# Kubernetes

# AWS EC2 Instance

# Open an SSH client.

# Locate your private key file (singaporekeypair.pem)

# # chmod 400 singaporekeypair.pem

# Now you will be able to SSH using your Public DNS/Public IP

# ssh -i "singaporekeypair.pem" [ec2-user@ec2-xx-xx-xx-xx.ap-south-1.compute.amazonaws.com](mailto:ec2-user@ec2-xx-xx-xx-xx.ap-south-1.compute.amazonaws.com)

# NB:- If SSH connectivity does not work, see changing the chmod to 600 (chmod 600 singaporekeypair.pem)

# [ec2-user@ip-xx-xx-xx-xx ~]$ sudo su

# # yum install -y git

# Steps to Check out from git repository

# # git init

# # git config --global user.email "mymail@email.com"

# # git config --global user.name " git user name"

# # mkdir gitrepo

# # cd gitrepo

# # git clone https://github.com/nevin-cleetus/kubernetes.git

# Ensure

# 1. Internet is working

# 2. Nobody should be connected to vpn or any other proxy.

# 3. Disable firewall if enabled.

# 

# 

# 

# Kubernetes Installation

$ login as: ec2-user

$ sudo su

$ 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=1

repo\_gpgcheck=0

gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg

EOF

$ cat <<EOF > /etc/sysctl.d/k8s.conf

net.bridge.bridge-nf-call-ip6tables = 1

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

EOF

$ sysctl --system

\* Applying /etc/sysctl.d/00-defaults.conf ...

\* Applying /etc/sysctl.d/k8s.conf ...

net.bridge.bridge-nf-call-ip6tables = 1

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

\* Applying /etc/sysctl.conf ...

$ setenforce 0

setenforce: SELinux is disabled

$ yum install -y kubelet kubeadm kubectl

Installed:

kubeadm.x86\_64 0:1.18.0-0 kubectl.x86\_64 0:1.18.0-0 kubelet.x86\_64 0:1.18.0-0

$ systemctl enable kubelet && systemctl start kubelet

Created symlink from /etc/systemd/system/multi-user.target.wants/kubelet.service to /usr/lib/systemd/system/kubelet.service.

# Kubernetes Master Node

$ sudo kubeadm init --pod-network-cidr=192.168.0.0/16 --ignore-preflight-errors=NumCPU

Your Kubernetes control-plane has initialized successfully!

……

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 172.31.41.123:6443 --token 7oy7kg.g2u9wmhewgxyl0zn \

--discovery-token-ca-cert-hash sha256:9e94b12d9391d5afa78bf32beece20ae129f8cff8d3a81b2085ff718a9274879

# mkdir -p $HOME/.kube

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

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

# 

# Kubernetes Worker Node

# kubeadm join xxx. xxx. xxx. xxx:6443 --token 7oy7kg.g2u9wmhewgxyl0zn \

--discovery-token-ca-cert-hash sha256:9e94b12d9391d5afa78bf32beece20ae129f8cff8d3a81b2085ff718a9274879

This node has joined the cluster:

\* Certificate signing request was sent to apiserver and a response was received.

\* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

# Kubernetes Master Node

[root@...]# kubectl cluster-info

Kubernetes master is running at https://172.31.41.123:6443

KubeDNS is running at https://172.31.41.123:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.

[root@...]# kubectl get nodes

NAME STATUS ROLES AGE VERSION

ip-X-X-X-X.ap-south-1.compute.internal NotReady <none> 2m51s v1.18.0

ip-X-X-X-X.ap-south-1.compute.internal NotReady master 7m19s v1.18.0

[root@...]# kubectl get nodes

[root@ip- kubernetes] sudo kubectl apply -f etcd.yaml

[root@ip- kubernetes] sudo kubectl apply -f rbac.yaml

[root@ip- kubernetes] sudo kubectl apply -f calico.yaml

[root@ip- kubernetes] kubectl get nodes

NAME STATUS ROLES AGE VERSION

ip- xxxxxxx Ready master 15m v1.18.0

ip-xxxxxxx Ready <none> 10m v1.18.0

kubectl get pods -o wide