7 th Jan abscent for more refer to video in mobile

**8-Jan-20:**

**Kubernetes**

<https://github.com/javahometech/kubernetes>

# Kubernetes on AWS using Kops

### 1. Launch Linux EC2 instance in AWS

### 2. Create and attach IAM role to EC2 Instance.

Kops need permissions to access

S3

EC2

VPC

Route53

Autoscaling

etc..

### 3. Install Kops on EC2

curl -LO https://github.com/kubernetes/kops/releases/download/$(curl -s https://api.github.com/repos/kubernetes/kops/releases/latest | grep tag\_name | cut -d '"' -f 4)/kops-linux-amd64

chmod +x kops-linux-amd64

sudo mv kops-linux-amd64 /usr/local/bin/kops

1st command is downloading kops binary

2nd is giving execute permissions

Moving the command to usr/local/bin

Kops is used to manage cluster like creating a master

### 4. Install kubectl -kubectl is to interact with cluster (lets say I wan t to create 5 replicas )

curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl

chmod +x ./kubectl

sudo mv ./kubectl /usr/local/bin/kubectl

moving my binary for above cmd

### 5. Create S3 bucket in AWS

S3 bucket is used by kubernetes to persist cluster state, lets create s3 bucket using aws cli **Note:** Make sure you choose bucket name that is uniqe accross all aws accounts

aws s3 mb s3://javahome.in.k8s --region ap-south-1

kops stores kuber state – how many masters how many nodes, how many pods are running ,how many replicas, all clustrere state is maintained in s3

we cant use same bucket name

### 6. Create private hosted zone in AWS Route53

1. Head over to aws Route53 and create hostedzone
2. Choose name for example (javahome.in)
3. Choose type as private hosted zone for VPC
4. Select default vpc in the region you are setting up your cluster
5. Hit create

### 7 Configure environment variables.

Open .bashrc file

vi ~/.bashrc

Add following content into .bashrc, you can choose any arbitary name for cluster and make sure buck name matches the one you created in previous step.

export KOPS\_CLUSTER\_NAME=javahome.in

export KOPS\_STATE\_STORE=s3://javahome.in.k8s

Then running command to reflect variables added to .bashrc

source ~/.bashrc

### 8. Create ssh key pair – used for connecting to cluster

This keypair is used for ssh into kubernetes cluster

ssh-keygen

(leave all defaults enter -enter)

### 9. Create a Kubernetes cluster definition.

kops create cluster \

--state=${KOPS\_STATE\_STORE} \

--node-count=2 \

--master-size=t2.micro \

--node-size=t2.micro \

--zones=ap-south-1a,ap-south-1b \

--name=${KOPS\_CLUSTER\_NAME} \

--dns private \

--master-count 1

### 10. Create kubernetes cluster

kops update cluster --yes

Above command may take some time to create the required infrastructure resources on AWS. Execute the validate command to check its status and wait until the cluster becomes ready

kops validate cluster

For the above above command, you might see validation failed error initially when you create cluster and it is expected behaviour, you have to wait for some more time and check again.

### 11. To connect to the master

ssh admin@api.javahome.in

# Destroy the kubernetes cluster

kops delete cluster --yes

# Optional (Create terraform scripts through kops)

https://github.com/kubernetes/kops/blob/master/docs/terraform.md

**printenv- to display environment variables**

* **POD:**

POD Is atomic unit of deployment in Kubernetes Cluster – we minimum needs a POD to deploy our application in to Kubernetes

In a POD we can have one or more containers

Most commonly we use one container in a POD in exceptional cases we might need 2 containers which works together.

Every pod gets an ip allocated by Kubernetes , this ip is private within a cluster

Containers in a same POD can communicate through local host.

