:~$ sudo systemctl enable --now kubelet
sudo apt-get install -y containerd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-ce-rootless-extras libltdl7 libslirp0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  runc
The following packages will be REMOVED:
  containerd.io docker-ce
The following NEW packages will be installed:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 143 not upgraded.
Need to get 47.2 MB of archives.
After this operation, 53.1 MB disk space will be freed.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu3.1 [8599 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.12-0ubuntu4.1 [38.6
MB]
Fetched 47.2 MB in 1s (63.8 MB/s)
(Reading database ... 68064 files and directories currently installed.)
Removing docker-ce (5:27.3.1-1~ubuntu.24.04~noble) ...
Removing containerd.io (1.7.22-1) ...
Selecting previously unselected package runc.
(Reading database ... 68044 files and directories currently installed.)
Preparing to unpack .../runc_1.1.12-0ubuntu3.1_amd64.deb ...
Unpacking runc (1.1.12-0ubuntu3.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../containerd_1.7.12-0ubuntu4.1_amd64.deb ...
```

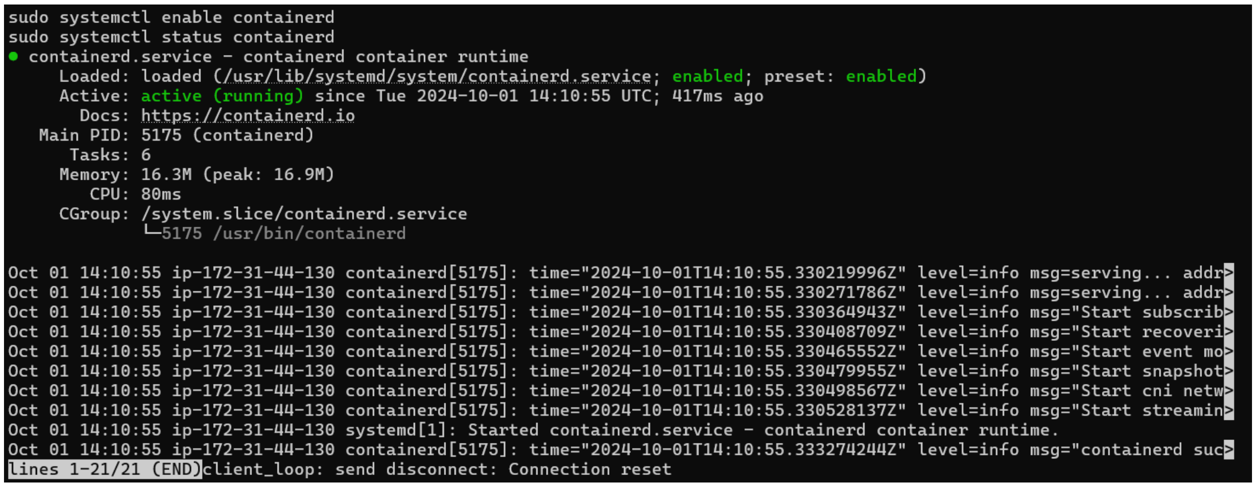
sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml

And

sudo systemctl restart containerd
sudo systemctl enable containerd
sudo systemctl status containerd

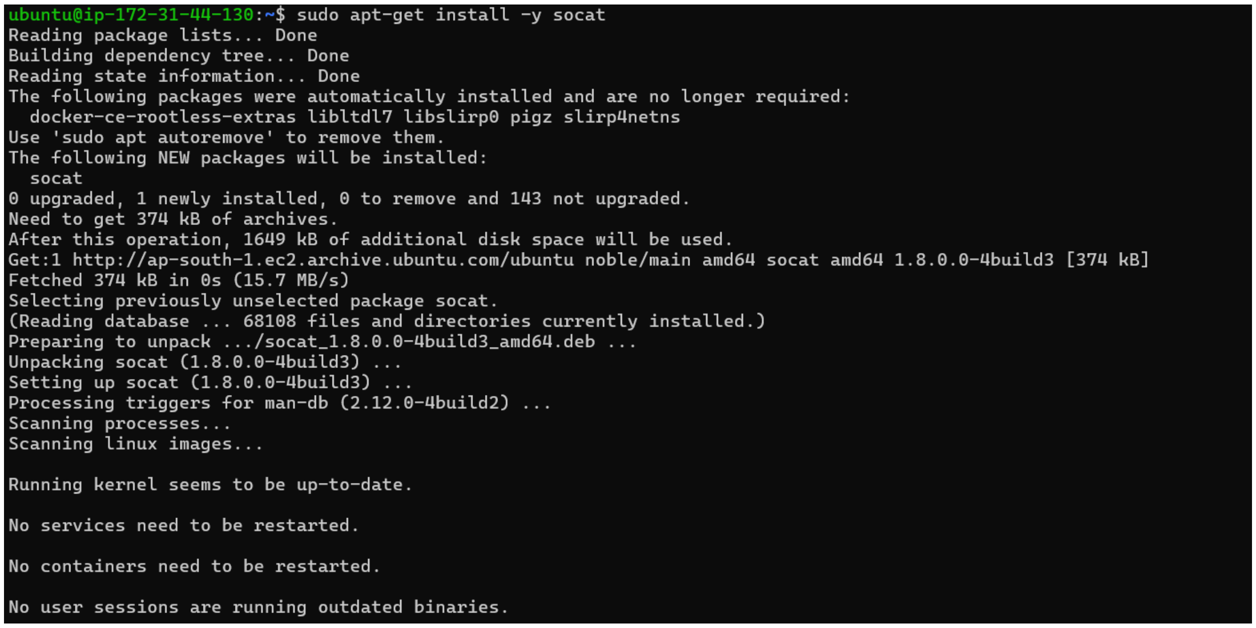


