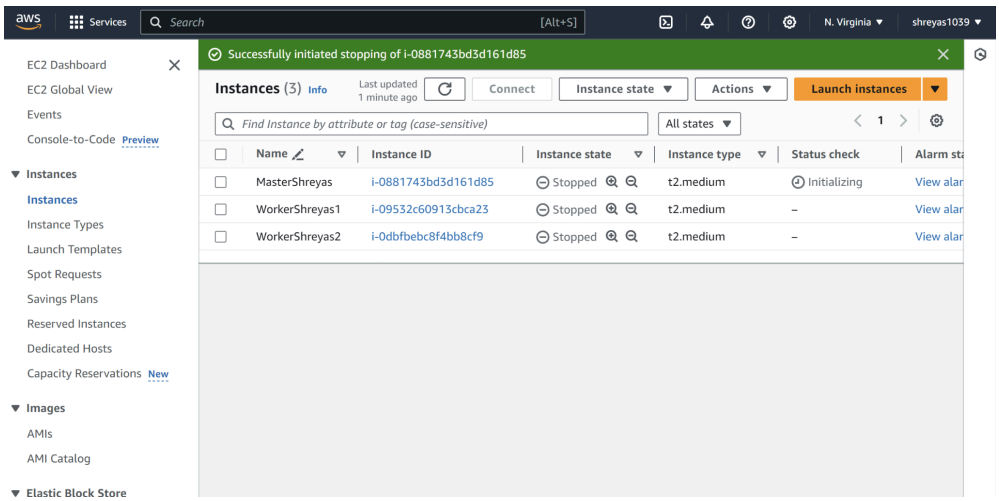
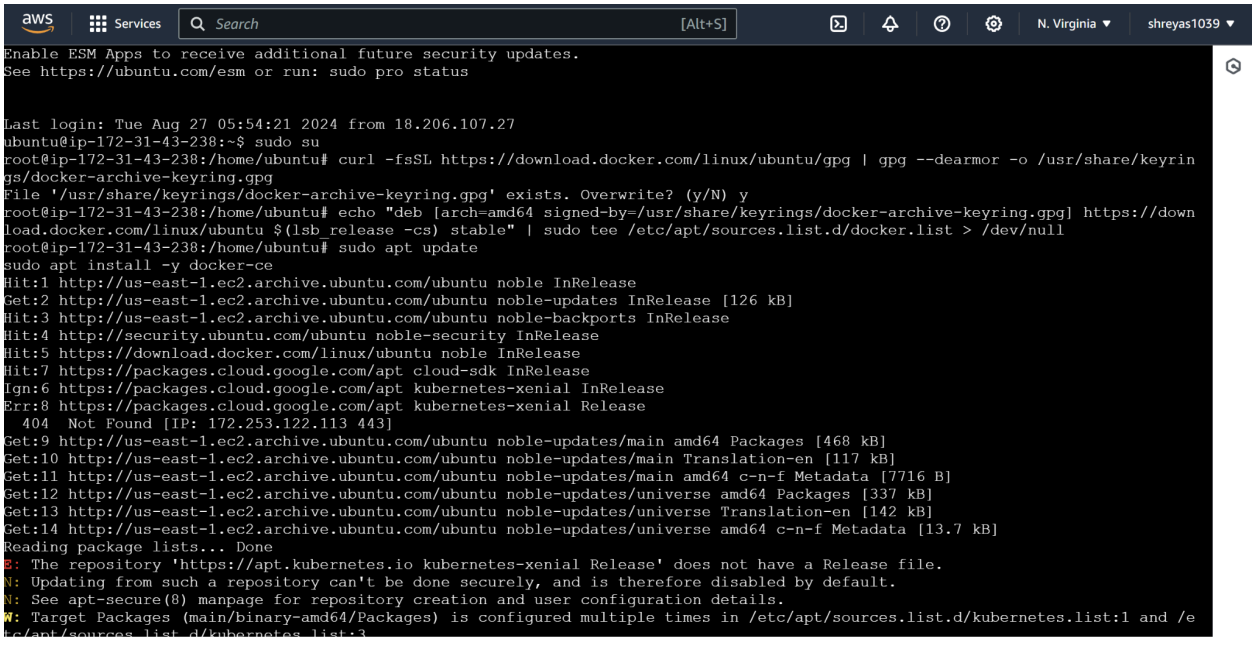
Steps:

