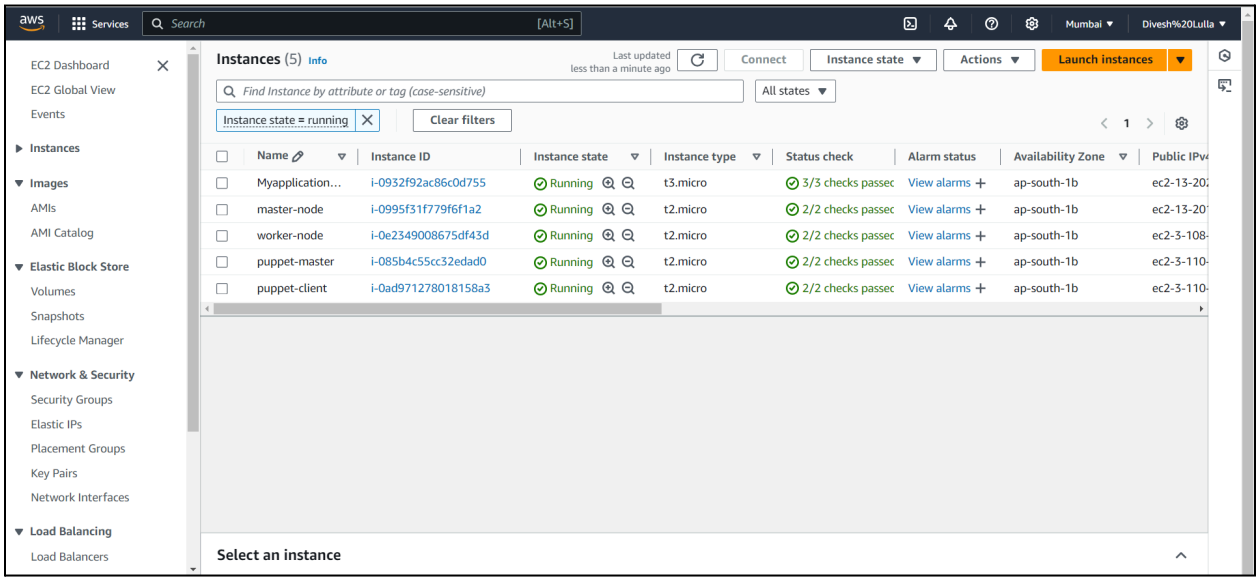
Experiment 3

Aim:

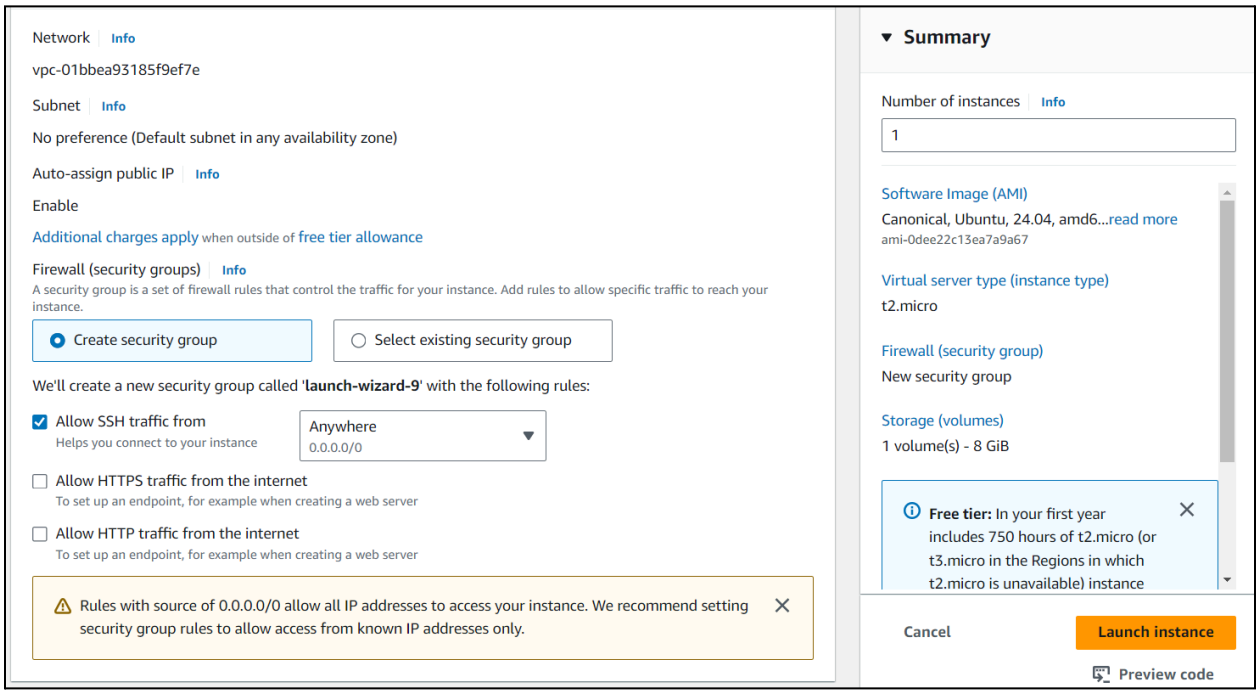
To understand the Kubernetes Cluster Architecture, install and Spin Up a KubernetesCluster on Linux Machines/Cloud Platforms.

