**Kubernetes Cluster Configuration:-**

Created-by: - Suraj Chopade Email: - [srj281191@gmail.com](mailto:srj281191@gmail.com)

* **Prerequisites:-**

1. Two nodes one as MASTER and second as Client
2. Curl Installed
3. Docker service installed
4. SWAP Should be off
   * **Steps:-**

* **Creating Master node:-**

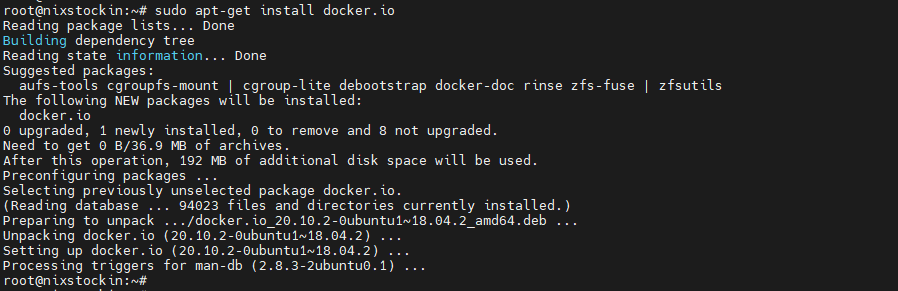
1. **Update the package list with the command**

# sudo apt-get update

1. **Install Docker and start the service**

# sudo apt-get install docker.io

# sudo systemctl start docker --now

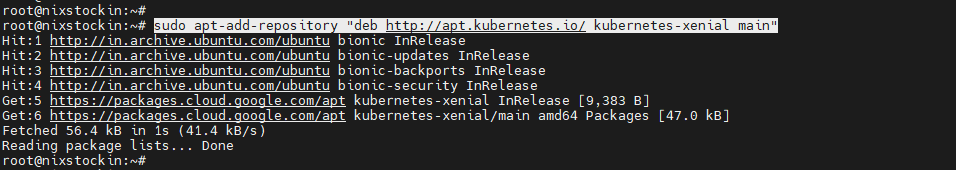


1. **Add Kubernetes Repository by adding kubernetes gpg key for kubernetes installation**

