sudo su

apt-get update

swapoff -a

nano /etc/fstab

nano /etc/hostname

nano /etc/hosts

set static ip

sudo apt-get install openssh-server

sudo su

apt-get update

apt-get install -y docker.io

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 http://apt.kubernetes.io/ kubernetes-xenial main
EOF

apt-get update

apt-get install -y kubelet kubeadm kubectl

nano /etc/systemd/system/kubelet.service.d/10-kubeadm.conf

Environment=”cgroup-driver=systemd/cgroup-driver=cgroupfs”

kubeadm init --apiserver-advertise-address=<ip-address-of-kmaster-vm> --pod-network-cidr=192.168.0.0/16

$ mkdir -p $HOME/.kube
$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
$ sudo chown $(id -u):$(id -g) $HOME/.kube/config

kubectl get pods -o wide --all-namespaces

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

kubectl proxy

http://localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/

kubectl create serviceaccount dashboard -n default

```
kubectl create clusterrolebinding dashboard-admin -n default \
  --clusterrole=cluster-admin \
  --serviceaccount=default:dashboard
```

```
kubectl get secret $(kubectl get serviceaccount dashboard -o jsonpath="{.secrets[0].name}") -o jsonpath="{.data.token}" | base64 --decode
```

sudo kubeadm join --apiserver-advertise-address=<ip-address-of-the master> --pod-network-cidr=192.168.0.0/16

```
kubectl taint nodes --all node-role.kubernetes.io/master-
```

```
kubectl apply -f https://docs.projectcalico.org/v2.6/getting-started/
kubernetes/installation/hosted/kubeadm/1.6/calico.yaml
```

```
kubectl create deployment nginx --image=nginx
```

```
kubectl get deployments
```

```
    kubectl create service nodeport nginx --tcp=80:80
```

```
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
```

```
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2 # tells deployment to run 2 pods matching the template
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9
        ports:
        - containerPort: 80
```

```
kubectl apply -f arun.yaml
```

```
kubectl describe deployment nginx-deployment
```

```
kubectl get pods -l app=nginx
```

```
kubectl delete deployment nginx-deployment
```