

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.

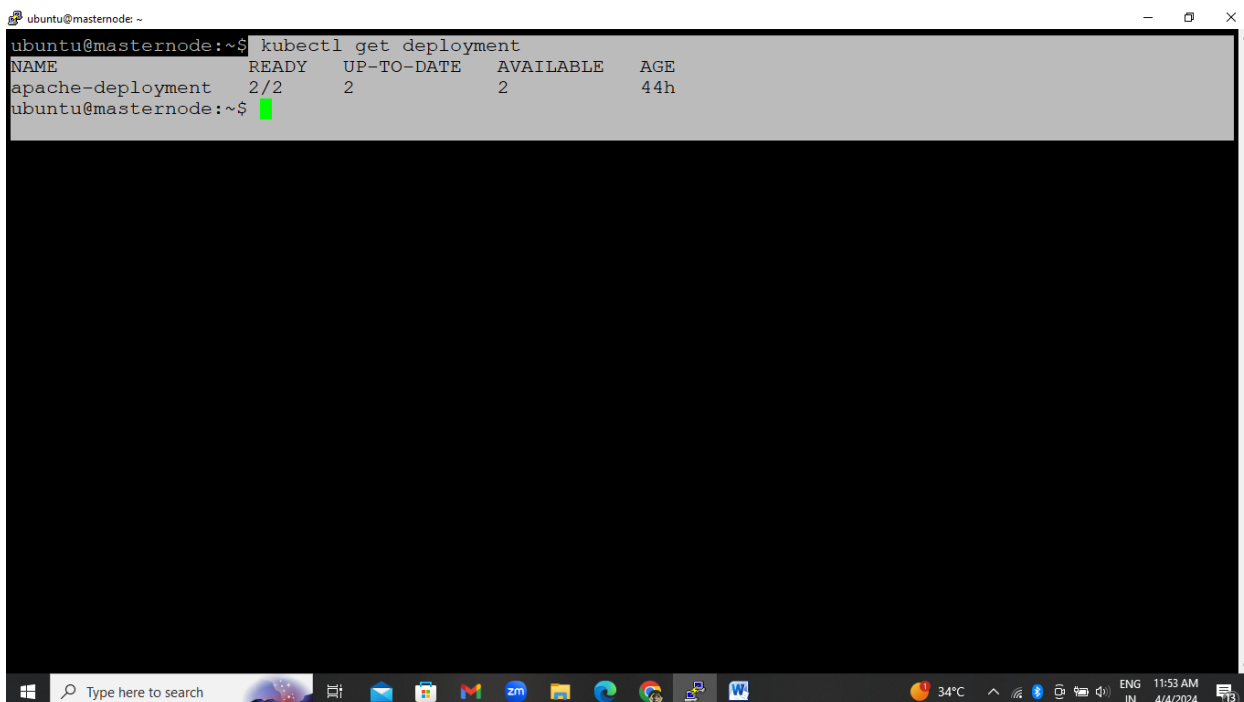
Following things have to be implemented:

