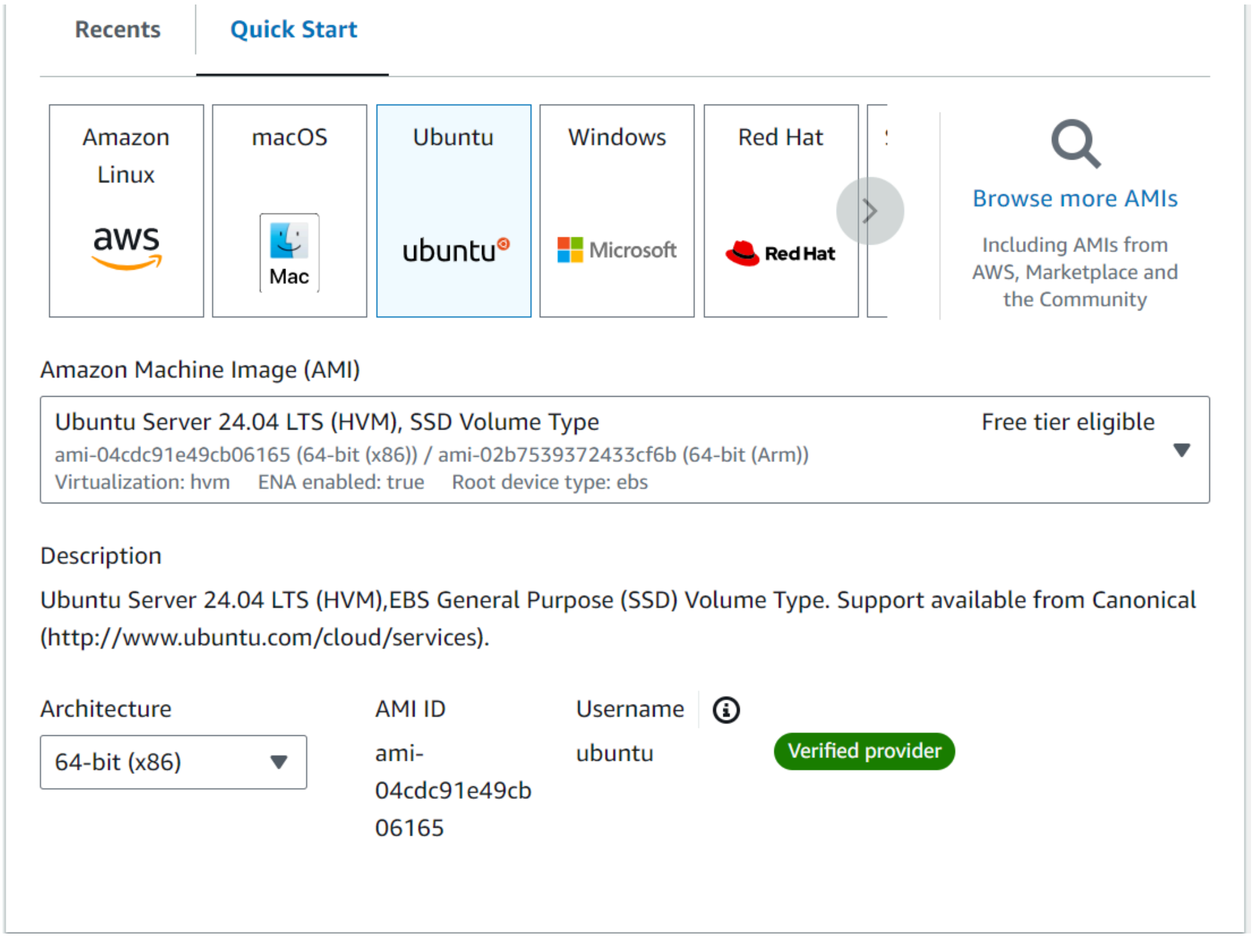
ADVANCE DEVOPS EXPERIMENT NO 3

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 revolutionized how development and operations teams test and deploy modern software. Containers allow companies to scale and deploy applications more efficiently, but they also introduce new challenges, adding complexity by creating a whole new infrastructure ecosystem. Today, both large and small software companies are deploying thousands of container instances daily. Managing this level of complexity at scale requires advanced tools. Enter Kubernetes. Originally developed by Google, Kubernetes is an open-source container orchestration platform designed to automate the deployment, scaling, and management of containerized applications. Kubernetes has quickly become the de facto standard for container orchestration and is the flagship project of the Cloud Native Computing Foundation (CNCF), supported by major players like Google, AWS, Microsoft, IBM, Intel, Cisco, and Red Hat. Kubernetes simplifies the deployment and operation of applications in a microservice architecture by providing an abstraction layer over a group of hosts. This allows development teams to deploy their applications while Kubernetes takes care of key tasks, including:
● Managing resource consumption by applications or teams
● Distributing application load evenly across the infrastructure
● Automatically load balancing requests across multiple instances of an application
● Monitoring resource usage to prevent applications from exceeding resource limits and automatically restarting them if needed
● Moving application instances between hosts when resources are low or if a host fails
● Automatically utilizing additional resources when new hosts are added to the cluster
● Facilitating canary deployments and rollbacks with ease
Necessary Requirements:
● EC2 Instance: The experiment required launching a t2.medium EC2 instance with 2 CPUs, as
Kubernetes demands sufficient resources for effective functioning.
● Minimum Requirements:
○ Instance Type: t2.medium
○ CPUs: 2
○ Memory: Adequate for container orchestration.
This ensured that the Kubernetes cluster had the necessary resources to function smoothly.

Step 1: Log in to your AWS Academy/personal account and launch 3 new Ec2 Instances. Select Ubuntu as AMI and t2.medium as Instance Type and create a key of type RSA with .pem extension and move the downloaded key to the new folder.We can use 3 Different keys or 1 common key also.
Note: A minimum of 2 CPUs are required so Please select t2.medium and do not forget to stop the instance after the experiment because it is not available in the free tier. Also Select Security groups from existing.

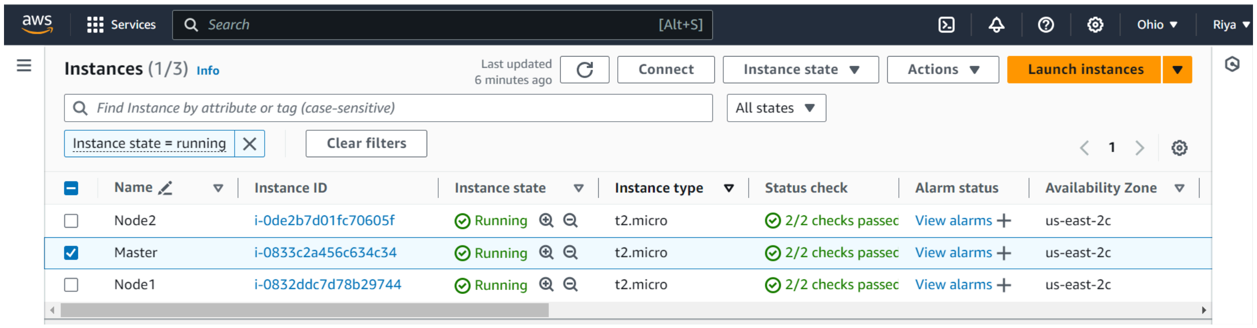


Amazon Machine Image (AMI)
Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-04cdc91e49cb06165 (64-bit (x86)) / ami-02b7539372433cf6b (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description
Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

