Creation of Kubernetes Cluster

Step 1 – Creating EC2 instances

To setup the cluster you need to create 3 EC2 instances wherein one instances will be ‘Master’ and 2 will be the ‘Worker Nodes’, worker nodes can be more than 2 based on the requirement.EC2 Instance

Specification of the Master Instance–

Instance type –t2.medium (2vCPUs, 4GB RAM)

Storage >= 10GB

Amazon Machine Image –Amazon Linux 2

Note: -The Master instance should have a minimum of 2vCPUs for the cluster to be operational EC2 Instance Specification of the Worker Instances–

Instance type –t2.small(1vCPUs, 2GB RAM)

Storage >= 10GB

Amazon Machine Image –Amazon Linux 2

Note: -The instance types mentioned here are indicative and can be modified based on the user/application requirement, only the minimum criteria for the Master Instance should be met.

Security Group Configuration to be used for the all the Instances –

Ssh port 22

http port 80

https 443

custom tcp port 10250

custom tcp port 6443

custom tcp port 31111

Step 2 –Initial Setup of the Instances

Once the instances are created, connect to the Master EC2 instance via an SSH client and perform the below initial steps as a root user–1.

Disable Swap on the Instance- Swap need to disable on the instance for the cluster to operate correctly.

This can be done by typing the below commands

swapoff -a

sed -i.bak -r 's/(.+ swap .+)/#\1/' /etc/fstab

2.Disable SELinux on the Instance–SELinux needs to be disabled as it has been known to cause compatibility issues with Kubernetes.

This can be done by typing the below commands –

setenforce 0

sed -i 's/enforcing/disabled/g' /etc/selinux/config

3. Add the Kubernetes Repository –To install Kubernetes and it’s packages the Kubernetes Repository needs to be added inside the YUM repository.

This is done by creating a repository file ‘kubernetes.repo’ in the path ‘/etc/yum.repos.d’ and adding the Kubernetes repository URL.You can do this by typing the below commands –

cat <<EOF > /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64

enabled=1

gpgcheck=0

EOF

4. Update the Repository –The repositories need to be updated as a pre-cursor. The command to update is –

yum update -y

5. Install Docker, Kubelet, Kubeadm and Kubectl.

Once the repositories are updated you need to install the container runtime engine docker and the kubernetes packages. This can be done by this command

yum install docker –y

yum install -y kubeadm kubelet kubectl

6. Start and Enable Docker daemon service –The docker service needs to be started and enabled. This can be done by the command –

systemctl enable docker && systemctl start docker

7. Start and Enable Kubelet Service –The Kubelet service needs to started and enables.

This can be done by the command –

systemctl enable kubelet && systemctl start kubelet

8. Steps1 to 7 has to be repeated for all the worker EC2 instances by connecting to them via SSH and performing the commands from points 1 to 7 as the root user.

Step 3 –Configure the Kubernetes ‘Master’ Node Instance

Once the above step has been completed for all the instances connect back to the master instance and give the below command to initialize the instance as the Kubernetes Master –

kubeadm init --pod-network-cidr=172.31.0.0/16

(Private IP range of the Instances)

This command takes a couple of minutes to execute, once the command execution is complete it will generate a kubeadm join command. This join command is to be noted down as it will be needed to join worker instances

Step 4 –Run kubectl as a ‘regular’ user –

To be able to run kubectl commands as a regular user execute the below 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 –

Installing Calico network-plug-in for cluster networking –

A network plugin needs to be installed to enable networking and communication inside the cluster this can be done by installing Calico network plugin by giving the below command –

**kubectl** apply -f https://docs.projectcalico.org/v3.8/manifests/calico.yaml

kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.24.1/manifests/tigera-operator.yaml

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

Step 6 –Join worker EC2 instances to the cluster–

Once the network plugin has been installed, you need to join the worker EC2 instances by executing the ‘kubeadm join’ command which was generated in Step 3 on each of the EC2 worker instances respectively.

Step 7 –Testing –

Once the worker instances have been joined successfully, you can verify that the cluster is ready by executing the below command to see the status of the Nodes in the cluster-

kubectl get nodes

The Cluster Setup for Kubernetes using AWS EC2 instances is now complete.