

INDEX

Tasks	Page No
Task:1 – Launched an Instance	02
Task:2 – Installing Kubernetes	02
Task:3 – Install Minikube	03
Task:4 – Cloning repository	04
Task:5 – Creating Dockerfile	05
Task:6 - Building the image	05
Task:7 - Pushing image on dockerhub	06
Task:8 - Create Apache2 deployment	07
Task:9 - Create ingress deployment	08
Task:10 - Forwarded Traffic	09
Task:11 - Create ingress customimage deployment	10
Task:12 - Forwarded Traffic	12

CASE STUDY - INTRODUCTION TO KUBERNETES

You have just joined a startup Ventura Software as a Devops Lead Engineer. The company relies on a Monolithic Architecture for its product. Recently, the senior management was hired. The new CTO insists on having a Microservice Architecture. The Development Team, is working on breaking the Monolith. Meanwhile, you have been asked to host a Test Application on Kubernetes, to understand how it works.

Following things have to be implemented:

1. Deploy an Apache2 deployment of 2 replicas
2. Sample code has been checked-in at the following Git-Hub repo;

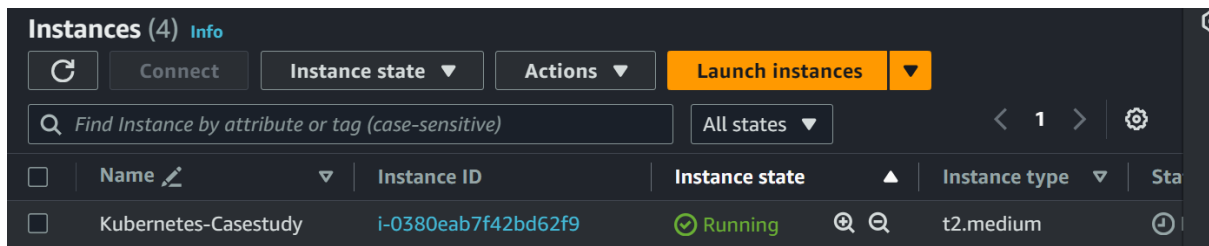
<https://github.com/hshar/website.git>.

You have to containerize this code, and push it to Docker Hub. Once done, deploy it on Kubernetes with 2 replicas

3. Deploy Ingress with the following rules:

- i) */apache* should point to the apache pods
 - ii) */custom* should point to the GitHub application
-

Task:1 – Launched an Instance.



Task:2 – Installing Kubernetes.

1. Now we will create a shell script to install the Kubernetes clusters.

```
sudo su –
```

```
vi kubernetes_install.sh
```

```
chmod +x kubernetes_install.sh
```

```
./kubernetes_install.sh
```

Command enters in the Kubernetes_install.sh

```
sudo apt-get update
```

```
sudo apt install docker.io -y
```

```
sudo apt-get install -y apt-transport-https ca-certificates curl gpg
```

```
sudo mkdir -p -m 755 /etc/apt/keyrings
```

```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
```

```
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
```

```
https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
```

```
sudo apt-get update
```

```
sudo apt-get install -y kubelet kubeadm kubectl
```

```
sudo systemctl enable --now Kubelet
```

```
ubuntu@ip-172-31-13-36:~$ sudo su -
root@ip-172-31-13-36:~# vi kubernetes_install.sh
root@ip-172-31-13-36:~# cat kubernetes_install.sh
sudo apt-get update
sudo apt install docker.io -y
sudo apt-get install -y apt-transport-https ca-certificates curl gpg
sudo mkdir -p -m 755 /etc/apt/keyrings
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ .' | sudo tee /etc/apt/sources.list.d/kubernetes.list
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo systemctl enable --now Kubelet
root@ip-172-31-13-36:~#
```

i-0380eab7f42bd62f9 (Kubernetes-Casestudy)

PublicIPs: 3.109.56.200 PrivateIPs: 172.31.13.36

```
root@ip-172-31-13-36:~# chmod +x kubernetes_install.sh
root@ip-172-31-13-36:~# ./kubernetes_install.sh
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 c-n-f Metadata [265 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [144 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/multiverse Translation-en [104 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 c-n-f Metadata [9136 B]
```

i-0380eab7f42bd62f9 (Kubernetes-Casestudy)

PublicIPs: 3.109.56.200 PrivateIPs: 172.31.13.36

Task:3 – Install Minikube.