**Deploying applications in Kubernetes Cluster:**

* Weneed create a POD 1st

**-**  we can do in 2 ways

Using command line

Using yaml document

<https://github.com/javahometech/kubernetes> - refer this doc

<https://github.com/javahometech/kubernetes/tree/master/pods>

kubectl create -f (pod.ymlfile) -created POD

kubecl get pods - to check pod is created or not

we can’t acess from internet , pod is having ip , ip is recognised inside the cluster but not outside

I am outside the cluster I want to test

Kubectl describe pod/nodeapp

Ssh [admin@api.javahome.in](mailto:admin@api.javahome.in)

I can ping pod now

Curl http://pasteip -- curl is used to send http request

Exposing pods to internet:

By default Pods run in a isolated environment i.e. they are reachable within kubernetes cluster, if you wanna reach your pod outside cluster, You have to expose it

$ kubectl expose pods/nodeapp --type="NodePort" --port 8080

This command creates service object, to that service object a pod is a target

Service object also has its own IP , it is a private

# CONTINATION AVAILABLE

# Kubernetes Pods – Commands

## Creating a Pod

### Create pod.yml with following content

Get the file [(pod.yml)](https://github.com/javahometech/kubernetes/blob/master/pods/pods.yml)

apiVersion: v1

kind: Pod

metadata:

name: nodeapp

labels:

app: nodeapp

spec:

containers:

- name: nodeapp

image: kammana/nodeapp:v1

ports:

- containerPort: 8080

$ kubectl create -f https://raw.githubusercontent.com/javahometech/kubernetes/master/pods/pods.yml

### Command to get all pods

$ kubectl get pods

### Command to describe pod details

**Syntax** - kubectl describe pods/POD\_NAME

$ kubectl describe pods/nodeapp

### Executing commands on pods

**Syntax** - kubectl exec POD\_NAME CMD\_TO\_EXECUTE

$ kubectl exec nodeapp printenv

### Getting into pods Terminal

**Syntax** - kubectl exec -it POD\_NAME bash

$ kubectl exec -it nodeapp bash

### Get logs from pods

**Syntax** - kubectl logs POD\_NAME

$ kubectl logs nodeapp

## Exposing Pods to Internet

By default Pods run in a isolated environment i.e. they are reachable within kubernetes cluster, if you wanna reach your pod outside cluster, You have to expose it

$ kubectl expose pods/nodeapp --type="NodePort" --port 8080

**Continuation from HERE**

# Refer fig 2

# Refer fig 3

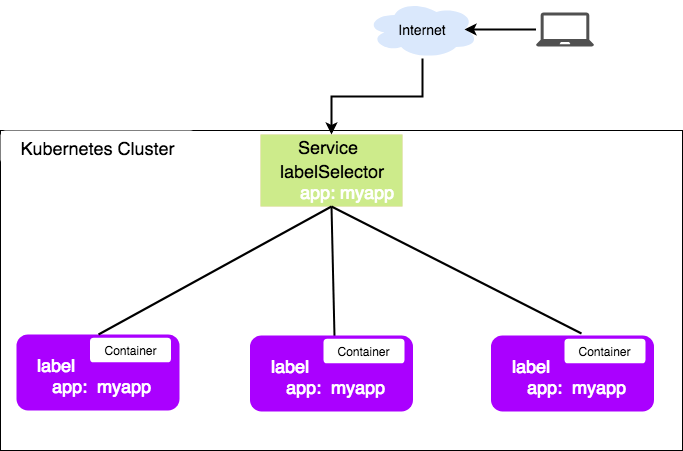
# Service also does load balancing

# How your service object identifies its targets(PODS) ?

# It is done using labels and label selectors

# Kubernetes Service Objects

* In kubernetes service is an object that is logically mapped to pods based on labels.
* Service can be single point of contact like loadbalancer for set of pods.
* By default Pod is not expose to the internet or outside of the cluster, by using service Pod is exposed to internet.
* Any Pod create in future is added to service dynamicaly if pod label is matching with Service lable selector.

[](https://github.com/javahometech/kubernetes/blob/master/images/service.png)

**09-01-2020**

Wget <https://raw.githubusercontent.com/javahometech/kubernetes/master/services/services.yml> (pastelink yml file)

Vi services.yml

kind: Service

apiVersion: v1

metadata:

name: nodeapp

spec:

selector:

app: nodeapp

ports:

- protocol: TCP

port: 80

targetPort: 8080

node port: 3030

type: NodePort

Kubectl create -f services.yml

It created a service

Kubectl get service nodeapp ( nodeapp is service name)

Cat services.yml

Kubectl describe service nodeapp

Kubectl get pods - it shows container creating if not yet created.

