

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:

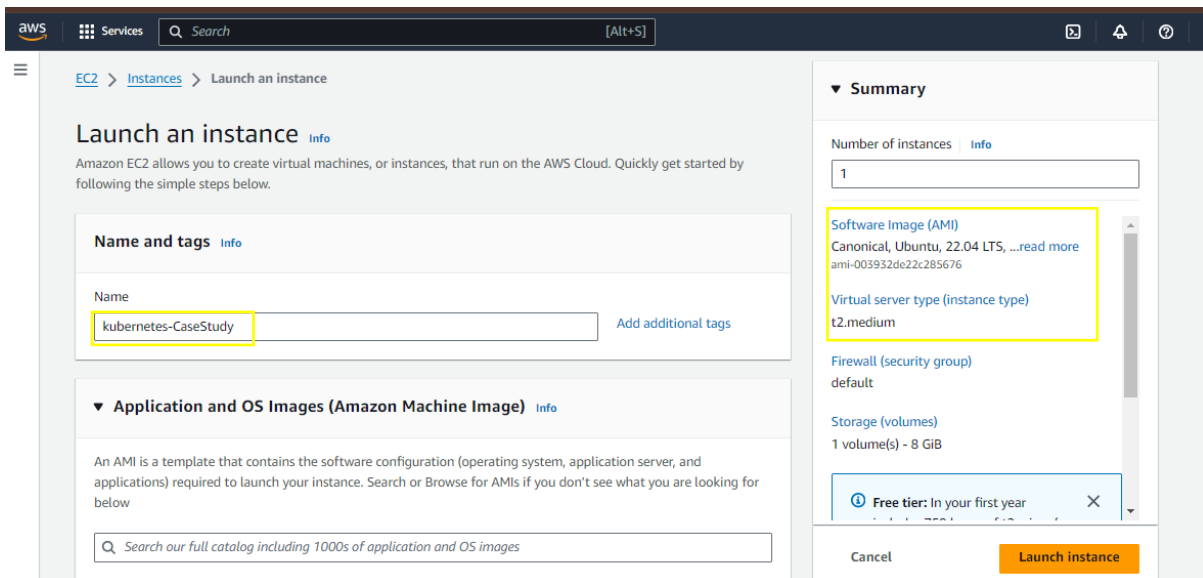
<https://github.com/hshar/website.git>.

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

Launch t2.medium instances for Kubernetes Assignment



aws Services Search [Alt+S]

EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

kubernetes-CaseStudy Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

▼ Summary

Number of instances Info

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more
ami-003932de22c285676

Virtual server type (instance type)

t2.medium

Firewall (security group)

default

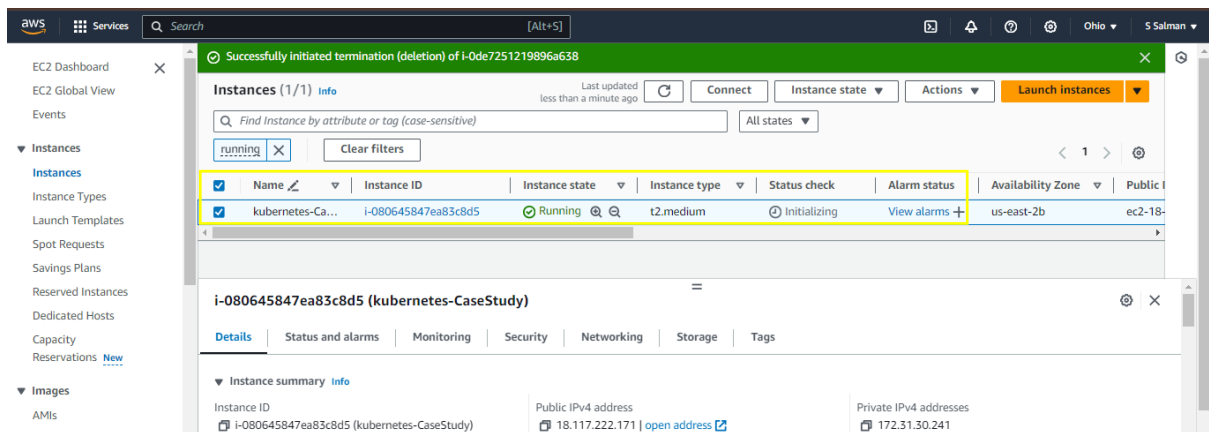
Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year