sudo su -

vi install.sh

chmod +x install.sh

./install.sh

Command enters in the install.sh file.

sudo apt update

sudo apt install docker.io

curl -LO

https://storage.googleapis.com/minikube/releases/latest/minikube_latest_amd64.deb

sudo dpkg -i minikube_latest_amd64.deb

sudo chmod 777 /var/run/docker.sock

minikube start

sudo snap install kubectl --classic

minikube addons enable ingress

minikube start

```
ubuntu@ip-172-31-13-36:~$ minikube start
* minikube v1.33.0 on Ubuntu 20.04 (xen/amd64)
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
```

i-0380eab7f42bd62f9 (Kubernetes-Casestudy)

PublicIPs: 3.110.124.29 PrivateIPs: 172.31.13.36

minikube status

```
ubuntu@ip-172-31-13-36:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

```
ubuntu@ip-172-31-13-36:~$ █
```

i-0380eab7f42bd62f9 (Kubernetes-Casestudy)

PublicIPs: 3.110.124.29 PrivateIPs: 172.31.13.36

Task:4 – Cloning repository.

```
root@ip-172-31-13-36:~# git clone https://github.com/hshar/website.git
Cloning into 'website'...
remote: Enumerating objects: 8, done.
remote: Total 8 (delta 0), reused 0 (delta 0), pack-reused 8
Unpacking objects: 100% (8/8), 82.67 KiB | 6.89 MiB/s, done.
root@ip-172-31-13-36:~# ls
kubernetes_install.sh  snap  website
root@ip-172-31-13-36:~# cd website
root@ip-172-31-13-36:~/website# ls
images  index.html
root@ip-172-31-13-36:~/website# cat index.html
<html>
```

i-0380eab7f42bd62f9 (Kubernetes-Casestudy)

PublicIPs: 3.109.56.200 PrivateIPs: 172.31.13.36

```
root@ip-172-31-13-36:~/website# cat index.html
<html>
<head>
<title> Intellipaat </title>
</head>
<body style = "background-image:url('images/github3.jpg'); background-size: 100%">
<h2 ALIGN=CENTER>Hello world!</h2>
</body>
</html>
root@ip-172-31-13-36:~/website#
```

i-0380eab7f42bd62f9 (Kubernetes-Casestudy)

PublicIPs: 3.109.56.200 PrivateIPs: 172.31.13.36

Task:5 – Creating Dockerfile.

```
root@ip-172-31-13-36:~/website# sudo nano Dockerfile
root@ip-172-31-13-36:~/website# cat Dockerfile
FROM ubuntu
RUN apt-get update
RUN DEBIAN_FRONTEND="noninteractive" apt-get install tzdata -y
RUN apt-get install apache2 -y
ADD . /var/www/html/
ENTRYPOINT apachectl -D FOREGROUND
root@ip-172-31-13-36:~/website#
```

i-0380eab7f42bd62f9 (Kubernetes-Casestudy)

PublicIPs: 3.109.56.200 PrivateIPs: 172.31.13.36

Task 6: Building the image.

docker build -t singh318/casestudy .

```
Running hooks in /etc/ca-certificates/update.d...
done.
Removing intermediate container 7423109b3a93
---> ecc39adbc0be
Step 5/6 : ADD . /var/www/html/
---> 77774c6f98b8
Step 6/6 : ENTRYPOINT apachectl -D FOREGROUND
---> Running in 20f0ff8cce07
Removing intermediate container 20f0ff8cce07
---> 02625c793254
Successfully built 02625c793254
Successfully tagged singh318/casestudy:latest
root@ip-172-31-13-36:~/website#
```

i-0380eab7f42bd62f9 (Kubernetes-Casestudy)

PublicIPs: 3.110.124.29 PrivateIPs: 172.31.13.36

```
root@ip-172-31-13-36:~/website# sudo docker images
REPOSITORY          TAG          IMAGE ID          CREATED           SIZE
singh318/casestudy   latest       02625c793254      About a minute ago 224MB
ubuntu               latest       de52d803b224      4 days ago       76.2MB
root@ip-172-31-13-36:~/website#
```

i-0380eab7f42bd62f9 (Kubernetes-Casestudy)

PublicIPs: 3.110.124.29 PrivateIPs: 172.31.13.36

Task 7: Pushing image on dockerhub.

sudo docker login