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



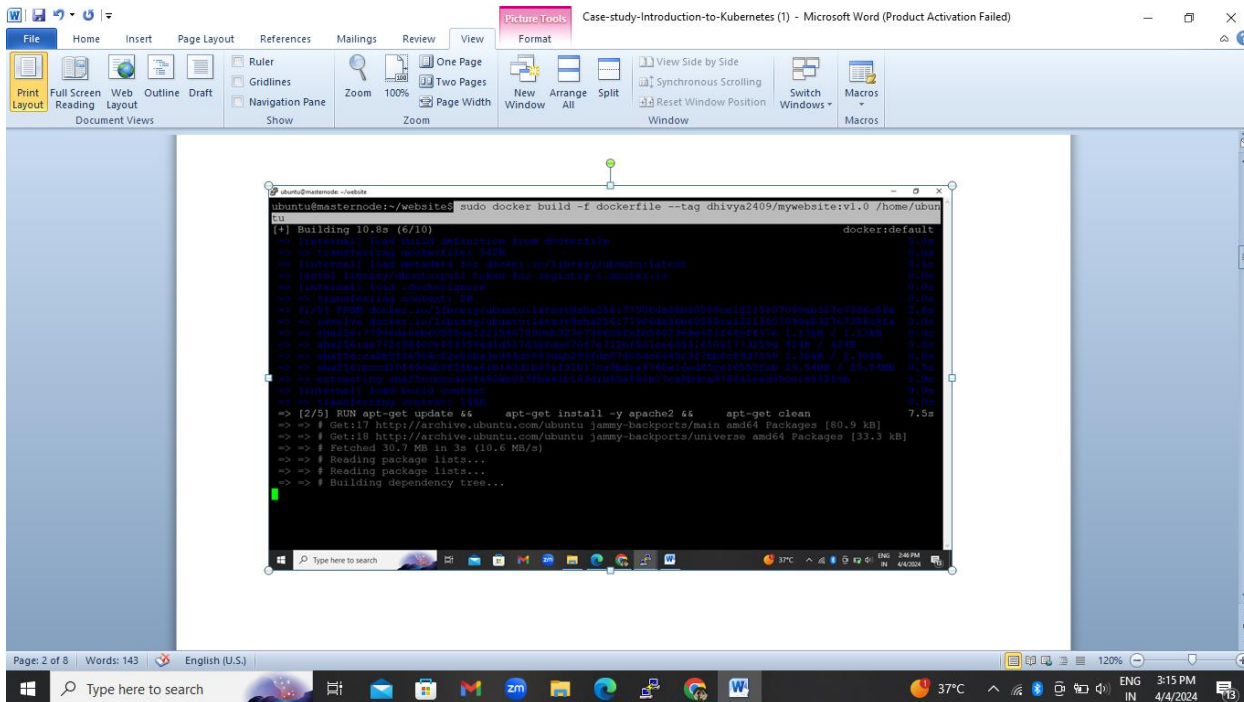
```
ubuntu@masternode: ~$ kubectl get deployment
NAME                READY    UP-TO-DATE    AVAILABLE    AGE
apache-deployment   2/2      2              2            44h
ubuntu@masternode: ~$
```

The screenshot shows a terminal window on a Linux system (ubuntu@masternode). The command `kubectl get deployment` has been executed, resulting in a table of deployment information. The table has five columns: NAME, READY, UP-TO-DATE, AVAILABLE, and AGE. There is one deployment listed: 'apache-deployment' with 2/2 ready replicas, 2 up-to-date replicas, 2 available replicas, and an age of 44h. The terminal prompt is `ubuntu@masternode: ~$`.

```
ubuntu@masternode:~$ kubectl get deployment
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
apache-deployment   2/2     2             2           44h

ubuntu@masternode:~$ kubectl get service
NAME                TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
apache-service      ClusterIP   10.96.187.84    <none>        3001/TCP    43h
kubernetes           ClusterIP   10.96.0.1       <none>        443/TCP     2d

ubuntu@masternode:~$
```



```
ubuntu@masternode: ~/website
ubuntu@masternode:~/website$ sudo docker login
Authenticating with existing credentials...
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
ubuntu@masternode:~/website$
```

```
ubuntu@masternode: ~/website
=> [auth] library/ubuntu:pull token for registry-1.docker.io 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [1/5] FROM docker.io/library/ubuntu:latest@sha256:77906da86b60585ce12215807090eb327e7386c8fa 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 108.08kB 0.0s
=> CACHED [2/5] RUN apt-get update && apt-get install -y apache2 && apt-get clean 0.0s
=> CACHED [3/5] COPY ./website/index.html /var/www/html/index.html 0.0s
=> CACHED [4/5] COPY ./website/github3.jpg /var/www/html/images/github3.jpg 0.0s
=> CACHED [5/5] WORKDIR /usr/sbin 0.0s
=> exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:b08dc91192631d5b21802c8ffee6688894dd827cc22e8e78dc83de36608ffla0 0.0s
=> => naming to docker.io/dhivya2409/mywebsite:v1.1 0.0s
ubuntu@masternode:~/website$ sudo docker login
Authenticating with existing credentials...
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
ubuntu@masternode:~/website$ sudo docker push dhivya2409/mywebsite:v1.1
The push refers to repository [docker.io/dhivya2409/mywebsite]
5f70bf18a086: Layer already exists
330a2123698d: Layer already exists
24e87fbc3017: Layer already exists
11bcf2fclffa: Layer already exists
5498e8c22f69: Layer already exists
v1.1: digest: sha256:0f1f1a0b5d9ab24c9b2dd1c9da85a1d0de9c99935866f39307bc98330304efe8 size: 1363
ubuntu@masternode:~/website$
```