Implementation:

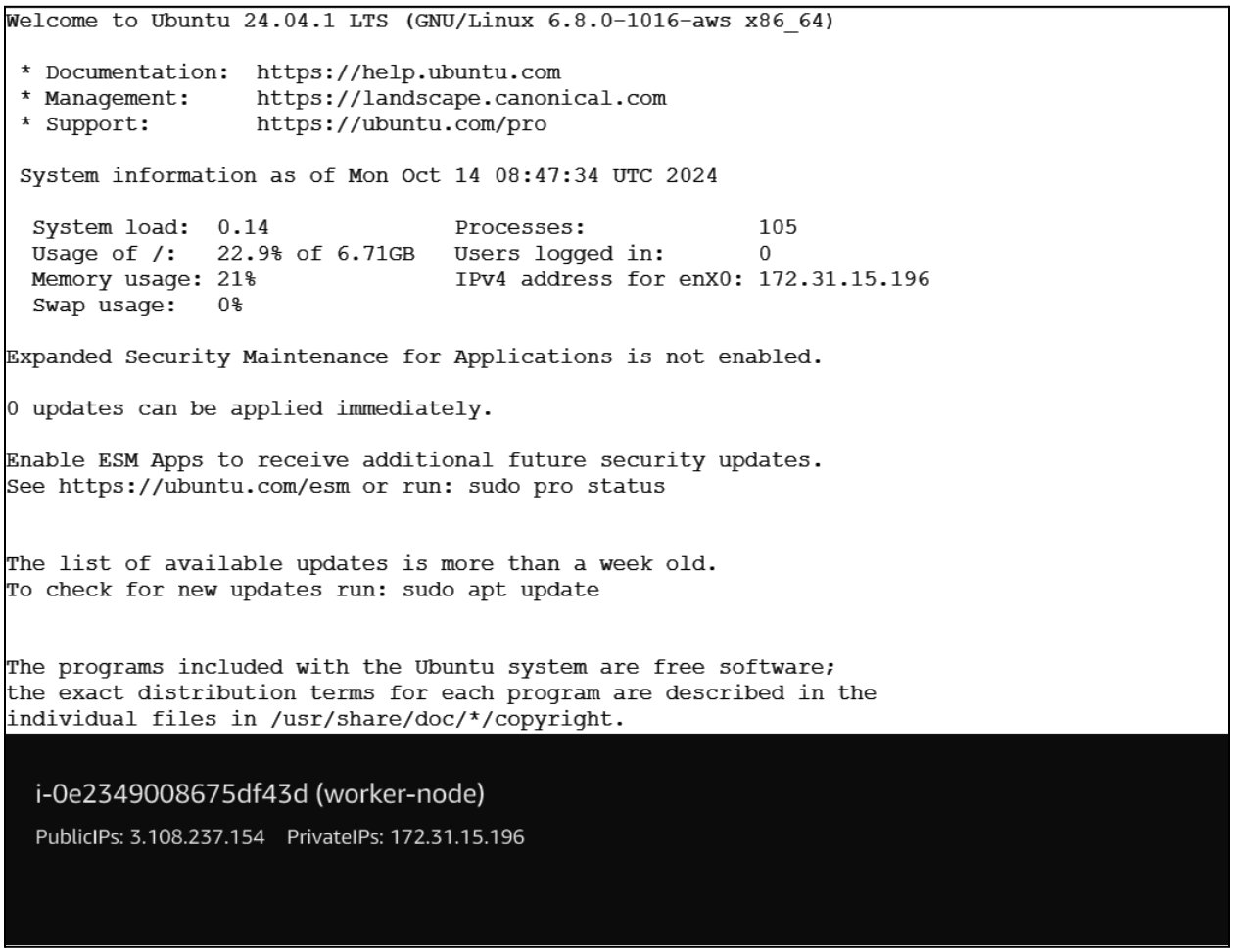
1) Create EC2 Ubuntu Instances on AWS. (Master and Worker)



