Docker and Kubernetes on AWS QuickStart

Kirk Kalvar

Updated 9/20/2016

# provision k8s-console

Amazon Linux AMI 2016.03.3 (HVM), SSD Volume Type - ami-6869aa05, Free tier eligible t2.micro, Free tier eligible which has the awscli tools pre-installed

# login into k8s-console using ssh and configure awscli

aws configure

AWS Access Key ID [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SSKA]:

AWS Secret Access Key [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IOxz]:

Default region name [us-east-1]:

Default output format [None]:

# add docker

sudo yum install -y docker

sudo service docker start

sudo usermod -a -G docker ec2-user

# test docker

sudo docker images

sudo docker run hello-world

# add git

sudo yum install -y git vim

# Dockerfile and kubernetes fixes

git clone https://github.com/kskalvar/kubernetes-lb.git

# install kubernetes v1.3.6

curl -sS https://get.k8s.io | sed '$d' | bash

# correct scripts

cp kubernetes-lb/scripts/common.sh kubernetes/cluster/common.sh

cp kubernetes-lb/scripts/config-test.sh kubernetes/cluster/aws/config-test.sh

cp kubernetes-lb/scripts/config-default.sh kubernetes/cluster/aws/config-default.sh

# test kubernetes

export PATH=/home/ec2-user/kubernetes/platforms/linux/amd64:$PATH

kubectl get nodes

# set kubernetes environment variables

export NUM\_NODES=2

export KUBE\_AWS\_INSTANCE\_PREFIX=k8s

export KUBE\_AWS\_ZONE=us-east-1a

export MASTER\_SIZE=t2.micro

export AWS\_S3\_REGION=us-east-1

export NODE\_SIZE=t2.micro

export KUBERNETES\_PROVIDER=aws

# start cluster (takes about 10 minutes)

kubernetes/cluster/kube-up.sh

# create container

cd kubernetes-lb/web

sudo docker build -t kskalvar/web .

sudo docker login

sudo docker push kskalvar/web

# create pod

kubectl run web --image=kskalvar/web --port=5000

# show pods running

kubectl get pods --output wide

# scale pod

kubectl scale deployment web --replicas=2

# create load balancer

kubectl expose deployment web --port=80 --target-port=5000 --type="LoadBalancer"

# get aws external load balancer external address

kubectl get service web --output wide

# test from browser

# kill application

kubectl delete deployment,service web

# shutdown cluster

kubernetes/cluster/kube-down.sh