**How to install kubernetes and run one pod using kubernetes such that the master node and worker node are on different virtual machines without using minikube?**

VM with IP Address 10.3.10.89 will have master node and VM with IP Address 10.3.10.90 will have worker node.

All the below commands can be implemented without using “sudo”, if in the beginning we use the command “sudo -s”.

**On worker’s node:**

ssh -p 9094 guest@180.151.3.44

ssh jyoti@10.3.10.90

**Update the system:**

sudo apt update

sudo apt upgrade

**Install Docker:**

sudo apt install docker.io

**Start and enable the Docker service:**

sudo systemctl start docker

sudo systemctl enable docker

**Install Kubernetes components:**

sudo apt install kubeadm kubelet kubectl

**On master node’s VM :**

ssh -p 9094 guest@180.151.3.44

ssh jyoti@10.3.10.89

All the below commands can be implemented without using “sudo”, if in the beginning we use the command “sudo -s”.

**Update the system:**

sudo apt update

sudo apt upgrade

**Install Docker:**

sudo apt install docker.io

**Start and enable the Docker service:**

sudo systemctl start docker

sudo systemctl enable docker

**Install Kubernetes components:**

sudo apt install kubeadm kubelet kubectl

**Initialize the Kubernetes cluster:**

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

This command will output a kubeadm join command, which you'll need later.

If any error comes during this command, do **kubeadm reset.**

**Set up the Kubernetes configuration for the current user:**

mkdir -p $HOME/.kube

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

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

**Install a network plugin (for networking between pods):**

kubectl apply -f <https://docs.projectcalico.org/v3.21/manifests/calico.yaml>

**Verify the status of the cluster:**

kubectl get nodes

You should see the master node in the "Ready" state.

**Join the worker node to the cluster (using the kubeadm join command from step 5):**

**(This step should be done on worker node’s VM)**

sudo kubeadm join <master-node-ip>:<master-node-port> --token <token> --discovery-token-ca-cert-hash <hash>

Replace <master-node-ip>, <master-node-port>, <token>, and <hash> with the values provided by the kubeadm init command.

**Verify that both the master and worker nodes are in the "Ready" state:**

kubectl get nodes

**Create a sample pod:**

Create a file named pod.yaml with the following contents:

apiVersion: v1

kind: Pod

metadata:

name: my-pod

spec:

containers:

- name: my-container

image: nginx:latest

resources:

limits:

cpu: "1"

memory: "512Mi"

requests:

cpu: "1"

memory: "256Mi"

**Deploy the pod:**

kubectl apply -f pod.yaml

**Verify the pod is running:**

kubectl get pods

**If this command shows pod with Status as ContainerCreating, then use command:**

kubectl get pods -o wide

This will show the pod with Status as Running.

Now, you can again check the Running status of pod by using **kubectl get nodes** command.

**To enter inside the pod :**

kubectl exec -it my-pod -- /bin/bash