2) Edit the Security Group Inbound Rules to allow SSH

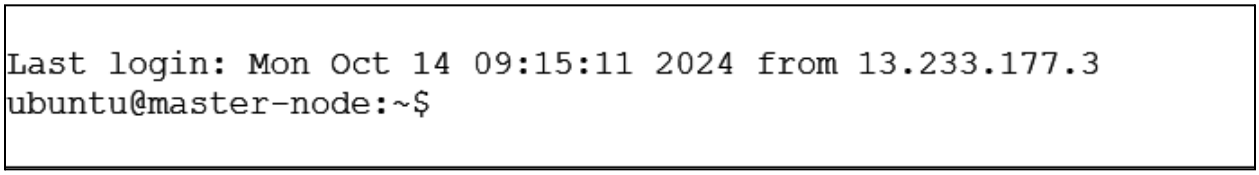


3) AWS CLI for master and worker instances.



4) Assign Unique Hostname for Each Server Node

$ sudo hostnamectl set-hostname master-node
$ sudo hostnamectl set-hostname worker-1



Set up Docker (both master and worker)

5) Install Docker

```
Last login: Mon Oct 14 09:15:17 2024 from 13.233.177.4
ubuntu@worker-1:~$ 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 [542 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [133 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [9048 B]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [386 kB]
Get:17 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [160 kB]
Get:18 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:19 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [15.0 kB]
```

6) $ sudo apt-get install docker.io

```
Reading package lists... Done
ubuntu@worker-1:~$ 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 12 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://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu3.1 [8599 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.12-0ubuntu4.1 [38.6 MB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dns-root-data all 2023112702~willsync1 [4450 B]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dnsmasq-base amd64 2.90-2build2 [375 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 24.0.7-0ubuntu4.1 [29.1 MB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]
Fetched 76.8 MB in 1s (65.0 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 67836 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
```

i-0e2349008675df43d (worker-node)