Vi pods.yml

kind: Service

apiVersion: v1

metadata:

name: nodeapp2

spec:

selector:

app: nodeapp

ports:

- protocol: TCP

port: 80

targetPort: 8080

node port: 3030

type: NodePort

above we are just changing name

Kubectl create -f pods.yml

(Kubectl apply -f pods.yml (to update)

Kubectl get pods

Kubectl describe service nodeapp

**Ssh** [**admin@api.javahome.in**](mailto:admin@api.javahome.in)

Kubectl get service nodeapp

Curl <http://100.65.139.144:80>

(Request Is going to node , node is routing traffic to service , service is routing traffic to pod)

**Service types:**

**Node port:-**

This allocates a port on every node in the cluster which is used by internet to access our service

**Cluster IP:-**

This will not expose service to the internet, it keeps your service private to the cluster.

Ex:- we want to attach mongodb pod to the service , this service should not be exposed to internet.

**Load balancer:**

This work only on cloud, when we choose this option it creates cloud native loadbalancer

kind: Service

apiVersion: v1

metadata:

name: nodeapp2

spec:

selector:

app: nodeapp

ports:

- protocol: TCP

port: 80

targetPort: 8080

node port: 3030

type: Loadbalancer

(Kubectl apply -f services.yml (to update)

kubectl get service nodeapp

* It shows loadbalancer

**External Name:**

We can configure our own custom names to the DNS.

**Replication Controller:**

Replication controller is kubernets resource, which is used for maintaining n number of replicas of specific pod alays oin the cluster

If we choose 5 rplicas it creates 5 PODS, If it finds more than 5 it deletes extra PODS.

If it finds less than 5 creates new PODS

**NOTE:**

* We should not directly deploy pods, it is not permitted
* There should be replication controller

Kubectl get pods

Kubectl delete pods nodeapp

**Replica set:**

There are 2 types of replicaset

<https://github.com/javahometech/kubernetes/tree/master/replicaset>

* The major difference is replica set supports set based selector.

For Ex:

matching expressions:

{key: env, operator: In, values: [“dev”,”prod”]}

Selector:

env: dev

env: dev

**Deployements:**

## (Deployments

It is a kubernetes object which the declarative updates for Pods and ReplicaSets

### Create Deployment

kubectl create -f https://raw.githubusercontent.com/javahometech/kubernetes/master/deployments/deployments.yml --record=true

### Check status of the current deployment

kubectl rollout status deployment nodeappdeployment

### Updating deployment

For example we want to change number of replicas, change replicas in yaml and run following command

kubectl apply -f https://raw.githubusercontent.com/javahometech/kubernetes/master/deployments/deployments.yml --record=true

### Kubernetes Deployment revisions

Kubernetes maintains deployment state of all versions

inorder to see deployment revision history

kubectl rollout history deployment nodeappdeployment

### Undo recent deployment

kubectl rollout undo deployment nodeappdeployment

### rollback to specific deployment revision

kubectl rollout undo deployment nodeappdeployment --to-revision=1

)

**Deployements:**

* **It**  provides additional functionality like rolling updates (roll backs)
* Deployement is a wrapper on replicas set

Undo current deployement in Kubernetes:

* kubectl rollout undo deployment nodeapp ( in place of nodeapp whatever the name u given )

To Rollback to specific version:

kubectl rollout undo deployment nodeappdeployment --to-revision=1

)

**IQ) How do you implement rolling updates in kubernetes:**

Using deployement object, it has builtin rolling update

**IQ) Tell me how you can do 0 down time.**

Rolling update –

Blue-green deployement

Caneary deployment

Refer: <https://github.com/javahometech/kubernetes/tree/master/deployments>

22-1-20

IQ) In Kubernetes how do you manage secrets like passwords, api tockens or any sensitive data (v.imp)

* we have to use Kubernetes secret object, it’s the object where you can store sensitive data

Ex: let’s say you have private docker image in docker hub we want to deploy that in to Kubernetes which requires username and passworod of dockerhub.

We wanna use secret object to store dockerhub credentials.

## 1. Pulling private images using Secrets

Create kubernetes secrets

kubectl create secret docker-registry regcred --docker-server=https://index.docker.io/v1/ --docker-username=kammana --docker-password=<your-password> --docker-email=hari.kammana@gmail.com

After create secret, lets create pod which uses private images.

apiVersion: v1

kind: Pod

metadata:

name: private-reg

spec:

containers:

- name: privateapp

image: kammana/privateapp:0.0.1

imagePullSecrets:

- name: regcred

When I run the above code it reads the secret object from the regcred and it works.