```
ubuntu@ip-172-31-44-130:~$ sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml
disabled_plugins = []
imports = []
oom_score = 0
plugin_dir = ""
required_plugins = []
root = "/var/lib/containerd"
state = "/run/containerd"
temp = ""
version = 2

[cgroup]
  path = ""

[debug]
  address = ""
  format = ""
  gid = 0
  level = ""
  uid = 0

[grpc]
  address = "/run/containerd/containerd.sock"
  gid = 0
  max_recv_message_size = 16777216
  max_send_message_size = 16777216
  tcp_address = ""
```



```
sudo systemctl enable containerd
sudo systemctl status containerd
● containerd.service - containerd container runtime
     Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
     Active: active (running) since Tue 2024-10-01 14:10:55 UTC; 417ms ago
       Docs: https://containerd.io
   Main PID: 5175 (containerd)
      Tasks: 6
     Memory: 16.3M (peak: 16.9M)
        CPU: 80ms
     CGroup: /system.slice/containerd.service
             └─5175 /usr/bin/containerd

Oct 01 14:10:55 ip-172-31-44-130 containerd[5175]: time="2024-10-01T14:10:55.330219996Z" level=info msg=serving... addr>
Oct 01 14:10:55 ip-172-31-44-130 containerd[5175]: time="2024-10-01T14:10:55.330271786Z" level=info msg=serving... addr>
Oct 01 14:10:55 ip-172-31-44-130 containerd[5175]: time="2024-10-01T14:10:55.330364943Z" level=info msg="Start subscrib>
Oct 01 14:10:55 ip-172-31-44-130 containerd[5175]: time="2024-10-01T14:10:55.330408709Z" level=info msg="Start recoveri>
Oct 01 14:10:55 ip-172-31-44-130 containerd[5175]: time="2024-10-01T14:10:55.330465552Z" level=info msg="Start event mo>
Oct 01 14:10:55 ip-172-31-44-130 containerd[5175]: time="2024-10-01T14:10:55.330479955Z" level=info msg="Start snapshot>
Oct 01 14:10:55 ip-172-31-44-130 containerd[5175]: time="2024-10-01T14:10:55.330498567Z" level=info msg="Start cni netw>
Oct 01 14:10:55 ip-172-31-44-130 containerd[5175]: time="2024-10-01T14:10:55.330528137Z" level=info msg="Start streamin>
Oct 01 14:10:55 ip-172-31-44-130 systemd[1]: Started containerd.service - containerd container runtime.
Oct 01 14:10:55 ip-172-31-44-130 containerd[5175]: time="2024-10-01T14:10:55.333274244Z" level=info msg="containerd suc>
lines 1-21/21 (END)client_loop: send disconnect: Connection reset
```

sudo apt-get install -y socat