Cancel **Launch instance**

Click on connect



Copy it

Example:

```
ssh -i "salman-Ohio.pem" ubuntu@ec2-18-117-222-171.us-east-2.compute.amazonaws.com
```

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

```
C:\Users\shaik\Downloads>ssh -i "salman-Ohio.pem" ubuntu@ec2-18-117-222-171.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-18-117-222-171.us-east-2.compute.amazonaws.com (18.117.222.171)' can't be established.
ED25519 key fingerprint is SHA256:f+PLENgXZSRdc0/pW7xMguEVxCf45BlviopmsxQyliA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-117-222-171.us-east-2.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1022-aws x86_64)
```

```
C:\WINDOWS\system32\cmd. X + v
ubuntu@ip-172-31-30-241:~$ sudo hostnamectl set-hostname Kubernetes
ubuntu@ip-172-31-30-241:~$ exit
logout
Connection to ec2-18-117-222-171.us-east-2.compute.amazonaws.com closed.
C:\Users\shaik\Downloads>ssh -i "salman-Ohio.pem" ubuntu@ec2-18-117-222-171.us-east-2.compute.amazonaws.com
```

```
ubuntu@Kubernetes: ~ X + v
ubuntu@Kubernetes:~$ sudo apt-get update
```

Script for installing dockers, Kubernetes, minikube

```
ubuntu@Kubernetes: ~ X + v
ubuntu@Kubernetes:~$ sudo nano a.sh
```

Script for installing dockers, Kubernetes, minikube

```
ubuntu@Kubernetes: ~  
GNU nano 6.2 a.sh *  
sudo apt-get update  
sudo apt-get install docker.io -y  
sudo systemctl enable --now=docker  
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64  
sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64  
sudo chmod 777 /var/run/docker.sock  
minikube start --force --driver=docker  
sudo snap install kubectl --classic  
minikube status  
minikube addons enable ingress
```

Execute the file

```
ubuntu@Kubernetes: ~  
ubuntu@Kubernetes:~$ sudo nano a.sh  
ubuntu@Kubernetes:~$ bash a.sh
```

```
ubuntu@Kubernetes: ~  
ubuntu@Kubernetes:~$ bash a.sh  
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Reading package lists... Done  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan  
Suggested packages:  
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils  
The following NEW packages will be installed:  
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan  
0 upgraded, 8 newly installed, 0 to remove and 48 not upgraded.  
Need to get 75.5 MB of archives.  
After this operation, 284 MB of additional disk space will be used.  
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]  
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu3 [34.4 kB]  
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 runc amd64 1.1.12-0ubuntu2~22.04.1 [8405 kB]  
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.7.12-0ubuntu2~22.04.1 [37.8 MB]  
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 dns-root-data all 2023112702-ubuntu0.22.04.1 [5136 B]  
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 dnsmasq-base amd64 2.90-0ubuntu0.22.04.1 [374 kB]  
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 docker.io amd64 24.0.7-0ubuntu2~22.04.1 [28.8 MB]  
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]  
Fetched 75.5 MB in 1s (69.4 MB/s)  
Preconfiguring packages ...  
Selecting previously unselected package pigz.  
(Reading database ... 65320 files and directories currently installed.)  
Preparing to unpack .../0-pigz_2.6-1_amd64.deb ...  
Unpacking pigz (2.6-1) ...  
Selecting previously unselected package bridge-utils.  
Preparing to unpack .../1-bridge-utils_1.7-1ubuntu3_amd64.deb ...  
Unpacking bridge-utils (1.7-1ubuntu3) ...
```

```
100 91.1M 100 91.1M 0 0 89.9M 0 0:00:01 0:00:01 --:--:-- 89.9M
minikube v1.33.1 on Ubuntu 22.04 (xen/amd64)
Automatically selected the docker driver. Other choices: none, ssh
Using Docker driver with root privileges
Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v0.0.44 ...
Downloading Kubernetes v1.30.0 preload ...
> preloaded-images-k8s-v18-v1...: 342.90 MiB / 342.90 MiB 100.00% 73.19 M
> gcr.io/k8s-minikube/kicbase...: 481.58 MiB / 481.58 MiB 100.00% 38.14 M
Creating docker container (CPUs=2, Memory=2200MB) ...
Preparing Kubernetes v1.30.0 on Docker 26.1.1 ...
  Generating certificates and keys ...
  Booting up control plane ...
  Configuring RBAC rules ...
Configuring bridge CNI (Container Networking Interface) ...
Verifying Kubernetes components...
  Using image gcr.io/k8s-minikube/storage-provisioner:v5
Enabled addons: storage-provisioner, default-storageclass
  kubectrl not found. If you need it, try: 'minikube kubectrl -- get pods -A'
Done! kubectrl is now configured to use "minikube" cluster and "default" namespace by default
error: cannot install "kubectrl", "-", "classic": invalid instance name: invalid snap name: "-"
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

  ingress is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
  Using image registry.k8s.io/ingress-nginx/controller:v1.10.1
  Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.1
  Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.1
Verifying ingress addon...
  The 'ingress' addon is enabled
ubuntu@Kubernetes:~$
```