Instances | EC2 | ChatGPT | Start Course | raw.githubusercontent.com | Intellipaat | Intellipaat | dhivya2409/mywebsite

hub.docker.com/repository/docker/dhivya2409/mywebsite/general

Gmail YouTube xdhivya (Dhivya) AWS Console Intellipat Freshers Jobs in India Naukri LinkedIn Instahyre Hackerrank Miro | Online White...

dockerhub Explore Repositories Organizations Search Docker Hub ctrl+k ? D

dhivya2409 / Repositories / mywebsite / General Using 0 of 1 private repositories. [Get more](#)

General Tags Builds Collaborators Webhooks Settings

Add a short description for this repository [Update](#)
The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.

dhivya2409/mywebsite [Public View](#)
Updated 3 minutes ago
This repository does not have a description

Docker commands
To push a new tag to this repository:
`docker push dhivya2409/mywebsite:tagname`

Tags
This repository contains 3 tag(s).

Tag	OS	Type	Pulled	Pushed
v1.1		Image	---	3 minutes ago

Automated Builds
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.
Available with Pro, Team and Business subscriptions. [Read more about automated builds](#)

Type here to search 37°C 3:17 PM 4/4/2024

```
ubuntu@masternode: ~$ kubectl apply -f mywebsite-deployment.yaml
deployment.apps/mywebsite-deployment created
ubuntu@masternode: ~$
```

Type here to search 37°C 2:55 PM 4/4/2024

```
ubuntu@masternode: ~  
ubuntu@masternode:~$ kubectl get deploy  
NAME                READY    UP-TO-DATE    AVAILABLE    AGE  
apache-deployment   2/2      2              2             47h  
mywebsite-deployment 2/2      2              2             42s  
ubuntu@masternode:~$
```

```
ubuntu@masternode: ~  
ubuntu@masternode:~$ kubectl get deploy  
NAME                READY    UP-TO-DATE    AVAILABLE    AGE  
apache-deployment   2/2      2              2             47h  
mywebsite-deployment 2/2      2              2             42s  
ubuntu@masternode:~$ kubectl get pods  
NAME                                READY    STATUS      RESTARTS    AGE  
apache-deployment-8c8bd99f6-7fbbc  1/1      Running    1 (45h ago)  47h  
apache-deployment-8c8bd99f6-9xfvg  1/1      Running    1 (45h ago)  47h  
mywebsite-deployment-5487487fd4-6trgq 1/1      Running    0            68s  
mywebsite-deployment-5487487fd4-hkw9c 1/1      Running    0            68s  
ubuntu@masternode:~$
```

```
ubuntu@masternode: ~  
ubuntu@masternode:~$ kubectl apply -f mywebsite-svc.yaml  
service/mywebsite-service created  
ubuntu@masternode:~$ kubectl get svc  
NAME                TYPE          CLUSTER-IP      EXTERNAL-IP  PORT(S)    AGE  
apache-service      ClusterIP     10.96.187.84    <none>       3001/TCP   46h  
kubernetes           ClusterIP     10.96.0.1       <none>       443/TCP    2d3h  
mywebsite-service   ClusterIP     10.105.202.211  <none>       4000/TCP   19s  
ubuntu@masternode:~$
```

```
ubuntu@masternode: ~  
ubuntu@masternode:~$ kubectl get svc --namespace ingress-nginx  
NAME                AGE      TYPE          CLUSTER-IP      EXTERNAL-IP  PORT(S)  
ingress-nginx-controller  3h47m    NodePort      10.109.185.51    <none>       80:30934/TCP,443:30554/TCP  
ingress-nginx-controller-admission  3h47m    ClusterIP     10.109.2.234     <none>       443/TCP  
ubuntu@masternode:~$
```

