EXPERIMENT 3

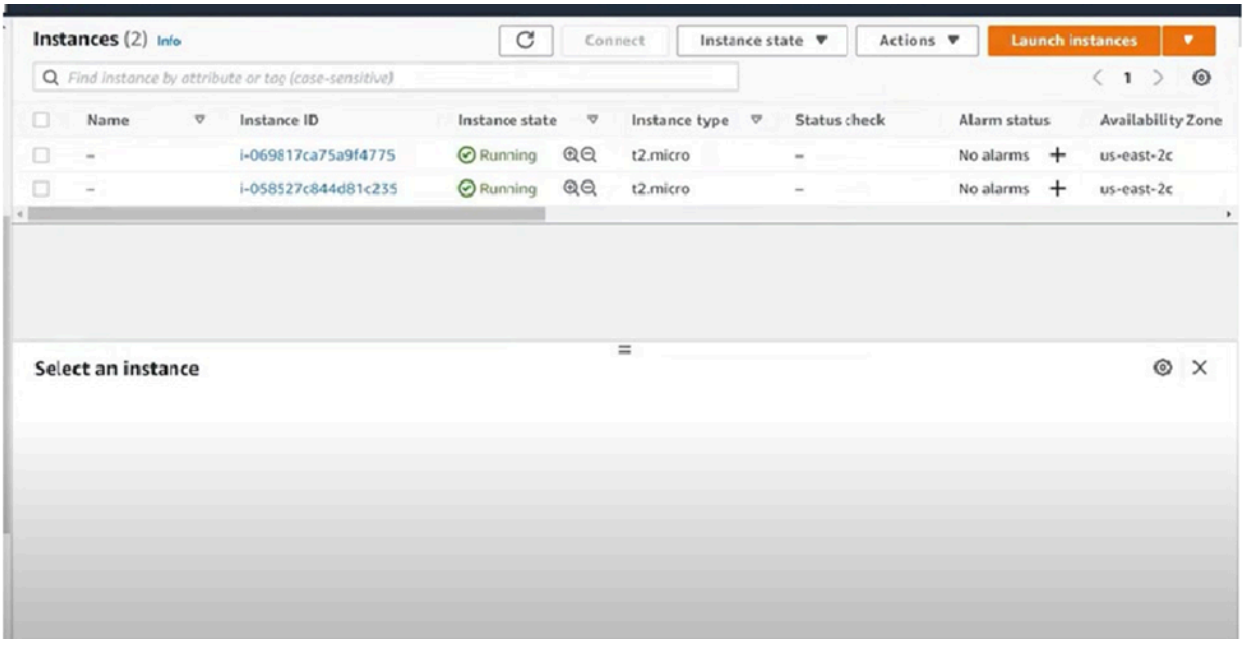
MANORATH ITAL
D10A/19

Aim

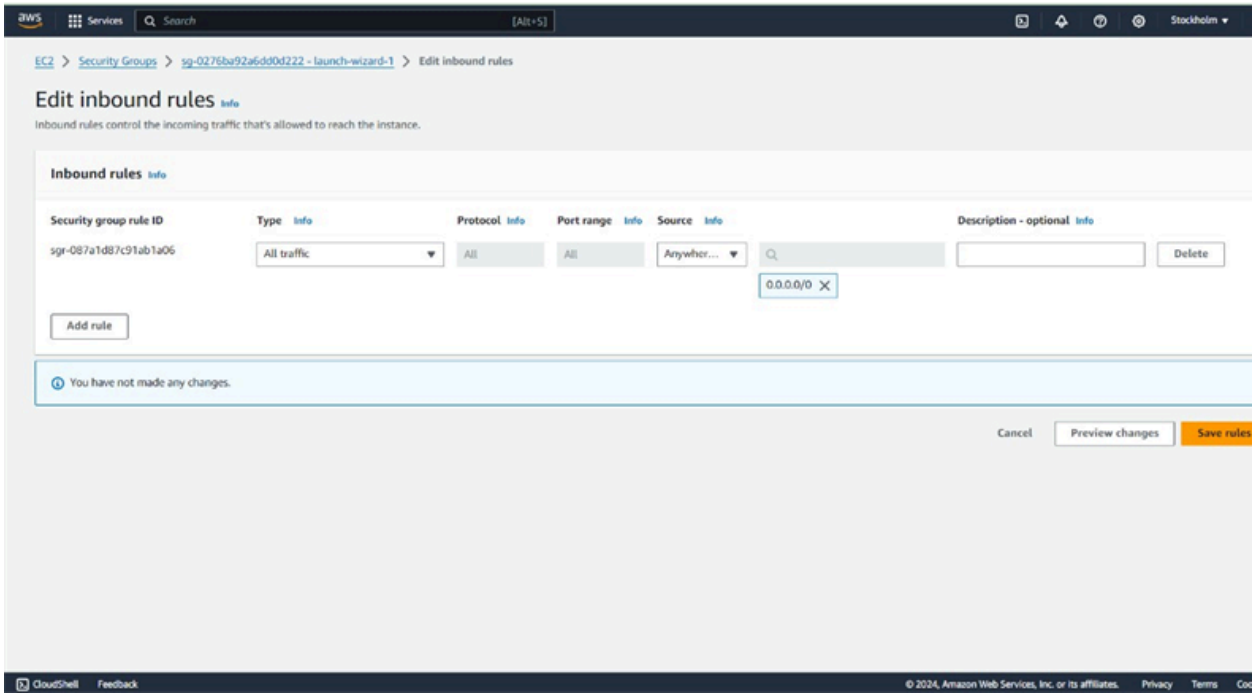
To