**Setting up three node Kubernetes cluster**

First of all, we should have three instances created that can connect over the public network. It doesn't matter how those instances are created, for example, they can either be Digital Ocean droplets or AWS EC2 instances.

**SSH into all the instances**

Once you are into those instances, the commands that are mentioned below should be run on all the instances

**Commands to run on all the nodes**

# Get sudo working

sudo -l

# update packages and their version

sudo apt-get update && sudo apt-get upgrade -y

# install curl and apt-transport-https

sudo apt-get update && sudo apt-get install -y apt-transport-https curl

# add key to verify releases

curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

# add kubernetes apt repo

cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list

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

EOF

# install kubelet, kubeadm and kubectl

sudo apt-get update

sudo apt-get install -y kubelet kubeadm kubectl

# install docker

sudo apt-get install docker.io

# apt-mark hold is used so that these packages will not be updated/removed automatically

sudo apt-mark hold kubelet kubeadm kubectl

After the above commands are successfully run on all the worker nodes. Below steps can be followed to initialize the Kubernetes cluster.

**On Leader Node**

Run the below command on the node that you want to make the leader node. Please make sure you replace the correct IP of the node with IP-of-Node

export MASTER\_IP=<IP-of-Node>

kubeadm init --apiserver-advertise-address=${MASTER\_IP} --pod-network-cidr=10.244.0.0/16

**Join worker nodes to the Leader node**

Once the command kubeadm init is completed on the leader node, below we would get a command like below in the output of kubeadm init that can be run on worker nodes to make them join the leader node.

kubeadm join 206.189.134.39:6443 --token dxxfoj.a2zzwbfrjejzir4h \

--discovery-token-ca-cert-hash sha256:110e853989c2401b1e54aef6e8ff0393e05f18d531a75ed107cf6c05ca4170eb

mkdir -p $HOME/.kube

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

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

**Install CNI plugin**

The below command can be run on the leader node to install the CNI plugin

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

if you are not getting ip after creating pod then use this command

curl <https://docs.projectcalico.org/manifests/calico.yaml> -O  
kubectl apply -f calico.yaml  
kubectl get pods -n kube-system

if pod are not getting ip address

git hub account= <https://github.com/suhasini7131/suhasini7131> password suha@7131

Setting up 3 node Kubernetes cluster version 1.22 from scratch, on Ubuntu instances

3,311 views

Aug 29, 2021

104

DISLIKE

SHARE

CLIP

SAVE

[https://yt3.ggpht.com/VlZWR5wsfab6PKAE9RwY-ptSbGdZKNrpLJoUEwrH3RAxA6Mq2kGCbKfgmeSNGA85tGht6GUo1Q=s48-c-k-c0x00ffffff-no-rj](https://www.youtube.com/channel/UCFrZ2AgmUq3AINd7VlTLOBw)

[Vivek Singh](https://www.youtube.com/channel/UCFrZ2AgmUq3AINd7VlTLOBw)

3.62K subscribers

SUBSCRIBE

Kubernetes version 1.22 was released recently and a lot of cloud providers still not supp

apiVersion: v1

kind: Pod

metadata:

name: pod1

labels:

disk: ssd

spec:

containers:

- name: c1

image: httpd

ports:

- containerPort: 80

kind: Pod

metadata:

name: pod2

labels:

disk: ssd

spec:

containers:

- name: c1

image: httpd

ports:

- containerPort: 80

- name: c2

image: httpd

- name: c3

image: httpd