---------------------------------------DOCKER-CE---------------------------------------------------

## Docker-ce install steps :

sudo apt-get update

sudo apt-get install \

apt-transport-https \

ca-certificates \

curl \

software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo apt-key fingerprint 0EBFCD88

## verify with :

## 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88

sudo add-apt-repository \

"deb [arch=amd64] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) \

stable"

sudo apt-get update

sudo apt-get install docker-ce

-------------------------------------------KUBERNETES------------------------------------------------------

## install kubelet kubectl kubeadm

## kubectl ==> commandline tool to interact with cluster

## kubeadm ==> kubernets cluster creator

## kubelet ==>

sudo swapoff -a

sudo apt-get update && apt-get install -y apt-transport-https curl  
 curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -  
 cat <<EOF >/etc/apt/sources.list.d/kubernetes.list  
 deb https://apt.kubernetes.io/ kubernetes-xenial main  
 EOF  
apt-get update  
apt-get install -y kubelet kubeadm kubectl  
apt-mark hold kubelet kubeadm kubectl

## check ports for free using command (( $ fuser PORT\_NO/tcp ))

## if port is not free kill the process using (( $ kill -9 PID ))

## ports== 6443\* 2379-2380 10250 10251 10252

----------------------Starting up cluster----------------------------------------

## starting up cluster ==>

## On Master node =>

kubeadm init

## configure kubectl

##on master

# 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

##On worker on normal user (not ROOT)

mkdir -p $HOME/.kube

scp yogesh@192.168.31.101:.kube/config .kube/config

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

## on worker nodes =>

# join the cluster using command displayed on master node

## checking nodes in cluster =>

kubectl get nodes

--------------------------Installing required packages for flannel------------------------

## install flannel to create pod network

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

## pod network cidr => ip address for nodes (must use for flannel)

## flannel install

exit

kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/bc79dd1505b0c8681ece4de4c0d86c5cd2643275/Documentation/kube-flannel.yml

## join nodes using

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

## get tokens using (on master node)

kubeadm token list

## create new token (tokens expire after 24 hr)

kubeadm token create

-------------------------------------kubeadm dashboard-------------------------------------------

kubectl create -f https://raw.githubusercontent.com/kubernetes/dashboard/master/src/deploy/recommended/kubernetes-dashboard.yaml

Alternative:

kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/master/aio/deploy/recommended/kubernetes-dashboard.yaml

## website address

<http://localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/>.

## Kubernetes Dashbaord Authentication using Token

1.Create a new ServiceAccount

kubectl create serviceaccount admin -n kube-system

2.Create a ClusterRoleBinding with Cluster Admin Privileges

kubectl create clusterrolebinding admin --clusterrole=cluster-admin --serviceaccount=kube-system:admin

3.Get the token

kubectl get secret -n kube-system | grep admin | cut -d " " -f1 | xargs -n 1 | xargs kubectl get secret -o 'jsonpath={.data.token}' -n kube-system | base64 --decode

------------------------------------Tear Down----------------------------------------------------------

## drain nodes

kubectl drain <node name> --delete-local-data --force --ignore-daemonsets

## remove node from clustor

kubectl delete node <node name>

## reset kubeadm

Kubeadm reset