```
ubuntu@ip-172-31-44-130:~$ sudo apt-get install -y socat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-ce-rootless-extras libltdl7 libslirp0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 143 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 socat amd64 1.8.0.0-4build3 [374 kB]
Fetched 374 kB in 0s (15.7 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68108 files and directories currently installed.)
Preparing to unpack .../socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Setting up socat (1.8.0.0-4build3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.
```

Step 6: Initialize the Kubecluster .Now Perform this Command only for Master. sudo kubeadm init --pod-network-cidr=10.244.0.0/16

If any errors in this command , run:-
1.) Enable IP Forwarding by running the following command:

```
sudo sysctl -w net.ipv4.ip_forward=1
```

2.) Make the Change Persistent (to ensure it remains active after a reboot):
Open the system control configuration file for editing:

```
sudo nano /etc/sysctl.conf
```

3.) Add or modify the following line:

```
net.ipv4.ip_forward=1
```

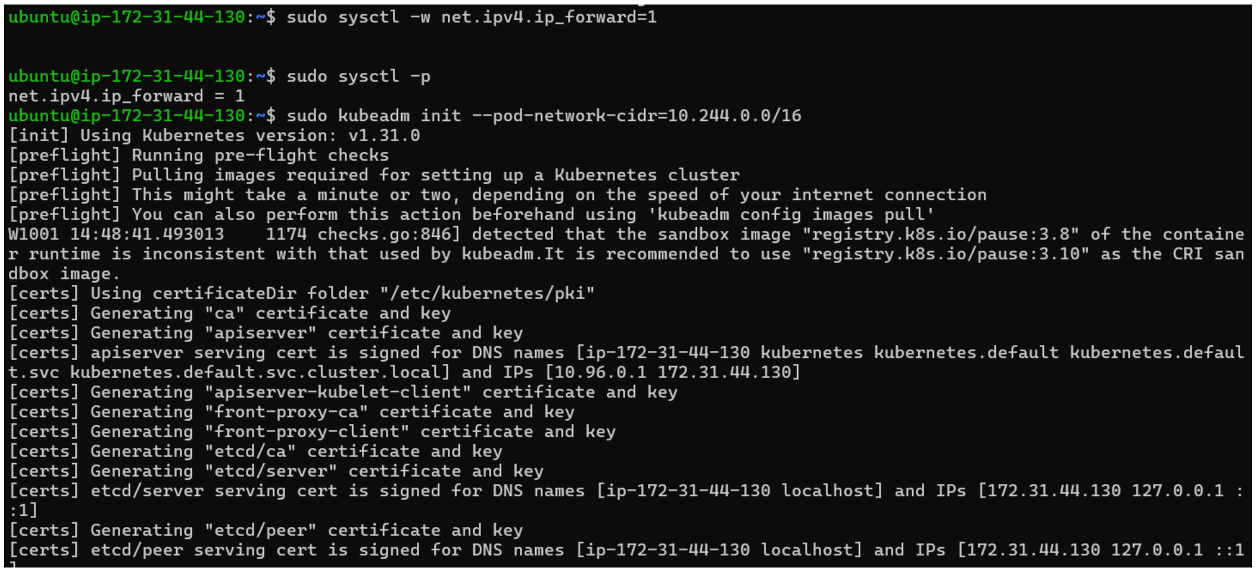
4.) Save the file and exit the editor.(Click ctrl+X,then Yes and then Enter)

5.) Apply the Changes:

