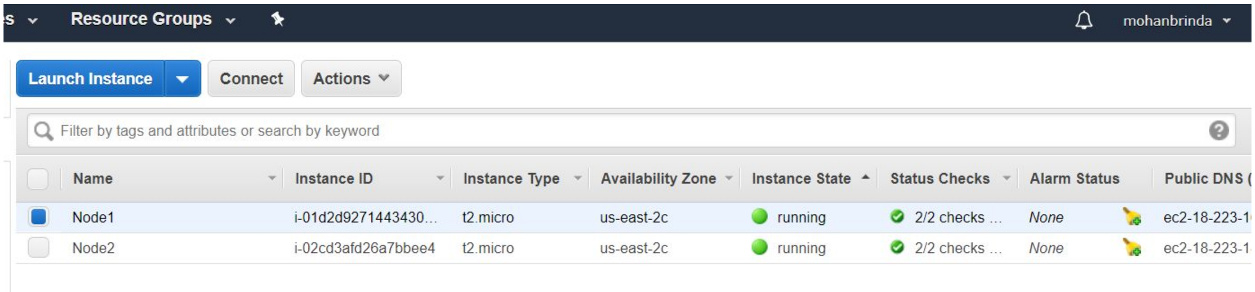
KUBERNETES

Kubernetes is an open source system for automating deployment, scaling and management of containerized application.

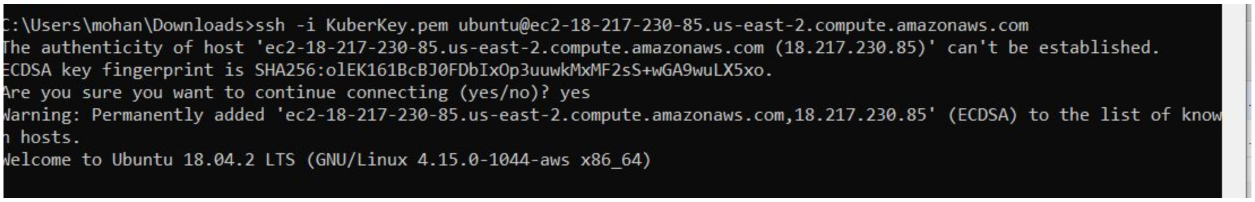
Create a kubernetes cluster. First, from the AWS management console launch two ubuntu instances and name it node1 and node2 respectively. Next install Kubernetes on the instances.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (
Node1	i-01d2d9271443430…	t2.micro	us-east-2c	● running	✓ 2/2 checks …	None	ec2-18-223-1
Node2	i-02cd3afd26a7bbee4	t2.micro	us-east-2c	● running	✓ 2/2 checks …	None	ec2-18-223-1

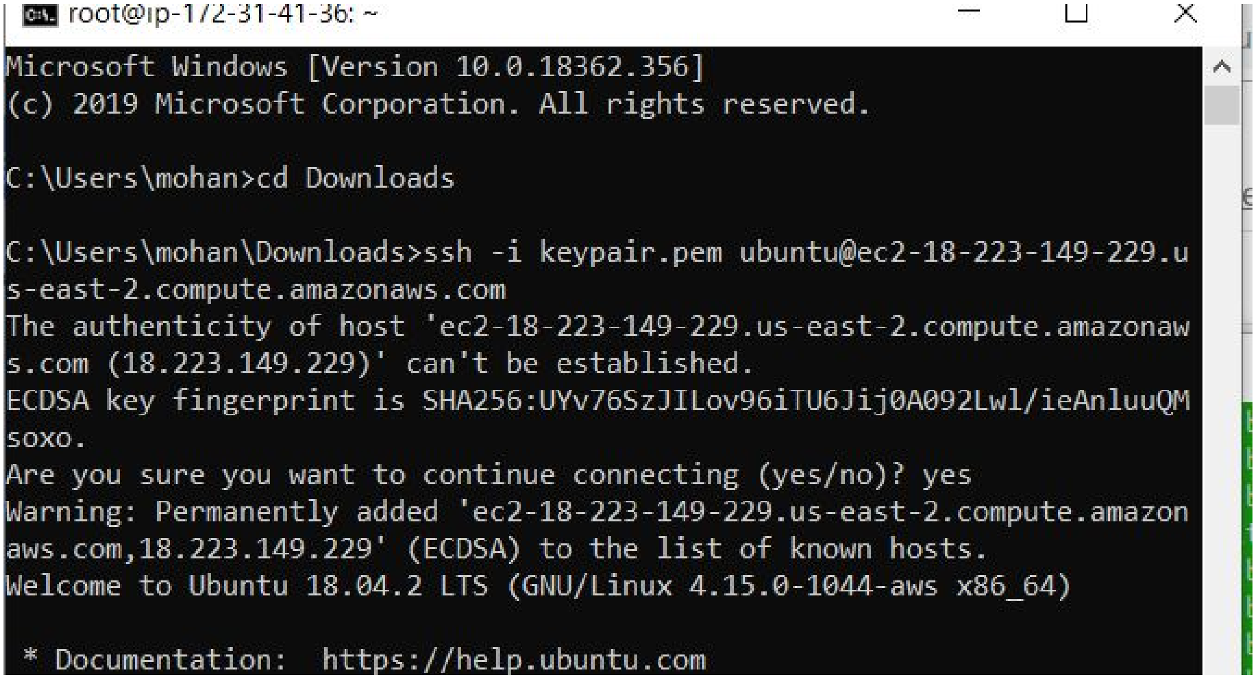
Install Kubernetes cluster and enable node1 to be the manager node. Since instance will be the manager node and will contain the cluster data that kubernetes needs to manage. The manager node is incharge of all the pods inside the worker nodes and will handle all the schedules of the worker nodes.