PublicIPs: 3.108.237.154 PrivateIPs: 172.31.15.196

7) $ docker —version

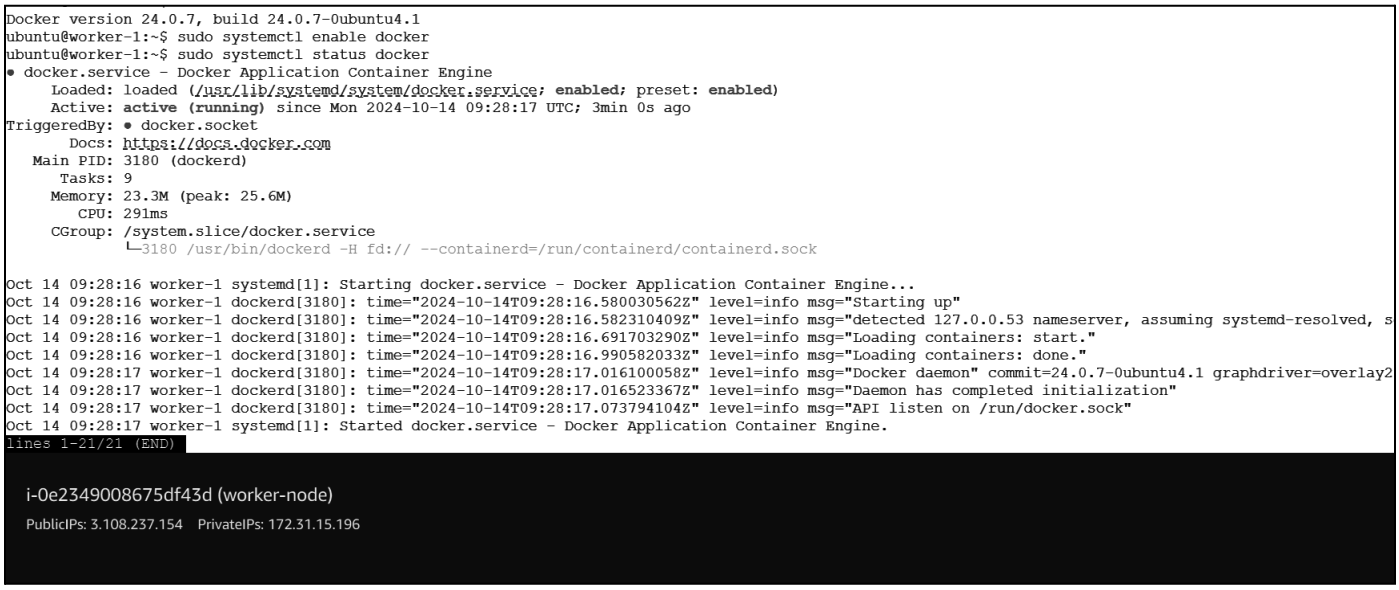
```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@master-node:~$ docker --version
Docker version 24.0.7, build 24.0.7-0ubuntu4.1
ubuntu@master-node:~$
```

i-0995f31f779f6f1a2 (master-node)

PublicIPs: 13.201.134.120 PrivateIPs: 172.31.0.46

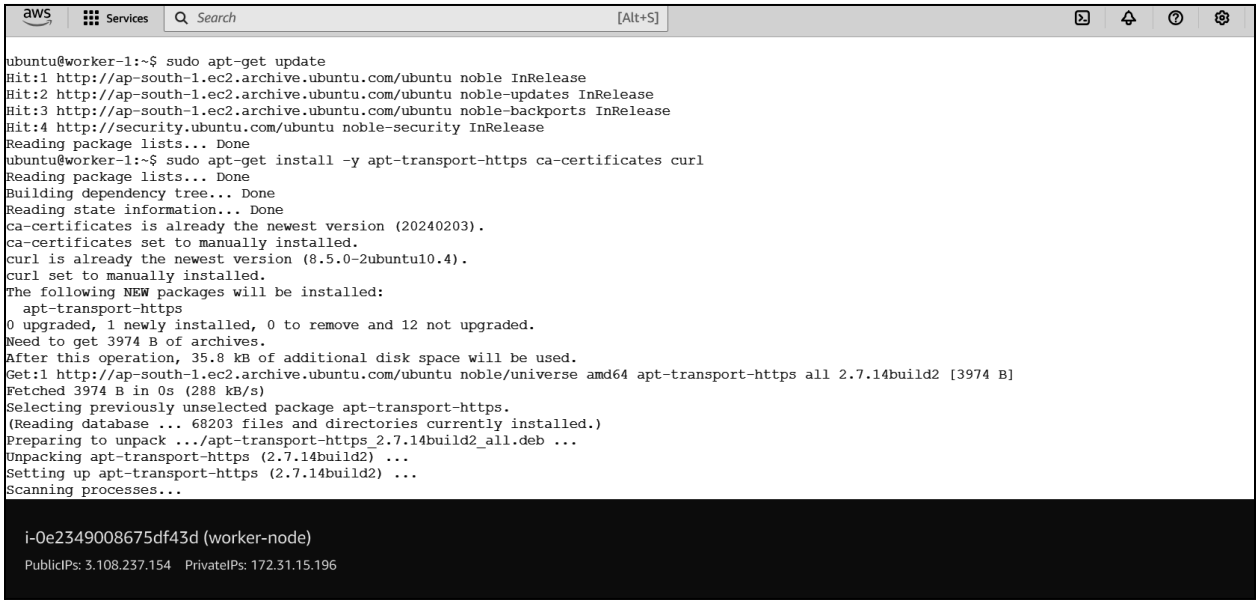
8) Start and Enable Docker

$ sudo systemctl enable docker
$ sudo systemctl status docker
$ sudo systemctl start docker