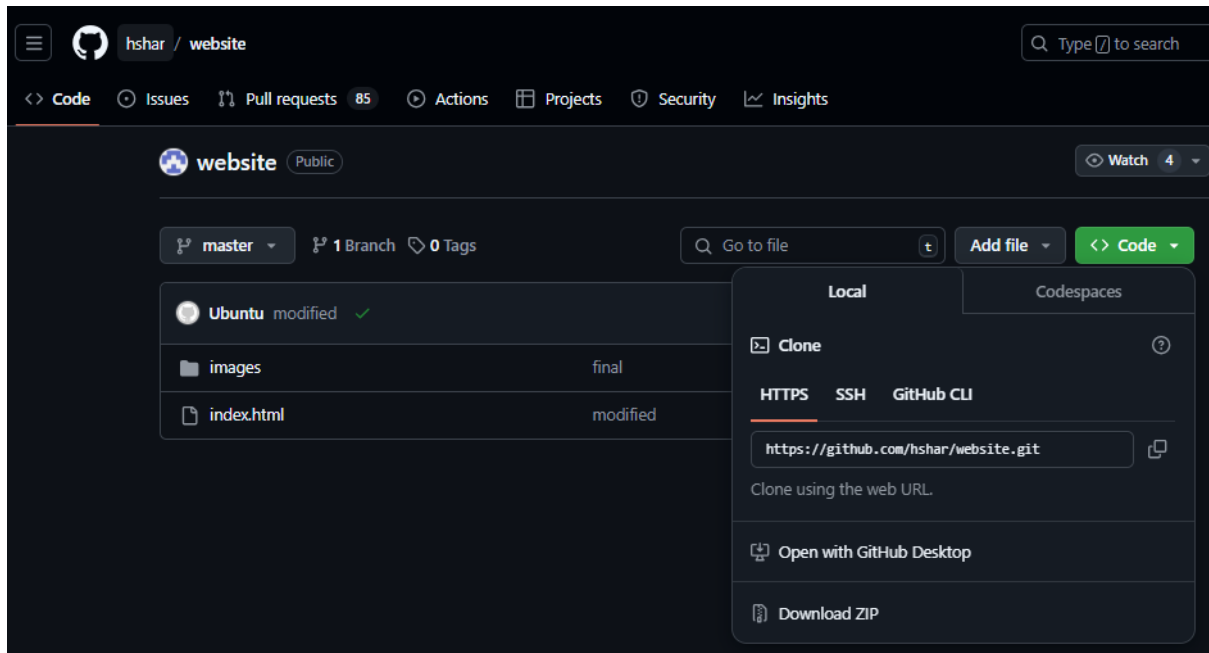
Its not installed kubectrl so we need to install

```
ubuntu@Kubernetes: ~$ sudo snap install kubectrl --classic
kubectrl 1.30.4 from Canonical✓ installed
ubuntu@Kubernetes:~$
```

For finding the nodes, run the below-given command

```
ubuntu@Kubernetes:~$ kubectrl get nodes
NAME          STATUS    ROLES          AGE    VERSION
minikube      Ready     control-plane  10m    v1.30.0
ubuntu@Kubernetes:~$
```