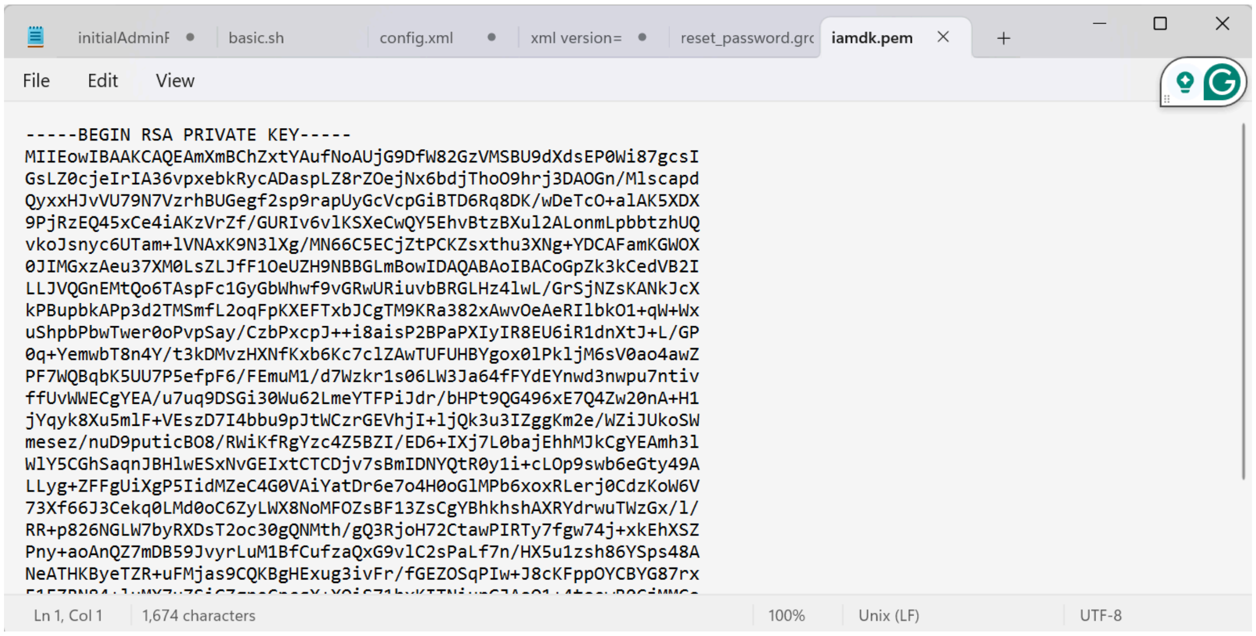
Architecture: 64-bit (x86)
AMI ID: ami-04cdc91e49cb06165
Username: ubuntu
Verified provider

Step 2: After creating the instances click on Connect & connect all 3 instances and navigate to SSH Client.



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☐	Node2	i-0de2b7d01fc70605f	Running	t2.micro	2/2 checks passec	View alarms +	us-east-2c
☑	Master	i-0833c2a456c634c34	Running	t2.micro	2/2 checks passec	View alarms +	us-east-2c
☐	Node1	i-0832ddc7d78b29744	Running	t2.micro	2/2 checks passec	View alarms +	us-east-2c

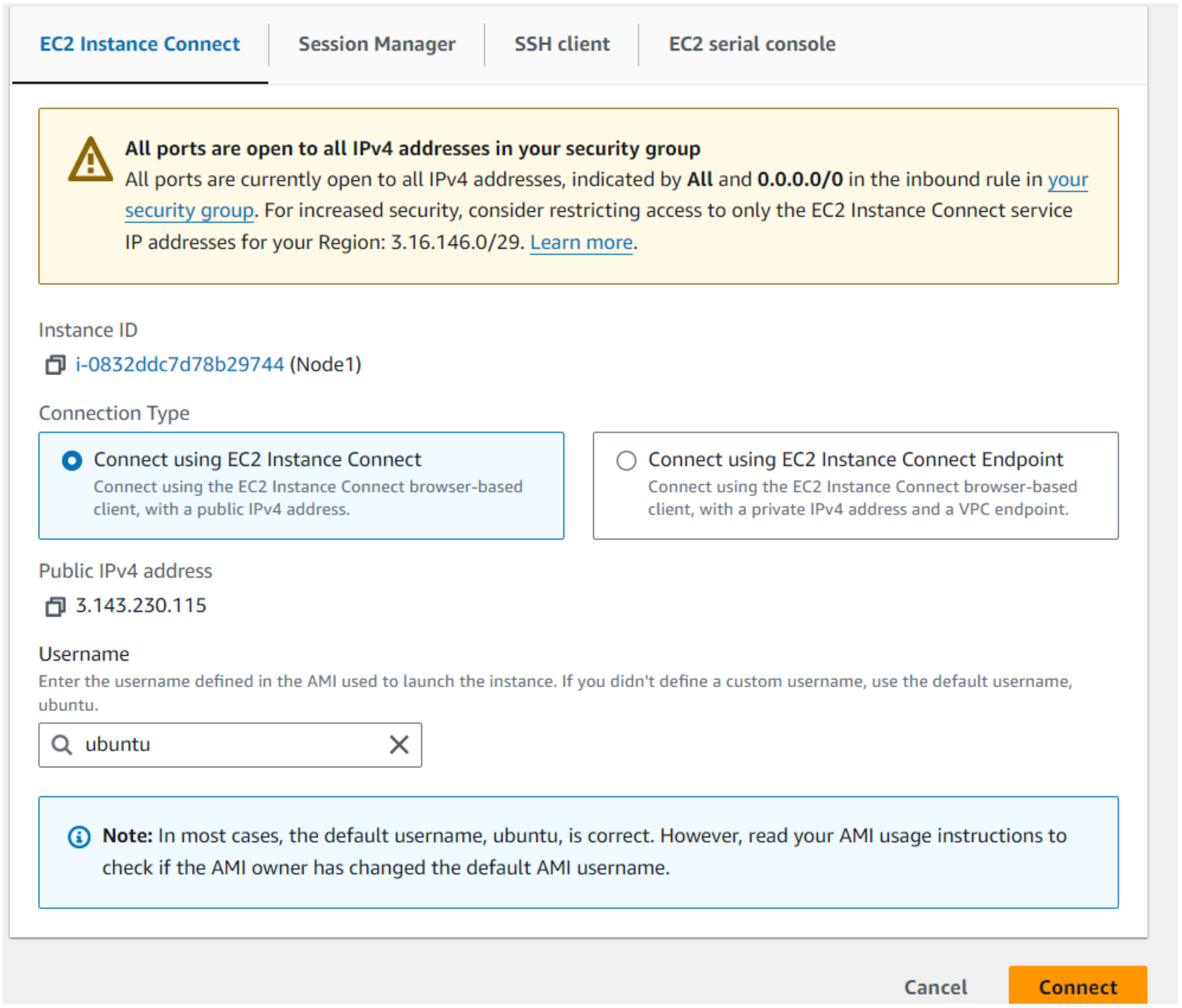
Downloaded key:



```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAmXmBChZxtYAufNoAUjG9DfW82GzVMSBU9dXdsEP0Wi87gcsI
GsLZ0cjeIrIA36vpxebkRycADaspLZ8rZOejNx6bdjThoO9hrj3DAOGn/Mlscapd
QyxxHJvVU79N7VzrhBUGegf2sp9rapUyGcVcpGiBTD6Rq8DK/wDeTcO+alAK5XDX
9PjRzEQ45xCe4iAKzVrZf/GURIv6vlKSXeCwQY5EhvBtzBXul2ALonmLpbbtzhUQ
vkoJsnyc6UTam+lVNAxK9N3lXg/MN66C5ECjZtPCKZsxthu3XNg+YDCAFamKGWOX
0JIMGxzAeu37XM0LsZLJfF1OeUZH9NBBGLmBowIDAQABAoIBACoGpZk3kCedVB2I
LLJVQGnEMtQo6TAspFc1GyGbWhwf9vGRwURiuvbBRGLHz4lwL/GrSjNZsKANkJcX
kPBupbkAPp3d2TMSmfL2oqFpKXEFTxbJCgTM9KRa382xAwvOeAeRIlbkO1+qW+Wx
uShpbPbwTwer0oPvpSay/CzbPxcpJ++i8aisP2BPaPXIyIR8EU6iR1dnXtJ+L/GP
0q+YemwbT8n4Y/t3kDMvzHXNfKxb6Kc7clZAwTUFUHBYgox0lPkljM6sV0ao4awZ
PF7WQBqbK5UU7P5efpF6/FEmuM1/d7Wzkr1s06LW3Ja64fFYdEYnwd3nwpu7ntiv
ffUvWWECgYEA/u7uq9DSGi30Wu62LmeYTFPiJdr/bHPt9QG496xE7Q4Zw20nA+H1
jYqyk8Xu5mlF+VEszD7I4bbu9pJtWCzrGEVhjI+ljQk3u3IZggKm2e/WZiJUkoSW
mesez/nuD9puticBO8/RWiKfRgYzc4Z5BZI/ED6+IXj7L0bajEhhMJkCgYEAmh3l
WlY5CGhSaqnJBHlwESxNvGEIxtCTCDjv7sBmIDNYQtR0y1i+cLOp9swb6eGty49A
LLyg+ZFFgUiXgP5IidMZeC4G0VAiYatDr6e7o4H0oGlMPb6xoxRLerj0CdzKoW6V
73Xf66J3Cekq0LMd0oC6ZyLWX8NoMFOZsBF13ZsCgYBhkhshAXRYdrwuTWzGx/l/
RR+p826NGLW7byRXDsT2oc30gQNMth/gQ3RjoH72CtawPIRTy7fgw74j+xkEhXSZ
Pny+aoAnQZ7mDB59JvyrLuM1BfCufzaQxG9vlC2sPaLf7n/HX5u1zsh86YSps48A
NeATHKByeTZR+uFMjas9CQKBgHExug3ivFr/fGEZOSqPIw+J8cKFppOYCBYG87rx
```

