kubernetes

1. what is kubernetes?

- Kubernetes is an open source system for
    - automating deployments,
    - scaling and management of containerized applications
- it is a container orchestration system
- using Kubernetes that means your application is following
    - 1. the microservices architecture and
    - 2. is already containerized

2. what does Orchestrator do ?

- 1. cloning container / replica creation
- 2. restarting container
- 3. destroying container
- 4. replacing existing container

3. which are the Container Orchestation tools

- 1. Docker Swarm
    - used when replica < 10,000
    - simpler than kubernetes
- 2. Kubernetes
    - used when replica(above) > 10,000
- 3. Apache Mesos
- 4. Marathon

4. work on K for developers

- 1. make K , application development ready ---> developer
    - here we have certification : (CKAD= Certified K Application Developer)
    - upload app to K
- 2. cluster administeration : setting,config cluster ---> administrator
    - here we have certification : (CKA : where Administrator) + (CKS : where S: security Specialist)

5. K history?

- developed by Google , in Borg project
- taken over by Cloud Native Computing Foundation (CNFC)
- open source/community

kubernetes ppt

yaml in docker ppt

Kubernetes Cluster demo

0. install virtual box

> https://www.virtualbox.org/wiki/Linux_Downloads

OR

> https://download.virtualbox.org/virtualbox/6.1.16/virtualbox-6.1_6.1.16-140961~Ubuntu~bionic_amd64.deb

1. install vagrant

> $ curl -O https://releases.hashicorp.com/vagrant/2.2.14/vagrant_2.2.14_x86_64.deb

> $ sudo apt install ./vagrant_2.2.14_x86_64.deb

1. Install kubernates from

> https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/

- OR use

```
    sudo apt-get update && sudo 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 | sudo tee /etc/apt/sources.list.d/kubernetes.list
    deb https://apt.kubernetes.io/ kubernetes-xenial main
    EOF
    sudo apt-get update
    sudo apt-get install -y kubelet kubeadm kubectl
    sudo apt-mark hold kubelet kubeadm kubectl
```

2. akms in pod.yaml file

- a : apiversion
- k : <kind of object>
- m : <metadata>
- spec : <specification of object>

4. create a directory for vagrant and use

```
# creates a Vagrantfile in the directory
vagrant init hashicorp/bionic64
```

- edit the vagrantFile as

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/bionic64"
  config.vm.provision "shell", path: 'init2.sh'
```

```
  config.vm.define "node1" do |app|
      app.vm.hostname = 'node1'
      app.vm.network "private_network", ip: '192.168.2.201'
      app.vm.provider "virtualbox" do |vb|
          vb.name = "node1"
      end
  end

  config.vm.define "node2" do |app|
      app.vm.hostname = 'node2'
      app.vm.network "private_network", ip: '192.168.2.202'
      app.vm.provider "virtualbox" do |vb|
          vb.name = "node2"
      end
  end
end
```

- create init.sh file

```
sudo apt-get update

sudo apt-get install -y \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg-agent \
  software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add
-

sudo add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) \
  stable"

sudo apt-get update

sudo apt-get install -y docker-ce=5:19.03.12~3-0~ubuntu-bionic

sudo apt-mark hold docker-ce

sudo swapoff -a

sudo sed -i '/ swap / s/^\(.*\)$/#\1/g' /etc/fstab

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

cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list
```

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

sudo apt-get update

sudo apt-get install -y kubelet=1.19.1-00 kubeadm=1.19.1-00 kubectl=1.19.1-
00

sudo apt-mark hold kubelet kubeadm kubectl
```

- run vagrant commands on terminals

```
$  vagrant up
$  vagrant ssh node1
# initiate kube on terminal 1
sudo kubeadm init --apiserver-advertise-address=<ip-address> --pod-network-
cidr=10.244.0.0/16
# get all nodes
$ sudo kubectl get nodes

$ sudo kubectl describe node1



# initiate kube on terminal 2
$ vagrant ssh node2

$ sudo kubeadm init --apiserver-advertise-address=<ip-address> --pod-
network-cidr=10.244.0.0/16
```

on terminal 1

- create a yaml file for pod creation

```
$ vim mypod.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod

spec:
  containers:
  - name: mypod-container
    image: nginx
    ports:
    - containerPort: 80
    name: http-port
```

```
    protocol: TCP
```

- create pod using

```
$ kubectl create -f mypod.yml
```

- get info of pod

```
$ kubectl describe pod mypod
```