Node1(manager) - log onto both instances and perfprm apt-get update on both instances



```
C:\Users\mohan\Downloads>ssh -i KuberKey.pem ubuntu@ec2-18-217-230-85.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-18-217-230-85.us-east-2.compute.amazonaws.com (18.217.230.85)' can't be established.
ECDSA key fingerprint is SHA256:olEK161BcBJ0FDbIxOp3uuwkMxMF2sS+wGA9wuLX5xo.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-18-217-230-85.us-east-2.compute.amazonaws.com,18.217.230.85' (ECDSA) to the list of know
n hosts.
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1044-aws x86_64)
```

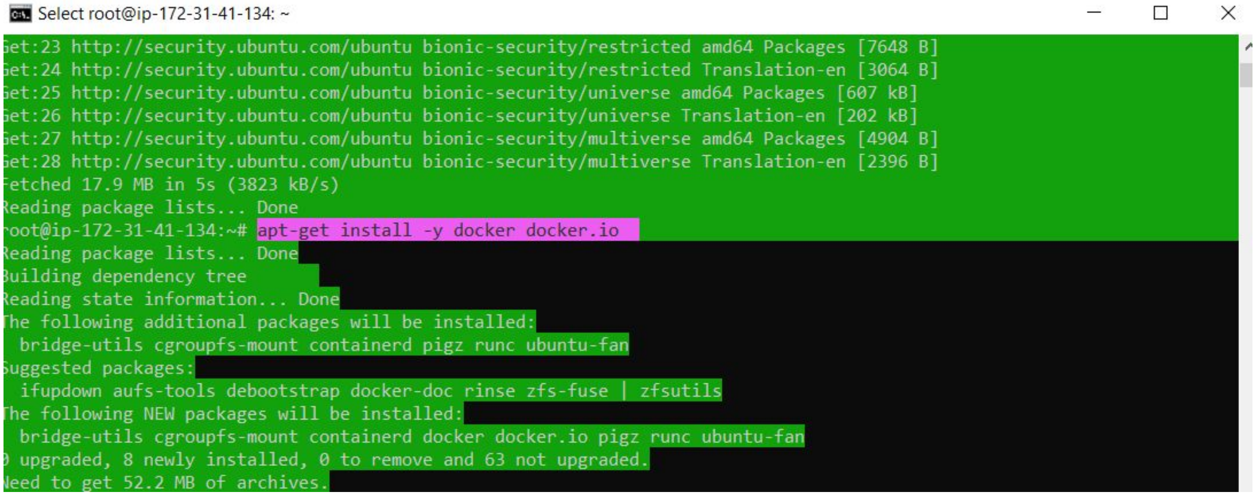
Node2



```
Microsoft Windows [Version 10.0.18362.356]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\mohan>cd Downloads

C:\Users\mohan\Downloads>ssh -i keypair.pem ubuntu@ec2-18-223-149-229.u
s-east-2.compute.amazonaws.com
The authenticity of host 'ec2-18-223-149-229.us-east-2.compute.amazonaw
s.com (18.223.149.229)' can't be established.
ECDSA key fingerprint is SHA256:UYv76SzJILov96iTU6Jij0A092Lwl/ieAnluuQM
soxo.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-18-223-149-229.us-east-2.compute.amazon
aws.com,18.223.149.229' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1044-aws x86_64)

 * Documentation:  https://help.ubuntu.com
```