Ln 1, Col 1 | 1,674 characters | 100% | Unix (LF) | UTF-8

Step 3: Now open the folder in the terminal 3 times for Master, Node1& Node 2 where our .pem key is stored and paste the Example command (starting with ssh -i .....) in the terminal.(ssh -i "Master_Ec2_Key.pem" ubuntu@ec2-54-196-129-215.compute-1.amazonaws.com)



All ports are open to all IPv4 addresses in your security group
All ports are currently open to all IPv4 addresses, indicated by **All** and **0.0.0.0/0** in the inbound rule in your security group. For increased security, consider restricting access to only the EC2 Instance Connect service IP addresses for your Region: 3.16.146.0/29. Learn more.

Instance ID
i-0832ddc7d78b29744 (Node1)

Connection Type
[✓ Connect using EC2 Instance Connect]
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

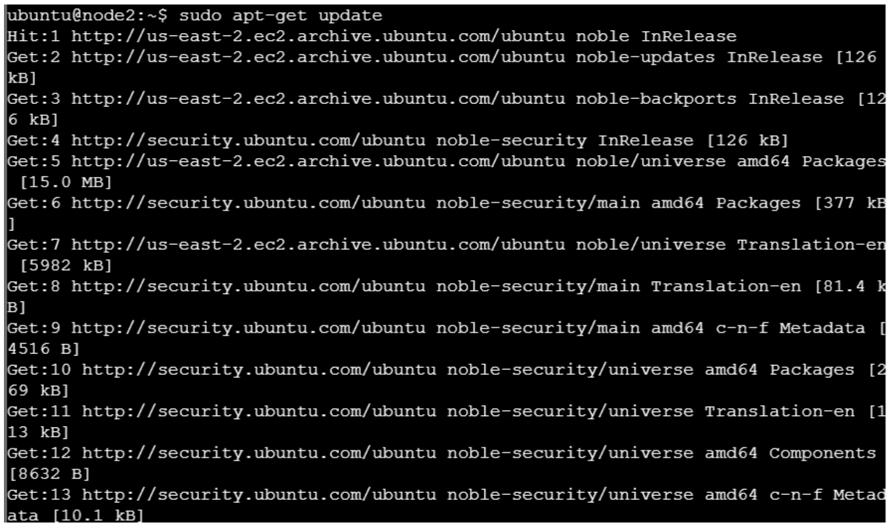
Public IPv4 address
3.143.230.115

Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.
ubuntu

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

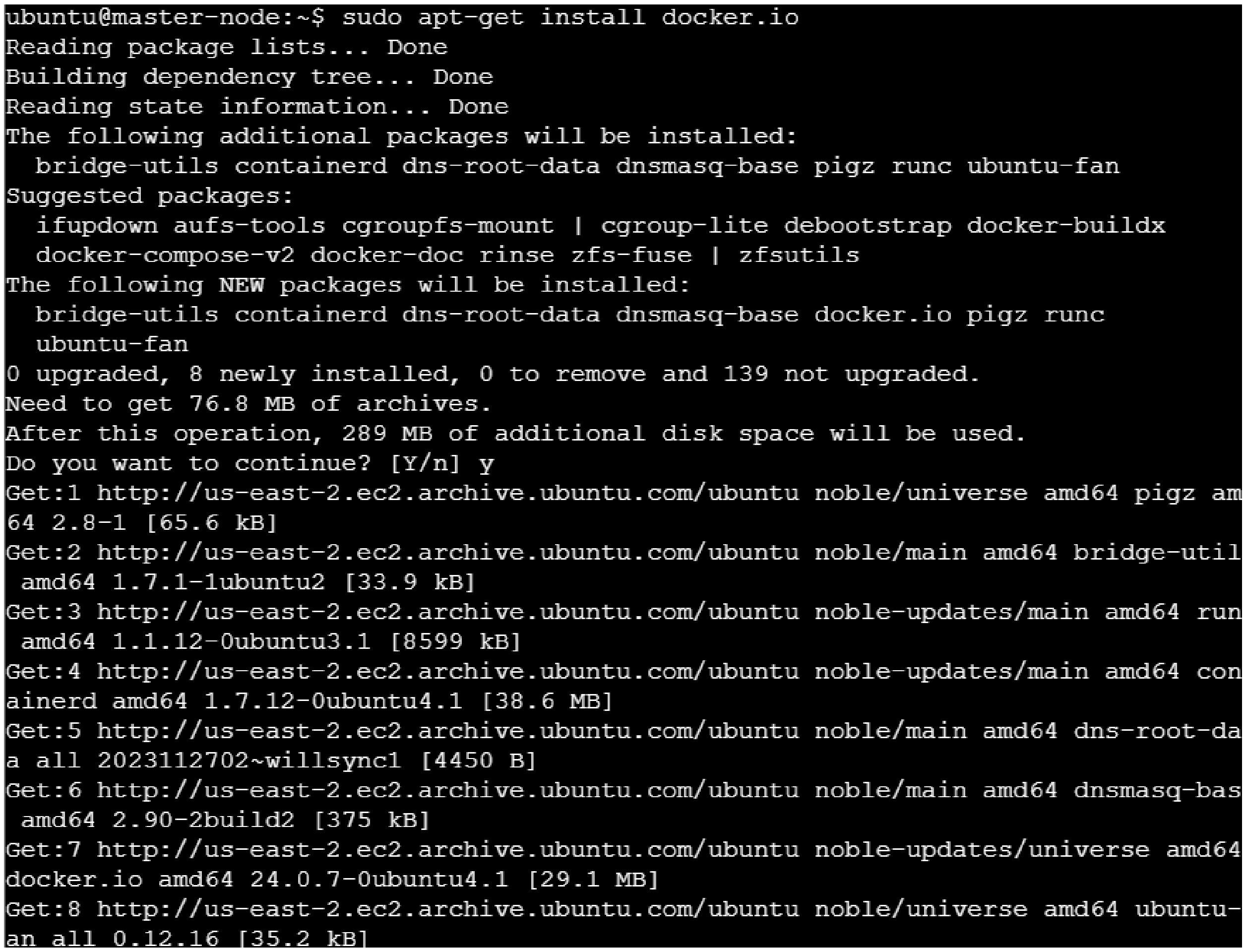
Step 4: Run on Master,Node 1,and Node 2 the below commands to install and setup Docker in Master, Node1, and Node2.