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud Platforms.

Steps

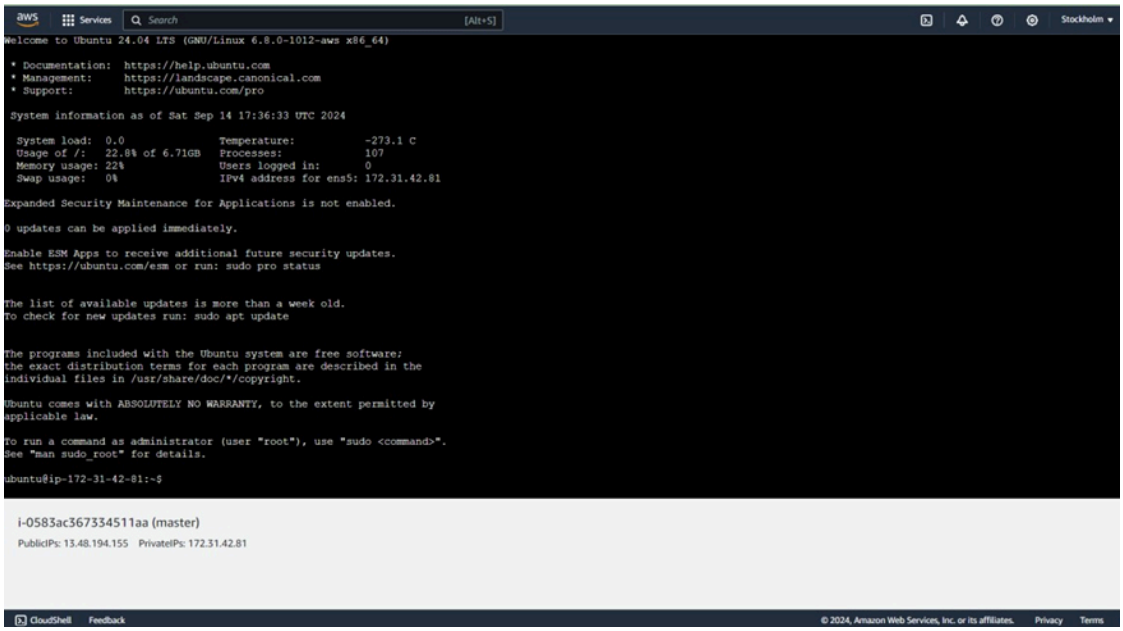
1) Create EC2 Ubuntu Instances on AWS. (Master and Worker)