Clone this repository in the “Git Hub Account”. Click on “Code” & Copy the URL



Type the below-given command in the EC2 Instance & press enter from the keyboard

It will be started cloning the website from Git.

```
ubuntu@Kubernetes: ~  
ubuntu@Kubernetes:~$ 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 (from 1)  
Receiving objects: 100% (8/8), 82.69 KiB | 3.31 MiB/s, done.  
Resolving deltas: 100% (1/1), done.  
ubuntu@Kubernetes:~$ ls  
a.sh  snap  website  
ubuntu@Kubernetes:~$
```

Type the below-given command to show how many directories are present

Write a Docker file & Push the Image to Docker Hub

Type the below-given command to create a Dockerfile

```
ubuntu@Kubernetes: ~/webs  
ubuntu@Kubernetes:~$ cd website  
ubuntu@Kubernetes:~/website$ ls  
images  index.html  
ubuntu@Kubernetes:~/website$ sudo nano Dockerfile
```

Type these **commands** to run “**apache2**” in the **containerized environment**

```
ubuntu@Kubernetes: ~/webs X + v
GNU nano 6.2 Dockerfile *
FROM ubuntu
RUN apt-get update
RUN apt-get install apache2 -y
RUN apt-get install apache2-utils -y
RUN apt-get clean
ENTRYPOINT apachectl -D FOREGROUND
ADD . /var/www/html/
```

Run the **below-given command** to **build** an **image** from the **Dockerfile**

```
ubuntu@Kubernetes: ~/webs X + v
ubuntu@Kubernetes:~$ cd website
ubuntu@Kubernetes:~/website$ ls
images  index.html
ubuntu@Kubernetes:~/website$ sudo nano Dockerfile
ubuntu@Kubernetes:~/website$ sudo docker build -t img .
```

Run the below-given command to build an image from the Dockerfile

```
ubuntu@Kubernetes: ~/webs X + v
ubuntu@Kubernetes:~/website$ sudo docker build -t img .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 252.9kB
Step 1/7 : FROM ubuntu
latest: Pulling from library/ubuntu
31e907dcc94a: Pull complete
Digest: sha256:8a37d68f4f73ebf3d4efafbcf66379bf3728902a8038616808f04e34a9ab63ee
Status: Downloaded newer image for ubuntu:latest
--> edbfe74c41f8
Step 2/7 : RUN apt-get update
--> Running in 5851f3a00d0d
Get:1 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [12.7 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [354 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [449 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [16.9 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [594 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [354 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [11.5 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [404 kB]
Get:17 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [337 kB]
Fetched 24.7 MB in 2s (14.1 MB/s)
Reading package lists...
Removing intermediate container 5851f3a00d0d
--> 5f55b00cc400
```

The image will be successfully created. Now run the below-given command to view the images in the website directory

```
ubuntu@Kubernetes: ~/webs X + v
ubuntu@Kubernetes:~/website$ docker images
REPOSITORY          TAG          IMAGE ID          CREATED           SIZE
img                  latest       ee7e137e3bad     24 seconds ago   231MB
ubuntu               latest       edbfe74c41f8     4 weeks ago      78.1MB
gcr.io/k8s-minikube/kicbase v0.0.44     5a6e59a9bdc0     3 months ago     1.26GB
ubuntu@Kubernetes:~/website$
```

Use the below-given command to use docker credentials the image

```
ubuntu@Kubernetes: ~/webs X + v
ubuntu@Kubernetes:~/website$ sudo docker tag img salmanfarsi/k8casestudy
ubuntu@Kubernetes:~/website$ docker images
REPOSITORY          TAG          IMAGE ID          CREATED           SIZE
img                  latest       ee7e137e3bad     9 minutes ago    231MB
salmanfarsi/k8casestudy latest       ee7e137e3bad     9 minutes ago    231MB
ubuntu               latest       edbfe74c41f8     4 weeks ago      78.1MB
gcr.io/k8s-minikube/kicbase v0.0.44     5a6e59a9bdc0     3 months ago     1.26GB
ubuntu@Kubernetes:~/website$
```