```
sudo sysctl -p
```

6.) Re-run the `kubeadm init` command:

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
```



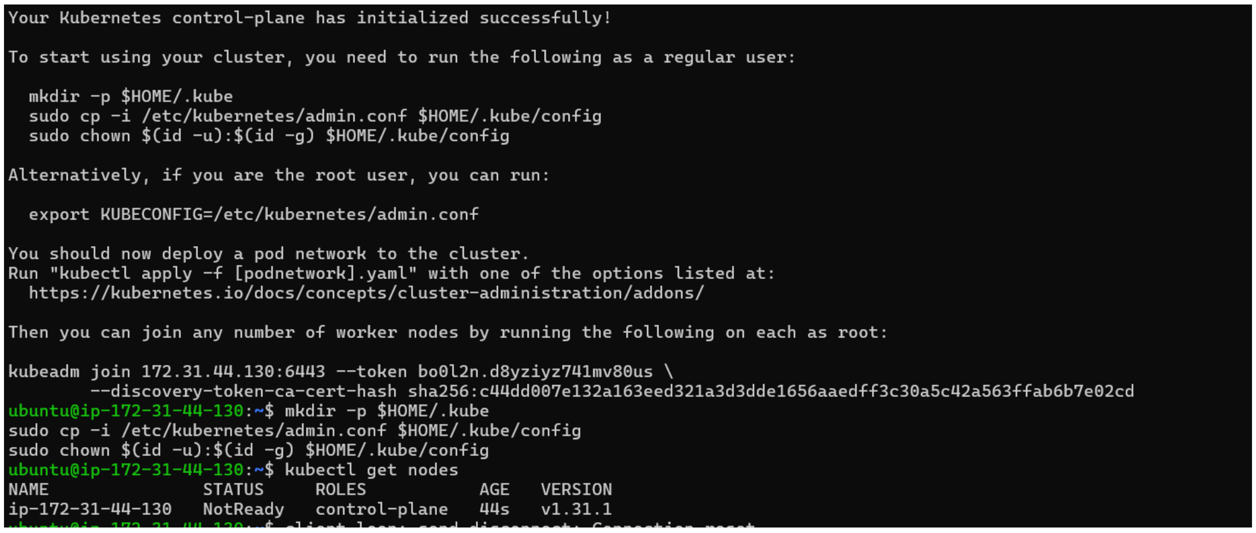
```
ubuntu@ip-172-31-44-130:~$ sudo sysctl -w net.ipv4.ip_forward=1


ubuntu@ip-172-31-44-130:~$ sudo sysctl -p
net.ipv4.ip_forward = 1
ubuntu@ip-172-31-44-130:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W1001 14:48:41.493013    1174 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the containe
r runtime is inconsistent with that used by kubeadm.It is recommended to use "registry.k8s.io/pause:3.10" as the CRI san
dbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-44-130 kubernetes kubernetes.default kubernetes.defaul
t.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.44.130]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-44-130 localhost] and IPs [172.31.44.130 127.0.0.1 :
:1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-44-130 localhost] and IPs [172.31.44.130 127.0.0.1 ::1
```

Run this command on master and also copy and save the Join command from below.

mkdir -p $HOME/.kube

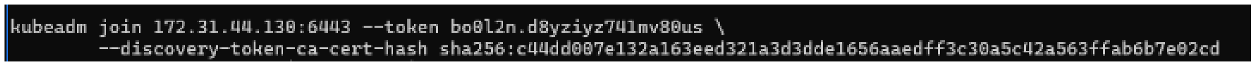
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config