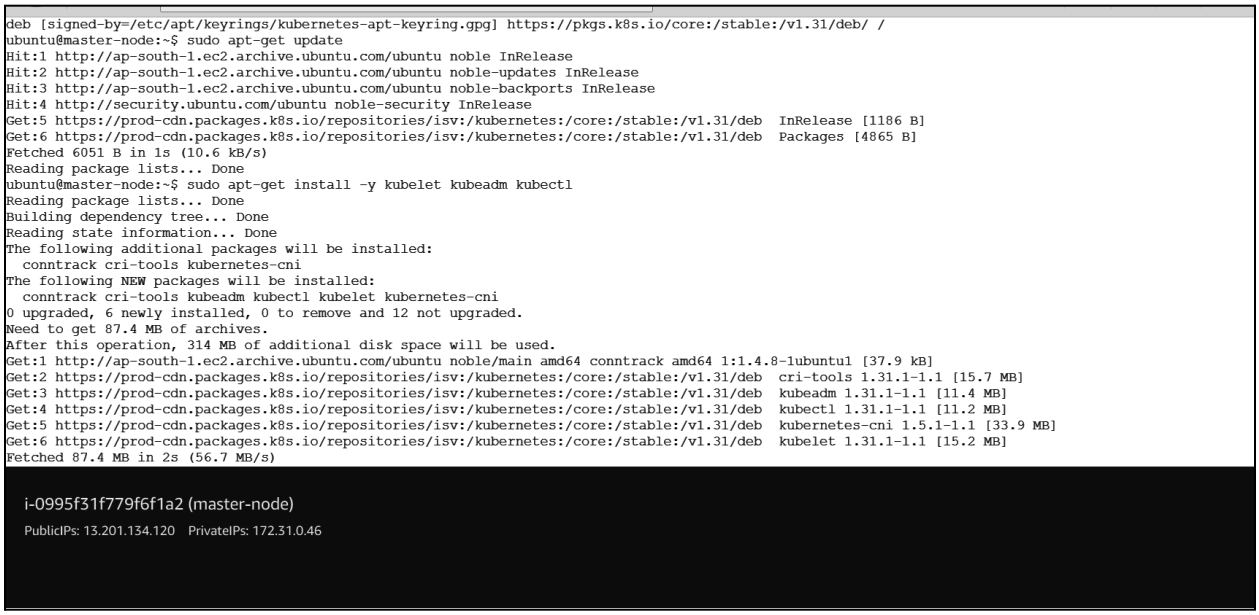
9) Install Kubernetes(both master and worker node)

$ sudo apt-get update
$ sudo apt-get install -y apt-transport-https ca-certificates cur



$ 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



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

```
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
ubuntu@worker-1:~$ 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
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 [4865 B]
Fetched 6051 B in 1s (7935 B/s)
Reading package lists... Done
ubuntu@worker-1:~$ 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 12 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]
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 2s (51.8 MB/s)
```

i-0e2349008675df43d (worker-node)

PublicIPs: 3.108.237.154 PrivateIPs: 172.31.15.196

Kubernetes Deployment (master only)

10) Begin Kubernetes Deployment $ sudo swapoff –a

11) Initialize Kubernetes on Master Node

$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16--ignorepreflight-errors=al

```
ubuntu@master-node:~$ kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml
namespace/kube-flannel created
```

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

kubeadm join 172.31.0.46:6443 --token m9nz62.2n5mbyt10eyamax1 \
        --discovery-token-ca-cert-hash sha256:08459ead16a1c35030f987b04013dce59dc38c9716489dbeac7c16416076da15
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) $HOME/.kube/config
ubuntu@master-node:~$ 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:~$
```

i-0995f31f779f6f1a2 (master-node)

PublicIPs: 13.201.134.120 PrivateIPs: 172.31.0.46

12) $ kubectl get pods --all-namespaces