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

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



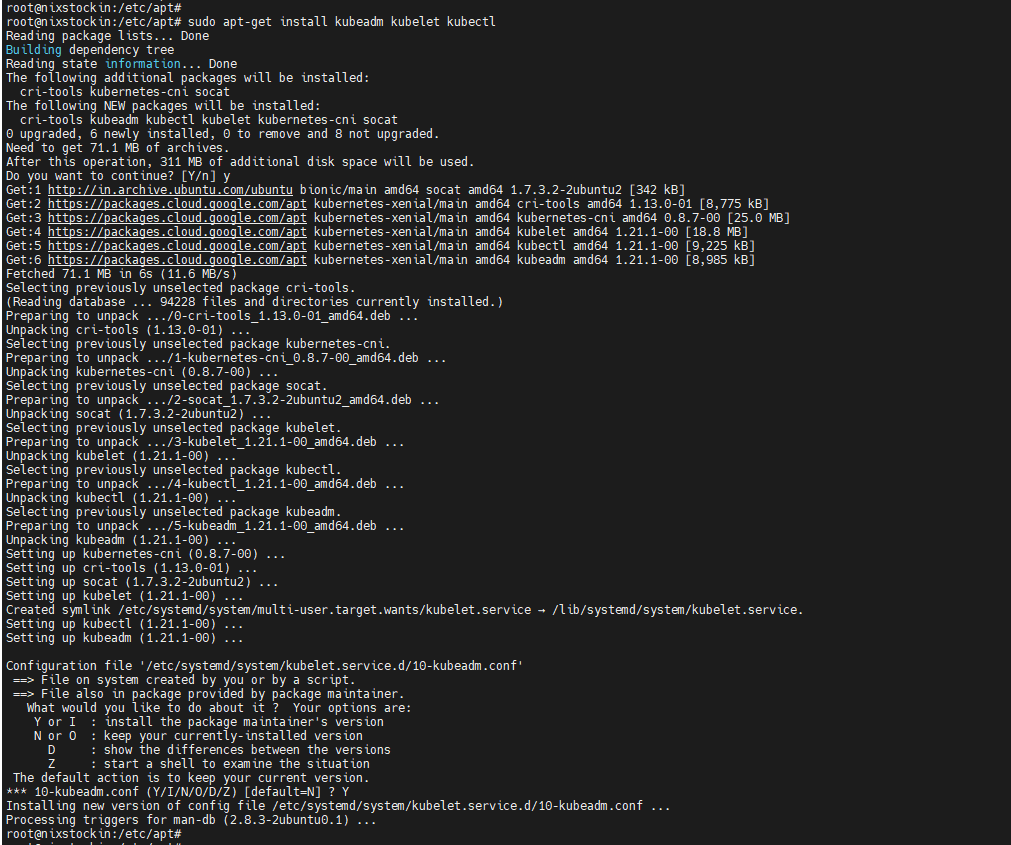


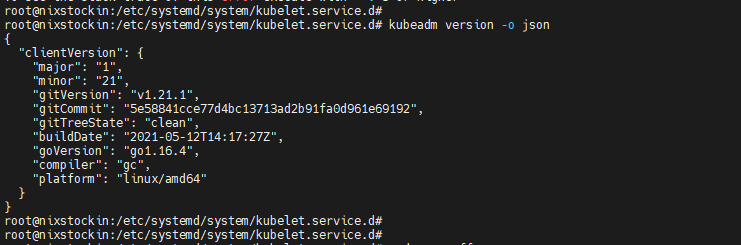
1. **Install Kubernetes tools**

# sudo apt-get install kubeadm kubelet kubectl

# sudo apt-mark hold kubeadm kubelet kubectl

# kubeadm version

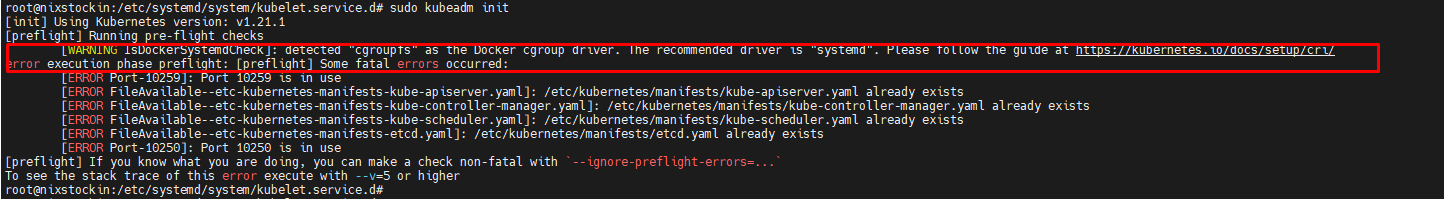




1. **Initialize Kubernetes on Master Node**

# sudo kubeadm init

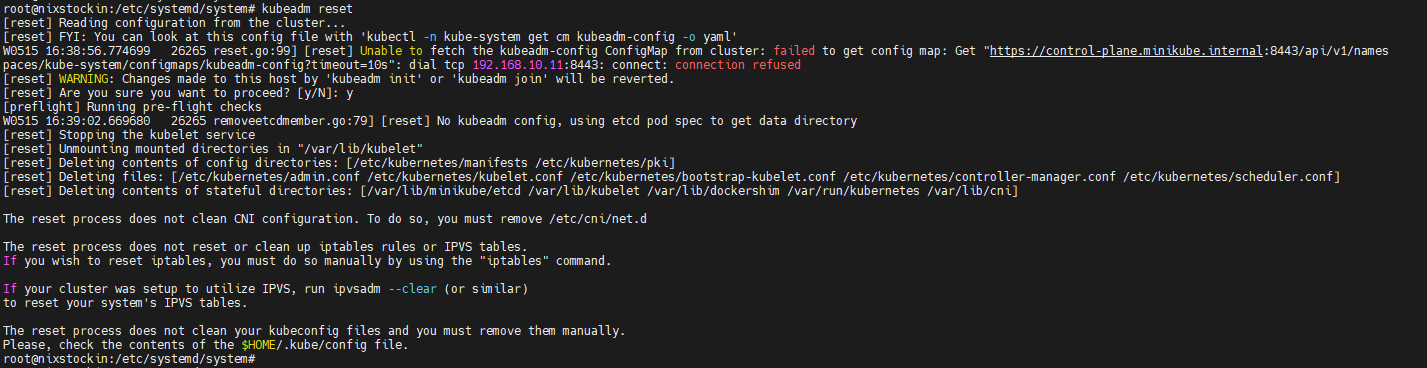
Note: - On this step you can receive an error as follows



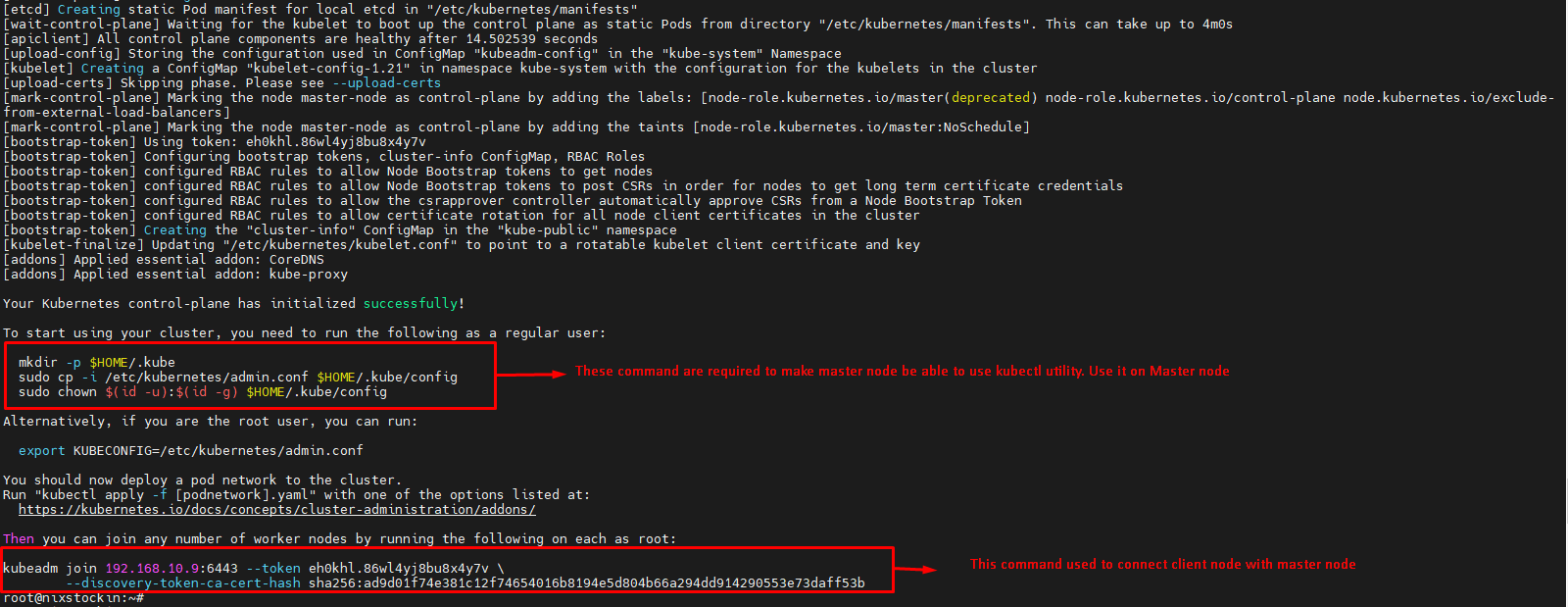
**Use following commands to fix above error**

# kubeadm reset

# echo 1 > /proc/sys/net/ipv4/ip\_forward



1. **Now go ahead with kubeadm init and you should receive following output**



Now fire commands mentioned in above snippet in red highlighted block as to make master node as Admin node

# mkdir -p $HOME/.kube

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

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

* **Configuring Client Node:-**

1. **Install the Kubernetes repository on client node as mentioned in step 3. Install the required tools with following commands**

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

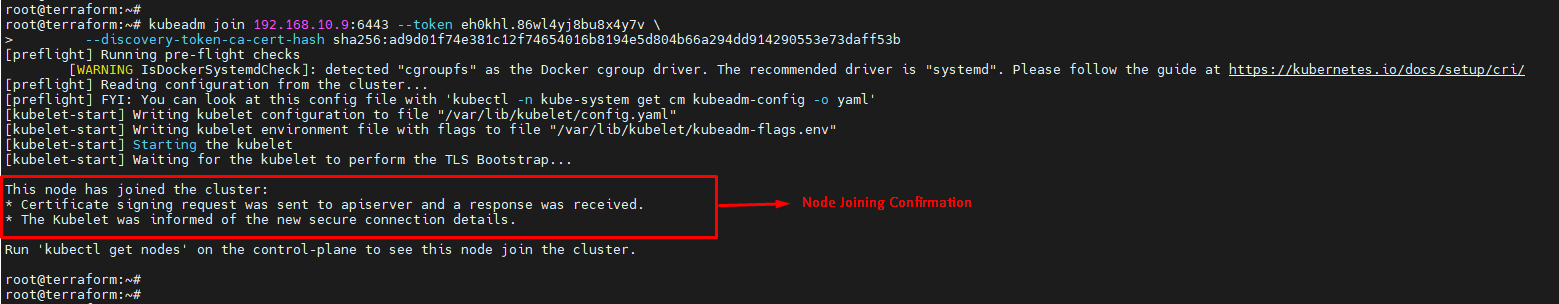
# sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"

# sudo apt-get install kubelet kubeadm kubectl docker.io

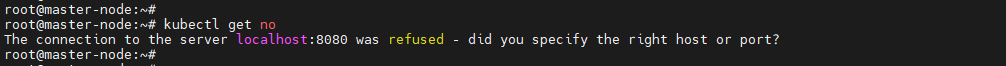
1. **Now use kubeadm join command to to join client node to master node**.

# kubeadm join 192.168.10.9:6443 --token eh0khl.86wl4yj8bu8x4y7v \

> --discovery-token-ca-cert-hash sha256:ad9d01f74e381c12f74654016b8194e5d804b66a294dd914290553e73daff53b



**You can try checking the number of nodes in cluster with “**kubectl get nodes**” but you will receive following error**

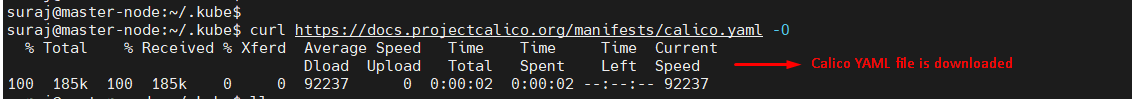


**Above error means you don’t network connectivity for pods. So following steps will help to establish connectivity**.

1. **Now to make pods communication you require overlay network. In our case we are using Calico Overlay network.**

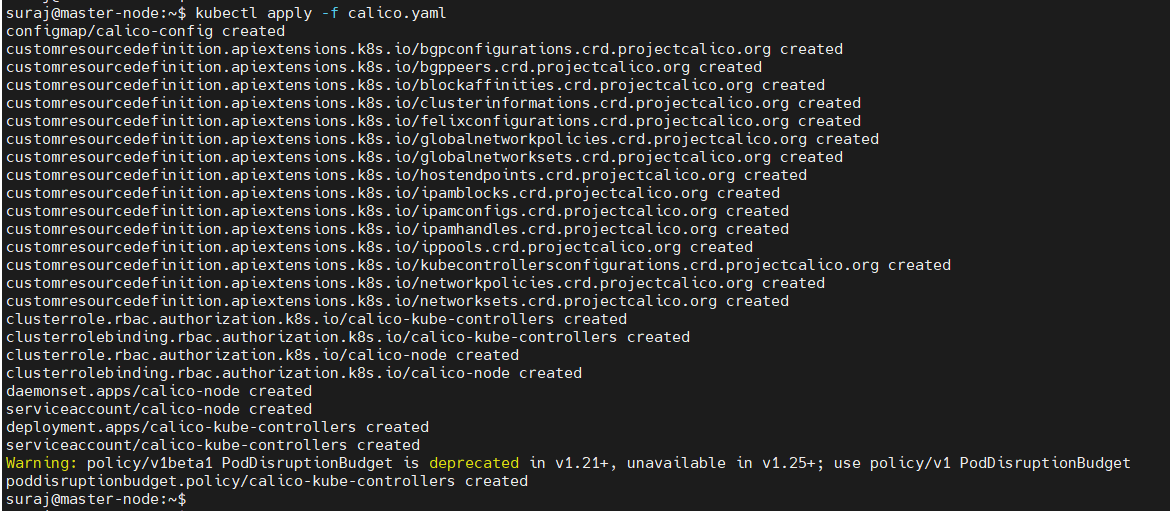
Download **Calico YAML** file

# curl https://docs.projectcalico.org/manifests/calico.yaml -O



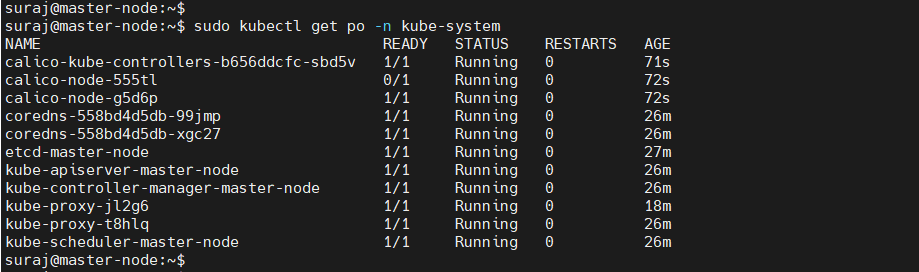
1. **Apply the Calico.YAML to create Overlay network pods**

# kubectl apply –f calico.yaml



1. **Verify the connected nodes and pods now with**

# kubectl get nodes –o wide



* **Important links:-**

<https://phoenixnap.com/kb/install-kubernetes-on-ubuntu>

<https://docs.projectcalico.org/getting-started/kubernetes/quickstart>

=========================================== **Cluster Configuration Completed** =========================================================