sudo apt install -y kubelet kubeadm kubectl docker  
Jst try installing docker 1.12...iam not getting solution with docker 1.17ce.( i think its not supported with kubelet)  
Make swap off by #swapoff -a  
Now reset kubeadm by #kubeadm reset  
Now try #kubeadm init  
after that check #systemctl status kubelet  
it will be working

sudo kubeadm init --pod-network-cidr=192.168.0.0/16 --upload-certs --cri-socket /run/cri-dockerd.sock --control-plane-endpoint=192.168.5.218:6443 --control-plane-endpoint=192.168.5.241:6443

sudo kubeadm init --control-plane-endpoint= --upload-certs --apiserver-advertise-address=192.168.5.248 --pod-network-cidr=192.168.0.0/16

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

mkdir -p $HOME/.kube

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

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

Alternatively, if you are the root user, you can run:

export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.

Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:

https://kubernetes.io/docs/concepts/cluster-administration/addons/

kubectl create -f https://docs.projectcalico.org/v3.15/manifests/calico.yaml

You can now join any number of the control-plane node running the following command on each as root:

frontend kubernetes

bind 192.168.5.218:6443

option tcplog

mode tcp

default\_backend kubernetes-master-nodes

backend kubernetes-master-nodes

mode tcp

balance roundrobin

option tcp-check

server kubernetes-master1 192.168.5.248:6443 check fall 3 rise 2

server kubernetes-master2 192.168.5.204:6443 check fall 3 rise 2

server kubernetes-master3 192.168.5.220:6443 check fall 3 rise 2

You can now join any number of the control-plane node running the following command on each as root:

kubeadm join 192.168.5.241:6443 --token vc3v16.fn0bes8msy942vol \

--discovery-token-ca-cert-hash sha256:bb9c02000754087315a8f9a18ab811a395d51e1c02a46aec44aaf8b564d00956 \

--control-plane --certificate-key aa7436cb4c4607c96a6caffd599ee2fb686c727793b3035a5f2fc8fe4f20b12f

Please note that the certificate-key gives access to cluster sensitive data, keep it secret!

As a safeguard, uploaded-certs will be deleted in two hours; If necessary, you can use

"kubeadm init phase upload-certs --upload-certs" to reload certs afterward.

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 192.168.5.241:6443 --token vc3v16.fn0bes8msy942vol \

--discovery-token-ca-cert-hash sha256:bb9c02000754087315a8f9a18ab811a395d51e1c02a46aec44aaf8b564d00956