Docker Login

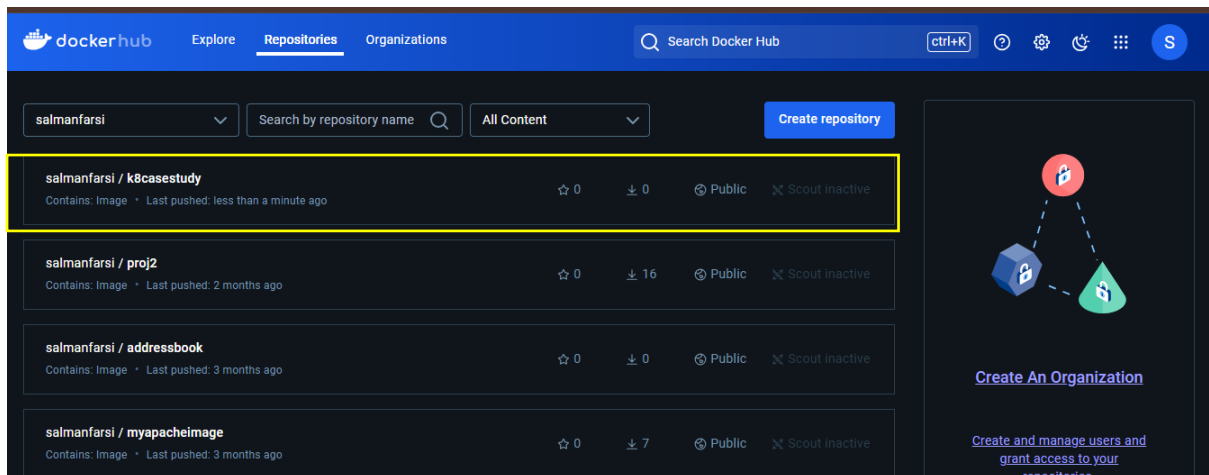
```
ubuntu@Kubernetes: ~/webs docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/

Username: salmanfarsi
Password:
WARNING! Your password will be stored unencrypted in /home/ubuntu/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
ubuntu@Kubernetes: ~/website$
```

Pushing the Image to Docker Account

```
ubuntu@Kubernetes: ~/webs docker push salmanfarsi/k8casestudy
Using default tag: latest
The push refers to repository [docker.io/salmanfarsi/k8casestudy]
8f79e5cf4fdb: Pushed
18d24e588286: Pushed
aef3e38a5bf7: Pushed
fd3314f966b4: Pushed
dccb0abec538: Pushed
f36fd4bb7334: Pushed
latest: digest: sha256:5ef5079a685bbb013d11981fb057cd75c1ff8c65b7adc6944b5dc0446bf1193b size: 1577
ubuntu@Kubernetes: ~/website$
```



Run the below-given command to create the custom image with 2 replicas

```
ubuntu@Kubernetes: ~/webs kubectl create deployment custom --image=salmanfarsi/k8casestudy --replicas=2 --port=80
deployment.apps/custom created
ubuntu@Kubernetes: ~/website$ kubectl get deploy -o wide
NAME      READY  UP-TO-DATE  AVAILABLE  AGE    CONTAINERS  IMAGES                SELECTOR
custom    2/2    2           2          22s   k8casestudy  salmanfarsi/k8casestudy  app=custom
ubuntu@Kubernetes: ~/website$
```


A Custom Deployment will be created with Two Pods

```
ubuntu@Kubernetes: ~/webs X + v
ubuntu@Kubernetes:~/website$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
custom-68c664f76b-jjjc1            1/1     Running   0           54s
custom-68c664f76b-jmc6k            1/1     Running   0           54s
ubuntu@Kubernetes:~/website$
```

Run the below-given command to create the deployment with 2 replicas:

A deployment will be successfully created on port 80.

When you run the below-given commands

Two Apache pods will be successfully created

```
ubuntu@Kubernetes: ~/webs X + v
ubuntu@Kubernetes:~/website$ kubectl create deployment apache --image=ubuntu/apache2 --replicas=2 --port=80
deployment.apps/apache created
ubuntu@Kubernetes:~/website$ kubectl get deploy -o wide
Command 'kubectl' not found, did you mean:
  command 'kubectl' from snap kubectl (1.30.4)
See 'snap info <snapname>' for additional versions.
ubuntu@Kubernetes:~/website$ kubectl get deploy -o wide
NAME      READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES           SELECTOR
apache    2/2     2            2           71s   apache2      ubuntu/apache2   app=apache
custom    2/2     2            2           2m57s k8casestudy  salmanfarsi/k8casestudy app=custom
ubuntu@Kubernetes:~/website$
```

When you run the below-given command

```
ubuntu@Kubernetes: ~/webs X + v
ubuntu@Kubernetes:~/website$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
apache-5b9b5bff7c-drhmd             1/1     Running   0           102s
apache-5b9b5bff7c-hmvfl             1/1     Running   0           102s
custom-68c664f76b-jjjc1            1/1     Running   0           3m28s
custom-68c664f76b-jmc6k            1/1     Running   0           3m28s
ubuntu@Kubernetes:~/website$
```

a. */apache* should point to the Apache pods

b. */custom* should point to the GitHub application.

