

KUBERNETES INSTALLATION WITH UBUNTU SERVER

===== COMMON FOR MASTER & SLAVE START =====

First, login as 'root' user because the following set of commands need to to be executed with 'sudo' permissions.

```
sudo su-
```

update pacage manager

```
apt-get update -y
```

install and enable docker

```
apt install docker.io -y
```

```
systemctl restart docker
```

```
systemctl enable docker.services
```

turn off swap spaces

```
swapoff -a
```

```
sed -i '/ swap / s/^\(.*\)$/#\1/g' /etc/fstab
```

Install required packages and apt keys

```
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 | tee /etc/opt/sources.list.d/kubernetes.list
```

```
deb https://apt.kubernetes.io/ kubernetes-xenial main
```

```
EOF
```

```
apt-get update -y
```

Install kubeadm, kubelet, kubectl

NOTE: both commands are same but cubectl not use in worker node

```
apt-get install -y kubelet kubeadm kubectl (use in master node )
```

```
apt-get install -y kubelet kubeadm (use in worker node )
```

Enable and start kubelet services

systemctl daemon-reload

systemctl start kubelet

systemctl enable kubelet.service

===== COMMON FOR MASTER & SLAVE END =====

===== IN MASTER NODE START =====

Steps only for kubernetes master

Switch to the root user

sudo su -

Initialize the kubernetes master by executing below commands

kubeadm init

exit as root user & execute as normal ubuntu user

exit

mkdir -p \$HOME/.kube

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

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

To find kubectl version

kubectl version

ls ---->>>>(To view the list)

to check the nodes

kubectl get nodes

Get token

kubeadm token create --print-join-command

Again to check the nodes

```
kubectl get nodes
```

To verify, if kubectl working or not, run the following commands

```
kubectl get pods -o wide -n kube-system
```

you will notice from the previous command, that all the pods are running except one: 'core-dns' , for resolving this we will install a "pod network". To install the weave pod network, run the following commands:

```
kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')"
```

===== IN MASTER NODE END =====

===== CONNECT TO THE CLUSTER FROM REMOTE MACHINE START =====

We create a linux server with aws

Download the latest kubectl with the command:

```
$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)
```

Make the kubectl binary executable.

```
chmod +x ./kubectl
```

Move the binary in to your PATH.

```
sudo mv ./kubectl /usr/local/bin/kubectl
```

To check the version fro kubectl

```
kubectl version
```

To view the nodes

```
kubectl get nodes ----->>>> (here we cant see any nodes because of there is no config file)
```

To creste a directory

mkdir ~/.kube

To view the manifest file

Vi ~/.kube/config ----->>>>(here config= file name)

----- NOW WE CAN GO TO THE MASTER NODE START -----

To view the config file

cat ~/.kube/config ----->>>>(copy the total config file)

----- NOW WE CAN GO TO THE MASTER NODE END -----

paste in the manest file then save that

To view the info

kubectl custer -info

No we check the node info

kubectl get nodes

this is the normal remote server but we connect to the cluster with the help of config file

By using this server we can create pods, nodes and delete pods everything in this server

===== CONNECT TO THE CLUSTER FROM REMOTE MACHINE END =====

===== CREATE A PODS IN REMOTE MACHINE(or) CLUSTER START =====

Containers managed by KUBERNETES

KUBERNETES can't create containers directly

KUBERNETES create a pod each pod consist of one or more containers

pod represents running process in the KUBERNETES

Name space is cluster inside cluster

View the namespace

kubectl get namespace

View the pods

kubectl get pods

we can list down all the pods, services, statefulsets, etc. in a namespace but not all the resources are listed using this command

kubectl get all

View the namespace

kubectl get ns ----->>>>>(ns= namespace)

>> default }

>> kube-system } these all are the namespaces

>> kube-public }

If we create a name space

kubectl create namespace navakoti ---->>>>>(navakoti is a name of namespace)

To run the pods with the namespace

kubectl get pods -n kube-system

this command shows deamons, replicasets, deployments, services

kubectl get all -n kube-system

===== CREATE A PODS IN REMOTE MACHINE(or) CLUSTER END =====

===== CREATE A CONTAINERS IN CLUSTER START =====

THERE IS TWO WAYS TO CREATE A CONTAINERS

1. INTERACTIVE WAY:- we can create a containers by using commands only
2. DECLARATIVE WAY:- we can create a containers by using yml files

To write a yml file

vi.javawebapppod.yml ---->>>(javawebapppod.yml is a name for this file)

To view the manifest or yml file

cat javawebapppod.yml

for checking the pods

kubectl get pods

#

kubectl apply -f javawebapppod.yml

for checking the pods

kubectl get pods ---->>>(now its running)

To show the entire process

kubectl get events

To see pod creation

kubectl describe pod javawebapppod.yml

To view the ipaddress of pod

kubectl get pods -o wide

With in the cluster we can access the using pod ip

curl pod ip:8080 ----->>>>(8080= port number)

===== CREATE A CONTAINERS IN CLUSTER END =====