==============================================================================================================================

Mithun Technologies Marathahalli, Banglore. Ph: +91-8296242028 / +91-9980923226

==============================================================================================================================

Agenda: Kubernetes Setup Using Kubeadm In AWS EC2 Ubuntu Servers

=======

Prerequisite:

=============

3 - Ubuntu Serves

1 - Manager (4GB RAM , 2 Core) t2.medium

2 - Workers (1 GB, 1 Core) t2.micro

Note: Open Required Ports In AWS Security Groups. For now we will open All trafic.

==========COMMON FOR MASTER & SLAVES START ====

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

sudo su -

# Update Package Manager

apt-get update -y

# Install And Enable Docker

apt install docker.io -y

usermod -aG docker ubuntu

systemctl restart docker

systemctl enable docker.service

#Turn Off Swap Space

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/apt/sources.list.d/kubernetes.list

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

EOF

apt-get update -y

#Install kubeadm, Kubelet And Kubectl

apt-get install -y kubelet kubeadm kubectl

# Enable and start kubelet service

systemctl daemon-reload

systemctl start kubelet

systemctl enable kubelet.service

==========COMMON FOR MASTER & SLAVES END=====

===========In Master Node Start====================

# Steps Only For Kubernetes Master

# Switch to the root user.

sudo su -

# Initialize Kubernates master by executing below commond.

kubeadm init

#exit as root user & exeucte 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 verify, if kubectl is working or not, run the following command.

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 command:

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

kubectl get nodes

kubectl get pods

kubectl get pods --all-namespaces

# Get token

kubeadm token create --print-join-command

=========In Master Node End====================

Add Worker Machines to Kubernates Master

=========================================

Copy kubeadm join token from and execute in Worker Nodes to join to cluster

kubectl commonds has to be executed in master machine.

Check Nodes

=============

kubectl get nodes

Deploy Sample Application

==========================

kubectl run nginx-demo --image=nginx --port=80

kubectl expose pod nginx-demo --port=80 --type=NodePort

Get Node Port details

=====================

kubectl get services

# Label node

kubectl label node <nodeName> node-role.kubernetes.io/worker=worker

https://github.com/kubernetes/kubeadm/issues/2272

WARNING: kubeadm cannot validate component configs for API groups [kubelet.config.k8s.io kubeproxy.config.k8s.io] · Issue #2272 · kubernetes/kubeadm

1. <http://mithuntechnologies.com/DevOpsMT/Aug22nd2020/Kubernetes/Nov_12th_Kubernetes_Introduction.html>
2. <http://mithuntechnologies.com/DevOpsMT/Aug22nd2020/Kubernetes/Nov_13th_Kubernetes_Setup.html>
3. <http://mithuntechnologies.com/DevOpsMT/Aug22nd2020/Kubernetes/Nov_14th_Kubernetes_POD_Services.html>
4. <http://mithuntechnologies.com/DevOpsMT/Aug22nd2020/Kubernetes/Nov_16th_Kubernetes_ReplicationController.html>