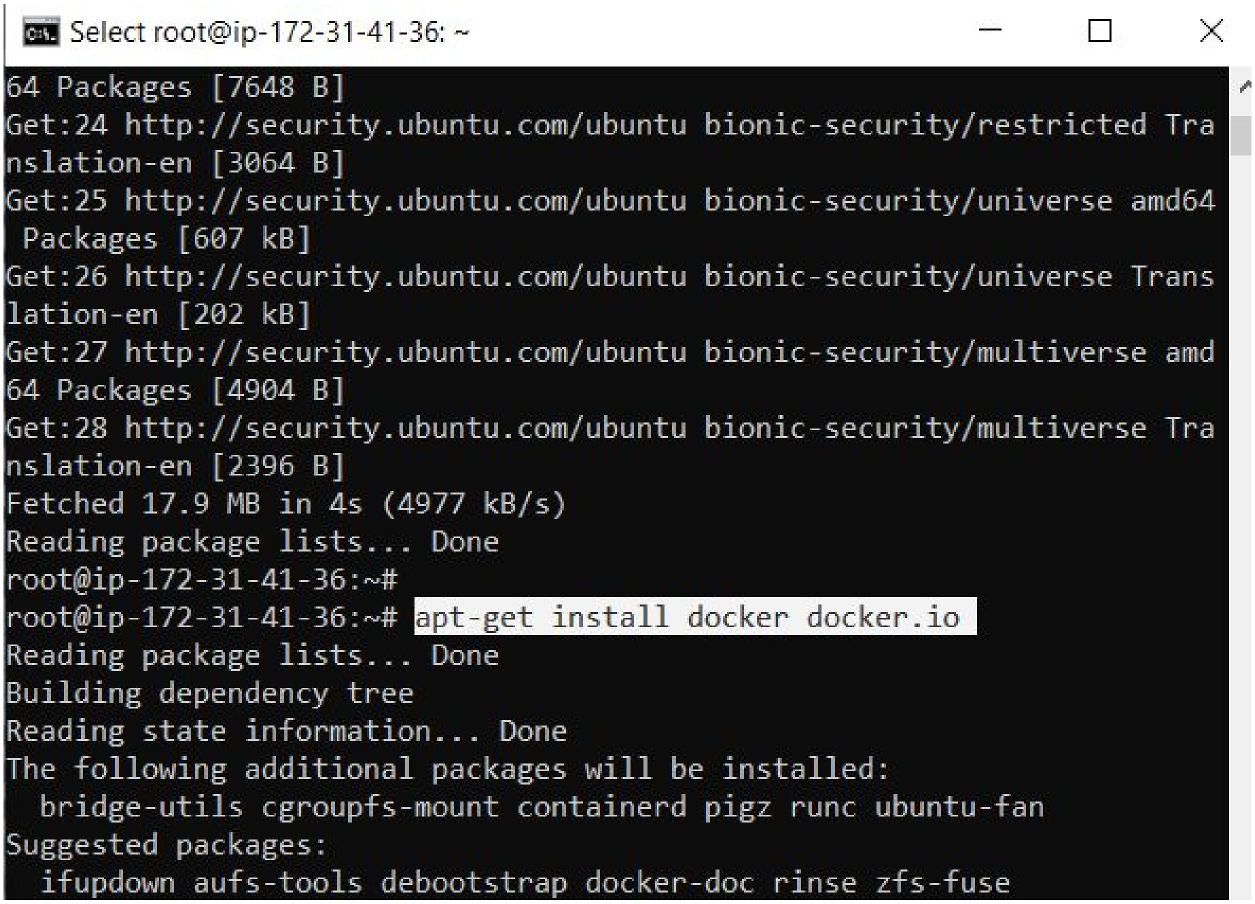
Install Kubernetes in both instances. Hence log into both instances. First docker needs to be installed as it is the runtime andlater kubernetes needs to be installed as it is the orchestrator. Run apt-get install docker on both machines

node1 (manager node)



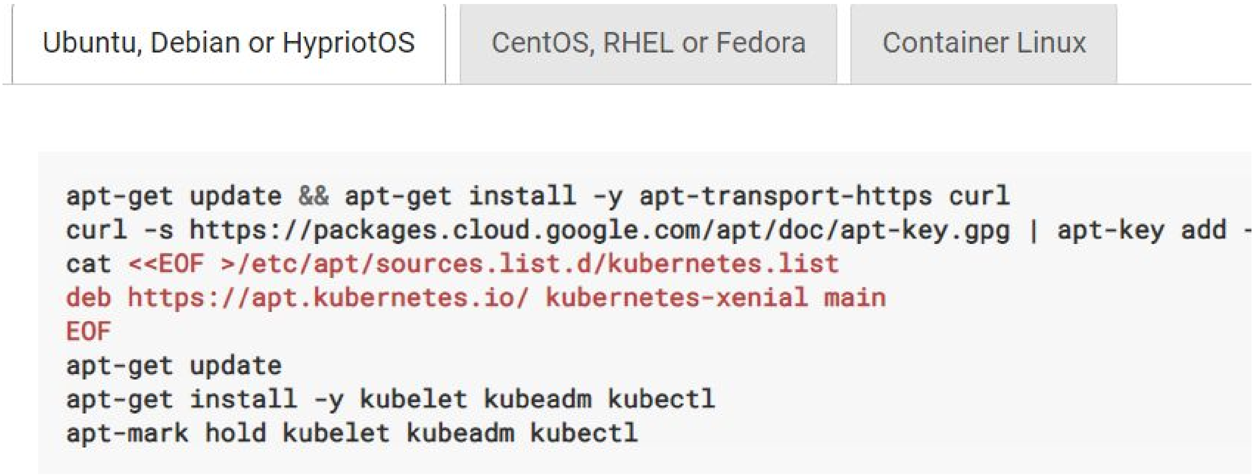
```
Get:23 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [7648 B]
Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [3064 B]
Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [607 kB]
Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [202 kB]
Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [4904 B]
Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [2396 B]
Fetched 17.9 MB in 5s (3823 kB/s)
Reading package lists... Done
root@ip-172-31-41-134:~# apt-get install -y docker docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils cgroupfs-mount containerd pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils cgroupfs-mount containerd docker docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 63 not upgraded.
Need to get 52.2 MB of archives.
```

Node 2



```
64 Packages [7648 B]
Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted Tra
nslation-en [3064 B]
Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe amd64
 Packages [607 kB]
Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Trans
lation-en [202 kB]
Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd
64 Packages [4904 B]
Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Tra
nslation-en [2396 B]
Fetched 17.9 MB in 4s (4977 kB/s)
Reading package lists... Done
root@ip-172-31-41-36:~#
root@ip-172-31-41-36:~# apt-get install docker docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils cgroupfs-mount containerd pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools debootstrap docker-doc rinse zfs-fuse
```