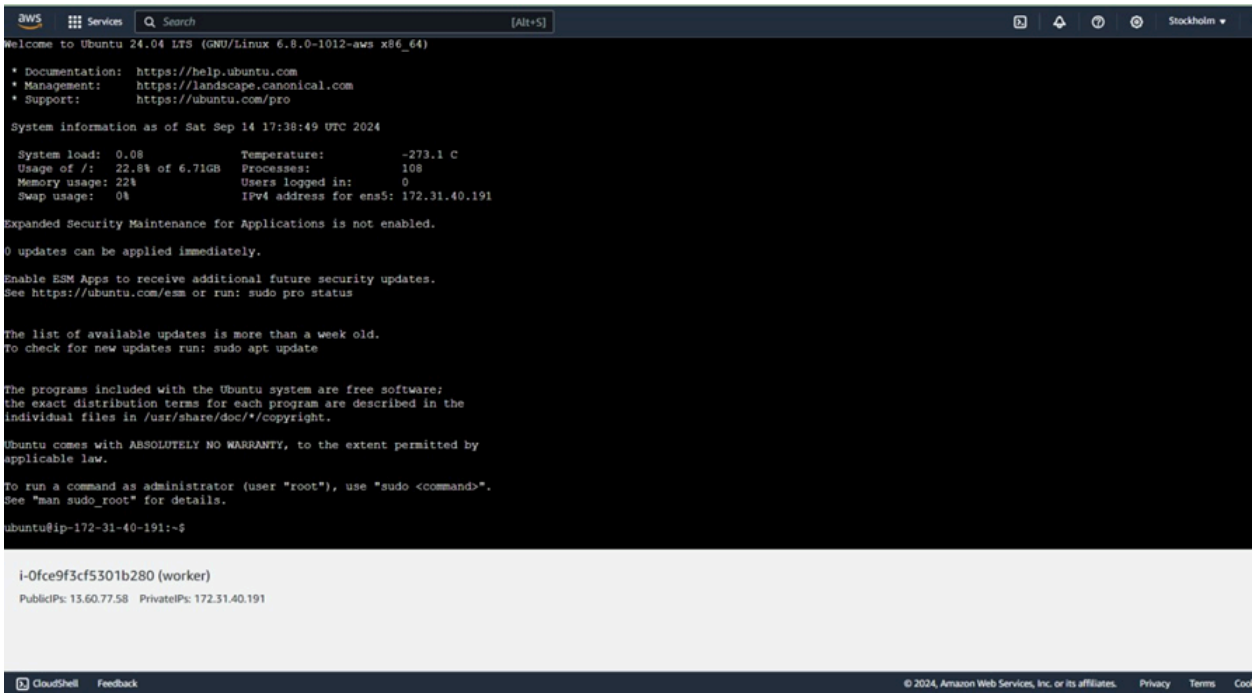


2) Edit the Security Group Inbound Rules to allow SSH



3) AWS CLI for master and worker instances





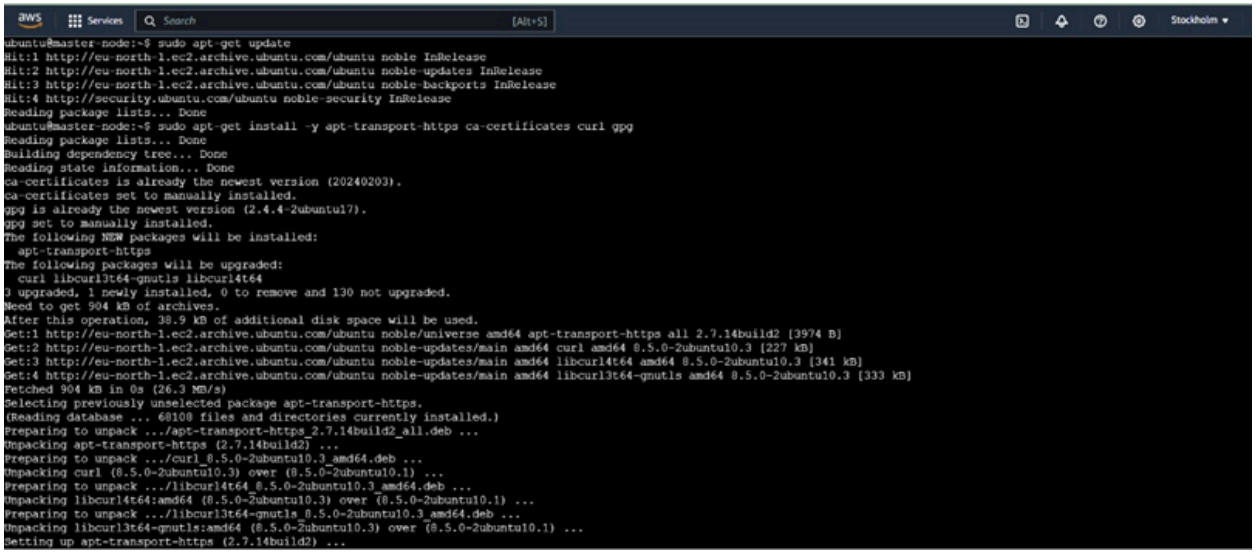