```
root@ip-172-31-13-36:~/website# sudo docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: singh318
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@ip-172-31-13-36:~/website#
```

i-0380eab7f42bd62f9 (Kubernetes-Casestudy)

PublicIPs: 3.110.124.29 PrivateIPs: 172.31.13.36

sudo docker push singh318/casestudy

```
Login Succeeded
root@ip-172-31-13-36:~/website# sudo docker push singh318/casestudy
Using default tag: latest
The push refers to repository [docker.io/singh318/casestudy]
ae7cce97967f: Pushed
132ce143c630: Pushed
29fa56ee8e87: Pushed
del1f08765fe4: Pushed
3e1ed584ae0e: Mounted from library/ubuntu
latest: digest: sha256:a02ef128e26fa149c3087511ed6f35da71ef5d831f8ad6f82ae33e8ba93bc041 size: 1373
root@ip-172-31-13-36:~/website#
```

i-0380eab7f42bd62f9 (Kubernetes-Casestudy)


PublicIPs: 3.110.124.29 PrivateIPs: 172.31.13.36

https://hub.docker.com/r/singh318/casestudy/tags

docker hub Explore Repositories Organizations

Search: singh318

Explore / singh318/casestudy

 **singh318/casestudy** ☆0

By [singh318](#) • Updated 3 minutes ago

Image

Manage Repository

Pulls 0

Overview **Tags**

Sort by Newest Filter Tags

TAG	Digest	OS/ARCH	Last pull	Compressed Size
latest	a02ef128e26f	linux/amd64	---	86.95 MB

Last pushed 3 minutes ago by [singh318](#)

docker pull singh318/casestudy:latest Copy

Task 8: Create Apache2 deployment.

kubectl create deployment apache2 --image=ubuntu/apache2 --port=80

```
ubuntu@ip-172-31-13-36:~$ kubectl create deployment apache2 --image=ubuntu/apache2 --port=80
deployment.apps/apache2 created
ubuntu@ip-172-31-13-36:~$
```

i-0380eab7f42bd62f9 (Kubernetes-Casestudy)

PublicIPs: 3.110.124.29 PrivateIPs: 172.31.13.36

```
ubuntu@ip-172-31-45-73:~/website$ kubectl get deploy
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
apache2   1/1     1             1           23s
```

```
ubuntu@ip-172-31-45-73:~/website$ kubectl expose deploy apache2 --type=NodePort --name=apache-np
service/apache-np exposed
ubuntu@ip-172-31-45-73:~/website$ kubectl get svc
NAME            TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
apache-np       NodePort    10.107.51.0   <none>        80:31130/TCP     4s
kubernetes      ClusterIP   10.96.0.1     <none>        443/TCP          7m47s
```

Task 9: Create ingress deployment.

```
sudo nano ingress.yaml
```

```
cat ingress.yaml
```

```
apiVersion: networking.k8s.io/v1
```

```
kind: Ingress
```

```
metadata:
```

```
  name: ingress
```

```
  annotations:
```

```
    nginx.ingress.kubernetes.io/rewrite-target: /
```

```
spec:
```

```
  ingressClassName: nginx
```

```
  rules:
```

```
    - http:
```

```
      paths:
```

```
        - path: /nginx
```

```
          pathType: Prefix
```

```
      backend:
```

```
        service:
```

```
          name: nginx
```

```
        port:
```

```
          number: 80
```



```

ubuntu@ip-172-31-45-73:~/website$ sudo nano ingress.yaml
ubuntu@ip-172-31-45-73:~/website$ cat ingress.yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  ingressClassName: nginx
  rules:
  - http:
      paths:
      - path: /apache
        pathType: Prefix
        backend:
          service:
            name: apache2
            port:
              number: 80

```

kubectl apply -f ingress.yaml

```

ubuntu@ip-172-31-45-73:~/website$ kubectl apply -f ingress.yaml
ingress.networking.k8s.io/ingress created

```

kubectl get svc -A

```

ubuntu@ip-172-31-45-73:~/website$ kubectl get svc -A

```

NAMESPACE	NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
default	apache-np	NodePort	10.107.51.0	<none>	80:31130/TCP	2m40s
default	kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	10m
ingress-nginx	ingress-nginx-controller	NodePort	10.107.7.81	<none>	80:32759/TCP, 443:31059/TCP	10m
ingress-nginx	ingress-nginx-controller-admission	ClusterIP	10.103.181.137	<none>	443/TCP	10m
kube-system	kube-dns	ClusterIP	10.96.0.10	<none>	53/UDP, 53/TCP, 9153/TCP	10m