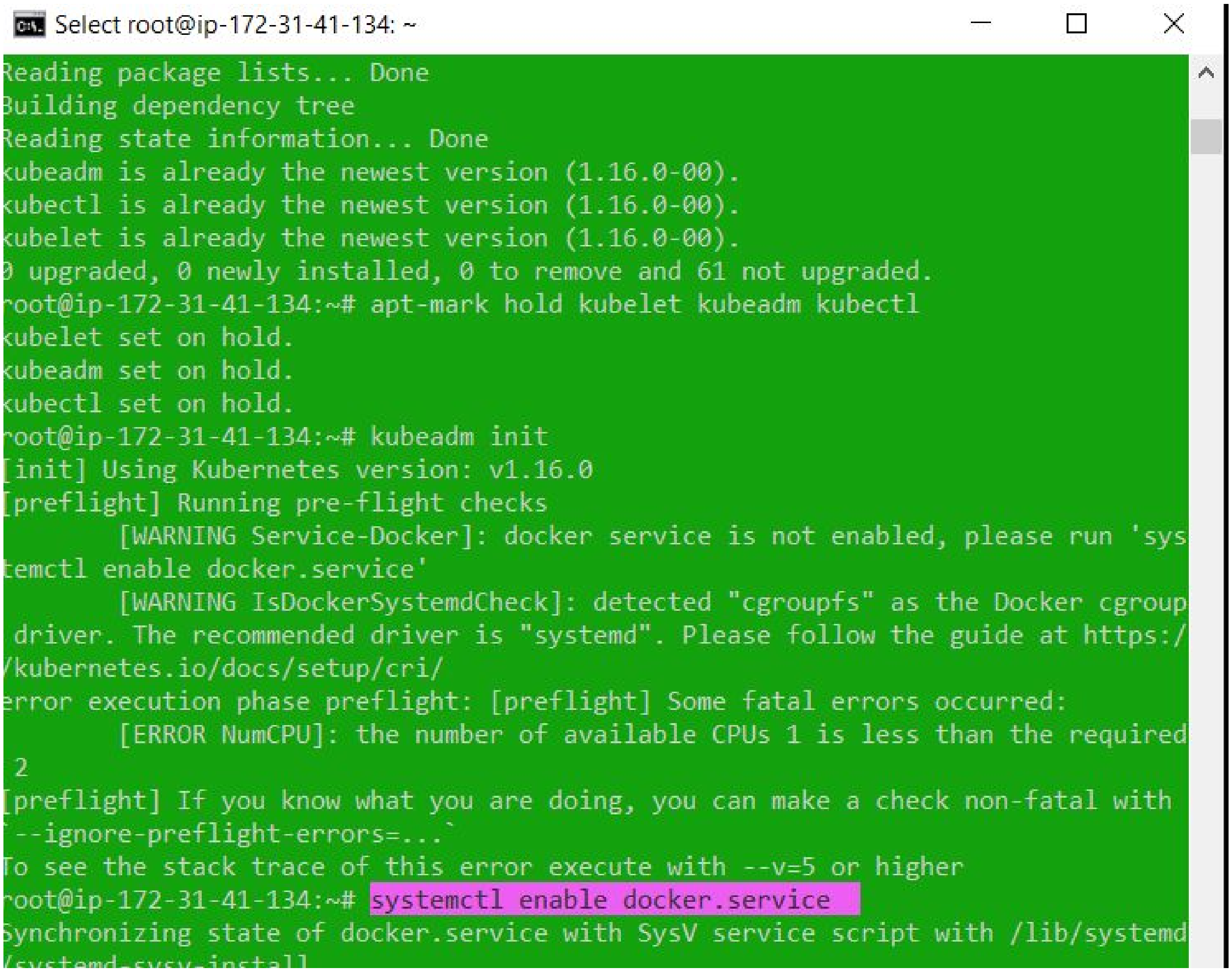
Once Docker installation is completed, then install kubernetes in both the machines from https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/
Follow the installation steps mentioned in kubernetes documentation