Follow these steps;

A. Expose Both the Replicas on “NodePort” for Creating a Service

First, we will expose “apache” on “Node Port”. Run the below-given command

```
ubuntu@Kubernetes: ~/webs X + v
ubuntu@Kubernetes:~/website$ kubectl expose deploy apache --type=NodePort
service/apache exposed
ubuntu@Kubernetes:~/website$
```

The apache service has been successfully created on “NodePort”.

Second, we will expose the “custom” on “NodePort”. Use the below-given command

```
ubuntu@Kubernetes: ~/webs X + v
ubuntu@Kubernetes:~/website$ kubectl expose deploy custom --type=NodePort
service/custom exposed
ubuntu@Kubernetes:~/website$
```

Run the below-given command to find the service

Both services are successfully exposed on “NodePort

```
ubuntu@Kubernetes: ~/webs X + v
ubuntu@Kubernetes:~/website$ kubectl get svc -o wide
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE	SELECTOR
apache	NodePort	10.107.98.205	<none>	80:31848/TCP	107s	app=apache
custom	NodePort	10.109.146.0	<none>	80:30121/TCP	17s	app=custom
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	45m	<none>

```
ubuntu@Kubernetes:~/website$
```

Create an ingress.yml file to point out the paths

Create an “ingress.yml” file using the below-given command

```
ubuntu@Kubernetes: ~/webs X + v
ubuntu@Kubernetes:~/website$ sudo nano ingress.yml
```

Paste the below-given content into the “ingress.yml” file

```
ubuntu@Kubernetes: ~/webs X + v
GNU nano 6.2
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: apache
            port:
              number: 80
      - path: /custom
        pathType: Prefix
        backend:
          service:
            name: custom
            port:
              number: 80
```

Create an “ingress.yaml” file using the below-given command

The “ingress” deployment will be created

Run the below-given command to view the ingress

Apply Port Forwarding to Open the Web Page on Separate Port

```
ubuntu@Kubernetes: ~/webs X + v
ubuntu@Kubernetes:~/website$ sudo nano ingress.yml
ubuntu@Kubernetes:~/website$ kubectl create -f ingress.yml
ingress.networking.k8s.io/ingress created
ubuntu@Kubernetes:~/website$ kubectl get ing
NAME      CLASS  HOSTS  ADDRESS      PORTS  AGE
ingress   nginx  *      192.168.49.2  80     22s
ubuntu@Kubernetes:~/website$
```

Paste the below-given command into “EC2 Machine

The port will be successfully forwarded

```
ubuntu@Kubernetes: ~/webs X +
ubuntu@Kubernetes:~/website$ kubectl port-forward service/ingress-nginx-controller -n ingress-nginx --address 0.0.0.0 :443
Forwarding from 0.0.0.0:33603 -> 443

Handling connection for 33603
Handling connection for 33603
Handling connection for 33603
Handling connection for 33603
Handling connection for 33603
Handling connection for 33603
Handling connection for 33603
Handling connection for 33603
Handling connection for 33603
Handling connection for 33603
Handling connection for 33603
Handling connection for 33603
```

kubernetes-Ca... i-080645847ea83c8d5 Running t2.medium 2/2 checks passed View alarms

i-080645847ea83c8d5 (kubernetes-CaseStudy)

Details Status and alarms Monitoring Security Networking Storage Tags

▼ Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-080645847ea83c8d5 (kubernetes-CaseStudy)	18.117.222.171 open address	172.31.30.241

Not secure https://18.117.222.171:33603/apache

Interfacing SIM900... Downloads New Tab Python Exercises: C... MEGA Women's Health: 25... Imported

Ubuntu Logo

Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Not secure https://18.117.222.171:33603/custom

Interfacing SIM900... Downloads New Tab Python Exercises: C... MEGA Women's Health: 25... Imported

Hello world!