**KUBERNETES MASTER SETUP IN AWS**

AWS AMI ID: ami-3158414e (owner: 601575382750)

OS: Ubuntu

Username: ubuntu

Password: “docker\_key\_pair.pem”

Installed Packages: Docker, Kubernetes

*Note:* the image has Docker and Kubernetes installed, but setting up Kubernetes master creates files that are tied up with the IP address of the master machine. Thus, after creating the instance, run below commands in order to setup the Kubernetes master.

Create an instance using the above image. After that, run below commands:

Step 1 # sign as root

sudo su -

Step 2 # pass bridged IPv4 traffic to iptables` chains which is required by certain CNI networks

sudo sysctl net.bridge.bridge-nf-call-iptables=1

Step 3 # initialize kubeadm (done only for master)

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

Step 4 # create kubeconfig so the user can run kubectl commands

mkdir -p $HOME/.kube

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

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

Step 5 # install flannel networking

kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/v0.9.1/Documentation/kube-flannel.yml

Step 6 # Only when you want to use master node to host pods (since we have 1 node cluster)

kubectl taint nodes --all node-role.kubernetes.io/master-

# Test

kubectl get pods --all-namespaces

**JOIN NODE AS WORKER**

AWS AMI ID: ami-3158414e (owner: 601575382750)

OS: Ubuntu

Username: ubuntu

Password: “docker\_key\_pair.pem”

Installed Packages: Docker, Kubernetes

*Note:* the image has Docker and Kubernetes installed, but setting up Kubernetes master creates files that are tied up with the IP address of the master machine. Thus, after creating the instance, run below commands in order to setup the Kubernetes master.

Create an instance using the above image. After that, run below commands:

Step 1 # sign as root

sudo su -

Step 2 # pass bridged IPv4 traffic to iptables` chains which is required by certain CNI networks

sudo sysctl net.bridge.bridge-nf-call-iptables=1

Step 3 # Get the token and sha digest from master and run in the new node

Run in master

kubeadm token create --print-join-command

This will output a command, run that command in new nodes to join the master as workers.

**1.1 APPENDIX (Docker and Kubernetes Installation)**

Steps to install Docker and Kubernetes (the one which is used to create the AMI). If you are using the provided AMI in the page 1, you do not need to follow these steps.

OS: Ubuntu

# update and upgrade the apt-get package manager

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

# install docker

sudo apt install docker.io

# start and enable docker engine

sudo systemctl start docker

sudo systemctl enable docker

# download and add the key to Kubernetes install

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

# Add the package to apt-get

echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" >> /etc/apt/sources.list.d/kubernetes.list

# apt-get update and install Kubernetes

sudo apt-get update && sudo apt-get install -y kubelet kubeadm kubectl kubernetes-cni