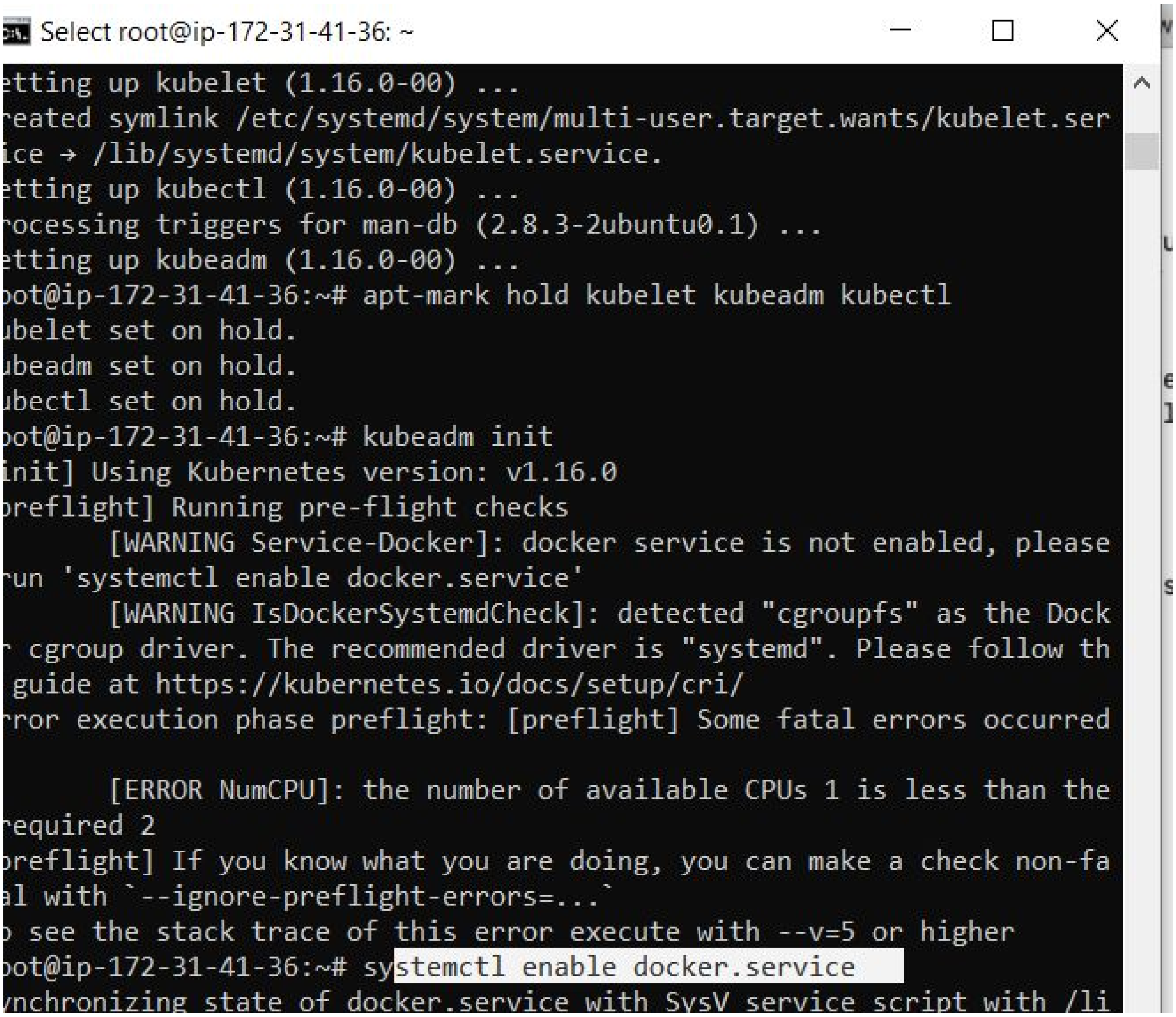
```
apt-get update && apt-get install -y apt-transport-https curl
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -
cat <<EOF >/etc/apt/sources.list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main
EOF
apt-get update
apt-get install -y kubelet kubeadm kubectl
apt-mark hold kubelet kubeadm kubectl
```

The above code installs three packages kubelet
(runs on all machines in the cluster and starts all pods and containers) kubeadm(to bootstrap

the cluster) and kubectl(the command line util to start the cluster, services etc.). The last line prevents the packages from uninstalling.