```

ubuntu@ip-172-31-45-73:~/website$

```

kubectl get ing

```

ubuntu@ip-172-31-45-73:~/website$ kubectl get ing

```

NAME	CLASS	HOSTS	ADDRESS	PORTS	AGE
ingress	nginx	*	192.168.49.2	80	4m54s

```

ubuntu@ip-172-31-45-73:~/website$

```

Task 10: Forwarded Traffic.

kubectl port-forward service/ingress-nginx-controller -n ingress-nginx --address 0.0.0.0 :443

```

ubuntu@ip-172-31-45-73:~/website$ kubectl port-forward service/ingress-nginx-controller -n ingress-nginx --address 0.0.0.0 :80
Forwarding from 0.0.0.0:35263 -> 80
Handling connection for 35263
Handling connection for 35263
Handling connection for 35263
Handling connection for 35263
Handling connection for 35263
Handling connection for 35263

```

kubectl create deployment customeimage --image=singh318/casestudy --port=80

```

^Cubuntu@ip-172-31-45-73:~/website$ kubectl create deployment customeimage --image=singh318/casestudy --port=80
deployment.apps/customeimage created
ubuntu@ip-172-31-45-73:~/website$ kubectl get deploy
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
apache2       1/1     1            1           34m
customeimage   1/1     1            1           12s
ubuntu@ip-172-31-45-73:~/website$ 

```

i-07c1ab491e66fea10 (Kubernetes-Casestudy)

PublicIPs: 13.232.248.42 PrivateIPs: 172.31.45.73

kubectl expose deploy customeimage --type NodePort

```

ubuntu@ip-172-31-45-73:~/website$ kubectl expose deploy customeimage --type NodePort
service/customeimage exposed
ubuntu@ip-172-31-45-73:~/website$ 

```

kubectl get svc

```

ubuntu@ip-172-31-45-73:~/website$ kubectl get svc
NAME          TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
apache-np     NodePort      10.107.51.0     <none>           80:31130/TCP     34m
customeimage   NodePort      10.108.241.156  <none>           80:31868/TCP     32s
kubernetes    ClusterIP     10.96.0.1       <none>           443/TCP          42m
ubuntu@ip-172-31-45-73:~/website$ 

```

Task 11: Create ingress customimage deployment.

sudo nano ingress.yaml

cat nano ingress.yaml

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: ingress

annotations:

nginx.ingress.kubernetes.io/rewrite-target: /

spec:

ingressClassName: nginx

rules:

- http:

paths:

- path: /apache

pathType: Prefix

backend:

service:

name: apache2

port:

number: 80

- path: /custome

pathType: Prefix

backend:

service:

name: customeimage

port:

number: 80

```
ubuntu@ip-172-31-45-73:~/website$ sudo nano ingress.yaml
ubuntu@ip-172-31-45-73:~/website$ cat nano ingress.yaml
cat: nano: No such file or directory
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  ingressClassName: nginx
  rules:
    - http:
        paths:
          - path: /apache
            pathType: Prefix
            backend:
              service:
                name: apache2
                port:
                  number: 80
          - path: /custome
            pathType: Prefix
            backend:
              service:
                name: customeimage
                port:
                  number: 80
ubuntu@ip-172-31-45-73:~/website$
```

kubectl apply -f ingress.yaml

```
ubuntu@ip-172-31-45-73:~/website$ kubectl apply -f ingress.yaml
ingress.networking.k8s.io/ingress configured
ubuntu@ip-172-31-45-73:~/website$
```

Task 12: Forwarded Traffic.

kubectrl port-forward service/ingress-nginx-controller -n ingress-nginx --address 0.0.0.0 :443

```
ubuntu@ip-172-31-45-73:~/website$ kubectrl port-forward service/ingress-nginx-controller -n ingress-nginx --address 0.0.0.0 :443
Forwarding from 0.0.0.0:46223 -> 443
Handling connection for 46223
Handling connection for 46223
Handling connection for 46223
Handling connection for 46223
Handling connection for 46223
Handling connection for 46223
```

As per problem statement Successfully executed Case Study-Kubernetes Assignment.

******* THE END *******