Kubernetes Setup

Install Docker first

```
sudo apt install docker.io
```

Enable Docker

```
sudo systemctl enable docker
```

Add Google Kubernetes Key

```
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add
```

Add Kubernetes to Deb

```
sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"
```

Install Kubernetes

```
sudo apt install kubeadm kubectl
```

Disable Memory Swap off

```
sudo swapoff -a
```

Setup Kubernetes Cluster & Start Master Node

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
```

Update the Kubernetes Configuration

```
mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Deploy a Kubernetes Network

Cilium
Big Cloud
WeaveNet
Romana
Flannel

Let us use Flannel , since it had some recent updates

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

For Slave Node

Install Docker first

```
sudo apt install docker.io
```

Enable Docker

```
sudo systemctl enable docker
```

Add Google Kubernetes Key

```
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo
apt-key add
```

Add Kubernetes to Deb

```
sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-
xenial main"
```

Install Kubernetes

```
sudo apt install kubeadm kubectl
```

Disable Memory Swap off

```
sudo swapoff -a
```

Join the cluster

This shall get generated once the master node setups the cluster successfully

```
sudo kubeadm join 172.1.1.10:6443 --token qdjnpd.5glu39uxr92xarsj --
discovery-token-ca-cert-hash
sha256:ed0684156c718caf425ceae6c85a56c05f7b49037cde3a2f1fd57430a4f58f89
```

Deploy an application

```
kubectl run --image=nginx nginx-server --port=80 --env="DOMAIN=cluster"
kubectl expose deployment nginx-server --port=80 --name=nginx-service
```

Deploy Dashboard

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/
dashboard/master/src/deploy/recommended/kubernetes-dashboard.yaml
```

Create a User Account

```
kubectl create clusterrolebinding kubernetes-dashboard –
clusterrole=cluster-admin – serviceaccount=kube-system:kubernetes-
dashboard

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
```

Access Kubernetes

```
kubectl proxy –address="cluster-info" -p 8001 --accept-hosts='^*$'
```

Teardown a Node

```
kubectl drain (Node Name) – delete-local-data – forcee – ignore-
daemonsets
```

Delete a Node

```
kubectl delete node (Node Name)
```

Reset the Cluster

```
sudo kubeadm reset
```

Run a Sample

via CLI

```
kubectl run hello-kubernetes --replicas=3 --image=paulbouwer/hello-kubernetes:1.5 --port=8080
```

Expose the Service

```
kubectl expose deployment hello-kubernetes --type=LoadBalancer --port=80 --target-port=8080 –name=hello-kubernetes
```

Or

```
kubectl run hello-world --replicas=2 --image=gcr.io/google-samples/node-hello:1.0  --port=8080

kubectl expose deployment hello-world --type=NodePort --name=hello-service
```

via YAML

```
kubectl apply -f yaml/hello-world.yaml
```

Scale

```
kubectl scale deployment hello-world  --replicas=3
```

AutoScale

```
Kubectl autoscale deployment hello-world –min=1 –max=5 –cpu-percent=75
```