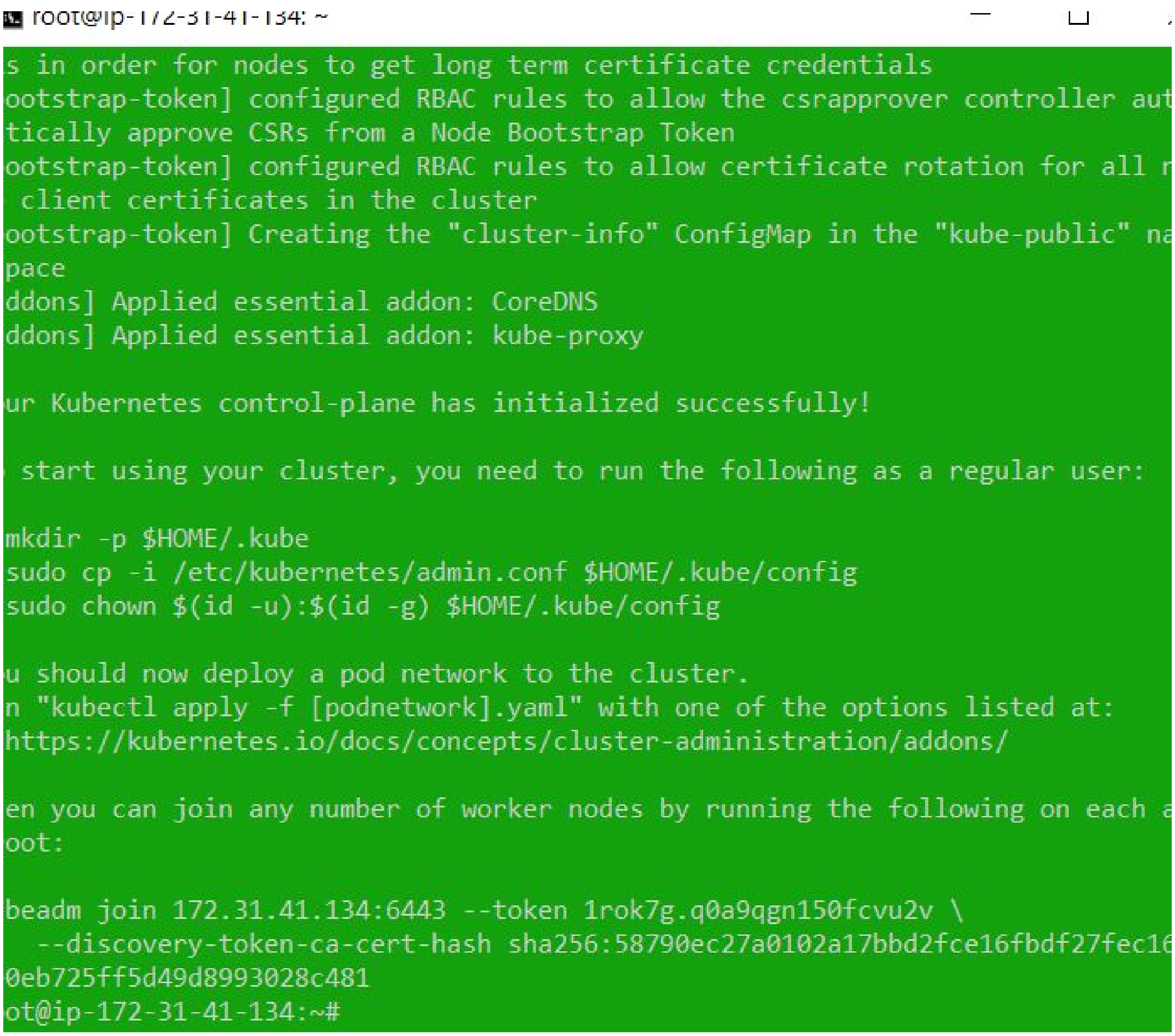
```
Reading package lists... Done
Building dependency tree
Reading state information... Done
kubeadm is already the newest version (1.16.0-00).
kubectl is already the newest version (1.16.0-00).
kubelet is already the newest version (1.16.0-00).
0 upgraded, 0 newly installed, 0 to remove and 61 not upgraded.
root@ip-172-31-41-134:~# apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
root@ip-172-31-41-134:~# kubeadm init
[init] Using Kubernetes version: v1.16.0
[preflight] Running pre-flight checks
        [WARNING Service-Docker]: docker service is not enabled, please run 'sys
temctl enable docker.service'
        [WARNING IsDockerSystemdCheck]: detected "cgroupfs" as the Docker cgroup
 driver. The recommended driver is "systemd". Please follow the guide at https:/
/kubernetes.io/docs/setup/cri/
error execution phase preflight: [preflight] Some fatal errors occurred:
        [ERROR NumCPU]: the number of available CPUs 1 is less than the required
 2
[preflight] If you know what you are doing, you can make a check non-fatal with
`--ignore-preflight-errors=...`
To see the stack trace of this error execute with --v=5 or higher
root@ip-172-31-41-134:~# systemctl enable docker.service
Synchronizing state of docker.service with SysV service script with /lib/systemd
/systemd-sysv-install
```



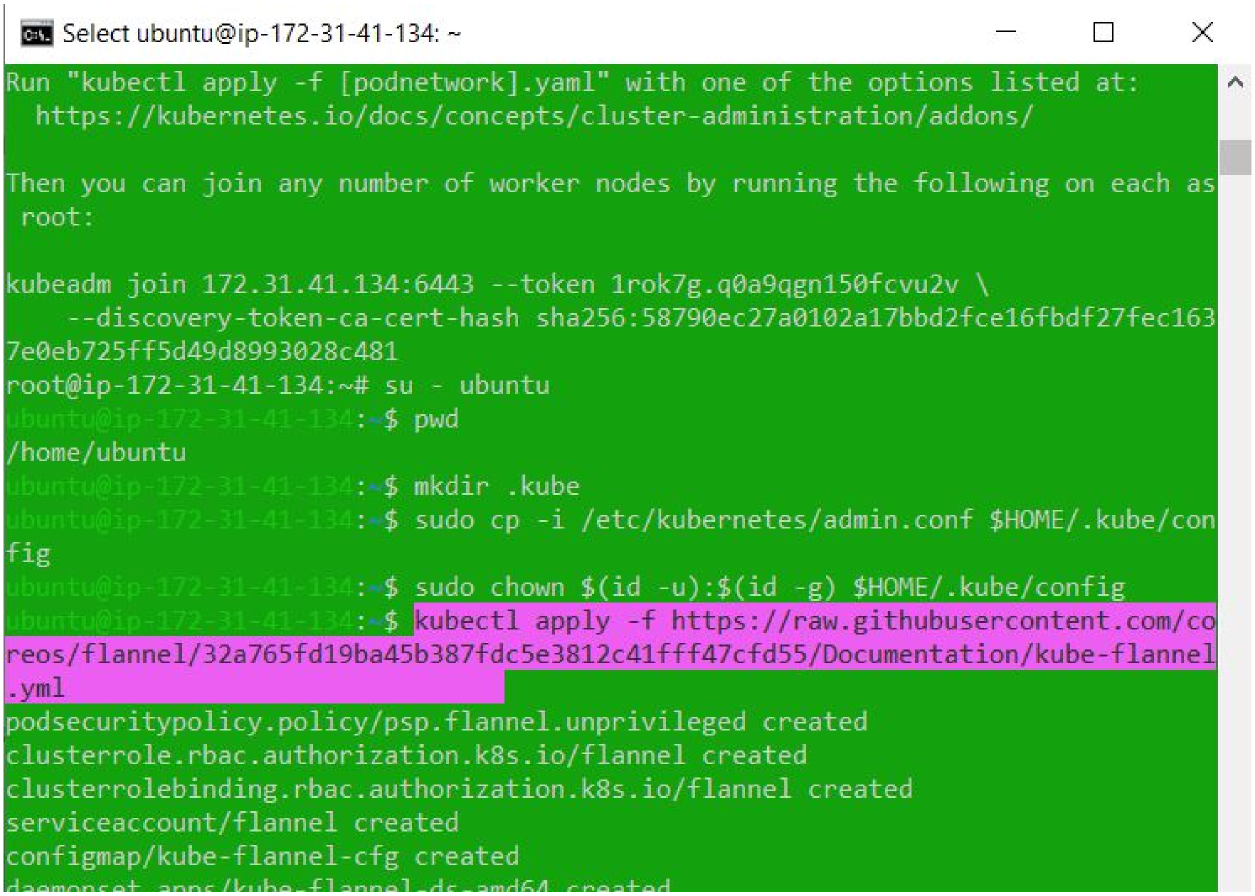
```
etting up kubelet (1.16.0-00) ...
reated symlink /etc/systemd/system/multi-user.target.wants/kubelet.ser
ice → /lib/systemd/system/kubelet.service.
etting up kubectl (1.16.0-00) ...
rocessing triggers for man-db (2.8.3-2ubuntu0.1) ...
etting up kubeadm (1.16.0-00) ...
oot@ip-172-31-41-36:~# apt-mark hold kubelet kubeadm kubectl
ubelet set on hold.
ubeadm set on hold.
ubectl set on hold.
oot@ip-172-31-41-36:~# kubeadm init
init] Using Kubernetes version: v1.16.0
preflight] Running pre-flight checks
        [WARNING Service-Docker]: docker service is not enabled, please
run 'systemctl enable docker.service'
        [WARNING IsDockerSystemdCheck]: detected "cgroupfs" as the Dock
r cgroup driver. The recommended driver is "systemd". Please follow th
 guide at https://kubernetes.io/docs/setup/cri/
rror execution phase preflight: [preflight] Some fatal errors occurred

        [ERROR NumCPU]: the number of available CPUs 1 is less than the
required 2
preflight] If you know what you are doing, you can make a check non-fa
al with `--ignore-preflight-errors=...`
o see the stack trace of this error execute with --v=5 or higher
oot@ip-172-31-41-36:~# systemctl enable docker.service
ynchronizing state of docker.service with SysV service script with /li
```

There are a lot of networks available in Kubernetes and for the purpose of the project we will install the flannel network. After installing docker run the the kubeadm init command along with the cidr 10.24.0.0/16 if flannel has been the chosen network. Install docker service and flannell network on both manager and worker node.