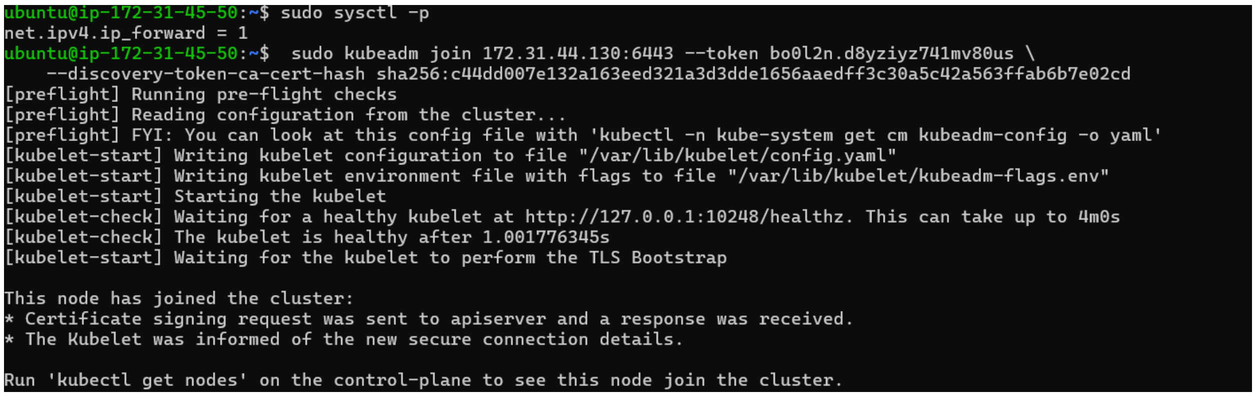
```
Your Kubernetes control-plane has initialized successfully!

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 172.31.44.130:6443 --token bo0l2n.d8yziyz741mv80us \
        --discovery-token-ca-cert-hash sha256:c44dd007e132a163eed321a3d3dde1656aaedff3c30a5c42a563ffab6b7e02cd
ubuntu@ip-172-31-44-130:~$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-44-130:~$ kubectl get nodes
NAME               STATUS     ROLES           AGE   VERSION
ip-172-31-44-130   NotReady   control-plane   44s   v1.31.1
```

Join command:-

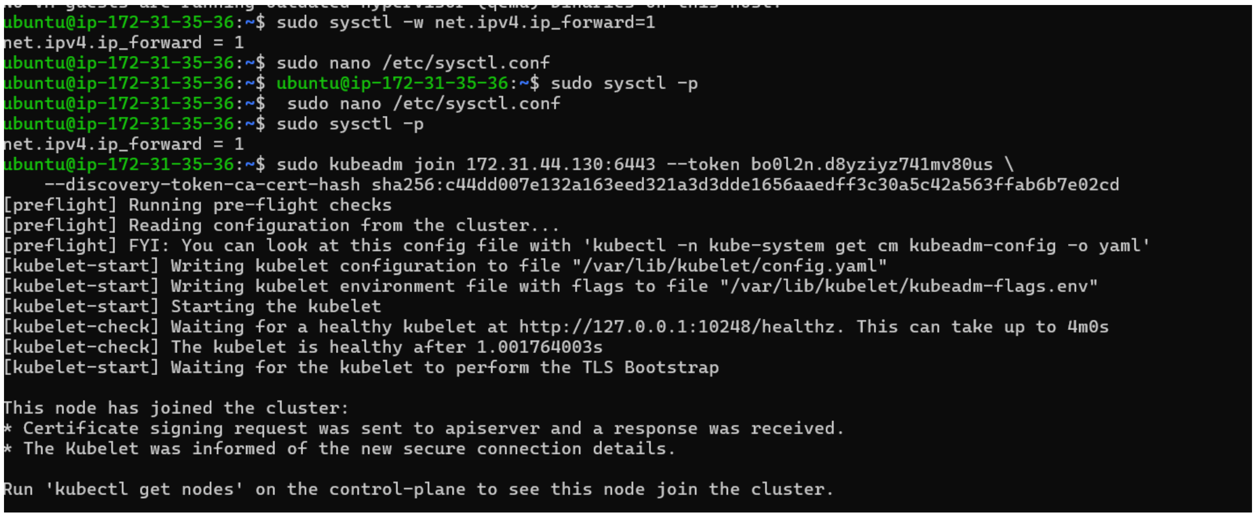


```
kubeadm join 172.31.44.130:6443 --token bo0l2n.d8yziyz741mv80us \
        --discovery-token-ca-cert-hash sha256:c44dd007e132a163eed321a3d3dde1656aaedff3c30a5c42a563ffab6b7e02cd
```

Step 8: Now Run the following command on Node 1 and Node 2 to Join to master. sudo kubeadm join <EC2 Instance Ip> --token <randomly_alloted_token>\ --discovery-token-ca-cert-hash sha256:d6fc5fb7e984c83e2807780047fec6c4f2acfe9da9184ecc028d77157608fbb6



```
ubuntu@ip-172-31-45-50:~$ sudo sysctl -p
net.ipv4.ip_forward = 1
ubuntu@ip-172-31-45-50:~$  sudo kubeadm join 172.31.44.130:6443 --token bo0l2n.d8yziyz741mv80us \
    --discovery-token-ca-cert-hash sha256:c44dd007e132a163eed321a3d3dde1656aaedff3c30a5c42a563ffab6b7e02cd
[preflight] Running pre-flight checks
[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 1.001776345s
[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.
```

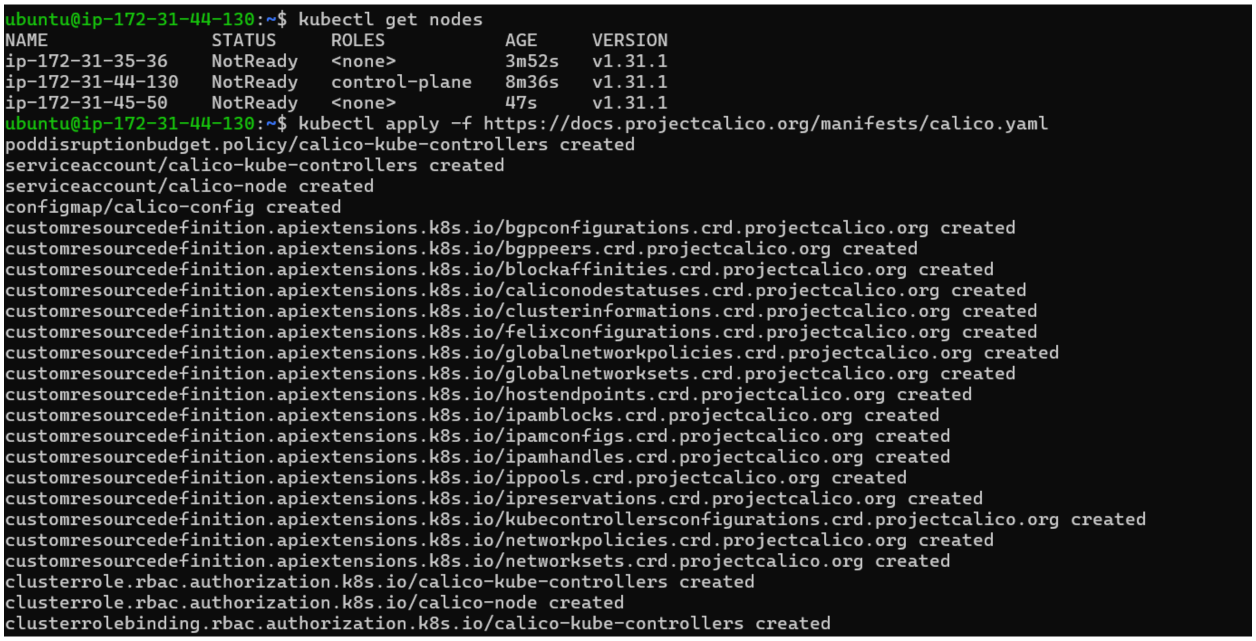


```
ubuntu@ip-172-31-35-36:~$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
ubuntu@ip-172-31-35-36:~$ sudo nano /etc/sysctl.conf
ubuntu@ip-172-31-35-36:~$ ubuntu@ip-172-31-35-36:~$ sudo sysctl -p
ubuntu@ip-172-31-35-36:~$  sudo nano /etc/sysctl.conf
ubuntu@ip-172-31-35-36:~$ sudo sysctl -p
net.ipv4.ip_forward = 1
ubuntu@ip-172-31-35-36:~$ sudo kubeadm join 172.31.44.130:6443 --token bo0l2n.d8yziyz741mv80us \
    --discovery-token-ca-cert-hash sha256:c44dd007e132a163eed321a3d3dde1656aaedff3c30a5c42a563ffab6b7e02cd
[preflight] Running pre-flight checks
[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 1.001764003s
[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.
```

Step 9: Now Run the command kubectl get nodes to see the nodes after executing Join command on nodes.

And

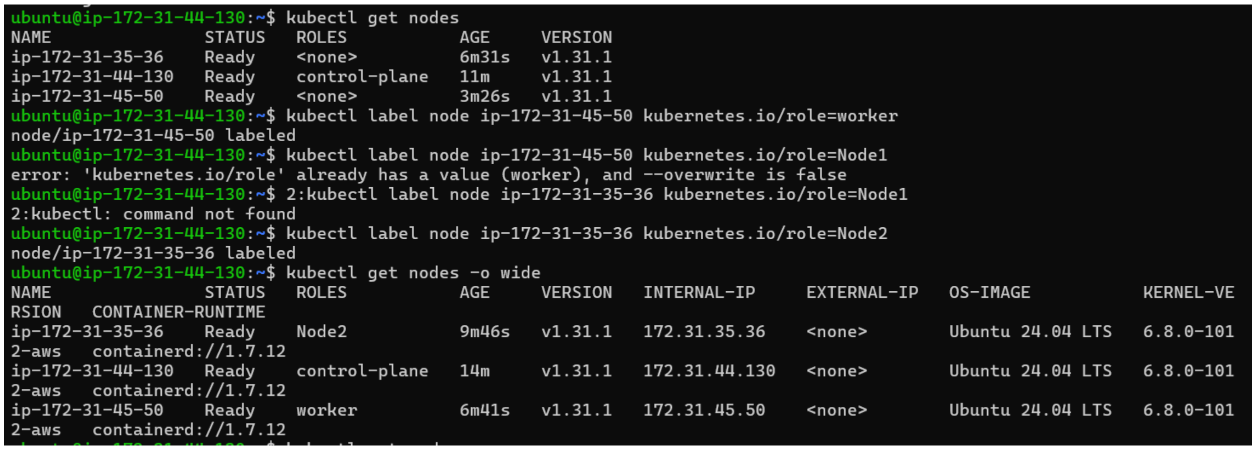
Step 10: Since Status is NotReady we have to add a network plugin. And also we have to give the name to the nodes. kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml



```
ubuntu@ip-172-31-44-130:~$ kubectl get nodes
NAME               STATUS     ROLES           AGE     VERSION
ip-172-31-35-36    NotReady   <none>          3m52s   v1.31.1
ip-172-31-44-130   NotReady   control-plane   8m36s   v1.31.1
ip-172-31-45-50    NotReady   <none>          47s     v1.31.1
ubuntu@ip-172-31-44-130:~$ kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml
poddisruptionbudget.policy/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
serviceaccount/calico-node created
configmap/calico-config created
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipreservations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
```