4) Assign Unique Hostname for Each Server Node

$ sudo hostnamectl set-hostname master

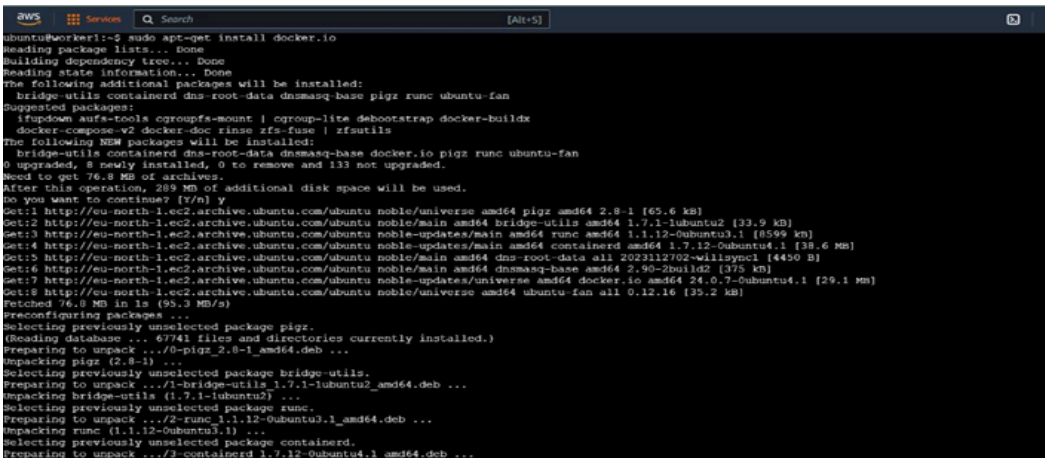
$ sudo hostnamectl set-hostname worker-1

Set up Docker (both master and worker)