1. Create 3 EC2 Ubuntu Instances on AWS.



Installing Docker



```
root@ip-172-31-43-238:/home/ubuntu# systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: active (running) since Tue 2024-08-27 06:27:36 UTC; 2min 15s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 775 (dockerd)
      Tasks: 7
     Memory: 98.4M (peak: 99.7M)
        CPU: 495ms
     CGroup: /system.slice/docker.service
             └─775 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Aug 27 06:27:34 ip-172-31-43-238 dockerd[775]: time="2024-08-27T06:27:34.900277570Z" level=info msg="Starting up"
Aug 27 06:27:34 ip-172-31-43-238 dockerd[775]: time="2024-08-27T06:27:34.915379249Z" level=info msg="detected 127.0.0.53 nameser
Aug 27 06:27:35 ip-172-31-43-238 dockerd[775]: time="2024-08-27T06:27:35.413459120Z" level=info msg="[graphdriver] trying confi
Aug 27 06:27:35 ip-172-31-43-238 dockerd[775]: time="2024-08-27T06:27:35.487609512Z" level=info msg="Loading containers: start."
Aug 27 06:27:36 ip-172-31-43-238 dockerd[775]: time="2024-08-27T06:27:36.488518335Z" level=info msg="Default bridge (docker0) i
Aug 27 06:27:36 ip-172-31-43-238 dockerd[775]: time="2024-08-27T06:27:36.645936049Z" level=info msg="Loading containers: done."
Aug 27 06:27:36 ip-172-31-43-238 dockerd[775]: time="2024-08-27T06:27:36.725358518Z" level=info msg="Docker daemon" commit=f952
Aug 27 06:27:36 ip-172-31-43-238 dockerd[775]: time="2024-08-27T06:27:36.729692762Z" level=info msg="Daemon has completed initi
Aug 27 06:27:36 ip-172-31-43-238 dockerd[775]: time="2024-08-27T06:27:36.880106945Z" level=info msg="API listen on /run/docker.
Aug 27 06:27:36 ip-172-31-43-238 systemd[1]: Started docker.service - Docker Application Container Engine.

root@ip-172-31-43-238:/home/ubuntu# sudo systemctl enable 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
root@ip-172-31-43-238:/home/ubuntu#
```

i-0881743bd3d161d85 (MasterShreyas)

PublicIPs: 18.209.172.155 PrivateIPs: 172.31.43.238

Install Kubernetes on all 3 machines

- curl -fsSL https://download.kubernetes.com/linux/ubuntu/gpg | gpg --dearmor -o /usr/share/keyrings/kubernetes-archive-keyring.gpg

- echo "deb [arch=amd64 signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://download.kubernetes.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/kubernetes.list

```
root@ip-172-31-43-238:/home/ubuntu# curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google
.com/apt/doc/apt-key.gpg
root@ip-172-31-43-238:/home/ubuntu# echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kuberne
tes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main
```

- sudo apt update
- sudo apt install -y kubectl kubelet kubeadm

```
root@ip-172-31-43-238:/home/ubuntu# sudo apt-get update
sudo apt-get install -y kubectl kubelet kubeadm
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu noble InRelease
Hit:7 https://packages.cloud.google.com/apt cloud-sdk InRelease
Ign:6 https://packages.cloud.google.com/apt kubernetes-xenial InRelease
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [463 kB]
Err:9 https://packages.cloud.google.com/apt kubernetes-xenial Release
  404  Not Found [IP: 142.251.163.138 443]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [337 kB]
Reading package lists... Done
E: The repository 'https://apt.kubernetes.io kubernetes-xenial Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package kubelet
E: Unable to locate package kubeadm
```

- Snap install kubectl –classic

```
root@ip-172-31-43-238:/home/ubuntu# snap install kubectl --classic
kubectl 1.30.4 from Canonical✓ installed
```

- Snap install kubelet –classic

```
root@ip-172-31-43-238:/home/ubuntu# snap install kubelet --classic
kubelet 1.30.4 from Canonical✓ installed
```

- Snap install kubeadm –classic

```
root@ip-172-31-43-238:/home/ubuntu# snap install kubeadm --classic
kubeadm 1.30.4 from Canonical✓ installed
```

- sudo swapoff -a
- echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
- sudo sysctl -p

```
root@ip-172-31-43-238:/home/ubuntu# sudo swapoff -a
root@ip-172-31-43-238:/home/ubuntu# echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
net.bridge.bridge-nf-call-iptables=1
root@ip-172-31-43-238:/home/ubuntu# sudo sysctl -p
net.bridge.bridge-nf-call-iptables = 1
```

-

```
root@ip-172-31-43-238:/home/ubuntu# kubelet --version
Kubernetes v1.30.4
root@ip-172-31-43-238:/home/ubuntu# sudo nano /etc/systemd/system/kubelet.service
root@ip-172-31-43-238:/home/ubuntu# sudo systemctl daemon-reload
root@ip-172-31-43-238:/home/ubuntu# sudo systemctl enable kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /etc/systemd/system/kubelet.service.
root@ip-172-31-43-238:/home/ubuntu# sudo systemctl start kubelet
root@ip-172-31-43-238:/home/ubuntu# sudo systemctl status kubelet
● kubelet.service - kubelet: The Kubernetes Node Agent
     Loaded: loaded (/etc/systemd/system/kubelet.service; enabled; preset: enabled)
     Active: activating (auto-restart) (Result: exit-code) since Sun 2024-08-25 14:03:50 UTC; 9s ago
       Docs: https://kubernetes.io/docs/home/
    Process: 52192 ExecStart=/usr/bin/kubelet (code=exited, status=203/EXEC)
   Main PID: 52192 (code=exited, status=203/EXEC)
        CPU: 1ms
```