sudo apt-get update



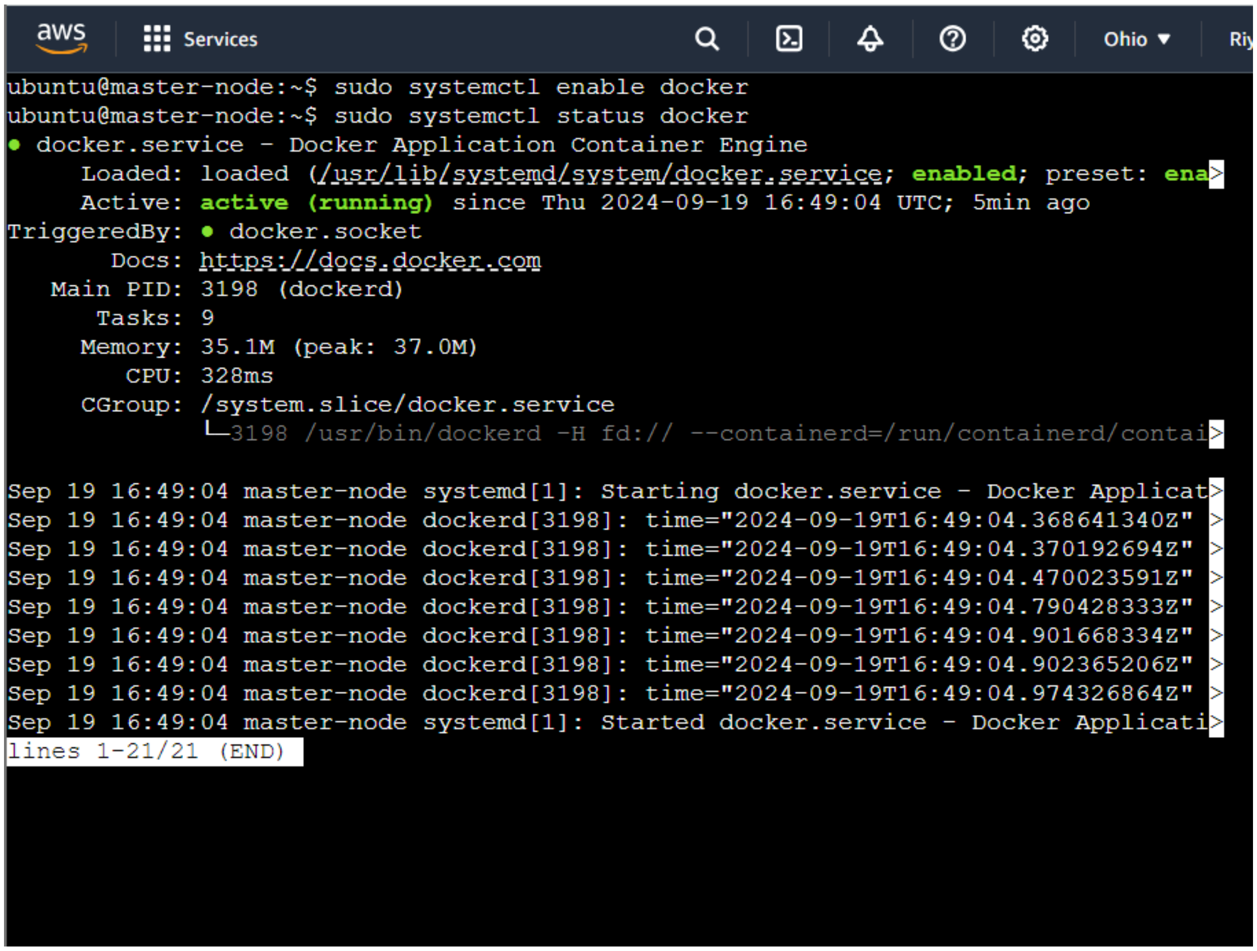
```
ubuntu@node2:~$ sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126
kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [12
6 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages
 [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [377 kB
]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en
 [5982 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [81.4 k
B]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [
4516 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [2
69 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [1
13 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components
[8632 B]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metad
ata [10.1 kB]
```

Sudo apt-get install docker.io



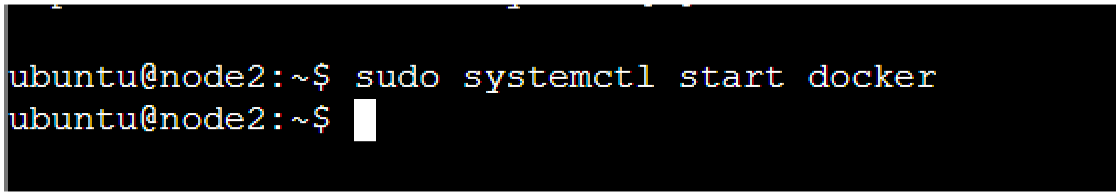
```
ubuntu@master-node:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  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 139 not upgraded.
Need to get 76.8 MB of archives.
After this operation, 289 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz am
64 2.8-1 [65.6 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 bridge-util
 amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 run
 amd64 1.1.12-0ubuntu3.1 [8599 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 con
ainerd amd64 1.7.12-0ubuntu4.1 [38.6 MB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dns-root-da
a all 2023112702~willsync1 [4450 B]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dnsmasq-bas
 amd64 2.90-2build2 [375 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64
docker.io amd64 24.0.7-0ubuntu4.1 [29.1 MB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 ubuntu-
an all 0.12.16 [35.2 kB]
```