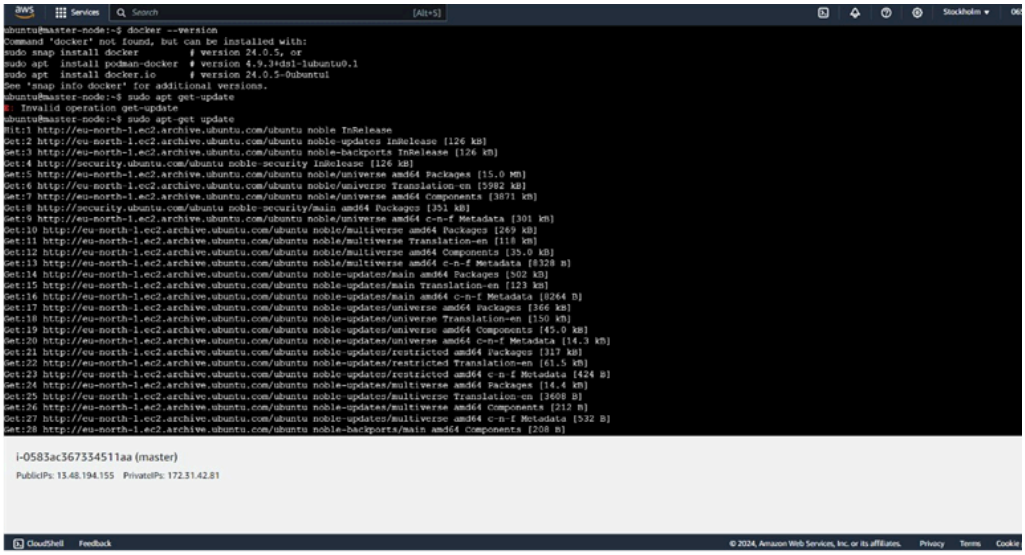
5) Install Docker



$ sudo apt-get update



$ sudo apt-get install docker.io

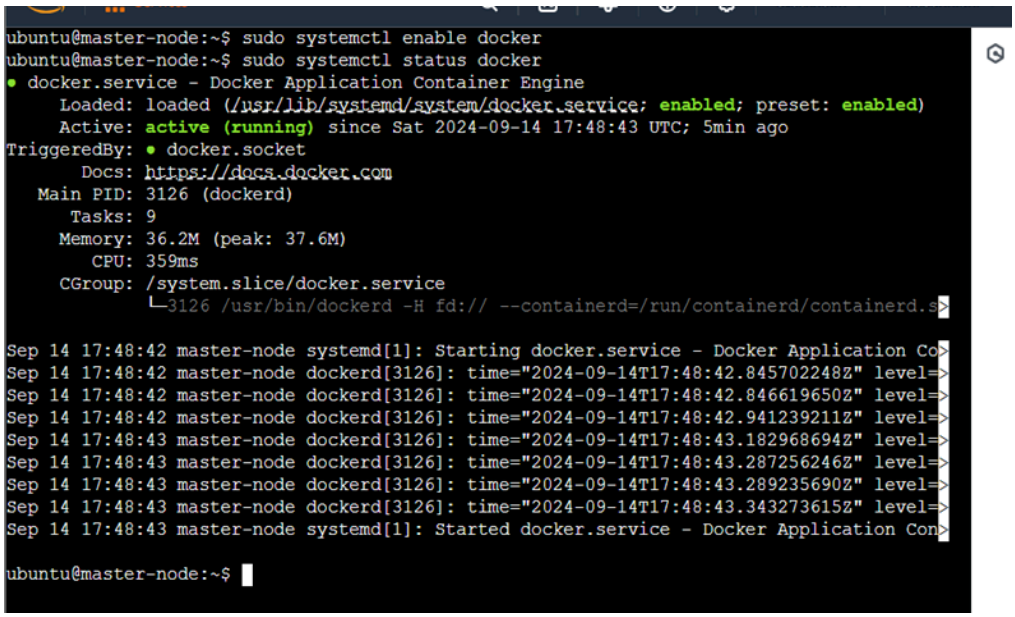


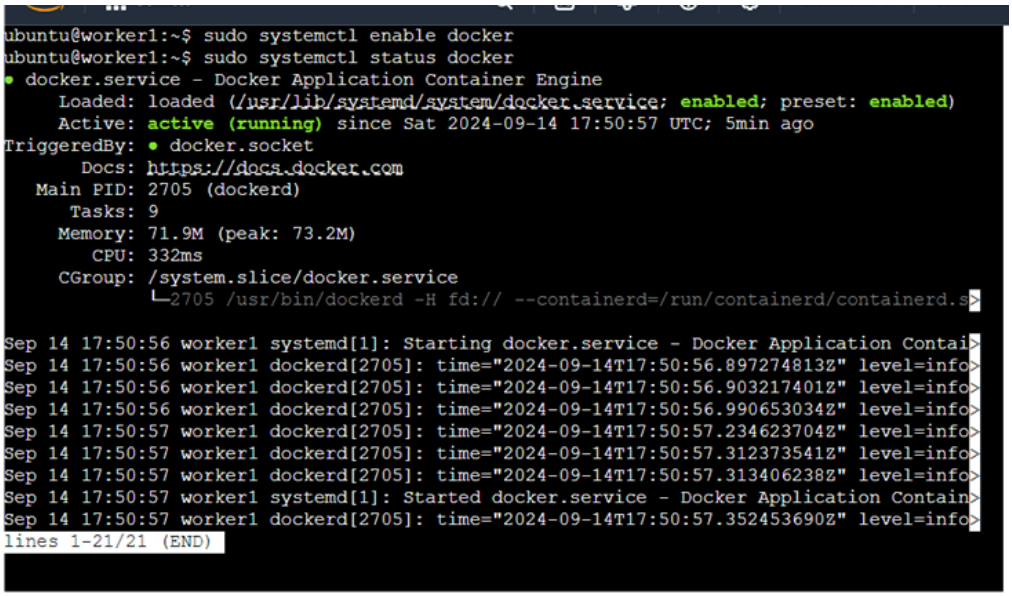
6.Start and Enable Docker

$ sudo systemctl enable docker