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#instances?v=3&case=tags:true%5C,client:false;\$regex=tags:false%5C,client:false

EC2 Dashboard

Instances (1/6) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
masternode	i-0d82dec701320537a	Running	t2.small	2/2 checks passed	View alarms	us-east-1a
workernode1	i-0fd4bb5782d6e9...	Running	t2.small	2/2 checks passed	View alarms	us-east-1a
workernode2	i-0c610c4348233ad...	Running	t2.small	2/2 checks passed	View alarms	us-east-1a

Instance: i-0d82dec701320537a (masternode)

Details | Status and alarms New | Monitoring | Security | Networking | Storage | Tags

Instance summary Info

Instance ID: i-0d82dec701320537a (masternode)

Public IPv4 address: 34.227.32.35 [open address](#)

Private IPv4 addresses: 172.31.88.113

Instance state: Running

Public IPv4 DNS: ec2-34-227-32-35.compute-1.amazonaws.com [open address](#)

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

37°C 3:00 PM 4/4/2024

Not secure 34.227.32.35:30934/apache

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

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|-- *.Load
```

Nift... 3:01 PM 4/4/2024

<http://34.227.32.35:30934/custom>

Hello world!



GitHub



```
ubuntu@masternode: ~$ cat ingres.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: apache-service
            port:
              number: 80
      - path: /custom
        pathType: Prefix
        backend:
          service:
            name: mywebsite-service
            port:
              number: 80
ubuntu@masternode:~$
```



```
ubuntu@masternode: ~  
ubuntu@masternode:~$ cat mywebsite-deployment.yaml  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: mywebsite-deployment  
spec:  
  replicas: 2  
  selector:  
    matchLabels:  
      app: mywebsite  
  template:  
    metadata:  
      labels:  
        app: mywebsite  
    spec:  
      containers:  
      - name: mywebsite  
        image: dhivya2409/mywebsite:v1.1  
ubuntu@masternode:~$
```

```
ubuntu@masternode: ~  
metadata:  
  name: mywebsite-deployment  
spec:  
  replicas: 2  
  selector:  
    matchLabels:  
      app: mywebsite  
  template:  
    metadata:  
      labels:  
        app: mywebsite  
    spec:  
      containers:  
      - name: mywebsite  
        image: dhivya2409/mywebsite:v1.1  
ubuntu@masternode:~$ cat mywebsite-svc.yaml  
apiVersion: v1  
kind: Service  
metadata:  
  name: mywebsite-service  
spec:  
  type: ClusterIP  
  selector:  
    app: mywebsite  
  ports:  
  - port: 4000  
    targetPort: 80  
ubuntu@masternode:~$
```

```
ubuntu@masternode: ~  
ubuntu@masternode:~$ vi apache2-deploy.yaml  
ubuntu@masternode:~$ cat apache2-deploy.yaml  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: apache-deployment  
spec:  
  replicas: 2  
  selector:  
    matchLabels:  
      app: apache  
  template:  
    metadata:  
      labels:  
        app: apache  
    spec:  
      containers:  
      - name: apache2  
        image: ubuntu/apache2:latest  
  
ubuntu@masternode:~$ ~
```

```
ubuntu@masternode: ~  
ubuntu@masternode:~$ cat apache-svc.yaml  
apiVersion: v1  
kind: Service  
metadata:  
  name: apache-service  
spec:  
  type: ClusterIP  
  selector:  
    app: apache  
  ports:  
  - port: 3001  
    targetPort: 80  
  
ubuntu@masternode:~$ ~
```