Create pod which uses private image

docker create -f https://raw.githubusercontent.com/javahometech/kubernetes/master/secrets/private-images-pod.yml

**VOLUMES:** <https://kubernetes.io/docs/concepts/storage/volumes/>

On-disk files in a Container are ephemeral, which presents some problems for non-trivial applications when running in Containers. First, when a Container crashes, kubelet will restart it, but the files will be lost - the Container starts with a clean state.

Kubernets deals with diff types of volumes depending on where we are running our cluster

For ex : it has integration with aws,azure,GCP.

EmptyDIR:

This volume is created when pod is created it is removed when a POD is terminated.

It is useful for containers in same POD to exchange data

HOST PATH:

A host path volume mounts a file or directory from the host nodes filesystem in to your POD.

**Persistent volume Claim:**

Persistent volume is a volume created by Kubernetes admins.

Persistent volume claim is requesting a portion of volume for our use.

Finaly Persistent volume client is used inside the pod.

**Service discovery:**

In microservices architecture we will deal with many microservices where one microservice wanna talk to another microservice.

The process of one microservice locating another microservice is called as service discovery.

1. By default one service talks to another service using its service name.

IQ) How can we constrain a POD to be scheduled only on specific nodes

There are diff ways to achieve this

* Node selector:
* Affinity & anti affinity:
* Taints & toleratons: -- explore om this

IQ)Which tool are you using for packaging & deploying Kubernetes resources.

* HELM

Helm helps you manage Kubernetes applications – helm charts helps you define, install and upgrade even the most complex kubernetes application

It is written in golang – google scripting

we create Kubernetes resources through yaml which are static files

in real time every time we deploy/upgrade an application we want to change certain information in yaml file dynamically. That behaviour is provided by HELM.

HELM allows us to embide yaml wth golag templates to mke our yaml fiels dynamic

(We need to install Helm to interact with Kubernetes cluster)

# Helm Kubernetes

Helm is for packaging and deploying kubernetes applications

### Install

curl -LO https://git.io/get\_helm.sh

chmod 700 get\_helm.sh

./get\_helm.sh

### To intial helm local repository

helm init

- initialize helm

### To create helm chart

helm create nodeapp-helm

- cmd to create nodeapp

### To create helm package

helm package nodeapp-helm

### To deploy helm package

helm install nodeapp-helm

### To deploy helm package with custom release name

If you do not mention name, helm chooses random names for release, if you want to have custom name for a relase, run the following command

helm install --name my-release nodeapp-helm

## NOTE - If you get error as follows when you run above command

configmaps is forbidden: User "system:serviceaccount:kube-system:default" cannot list resource "configmaps" in API group "" in the namespace "kube-system"

### deploy tiller service account RBAC to avoid above error

kubectl --namespace kube-system create serviceaccount tiller

kubectl create clusterrolebinding tiller-cluster-rule --clusterrole=cluster-admin --serviceaccount=kube-system:tiller

kubectl --namespace kube-system patch deploy tiller-deploy -p '{"spec":{"template":{"spec":{"serviceAccount":"tiller"}}}}'

### Upgrade helm release

helm upgrade --set-string image.tag=v2

helm upgrade --set-string image.tag=v2 nodeapp nodeapp

### To list out helm releases which are in deployed state

helm ls

### To list out helm releases which are deployed and deleted state

helm ls --all

### To delete helm release

To delete release completely run following two commands

helm delete helm-release-name

helm del --purge helm-release-name

**Helm Charts : refer notes:**

**Helm is a package manager for Kubernetes**

after packing u can store the package in nexus or private artifatorys

tiller is a pod which runs inside the cluster

is there any alternative for Helm

Instead of Helm ansible is an option to pakage and deploy Kubwernetes resources

**IQ) How do you secure Kubernetes Cluster**

**For Ex: 5 deveops engineer , I want to give access to kube cluster for only one guy**

**Using RBAC ( role based access control) we can configure different users with different levels of permissions.**

**IQ) (Kubernetes ingres controller)**

**KUBERNETES NAME SPACES:**

* Using name spacewe can divide kubernetes cluster into logical work spaces.

Ex: we can have name spaces like UAT, dev, test, production, monitoring etc: