K8SC

INSTALL AWS CLI: curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip" unzip awscliv2.zip sudo ./aws/install yum install vim wget -y vim .bashrc export PATH=$PATH:/usr/local/bin/ source .bashrc aws --version

curl -LO "https://dl.k8s.io/release/$(curl -L -s

https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

wget https://github.com/kubernetes/kops/releases/download/v1.24.1/kops-linux-amd64

wget https://github.com/kubernetes/kops/releases/download/v1.21.1/kops-linux-amd64

m

chmod +x kops-linux-amd64 kubectl mv kops-linux-amd64 /usr/local/bin/kops mv kubectl /usr/local/bin/kubectl aws --version kubectl version kops

======================================

Console -- > uername -- > security credentials -- > Access key -- > Downloads aws configure

aws s3api create-bucket --bucket zomato.k8s.local--region us-east-1

aws s3api put-bucket-versioning --bucket zomato.k8s.local--region us-east-1 --versioning-configuration Status=Enabled

export KOPS\_STATE\_STORE=s3://zomato.k8s.local

ssh-keygen

kops create secret sshpublickey admin -i ~/.ssh/id\_rsa.pub --name zomato.k8s.local--state s3://bhavuk.k8s.local

kops create cluster --name zomato.k8s.local--zones us-east-1a --master-size t2.medium --node-size t2.micro

# no use: kops edit cluster --name bhavuk.k8s.local

kops update cluster --name zomato.k8s.local--yes --admin

kubectl get nodes

kops validate cluster --name zomato.k8s.localinitially it will fail then later use same command after sometime it will be ready

kubectl get nodes

EXTRA IF FAILS: kops export kubecfg

PODS:

pods is the smallest unit of deployment to don in Kubernetes. K8s cant manager containers directly if u add some metadata to your running container then its a pod within a pod we can have multiple containers. but mostly we have one container in one pod

only very few cases we have multiple containers in a pod if u have a wokernode of 2 gb of ram then u can get 2 pods of 1 gb ram

REPLICASET:

it is nothing but Group of pods if one pod crashes then by using replicaset another pod is recreated automatically.

DEPLOYMENT:

It has features of Replicaset and some other extra features

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ kubectl config view : To show config settings kubectl config get-context :

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

.kube dir: used to interact with one node to another node

here we can find the dir inside this dir there is a config for a particular user to authenticate kubeapi server. It is also on master node as well by using this we can interact with Kube API.

We can do all activites from the main node created initially to created the cluster.

We are going to create a few resources so for that we can use my predefined templates.

Yum install git docker -y ( if u don’t install docker also u can do things on pod level not on container level)

Git clone <https://github.com/rhavukm/k8s.git>

Cd k8s

Pods :

pods is the smallest unit of deployment to don in Kubernetes.

We can create any resources in 2 ways

1: by running the commands (ex: kubectl run nginx - -image nginx:alpine)

Kubectl get pods : for pods list

Kubectl get pods -o wide : for additional info

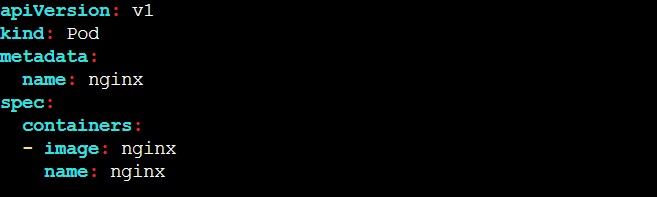
If you want to schedule a pod on particular node in that case we need to create own scheduler.

We use default scheduler it will decide on which node the pod is created.

The above method is not recommended in real time.

For that we use yaml file

2: by using a yaml manifest file



Api: version of Api for creating a pod

Kind : type of service

Name: name of the pod

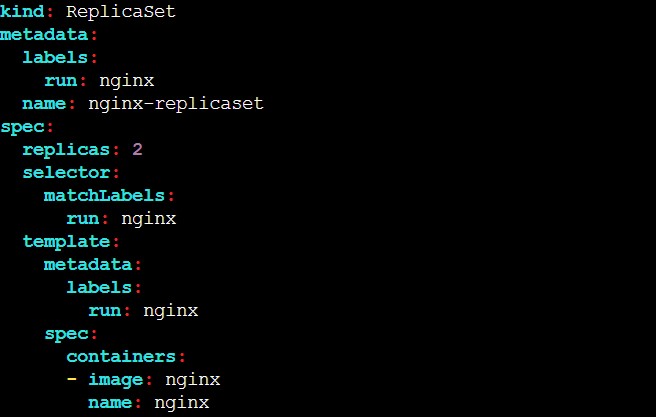
Spec: specification

Image: the image you want to give to container kubectl create -f pod-nginx.yaml : command to create a pod from file

The main dis advantage is if we delete that pod we are not able to access So if we work on the realtime production then we need to use replicaset.