$ sudo systemctl status docker

$ sudo systemctl start docker

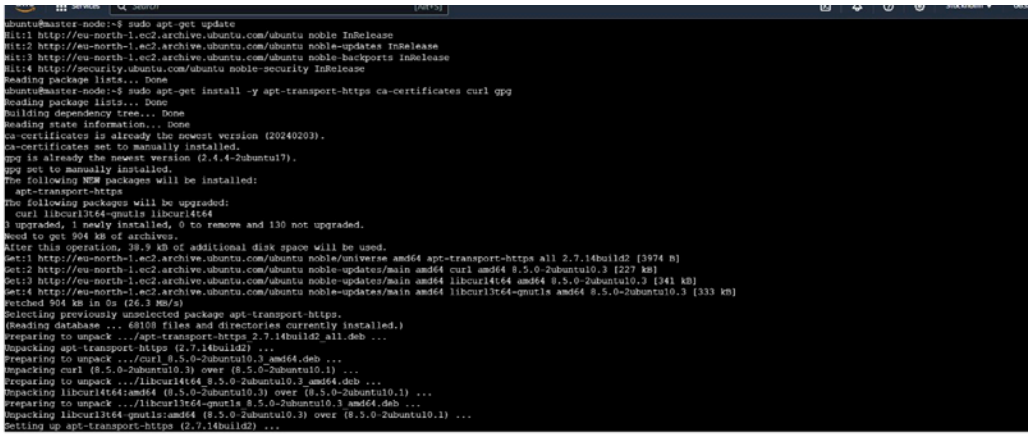




7) Install Kubernetes(both master and worker node)

$ sudo apt-get update

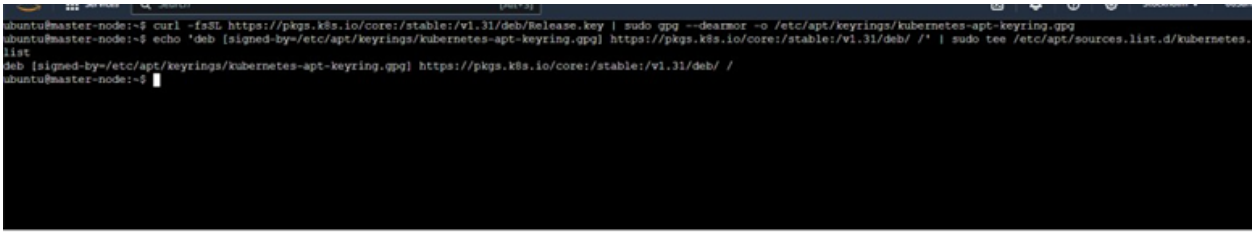
$ sudo apt-get install -y apt-transport-https ca-certificates curl