- sudo nano /etc/systemd/system/kubelet. service

  Manually add content to file

-

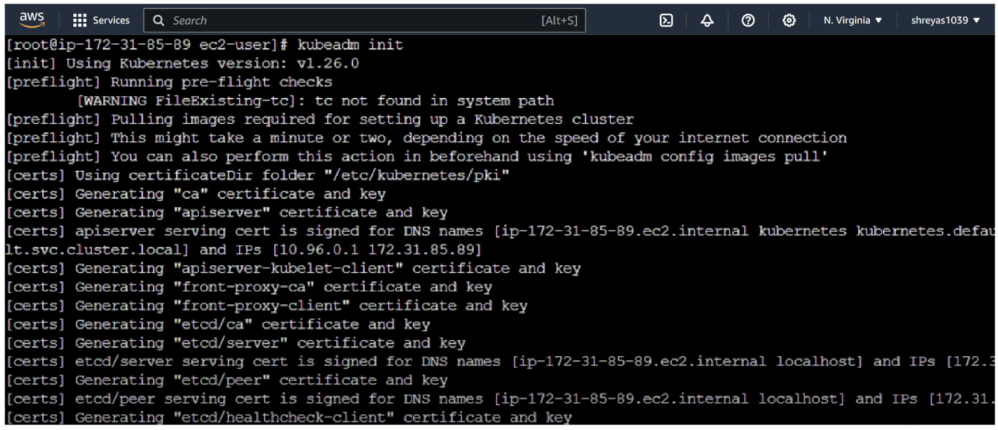
Add the following content to the file:

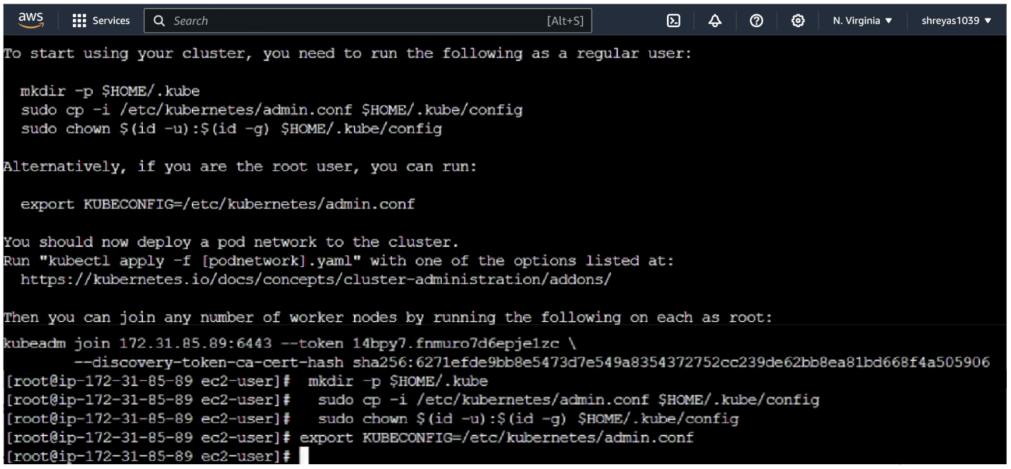
ini

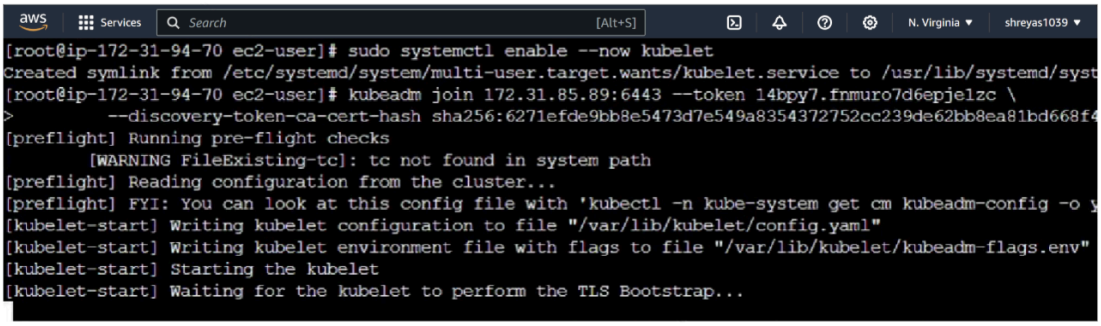
```
[Unit]
Description=kubelet: The Kubernetes Node Agent
Documentation=https://kubernetes.io/docs/home/
Wants=network-online.target
After=network-online.target

[Service]
ExecStart=/usr/bin/kubelet
Restart=always
StartLimitInterval=0
RestartSec=10

[Install]
WantedBy=multi-user.target
```