Sudo systemctl enable docker
Sudo systemctl status docker



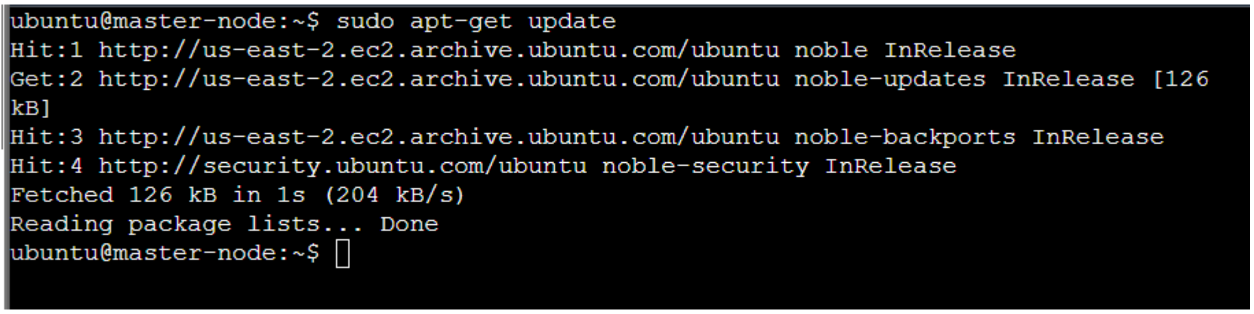
```
ubuntu@master-node:~$ sudo systemctl enable docker
ubuntu@master-node:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: ena>
     Active: active (running) since Thu 2024-09-19 16:49:04 UTC; 5min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 3198 (dockerd)
      Tasks: 9
     Memory: 35.1M (peak: 37.0M)
        CPU: 328ms
     CGroup: /system.slice/docker.service
             └─3198 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contai>

Sep 19 16:49:04 master-node systemd[1]: Starting docker.service - Docker Applicat>
Sep 19 16:49:04 master-node dockerd[3198]: time="2024-09-19T16:49:04.368641340Z" >
Sep 19 16:49:04 master-node dockerd[3198]: time="2024-09-19T16:49:04.370192694Z" >
Sep 19 16:49:04 master-node dockerd[3198]: time="2024-09-19T16:49:04.470023591Z" >
Sep 19 16:49:04 master-node dockerd[3198]: time="2024-09-19T16:49:04.790428333Z" >
Sep 19 16:49:04 master-node dockerd[3198]: time="2024-09-19T16:49:04.901668334Z" >
Sep 19 16:49:04 master-node dockerd[3198]: time="2024-09-19T16:49:04.902365206Z" >
Sep 19 16:49:04 master-node dockerd[3198]: time="2024-09-19T16:49:04.974326864Z" >
Sep 19 16:49:04 master-node systemd[1]: Started docker.service - Docker Applicati>
lines 1-21/21 (END)
```

Sudo systemctl start docker



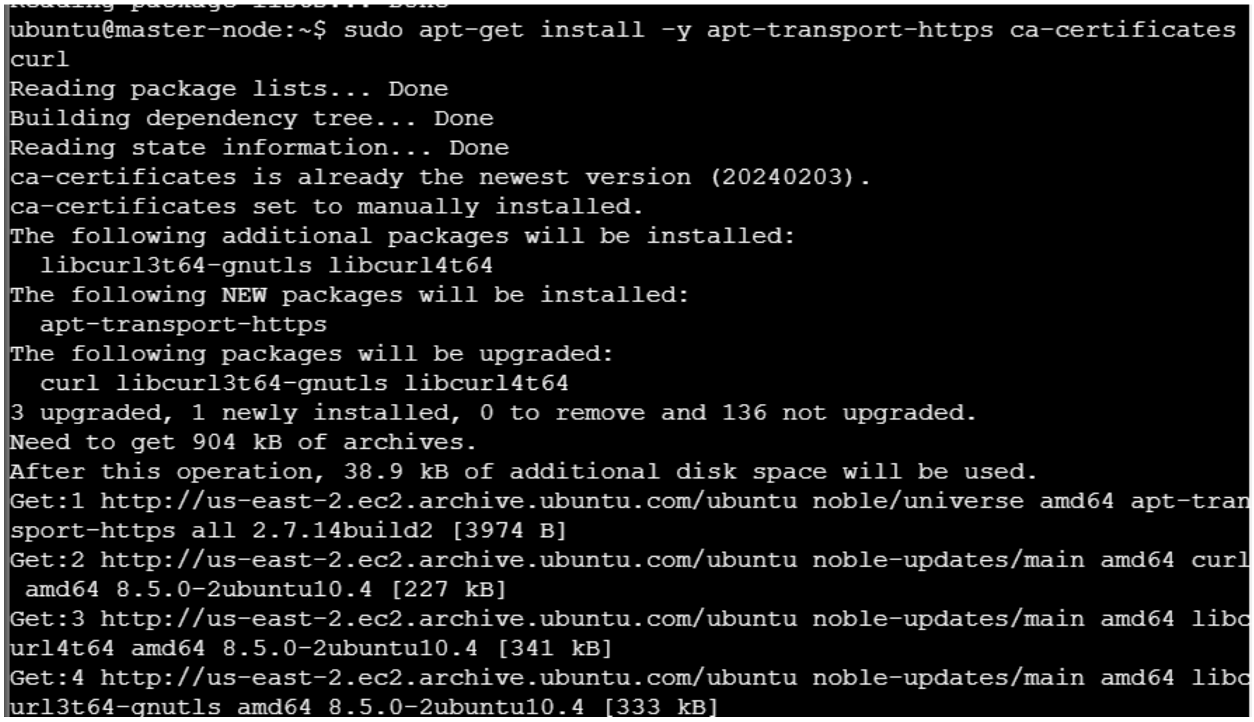
```
ubuntu@node2:~$ sudo systemctl start docker
ubuntu@node2:~$ 
```

Step 5: Run the below command to install Kubernets.



```
ubuntu@master-node:~$ sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126
kB]
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Fetched 126 kB in 1s (204 kB/s)
Reading package lists... Done
ubuntu@master-node:~$ 
```

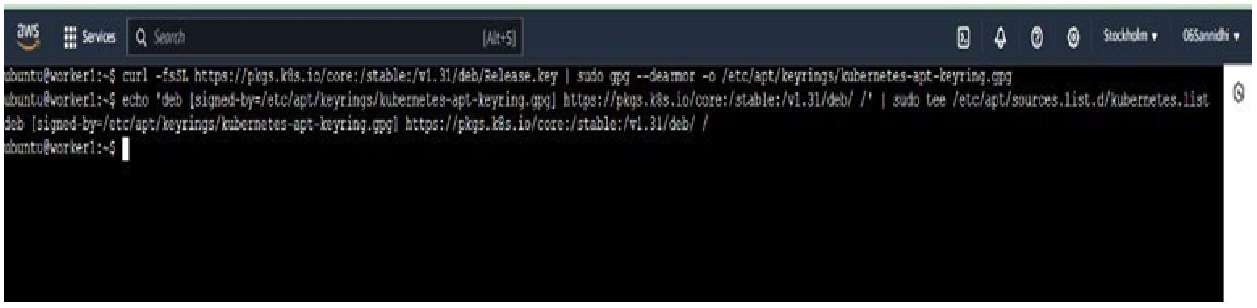
sudo apt-get install -y apt-transport-https ca-certificates curl



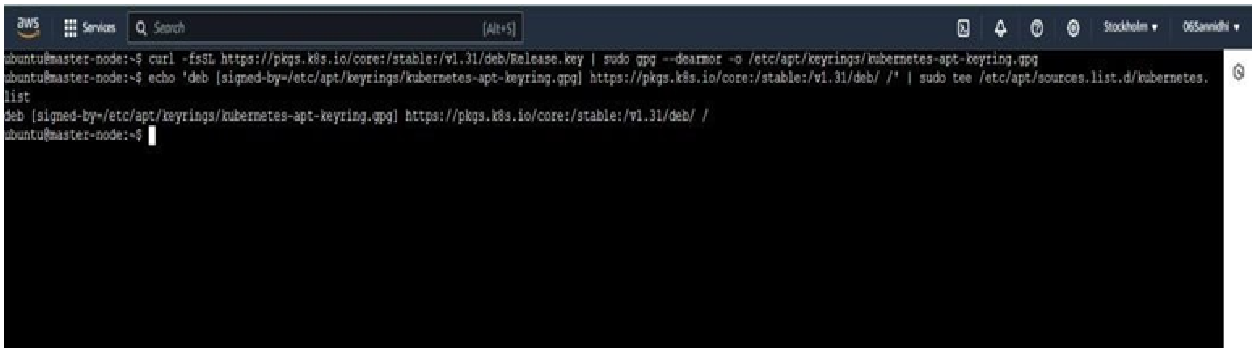
```
ubuntu@master-node:~$ sudo apt-get install -y apt-transport-https ca-certificates
curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
The following 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 136 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://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 apt-tran
sport-https all 2.7.14build2 [3974 B]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 curl
 amd64 8.5.0-2ubuntu10.4 [227 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libc
url4t64 amd64 8.5.0-2ubuntu10.4 [341 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libc
url3t64-gnutls amd64 8.5.0-2ubuntu10.4 [333 kB]
```

$ sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg
(download the google cloud public signing key)
(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=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee/etc/apt/sources.list.d/kubernetes.list (add the Kubernetes apt repository:)
(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
)



```
ubuntu@worker1:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
ubuntu@worker1:~$ 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@worker1:~$ 
```

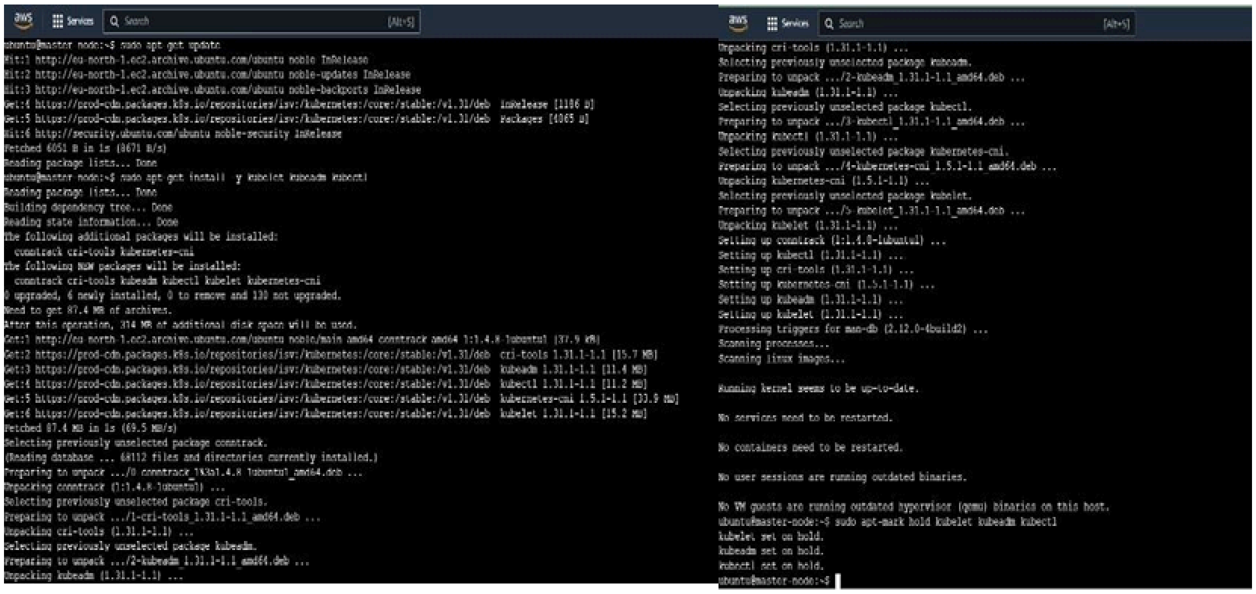


```
ubuntu@master-node:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
ubuntu@master-node:~$ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | 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@master-node:~$
```

$ sudo apt-get update

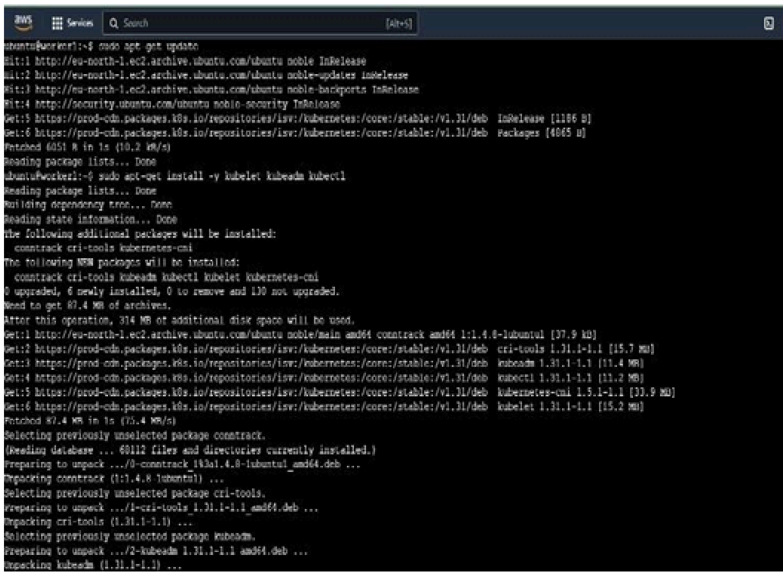
$ sudo apt-get install -y kubelet kubeadm kubectl

$ sudo apt-mark hold kubelet kubeadm kubectl

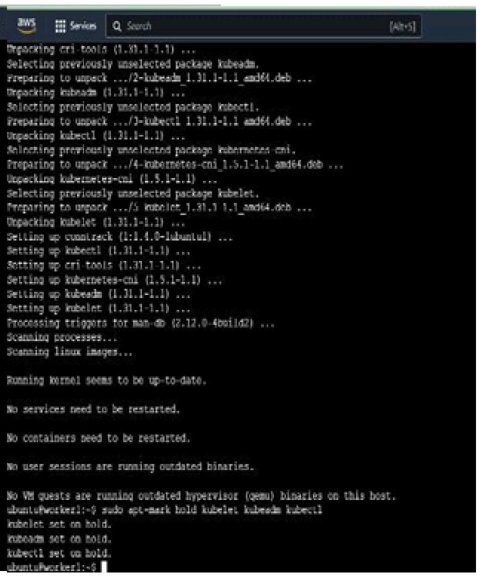


```
ubuntu@master node:~$ sudo apt get update
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  InRelease [1186 B]
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  Packages [4065 B]
Hit:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Fetched 6051 B in 1s (8671 B/s)
Reading package lists... Done
ubuntu@master node:~$ sudo apt get install  y kubelet kubeadm kubectl
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 130 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://eu north-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]
Get:3 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  kubeadm 1.31.1-1.1 [11.4 MB]
Get:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  kubectl 1.31.1-1.1 [11.2 MB]
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  kubernetes-cni 1.5.1-1.1 [33.9 MB]
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  kubelet 1.31.1-1.1 [15.2 MB]
Fetched 87.4 MB in 1s (69.5 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 68112 files and directories currently installed.)
Preparing to unpack .../0 conntrack_1%3a1.4.8 1ubuntu1_amd64.deb ...
Unpacking conntrack (1:1.4.8-1ubuntu1) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.31.1-1.1_amd64.deb ...
Unpacking cri-tools (1.31.1-1.1) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../2-kubeadm_1.31.1-1.1_amd64.deb ...
Unpacking kubeadm (1.31.1-1.1) ...
```

```
Unpacking cri-tools (1.31.1-1.1) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../2-kubeadm_1.31.1-1.1_amd64.deb ...
Unpacking kubeadm (1.31.1-1.1) ...
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.
ubuntu@master-node:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@master-node:~$
```



```
ubuntu@worker1:~$ sudo apt get update
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  InRelease [1186 B]
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  Packages [4065 B]
Fetched 6051 B in 1s (10.2 kB/s)
Reading package lists... Done
ubuntu@worker1:~$ sudo apt-get install -y kubelet kubeadm kubectl
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 130 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://eu-north-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]
Get:3 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  kubeadm 1.31.1-1.1 [11.4 MB]
Get:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  kubectl 1.31.1-1.1 [11.2 MB]
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  kubernetes-cni 1.5.1-1.1 [33.9 MB]
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  kubelet 1.31.1-1.1 [15.2 MB]
Fetched 87.4 MB in 1s (75.4 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 68112 files and directories currently installed.)
Preparing to unpack .../0-conntrack_1%3a1.4.8-1ubuntu1_amd64.deb ...
Unpacking conntrack (1:1.4.8-1ubuntu1) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.31.1-1.1_amd64.deb ...
Unpacking cri-tools (1.31.1-1.1) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../2-kubeadm_1.31.1-1.1_amd64.deb ...
Unpacking kubeadm (1.31.1-1.1) ...
```



```
Unpacking cri-tools (1.31.1-1.1) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../2-kubeadm_1.31.1-1.1_amd64.deb ...
Unpacking kubeadm (1.31.1-1.1) ...
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.
ubuntu@worker1:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@worker1:~$
```