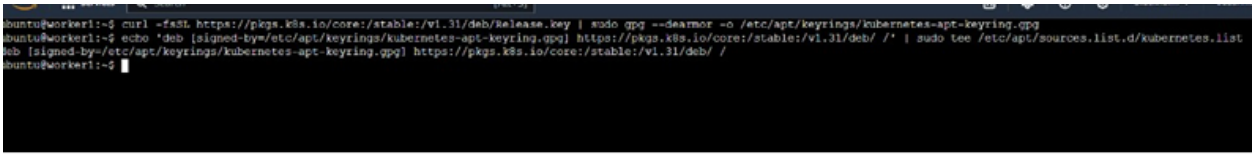


$ sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg

$ 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

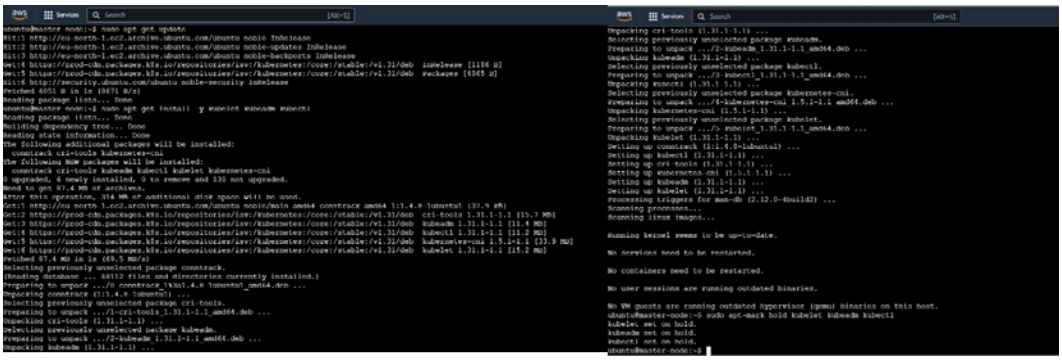


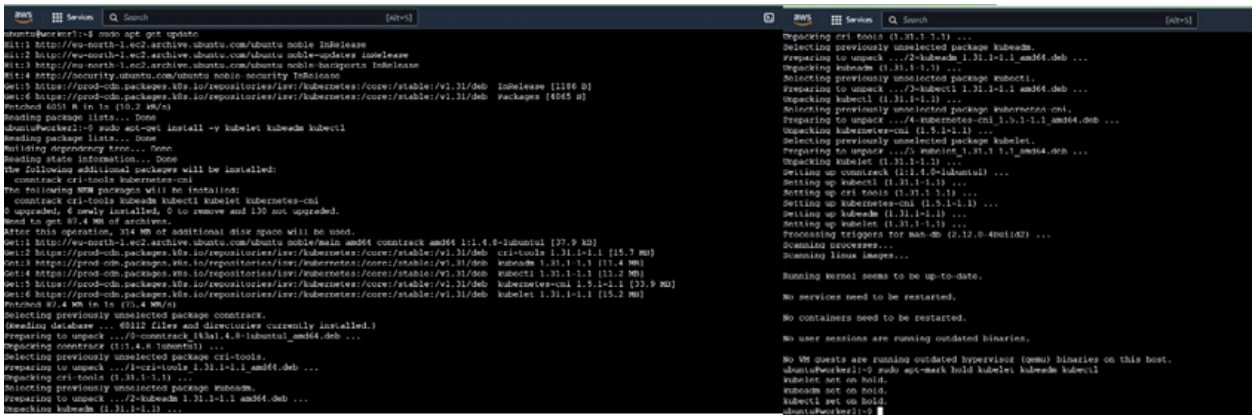


$ sudo apt-get update

$ sudo apt-get install -y kubelet kubeadm kubectl

$ sudo apt-mark hold kubelet kubeadm kubectl





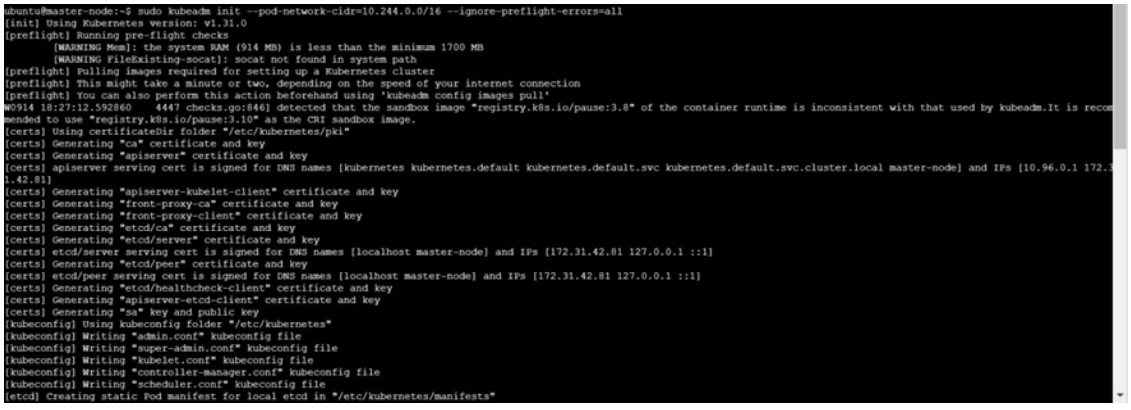
Kubernetes Deployment (master only)

8) Begin Kubernetes Deployment

$ sudo swapoff –a

9) Initialize Kubernetes on Master Node

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



10) 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

```
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
```

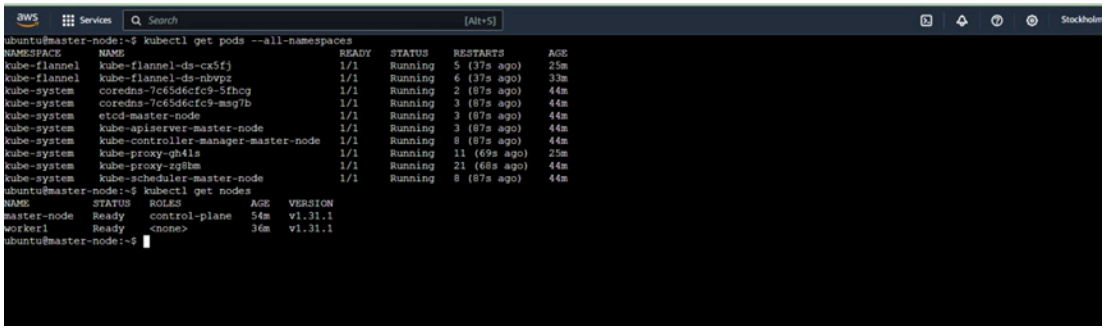
$ kubectl get pods --all-namespaces

```
ubuntu@master-node:~$ kubectl get pods --all-namespaces
NAMESPACE      NAME                                 READY   STATUS    RESTARTS       AGE
kube-flannel   kube-flannel-ds-cx5fj                1/1     Running   5 (37s ago)    25m
kube-flannel   kube-flannel-ds-nbvpz                1/1     Running   6 (37s ago)    33m
kube-system    coredns-7c65d6cfc9-5fhcg             1/1     Running   2 (87s ago)    44m
kube-system    coredns-7c65d6cfc9-msg7b             1/1     Running   3 (87s ago)    44m
kube-system    etcd-master-node                     1/1     Running   3 (87s ago)    44m
kube-system    kube-apiserver-master-node           1/1     Running   3 (87s ago)    44m
kube-system    kube-controller-manager-master-node  1/1     Running   8 (87s ago)    44m
kube-system    kube-proxy-gh4ls                     1/1     Running   11 (69s ago)   25m
kube-system    kube-proxy-zg8bm                     1/1     Running   21 (68s ago)   44m
kube-system    kube-scheduler-master-node           1/1     Running   8 (87s ago)    44m
ubuntu@master-node:~$
```

Join Worker Node to Cluster (on worker node)

11) $ kubeadm join 172.31.42.81:6443 --token 1evaro.2xdo7fco07hizfja --discovery-token-ca-cert-hash sha256:be87f981c53e5ea6471eef94af12957cbdd291711150e58ceaf6a56b0f768d5e --ignore-preflight-errors=all

12) $ kubectl get nodes (on 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.