```
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
NAMESPACE      NAME                                READY   STATUS             RESTARTS      AGE
kube-flannel   kube-flannel-ds-f77qk               1/1     Running            0             43s
kube-system    coredns-7c65d6cfc9-qxxp2            1/1     Running            0             7m36s
kube-system    coredns-7c65d6cfc9-wxhn2            1/1     Running            0             7m36s
kube-system    etcd-master-node                    1/1     Running            0             7m41s
kube-system    kube-apiserver-master-node          1/1     Running            0             7m41s
kube-system    kube-controller-manager-master-node 1/1     Running            0             7m41s
kube-system    kube-proxy-4ls8p                    0/1     CrashLoopBackOff   4 (52s ago)   7m37s
kube-system    kube-scheduler-master-node          1/1     Running            0             7m41s
ubuntu@master-node:~$
```

i-0995f31f779f6f1a2 (master-node)

PublicIPs: 13.201.134.120 PrivateIPs: 172.31.0.46

Join Worker Node to Cluster (on worker node)

13) sudo kubeadm join 172.31.40.240:6443 --token i0zoaj.tblkx57b8mg41aq3 --discovery-token-ca-cert-hash sha256:b66cf6a507714d87b3012ab879b7af89f0d484df29bd6bccc7808e713a1c52fa –ignore-preflight-errors=al

```
 user sessions are running outdated binaries.

 VM guests are running outdated hypervisor (qemu) binaries on this host.
untu@worker-1:~$ sudo kubeadm join 172.31.0.46:6443 --token m9nz62.2n5mbyt10eyamax1 \
  --discovery-token-ca-cert-hash sha256:08459ead16a1c35030f987b04013dce59dc38c9716489dbeac7c16416076da15 \
  --ignore-preflight-errors=all
reflight] Running pre-flight checks
      [WARNING FileExisting-socat]: socat not found in system path
reflight] Reading configuration from the cluster...
reflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
ubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
ubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
ubelet-start] Starting the kubelet
ubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
ubelet-check] The kubelet is healthy after 1.502221294s
ubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

is 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.

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

untu@worker-1:~$
```

i-0e2349008675df43d (worker-node)

PublicIPs: 3.108.237.154 PrivateIPs: 172.31.15.196

14) $ kubectl get nodes (on master node)

```
kubeadm join 172.31.0.46:6443 --token m9nz62.2n5mbyt10eyamax1 \
        --discovery-token-ca-cert-hash sha256:08459ead16a1c35030f987b04013dce59dc38c9716489dbeac7c16416076da15
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) $HOME/.kube/config
ubuntu@master-node:~$ 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
NAMESPACE      NAME                                 READY   STATUS             RESTARTS      AGE
kube-flannel   kube-flannel-ds-f77qk                1/1     Running            0             43s
kube-system    coredns-7c65d6cfc9-qxxp2             1/1     Running            0             7m36s
kube-system    coredns-7c65d6cfc9-wxhn2             1/1     Running            0             7m36s
kube-system    etcd-master-node                     1/1     Running            0             7m41s
kube-system    kube-apiserver-master-node           1/1     Running            0             7m41s
kube-system    kube-controller-manager-master-node  1/1     Running            0             7m41s
kube-system    kube-proxy-4ls8p                     0/1     CrashLoopBackOff   4 (52s ago)   7m37s
kube-system    kube-scheduler-master-node           1/1     Running            0             7m41s
ubuntu@master-node:~$ ^C
ubuntu@master-node:~$ kubectl get nodes
NAME          STATUS   ROLES           AGE   VERSION
master-node   Ready    control-plane   11m   v1.31.1
worker-1      Ready    <none>          40s   v1.31.1
ubuntu@master-node:~$
```

we now have a Kubernetes cluster running across AWS EC2 Instances. This cluster can be used to further deploy applications and their loads being distributed across these machines.

Conclusion:

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