* There is three way to use EKS in aws.

1. webUI
2. CLI
   1. Aws eks
   2. Eksctl
      1. This command provides some extra services.
3. API (terraform)

**Eksctl**

🡪Download it from google

🡪Copy in minikube folder (You can copy anywhere).

🡪Add to the path variable.

🡪Create IAm with admin power for using this.

🡪why to use eksctl?

🡪 aws eks command don’t have so much option.

🡪using this command, you cannot do lot of customization.

🡪suppose you need 2 t2.micro and 1 t2.small for worker node.This thing you can’t do using aws eks command.

🡪For doing this setup you have to tell how many nodes and how many nodegroup you need.

🡪node group is group of same nodes.

🡪One node group for 2 t2.micro, one group for 1 t2.small

🡪notepad “cluster.yml”

🡪eksctl create cluster -f cluster.yml

🡪They will automatically launch complete cluster for you.

🡪eksctl internally creating CloudFormation program.

🡪This program launch cluster.

🡪CloudFormation program is known as stack.

🡪In aws you want to automate anything you have to use cloud formation.

🡪This entire cluster launched in mumbai.

🡪Here eksctl try to launch 3 worker nodes in 3 different datacenter.

🡪For local connect we have to use kubectl.

🡪Here aws fully manage master node.

🡪So, you don’t have to worry about disaster.

🡪They will launch master in multiple AZ.

🡪aws eks update-kubeconfig --name akcluster

🡪This command will update the file and if there is no file then this command create config file for you.

🡪In your root folder, inside .kube file.

🡪Now you can use Kubectl from local system or any other system.

🡪aws get nodes

🡪kubectl describe nodes <id>

🡪kubectl create namespace akshit

🡪kubectl get ns

🡪This provides different and isolated environment to different teams.

🡪kubectl config set-context --current --namespace=akshit

🡪Now namespace changed to akshit

🡪Kubectl create deployment myweb --image=vimal13/apache-webserver-php

🡪Kubectl scale deployment myweb --replicas=3

🡪Now it will launch three pods.

🡪we use deployment so whenever any pod fail, it will create one more pod.

🡪If you use load balancer service and LoadBalancer type they know they have external load balancer ELB.

🡪kubectl expose deployment <myweb> --type=LoadBalancer --port=80.

🡪as soon as you run this command one ELB launch for you.

🡪ELB can do two things.

🡪Do the load balancing.

🡪Provide public IP.

🡪Here you don’t have to configure ELB.

🡪Here internally when you connect using public id then first it connects to ELB, ELB connect to the kubenetes (EKS) LoadBalancer and then connect to pods.

🡪kubectl describe service/myweb

🡪kubectl cp index.php <myweb-89ssdbnmdbz:/var/www/html/index.php>

🡪For using persistent storage, we have to use kubernetes PVC concept.

🡪Now a days we use dynamic PVC.

🡪storage class is already created by EKS.

🡪It by default takes storage from EBS.

🡪By default, created SC name is gp2 (Volume type).

🡪notepad pvc.yml

🡪kubectl create -f pvc.yml

🡪Here pvc created.

🡪Here as soon as pod request for PVC, PV created.

🡪Here SC is created by EKS and it is mentioned in EKS that volumeBindingMode=waitForFirstConsumer so, whenever pod request for PVC then only PV created.

🡪Consumer=pod

🡪kubectl get pv

🡪To check PV is not created.

🡪kubectl edit deployment <myweb>

🡪Edit command on the fly edit content of working pod.

🡪This will open yml file for you.

🡪Below spec, inside spec-part add this.

🡪volumes:

* name: web-vol1

persistentVolumeClaim:

claimName: akpvc

🡪In containers part, below image add this.

🡪volumeMounts:

* mountPath: /var/www/html

name: web-vol1

🡪save this file, you edit the file so, pods are recreated.

🡪this time PV created.

🡪kubectl get pv

🡪kubectl get sc

🡪Here, volume type = gp2, you can also create your own SC file.

🡪Here reclaimPolicy= DELETE

🡪So as soon as you delete PVC deleted, PV delete.

🡪and this deleted PV contact to SC and delete hard disk (EBS) also.

🡪While creating PVC we do not write storage class so by default it uses default SC.

🡪notepad “sc.yml”

🡪kubectl create –f sc.yml

🡪kubectl create –f pvc2.yml

🡪Here storage class mentioned,reclaimpolicy=retain.

🡪Here you don’t have use volumeBindingMode so as soon as you create pvc, pv created

#🡪kubectl edit sc gp2

# 🡪Copy the annotation part

#🡪kubectl edit sc aksc

# 🡪paste here inside metadata. You can also make this default, and #change whatever you want

#You can change this setting, but we want so I have comment it.

🡪eksctl delete cluster –f cluster.yml

* aws eks help
* Aws eks list-clusters
* Aws eks create-cluster

🡪This command don’t have so much option.

🡪using this command you can not do lot of customization.

* Eksctl version

🡪It behind the seen use aws login details.

🡪To check this use aws configure command.

* Eksctl get cluster
* eksctl create cluster -f cluster.yml
* aws eks update-kubeconfig --name akcluster

🡪This will automatically update config file for Kubectl.

🡺 kubectl create namespace akshit

🡺kubectl get ns

* Kubectl create deployment myweb --image=vimal13/apache-webserver-php
* Kubectl get pods –o wide

🡪This gives some extra details – where node is running.

* Kubectl scale deployment myweb --replicas=3
* kubectl expose deployment <myweb> --type=LoadBalancer --port=80
* kubectl describe service/myweb
* kubectl cp index.php <myweb-89ssdbnmdbz:/var/www/html/index.php>
* kubectl get sc
* kubectl edit sc aksc
* eksctl delete cluster –f cluster.yml