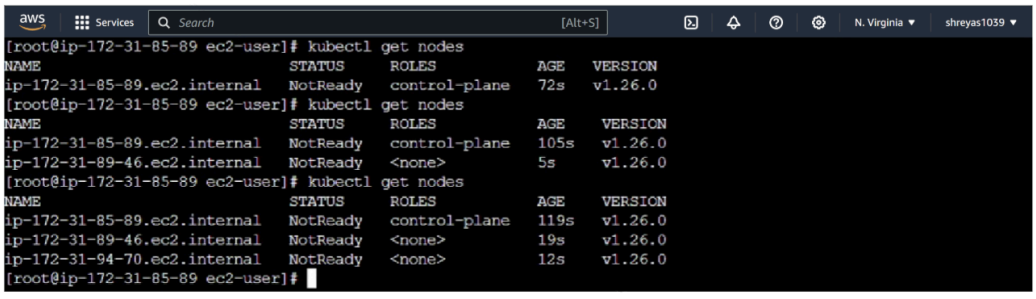
- sudo systemctl daemon-reload
- sudo systemctl enable kubelet
- sudo systemctl start kubelet

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


aws
Services
Search
[Alt+S]
N. Virginia
shreyas1039
[root@ip-172-31-85-89 ec2-user]# kubeadm init
[init] Using Kubernetes version: v1.26.0
[preflight] Running pre-flight checks
[WARNING FileExisting-tc]: tc 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 in beforehand using 'kubeadm config images pull'
[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-172-31-85-89.ec2.internal kubernetes kubernetes.defau
lt.svc.cluster.local] and IPs [10.96.0.1 172.31.85.89]
[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-172-31-85-89.ec2.internal localhost] and IPs [172.3
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-85-89.ec2.internal localhost] and IPs [172.31.
[certs] Generating "etcd/healthcheck-client" certificate and key


aws
Services
Search
[Alt+S]
N. Virginia
shreyas1039
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 172.31.85.89:6443 --token 14bpy7.fnmuro7d6epje1zc \
--discovery-token-ca-cert-hash sha256:6271efde9bb8e5473d7e549a8354372752cc239de62bb8ea81bd668f4a505906
[root@ip-172-31-85-89 ec2-user]# mkdir -p $HOME/.kube
[root@ip-172-31-85-89 ec2-user]# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
[root@ip-172-31-85-89 ec2-user]# sudo chown $(id -u):$(id -g) $HOME/.kube/config
[root@ip-172-31-85-89 ec2-user]# export KUBECONFIG=/etc/kubernetes/admin.conf
[root@ip-172-31-85-89 ec2-user]#


aws
Services
Search
[Alt+S]
N. Virginia
shreyas1039
[root@ip-172-31-94-70 ec2-user]# sudo systemctl enable --now kubelet
Created symlink from /etc/systemd/system/multi-user.target.wants/kubelet.service to /usr/lib/systemd/syst
[root@ip-172-31-94-70 ec2-user]# kubeadm join 172.31.85.89:6443 --token 14bpy7.fnmuro7d6epje1zc \
> --discovery-token-ca-cert-hash sha256:6271efde9bb8e5473d7e549a8354372752cc239de62bb8ea81bd668f4
[preflight] Running pre-flight checks
[WARNING FileExisting-tc]: tc 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 y
[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-start] Waiting for the kubelet to perform the TLS Bootstrap...


aws
Services
Search
[Alt+S]
N. Virginia
shreyas1039
[root@ip-172-31-85-89 ec2-user]# kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-85-89.ec2.internal NotReady control-plane 72s v1.26.0
[root@ip-172-31-85-89 ec2-user]# kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-85-89.ec2.internal NotReady control-plane 105s v1.26.0
ip-172-31-89-46.ec2.internal NotReady <none> 5s v1.26.0
[root@ip-172-31-85-89 ec2-user]# kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-85-89.ec2.internal NotReady control-plane 119s v1.26.0
ip-172-31-89-46.ec2.internal NotReady <none> 19s v1.26.0
ip-172-31-94-70.ec2.internal NotReady <none> 12s v1.26.0
[root@ip-172-31-85-89 ec2-user]#