FOR CONTROL PLANE:

==================

curl https://get.docker.com/ | bash

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

kubeadm init --pod-network-cidr 192.168.0.0/16

BELOW COMMAND NEEDED TO BE EXECUTED IN THE WORKER NODE:

kubeadm join 10.1.1.81:6443 --token j1dmnn.5p2vhbau96eupjdl \

--discovery-token-ca-cert-hash sha256:cf2b3dc616f30fde87730fa1955a14b391eaf15d737be54d2d3a7c32634bc6ec

mkdir -p $HOME/.kube

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

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

cp /usr/bin/kubectl /usr/bin/ku

ku taint node ip-10-1-1-20 node-role.kubernetes.io/master:NoSchedule-

ku get pods -o=custom-columns="POD\_NAME:.metadata.name,POD\_IP:.status.podIP,POD\_HOST:.status.hostIP"

root@ip-10-1-1-81:~# kubectl get nodes

NAME STATUS ROLES AGE VERSION

ip-10-1-1-81 NotReady master 2m16s v1.18.3

kubectl apply -f https://docs.projectcalico.org/v3.14/manifests/calico.yaml

root@ip-10-1-1-81:~# ku get nodes

NAME STATUS ROLES AGE VERSION

ip-10-1-1-81 Ready master 5m47s v1.18.3

ON WORKER NODE:

==============

curl https://get.docker.com/ | bash

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

kubeadm join 10.1.1.81:6443 --token j1dmnn.5p2vhbau96eupjdl \

--discovery-token-ca-cert-hash sha256:cf2b3dc616f30fde87730fa1955a14b391eaf15d737be54d2d3a7c32634bc6ec

This node has joined the cluster:

\* Certificate signing request was sent to apiserver and a response was received.

\* The Kubelet was informed of the new secure connection details.

AGAIN ON THE MASTER NODE:

=========================

root@ip-10-1-1-81:~# kubectl get nodes

NAME STATUS ROLES AGE VERSION

ip-10-1-1-81 Ready master 7m50s v1.18.3

ip-10-1-1-99 Ready <none> 2m14s v1.18.3

root@ip-10-1-1-81:~# kubectl label node ip-10-1-1-99 node-role.kubernetes.io/worker=worker

node/ip-10-1-1-99 labeled

root@ip-10-1-1-81:~# ku get nodes

NAME STATUS ROLES AGE VERSION

ip-10-1-1-81 Ready master 8m19s v1.18.3

ip-10-1-1-99 Ready worker 2m43s v1.18.3

=====================================================================================================================================

mkdir -p $HOME/.kube

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

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

kubectl apply -f calico.yaml

kubeadm init --pod-network-cidr "10.100.0.0/16" --service-cidr "10.200.0.0/16" --service-dns-domain "mavrick.local"

kubeadm init --pod-network-cidr "10.244.0.0/16"

mkdir -p $HOME/.kube

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

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

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

kubectl create -f canal.yaml

If you are using pod CIDR 10.244.0.0/16, skip to the next step. If you are using a different pod CIDR with kubeadm, no changes are required - Calico will automatically detect the CIDR based on the running configuration. For other platforms, make sure you uncomment the CALICO\_IPV4POOL\_CIDR variable in the manifest and set it to the same value as your chosen pod CIDR.

kubectl apply -f canal.yaml # If you dont use pod CIDR other than 10.244.0.0/16, you need to update this file.

To delete the controller, kubeadm reset --force

rm -rf $HOME/.kube/config

rm -rf /etc/cni/net.d

kubelet cert are valid for an year.

kubeadm alpha certs check-expiration.

kubeadm token generate

kubeadm token create <generated-token> --print-join-command --ttl=0

Make sure you save the snapshot to a diffrent folder apart from /tmp/. Save it to /etc/backups

apt install -y etcd

mkdir /etc/backups #Dont put in /tmp as the data will be lost after reboot.

sudo ETCDCTL\_API=3 etcdctl --endpoints https://127.0.0.1:2379 --cert=/etc/kubernetes/pki/etcd/server.crt --key=/etc/kubernetes/pki/etcd/server.key --cacert=/etc/kubernetes/pki/etcd/ca.crt snapshot save /etc/backups/etcdbackup.db

ETCDCTL\_API=3 etcdctl --write-out=table snapshot status /etc/backups/etcdbackup.db

Reboot the master and check everything comes back. There wont be any impact incase of a reboot. Next we will delta etcd and try.

Perform the below and waint until nothing shows up and then perform the restore.

rm -rf /var/lib/etcd

ETCDCTL\_API=3 etcdctl \

--endpoints=https://127.0.0.1:2379 \

--cacert=/etc/kubernetes/pki/etcd/ca.crt \

--name=master \

--cert=/etc/kubernetes/pki/etcd/server.crt \

--key=/etc/kubernetes/pki/etcd/server.key \

--data-dir /var/lib/etcd \

--initial-cluster=master=https://127.0.0.1:2380 \

--initial-cluster-token etcd-cluster-new \

--initial-advertise-peer-urls=https://127.0.0.1:2380 \

snapshot restore /etc/backups/etcdbackup.db