

Q) * Use case of Deploying a website on AWS using EKS (Elastic K8s Service Cluster) only

- Steps:-

- 1) Create an EKS cluster! -

- (i) EKS is acts as container manager to run our website.

- (ii) To create cluster using command -

```
eksctl create cluster -
```

- This will take 10-15 min to set up.

- 2) Deploy website in k8s cluster! -

- (i) We will use "docker" to package your website into a container.

- (ii) Then we used to "k8s deployment" to run website.

• Create a website deployment yaml file

Then apply this file using command

"kubectl apply -f website-deployment.yaml"

3) Expose our website over the internet:-

(i) To allow our website accessible over the internet then used "loadbalancer".

(ii) Create service (loadbalancer) to make website accessible

(iii) Create service.yaml file

(iv) Apply file with kubectl apply -f service.yaml
website-service.yaml

(4) check service if running or not:-

check by "kubectl get service"

* copy the External-IP and open it into a new browser!

then Your "E-commerce website" is now live

website-deployment.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

name: ecommerce-website

Spec:

replicas: 3 # run 3 copies for reliability.

Selector:

matchLabels:

app: ecommerce

template:

metadata:

labels:

app: ecommerce

Spec:

containers:

- name: ecommerce-container

image: my-docker-hub/ecommerce-app:v1

ports:

- containerPort: 80

* @website-service.yaml:-

apiVersion: v1

kind: service

metadata:

name: ecommerce-service

spec:

type: LoadBalancer.

Selector:

app: ecommerce

ports:

- protocol: TCP

port: 80

targetPort: 80