**Create a Kubernetes cluster using kubeadm**

We are creating a Kubernetes cluster using kubeadm.

For this we will have need of two Linux machines(may be Virtual machines) with static IP address and after this have to disable all swap nodes as follow:

$ sudo swapoff -a

$ sudo sed -i '/ swap / s/^/#/' /etc/fstab

Updating sysctl settings for Kubernetes networking

cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf

br\_netfilter

EOF

sudo modprobe br\_netfilter

cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf

net.bridge.bridge-nf-call-ip6tables = 1

net.bridge.bridge-nf-call-iptables = 1

EOF

sudo sysctl –system

Now I’ll install Docker (if not already installed)

sudo apt-get update

sudo apt-get install -y apt-transport-https ca-certificates curl software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"

sudo apt-get update

sudo apt-get install -y docker-ce docker-ce-cli containerd.io

sudo systemctl enable docker

sudo systemctl start docker

Along with Docker has been started, now docker is in running condition

Again installing kubeadm, kubelet, and kubectl on all nodes(if not installed)

sudo apt-get update && sudo apt-get install -y apt-transport-https curl

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

cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list

deb https://apt.kubernetes.io/ kubernetes-xenial main

EOF

sudo apt-get update

sudo apt-get install -y kubelet kubeadm kubectl

sudo apt-mark hold kubelet kubeadm kubectl

Initialize the Kubernetes master node

For the master node:

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

After initialization completes, setting up kubeconfig for the `kubectl` command:

mkdir -p $HOME/.kube

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

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

Now installing a pod network add-on

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

Now join worker nodes to the cluster

On the master node, after `kubeadm init`, we will see a command like:

kubeadm join <master-ip>:6443 --token <token> --discovery-token-ca-cert-hash sha256:<hash>

Running this command on each worker node to join the cluster.

Final verifying the cluster

On the master node:

$ kubectl get nodes

Now as everything have been in suitable manner, there should be the master and worker nodes listed with status `Ready`