**Kubernets cluster setup with KOPS in AWS**

***Update the ubuntu with the following which will refresh the repository***

apt-get update

***Install the awscli package in local VM or AES VM***

apt-get install awscli

***Create the aws user with full access and Copy the ACCESS KEY and SECRET KEY and keep it in not pad.***

***Configure the credentials in VM.***

aws configure

***Download the kubectl package***

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

***Download the kops package***

wget https://github.com/kubernetes/kops/releases/download/1.6.1/kops-linux-amd64

chmod +x kops-linux-amd64

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

***Create the S3 bucket to store the cluster state and configuration files and export it.***

export KOPS\_STATE\_STORE=s3://mykishtest

***Generate RSA key to login to the Kubernetes Cluster from VM and the create the secret.***

ssh-keygen -t rsa

kops create secret --name my.microboat.co.in sshpublickey admin -i /root/.ssh/id\_rsa.pub --state=s3://mykishtest

***Create the Cluster with KOPS command without --yes option***

kops create cluster --name=my.microboat.co.in --state=s3://mykishtest --zones=ap-south-1a --node-count=1 --node-size=t2.medium --master-size=t2.medium --master-count=1 --dns-zone=microboat.co.in

***Edit/update the cluster configuration with following parameters, add the lines after "Spec: " with proper indentation***

kops edit cluster my.microboat.co.in

***add these lines after Spec:***

kubeAPIServer:

admissionControl:

- NamespaceLifecycle

- LimitRanger

- ServiceAccount

- PersistentVolumeLabel

- DefaultStorageClass

- DefaultTolerationSeconds

- MutatingAdmissionWebhook

- ValidatingAdmissionWebhook

- ResourceQuota

- NodeRestriction

- Priority

***Update/create the Cluster now and*** Wait 10 minutes to come up the cluster .

kops update cluster my.microboat.co.in --yes

kops validate cluster

kubectl get nodes

kubectl get nodes --show-labels

kubectl get pods --all-namespaces

***Login to master node to check kubernetes cluster login.***

ssh -i ~/.ssh/id\_rsa admin@api.my.microboat.co.in

***Test with kuberbetes dash board is it working or not.***

wget https://raw.githubusercontent.com/kubernetes/dashboard/v1.10.1/src/deploy/alternative/kubernetes-dashboard.yaml

vi kubernetes-dashboard.yaml

kubectl apply -f kubernetes-dashboard.yaml

***Install the istio now.***

wget https://github.com/istio/istio/releases/download/1.0.2/istio-1.0.2-linux.tar.gz

tar -xzvf istio-1.0.2-linux.tar.gz

cd istio-1.0.2/

export PATH=$PWD/bin:$PATH

istioctl

cd istio-1.0.2/

kubectl apply -f install/kubernetes/helm/istio/templates/crds.yaml

kubectl get crds

kubectl get pods --all-namespaces

kubectl apply -f install/kubernetes/istio-demo.yaml

kubectl get pods --all-namespaces

***Clone the following repo for sample applications.***

git clone https://github.com/kishoredpt/kubernetes-course.git

cd kubernetes-course/istio

kubectl apply -f <(istioctl kube-inject -f helloworld.yaml)

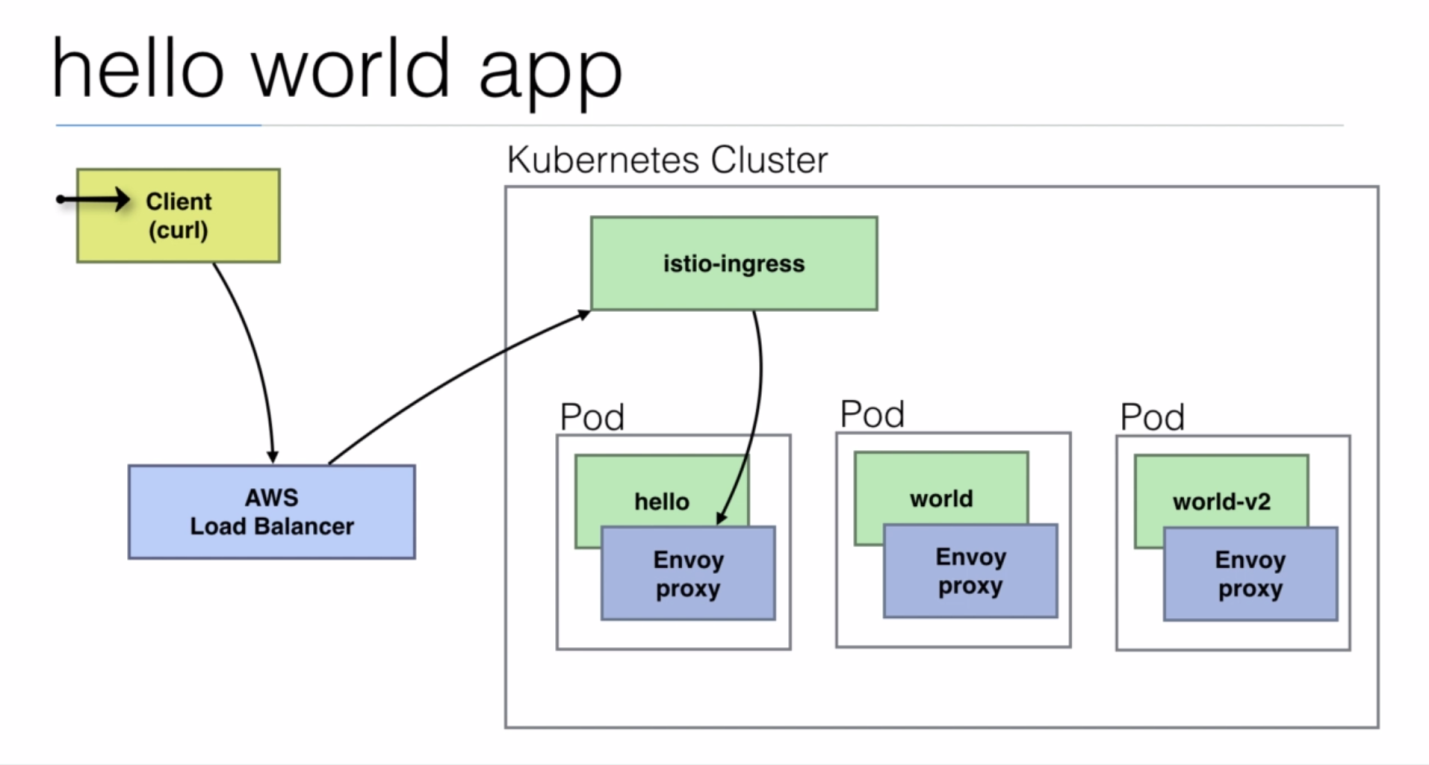
kubectl apply -f helloworld-gw.yaml

kubectl get pods

kubectl get svc --all-namespaces

curl http://aecab95ed748511e9b0bb0200ca7af54-2113117283.ap-south-1.elb.amazonaws.com/hello

curl http://aecab95ed748511e9b0bb0200ca7af54-2113117283.ap-south-1.elb.amazonaws.com/



***Edit the following services to access it from public browser.***

***Change type from "*ClusterIP*" to "*Load Balancer*"***

kubectl get svc --all-namespace -o wide

kubectl edit svc jaeger-query -n istio-system

kubectl edit svc grafana-n istio-system

kubectl edit svc prometheus -n istio-system

Copy the aws laod balancer link from service and access it from browser.

Github Links

https://github.com/kishoredpt/kubernetes-course/tree/master/istio

https://github.com/kishoredpt/kubernetes-course