**Sudo su……….for root user**

**2. Update the Package List:**

bash

Copy code

sudo apt-get update

**3. Install Docker:**

bash

Copy code

sudo apt-get install -y docker.io

**4. Add Kubernetes GPG Key:**

bash

Copy code

sudo mkdir -p /etc/apt/keyrings

curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

**5. Add Kubernetes APT Repository:**

bash

Copy code

echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.30/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

**6. Update the Package List Again:**

bash

Copy code

sudo apt-get update

**7. Install Kubernetes Components:**

bash

Copy code

sudo apt-get install -y kubelet kubeadm kubectl

**8. Prevent Automatic Updates to Kubernetes Components:**

bash

Copy code

sudo apt-mark hold kubelet kubeadm kubectl

**9. Enable and Start the kubelet Service:**

bash

Copy code

sudo systemctl enable --now kubelet

**10. Disable Swap:**

bash

Copy code

sudo swapoff -a

**11. Remove Swap Entry from /etc/fstab:**

bash

Copy code

sudo sed -i '/swap/d' /etc/fstab

**12. Reload /etc/fstab:**

bash

Copy code

sudo mount -a

**13. Verify Memory Status:**

bash

Copy code

free -h

**14. Pull Kubernetes Images:**

bash

Copy code

sudo kubeadm config images pull

**15. Check Docker Status:**

bash

Copy code

sudo systemctl status docker

**16. Check kubelet Status:**

bash

Copy code

sudo systemctl status kubelet

**17. Initialize the Kubernetes Cluster:**

Replace 172.31.39.252 with your master node's private IP.

bash

Copy code

sudo kubeadm init --apiserver-advertise-address=172.31.39.252 --pod-network-cidr=192.168.0.0/16 --ignore-preflight-errors=all

**18. Set Up kubectl for the Default User:**

bash

Copy code

mkdir -p $HOME/.kube

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

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

**19. Export KUBECONFIG Environment Variable (optional):**

bash

Copy code

export KUBECONFIG=/etc/kubernetes/admin.conf

**20. Install the Calico Network Plugin:**

bash

Copy code

kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/tigera-operator.yaml

bash

Copy code

kubectl create -f <https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/custom-resources.yaml>

kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml

poddisruptionbudget.policy/calico-kube-controllers created

curl -I <https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/tigera-operator.yaml>

These commands will help you set up a Kubernetes master node with Calico as the network plugin.

4o

**You said:**

now give me commands to set up slave

To set up a Kubernetes slave (worker) node, follow these steps:

**1. Update the Package List:**

bash

Copy code

sudo apt-get update

**2. Install Docker:**

bash

Copy code

sudo apt-get install -y docker.io

**3. Add Kubernetes GPG Key:**

bash

Copy code

sudo mkdir -p /etc/apt/keyrings

curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

**4. Add Kubernetes APT Repository:**

bash

Copy code

echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.30/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

**5. Update the Package List Again:**

bash

Copy code

sudo apt-get update

**6. Install Kubernetes Components:**

bash

Copy code

sudo apt-get install -y kubelet kubeadm kubectl

**7. Prevent Automatic Updates to Kubernetes Components:**

bash

Copy code

sudo apt-mark hold kubelet kubeadm kubectl

**8. Enable and Start the kubelet Service:**

bash

Copy code

sudo systemctl enable --now kubelet

**9. Disable Swap:**

bash

Copy code

sudo swapoff -a

**10. Remove Swap Entry from /etc/fstab:**

bash

Copy code

sudo sed -i '/swap/d' /etc/fstab

**11. Reload /etc/fstab:**

bash

Copy code

sudo mount -a

**12. Verify Memory Status:**

bash

Copy code

free -h

**13. Join the Worker Node to the Kubernetes Cluster:**

Use the join command generated by the master node when you ran kubeadm init. It should look something like this:

bash

Copy code

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

Replace <master-ip>, <token>, and <hash> with the appropriate values.

**14. Check Node Status (from the master node):**

After joining the worker node, you can check the status from the master node:

bash

Copy code

kubectl get nodes