REPLICASET:

it is nothing but Group of pods if one pod crashes then by using replicaset another pod is recreated automatically.



Labels : mandatory to create RS (if u have 100 pods in a cluster we can inform which pods we need to take care by using labels) if u labeled some pods a raham then all the pods with label raham will be managed

Replicas: Number of pod copies we need to create Matchelabel: the label we want to match with pods

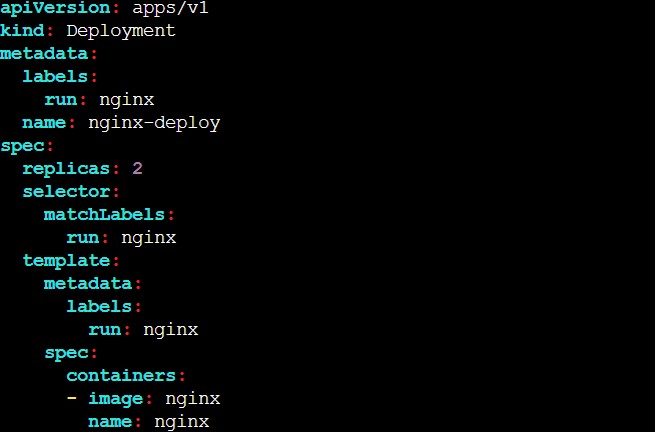
Template: This is the template of pod. kubectl create -f replicaset-nginx.yaml kubectl get replicaset -o wide kubectl get all

Now delete a pod and do list now it will be created automatically

Kubectl delete po nginx-replicaset-f2zkq Kubectl get all or Kubectl get rs kubectl describe pod/nginx-replicaset-g9vcw : to see additional info this command is used to see the events where we can see error log kubectl delete rs nginx-replicaset : to delte replicaset so no pods will create now automatically

DEPLOYMENT:

It has features of Replicaset and some other extra features like updating and rollbacking to a particular version we want without downtime\*.



kubectl create -f deployment-nginx.yaml deployment created replicaset and replicaset create pod here application will reside on pod to manage and update& manage that pod we used deployment kubectl delete rs nginx-deploy-7dc4b48974 kubectl get rs : now u will get replicasets automatically because it is managed by deployment

NAMESPACE:

It is a way to filter out the stuff for different users in a single machine.

Kubectl get ns : we can list name spaces by default it is four

Default : if we don’t specify any namespace the n default is used.

Kubectl get all -n default

Kube-node-lease: It is used for the lease objects associated with each node that improves the performance of the node heartbeats as the cluster scales.

Kube-public: To create any object that is publicly available to users.

Kube-system: Kubernetes uses this for creating its own objects.

kubectl get all -n kube-system kubectl create ns raham kubectl get ns kubectl config set-context $(kubectl config current-context) --namespace=raham to change the resources from to desired name space kubectl get all

Now lets create a new pod and lets see on which name space it will go kubectl create -f pod-nginx.yaml kubectl describe pod nginx | grep -i namespace

if you are cluster admin we can get different namespaces for different users then you can create different namespaces for different users.

GETTING INSIDE CLUSTER USING SSH:

kops create secret sshpublickey admin -i ~/.ssh/id\_rsa.pub --name zomato.k8s.local--state s3://zomato.k8s.localwe need to update the cluster now kops update cluster rahammc.k8s.local –yes kops rolling-update cluster : optional now we have attached a key to connect to my master node to access we need to use check from 50: minutes on part-2 kops get cluster kops delete cluster --name=rahammc.k8s.local --yes

specific k8s version:

kops edit cluster --name zomato.k8s.local#edit the Kubernetes version & save and exit kops update cluster --name zomato.k8s.local--yes --admin

kops validate cluster --name bhavuk.k8s.local

latest k8s version:

kops upgrade cluster --name zomato.k8s.local--yes kops update cluster --name zomato.k8s.local--yes --admin kops validate cluster --name bhavuk.k8s.local

delete k8s cluster: kops delete cluster --name zomato.k8s.local--yes

https://github.com/bhavukm/k8s git clone https://github.com/bhavukm/k8s To SSH to Master and Worker Nodes in KOPS:

kops update cluster zomato.k8s.local--yes kops rolling-update cluster