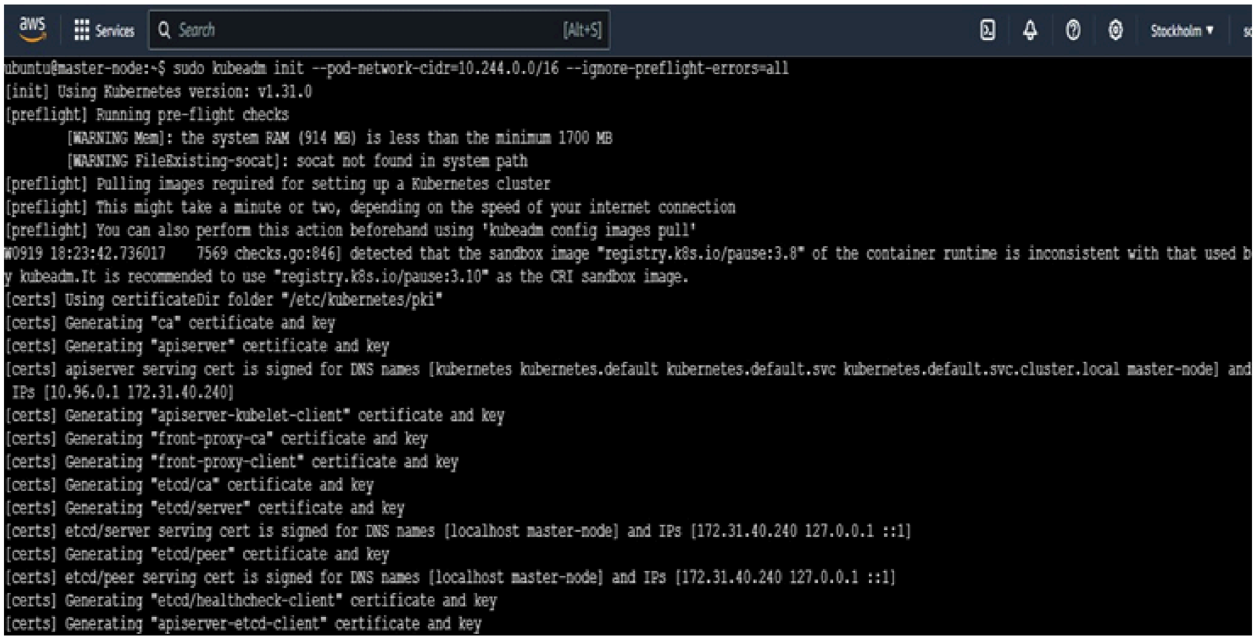
Kubernetes Deployment (master only)

Begin Kubernetes Deployment

```
$ sudo swapoff –a
```

Initialize Kubernetes on Master Node

```
$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16--ignore-  preflight- errors=all
```



```
ubuntu@master-node:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
        [WARNING Mem]: the system RAM (914 MB) is less than the minimum 1700 MB
        [WARNING FileExisting-socat]: socat not found in system path
[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'
W0919 18:23:42.736017    7569 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used b
y kubeadm.It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox 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 [kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local master-node] and
 IPs [10.96.0.1 172.31.40.240]
[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 [localhost master-node] and IPs [172.31.40.240 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [localhost master-node] and IPs [172.31.40.240 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
```

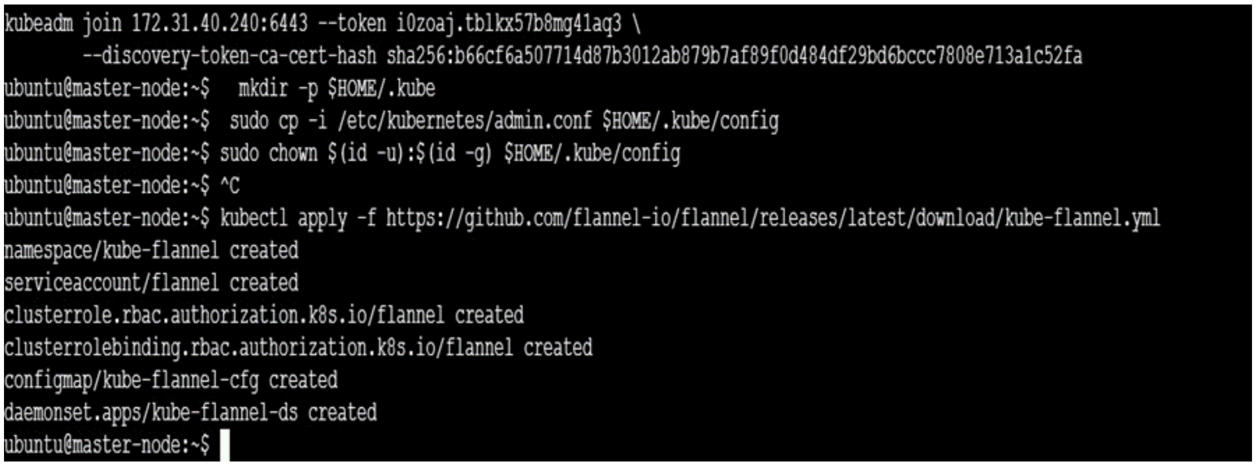
Deploy Pod Network to Cluster

```
$ mkdir -p $HOME/.kube
```

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

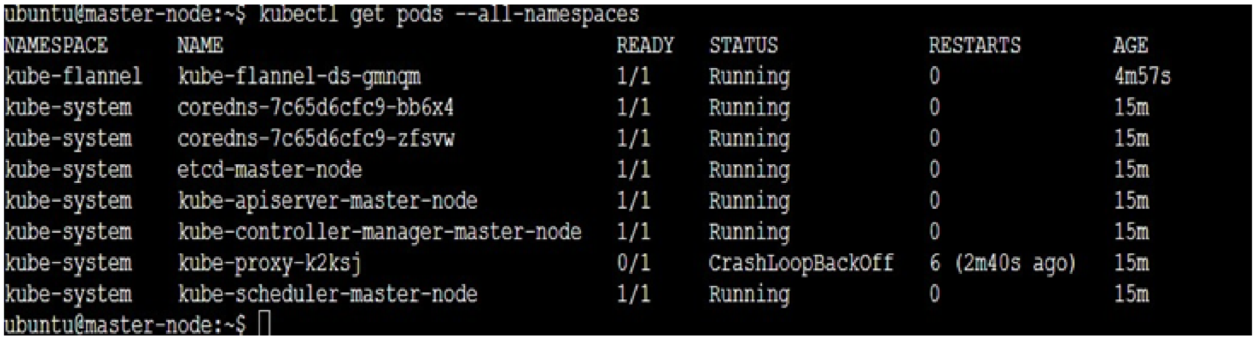
$ sudo chown $(id -u):$(id -g) $HOME/.kube/config

$ kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube
-flannel.yml



```
kubeadm join 172.31.40.240:6443 --token i0zoaj.tblkx57b8mg41aq3 \
        --discovery-token-ca-cert-hash sha256:b66cf6a507714d87b3012ab879b7af89f0d484df29bd6bccc7808e713a1c52fa
ubuntu@master-node:~$   mkdir -p $HOME/.kube
ubuntu@master-node:~$  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@master-node:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@master-node:~$ ^C
ubuntu@master-node:~$ kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml
namespace/kube-flannel created
serviceaccount/flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
ubuntu@master-node:~$
```

$ kubectl get pods --all-namespaces

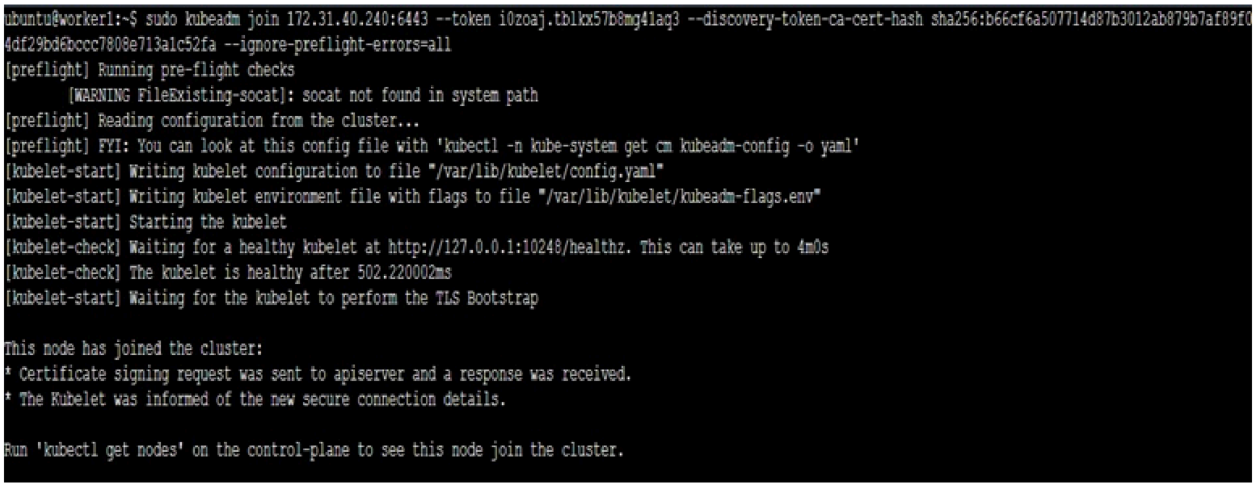


```
ubuntu@master-node:~$ kubectl get pods --all-namespaces
NAMESPACE      NAME                                  READY   STATUS             RESTARTS        AGE
kube-flannel   kube-flannel-ds-gmnqm                 1/1     Running            0               4m57s
kube-system    coredns-7c65d6cfc9-bb6x4              1/1     Running            0               15m
kube-system    coredns-7c65d6cfc9-zfsvw              1/1     Running            0               15m
kube-system    etcd-master-node                      1/1     Running            0               15m
kube-system    kube-apiserver-master-node            1/1     Running            0               15m
kube-system    kube-controller-manager-master-node   1/1     Running            0               15m
kube-system    kube-proxy-k2ksj                      0/1     CrashLoopBackOff   6 (2m40s ago)   15m
kube-system    kube-scheduler-master-node            1/1     Running            0               15m
ubuntu@master-node:~$
```

Join Worker Node to Cluster (on worker node)

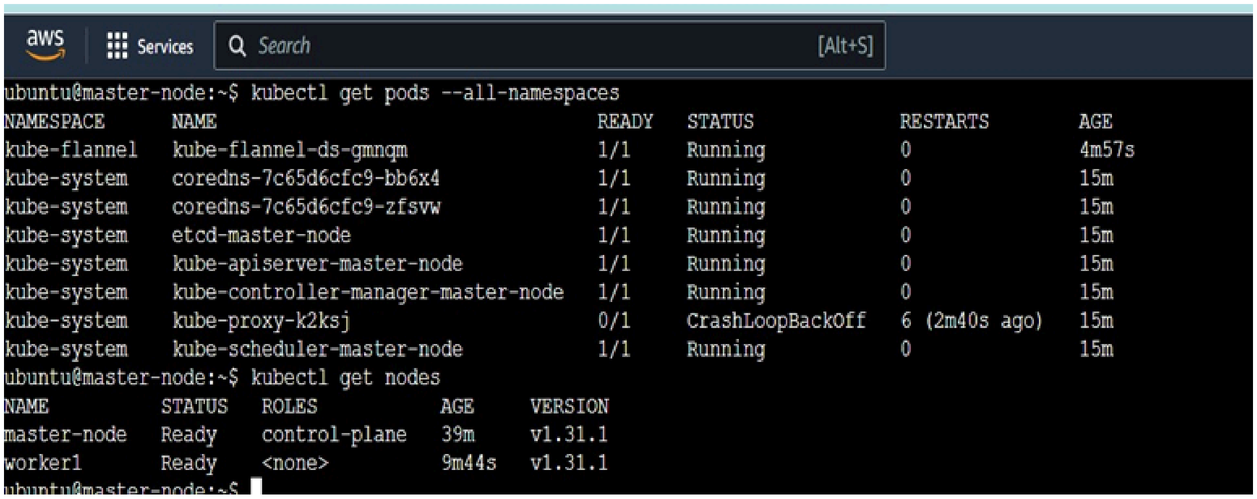
sudo kubeadm join 172.31.40.240:6443 --token i0zoaj.tblkx57b8mg41aq3 --discovery-token-ca-cert-hash

sha256:b66cf6a507714d87b3012ab879b7af89f0d484df29bd6bccc7808e713a1c52fa –ignore-preflight-errors=all



```
ubuntu@worker1:~$ sudo kubeadm join 172.31.40.240:6443 --token i0zoaj.tblkx57b8mg41aq3 --discovery-token-ca-cert-hash sha256:b66cf6a507714d87b3012ab879b7af89f0
4df29bd6bccc7808e713a1c52fa --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 502.220002ms
[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.
```

$ kubectl get nodes (on master node)



```
ubuntu@master-node:~$ kubectl get pods --all-namespaces
NAMESPACE      NAME                                 READY   STATUS             RESTARTS        AGE
kube-flannel   kube-flannel-ds-gmnqm                1/1     Running            0               4m57s
kube-system    coredns-7c65d6cfc9-bb6x4             1/1     Running            0               15m
kube-system    coredns-7c65d6cfc9-zfsvw             1/1     Running            0               15m
kube-system    etcd-master-node                     1/1     Running            0               15m
kube-system    kube-apiserver-master-node           1/1     Running            0               15m
kube-system    kube-controller-manager-master-node  1/1     Running            0               15m
kube-system    kube-proxy-k2ksj                     0/1     CrashLoopBackOff   6 (2m40s ago)   15m
kube-system    kube-scheduler-master-node           1/1     Running            0               15m
ubuntu@master-node:~$ kubectl get nodes
NAME          STATUS   ROLES           AGE    VERSION
master-node   Ready    control-plane   39m    v1.31.1
worker1       Ready    <none>          9m44s  v1.31.1
ubuntu@master-node:~$
```

Conclusion

Successfully understood the Kubernetes cluster architecture and deployed a Kubernetes cluster on Linux machines/cloud platforms, demonstrating seamless setup and orchestration.