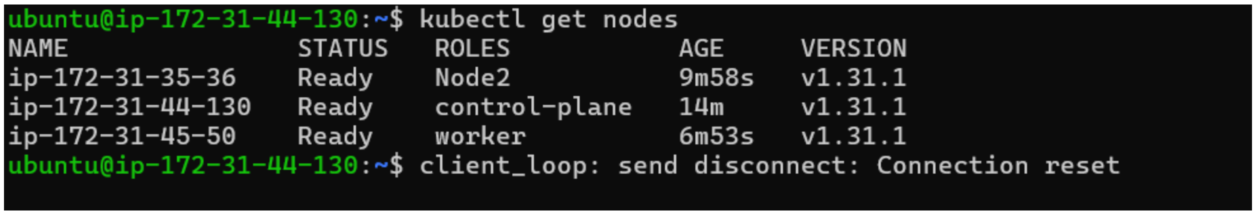
Now Run command kubectl get nodes -o wide
Or
kubectl get nodes so
we can see Status is ready.

Renaming:-
Rename to Node 1:kubectl label node ip-172-31-45-50 kubernetes.io/role=Node1
Rename to Node 2:kubectl label node ip-172-31-35-36 kubernetes.io/role=Node2



```
ubuntu@ip-172-31-44-130:~$ kubectl get nodes
NAME               STATUS   ROLES           AGE     VERSION
ip-172-31-35-36    Ready    <none>          6m31s   v1.31.1
ip-172-31-44-130   Ready    control-plane   11m     v1.31.1
ip-172-31-45-50    Ready    <none>          3m26s   v1.31.1
ubuntu@ip-172-31-44-130:~$ kubectl label node ip-172-31-45-50 kubernetes.io/role=worker
node/ip-172-31-45-50 labeled
ubuntu@ip-172-31-44-130:~$ kubectl label node ip-172-31-45-50 kubernetes.io/role=Node1
error: 'kubernetes.io/role' already has a value (worker), and --overwrite is false
ubuntu@ip-172-31-44-130:~$ 2:kubectl label node ip-172-31-35-36 kubernetes.io/role=Node1
2:kubectl: command not found
ubuntu@ip-172-31-44-130:~$ kubectl label node ip-172-31-35-36 kubernetes.io/role=Node2
node/ip-172-31-35-36 labeled
ubuntu@ip-172-31-44-130:~$ kubectl get nodes -o wide
NAME               STATUS   ROLES           AGE     VERSION   INTERNAL-IP     EXTERNAL-IP   OS-IMAGE           KERNEL-VE
RSION   CONTAINER-RUNTIME
ip-172-31-35-36    Ready    Node2           9m46s   v1.31.1   172.31.35.36    <none>        Ubuntu 24.04 LTS   6.8.0-101
2-aws   containerd://1.7.12
ip-172-31-44-130   Ready    control-plane   14m     v1.31.1   172.31.44.130   <none>        Ubuntu 24.04 LTS   6.8.0-101
2-aws   containerd://1.7.12
ip-172-31-45-50    Ready    worker          6m41s   v1.31.1   172.31.45.50    <none>        Ubuntu 24.04 LTS   6.8.0-101
2-aws   containerd://1.7.12
```

run kubectl get nodes



```
ubuntu@ip-172-31-44-130:~$ kubectl get nodes
NAME               STATUS   ROLES           AGE     VERSION
ip-172-31-35-36    Ready    Node2           9m58s   v1.31.1
ip-172-31-44-130   Ready    control-plane   14m     v1.31.1
ip-172-31-45-50    Ready    worker          6m53s   v1.31.1
ubuntu@ip-172-31-44-130:~$ client_loop: send disconnect: Connection reset
```

Conclusion:

In this experiment, we learned how to install Kubernetes create a Kubernetes Cluster in AWS EC2 instances and get them up and running.