**Optional:**

Install kubeadm, kubectl and kubelet:

<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>

On All the nodes:

1. Update the apt package index and install packages needed to use the Kubernetes apt repository:

sudo apt-get update

sudo apt-get install -y apt-transport-https ca-certificates curl

1. Download the Google Cloud public signing key:

sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg

1. Add the Kubernetes apt repository:

echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list

1. Update apt package index, install kubelet, kubeadm and kubectl, and pin their version:

sudo apt-get update

1. sudo apt-get install -y kubelet kubeadm kubectl

**On Master**

**Initialize the cluster**

sudo kubeadm init

sudo mkdir -p $HOME/.kube

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

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

**Install Networking**

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

**Verify that cluster is up:**

sudo kubectl get nodes

You should see one master node in “ready” state

**On Worker Nodes:**

Copy the join command from kubeadm init output and execute on the worker nodes. Optional: specify Node Name for your worker nodes

Example:

sudo kubeadm join <ip address:port> --token <Some-token> --discovery-token-ca-cert-hash <SomeHash> --node-name <worker1 or worker2>

**On Master**

sudo kubectl get pods -A

All components should be in “Running” state

Note: If you have lost the join command, please use the below command to retrieve it:

sudo kubeadm token create --print-join-command

**Optional (auto completion)**

echo 'source <(kubectl completion bash)' >>~/.bashrc

source .bashrc