The kubernetes cluster has been created

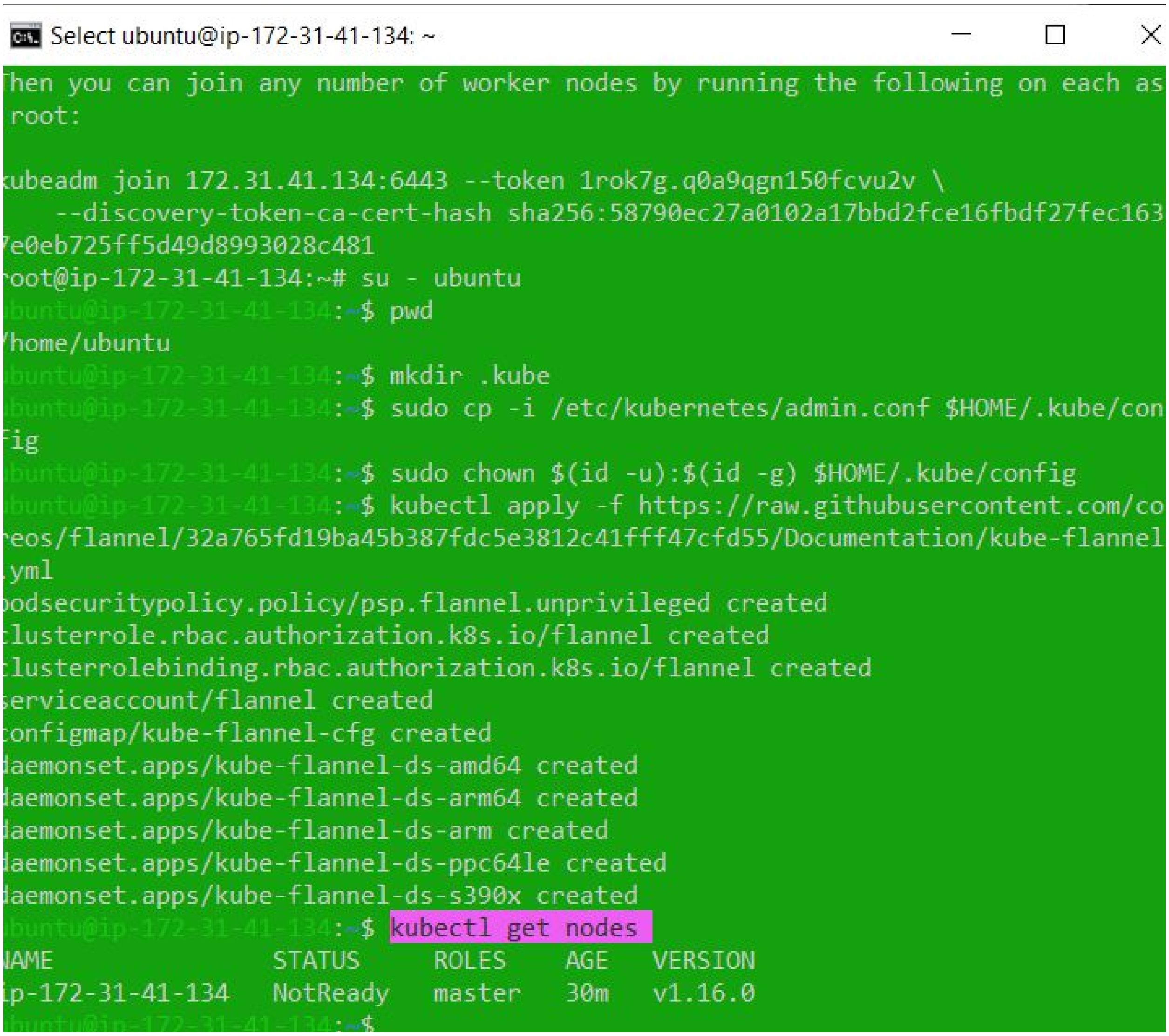


```
s in order for nodes to get long term certificate credentials
ootstrap-token] configured RBAC rules to allow the csrapprover controller aut
tically approve CSRs from a Node Bootstrap Token
ootstrap-token] configured RBAC rules to allow certificate rotation for all r
 client certificates in the cluster
ootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" na
pace
ddons] Applied essential addon: CoreDNS
ddons] Applied essential addon: kube-proxy

ur Kubernetes control-plane has initialized successfully!

 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

u should now deploy a pod network to the cluster.
n "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/

en you can join any number of worker nodes by running the following on each a
oot:

beadm join 172.31.41.134:6443 --token 1rok7g.q0a9qgn150fcvu2v \
  --discovery-token-ca-cert-hash sha256:58790ec27a0102a17bbd2fce16fbdf27fec16
0eb725ff5d49d8993028c481
ot@ip-172-31-41-134:~#
```



```
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 172.31.41.134:6443 --token 1rok7g.q0a9qgn150fcvu2v \
    --discovery-token-ca-cert-hash sha256:58790ec27a0102a17bbd2fce16fbdf27fec163
7e0eb725ff5d49d8993028c481
root@ip-172-31-41-134:~# su - ubuntu
ubuntu@ip-172-31-41-134:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-41-134:~$ mkdir .kube
ubuntu@ip-172-31-41-134:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/con
fig
ubuntu@ip-172-31-41-134:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-41-134:~$ kubectl apply -f https://raw.githubusercontent.com/co
reos/flannel/32a765fd19ba45b387fdc5e3812c41fff47cfd55/Documentation/kube-flannel
.yml
podsecuritypolicy.policy/psp.flannel.unprivileged created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds-amd64 created
```

Run the following ‘getnode’ commands in order to view the number of nodes in Kubernetes



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

kubeadm join 172.31.41.134:6443 --token 1rok7g.q0a9qgn150fcvu2v \
    --discovery-token-ca-cert-hash sha256:58790ec27a0102a17bbd2fce16fbdf27fec163
7e0eb725ff5d49d8993028c481
root@ip-172-31-41-134:~# su - ubuntu
ubuntu@ip-172-31-41-134:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-41-134:~$ mkdir .kube
ubuntu@ip-172-31-41-134:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/con
fig
ubuntu@ip-172-31-41-134:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-41-134:~$ kubectl apply -f https://raw.githubusercontent.com/co
reos/flannel/32a765fd19ba45b387fdc5e3812c41fff47cfd55/Documentation/kube-flannel
.yml
podsecuritypolicy.policy/psp.flannel.unprivileged created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds-amd64 created
daemonset.apps/kube-flannel-ds-arm64 created
daemonset.apps/kube-flannel-ds-arm created
daemonset.apps/kube-flannel-ds-ppc64le created
daemonset.apps/kube-flannel-ds-s390x created
ubuntu@ip-172-31-41-134:~$ kubectl get nodes
NAME               STATUS     ROLES    AGE   VERSION
ip-172-31-41-134   NotReady   master   30m   v1.16.0
ubuntu@ip-172-31-41-134:~$
```

Clusters have been successfully created.

REFERENCES

https://kubernetes.io/