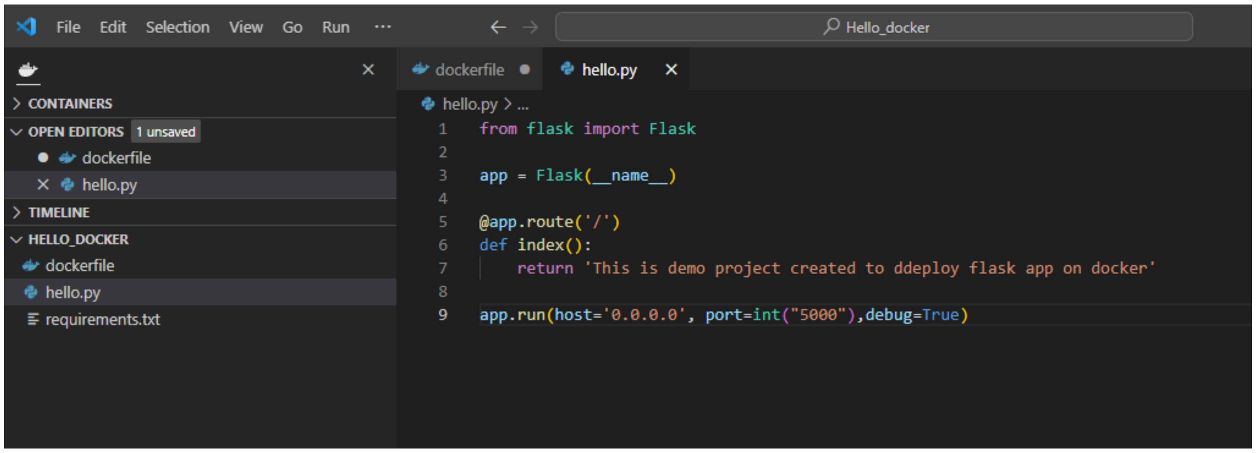
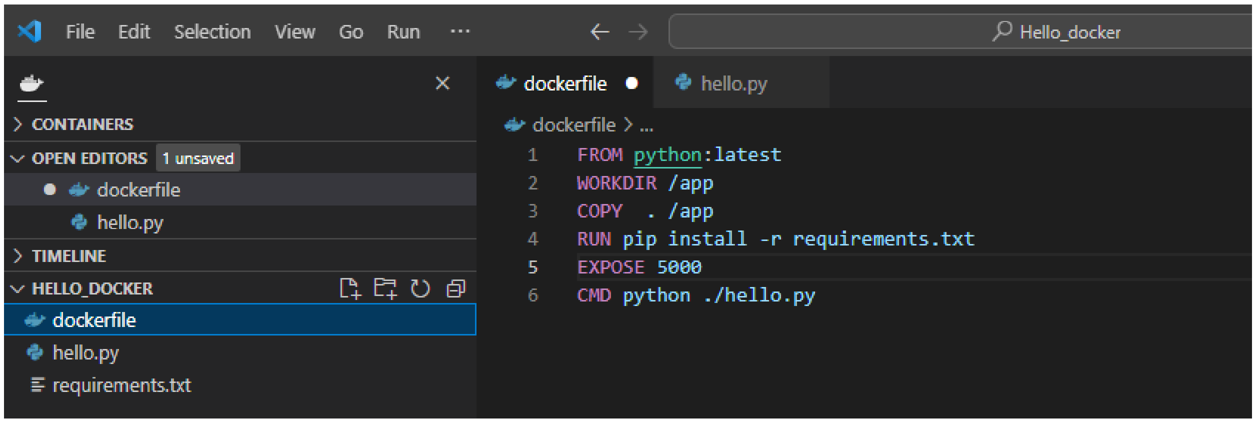
PROJECT DIPLOYMENT USING KUBERNETES

Step1: create project file



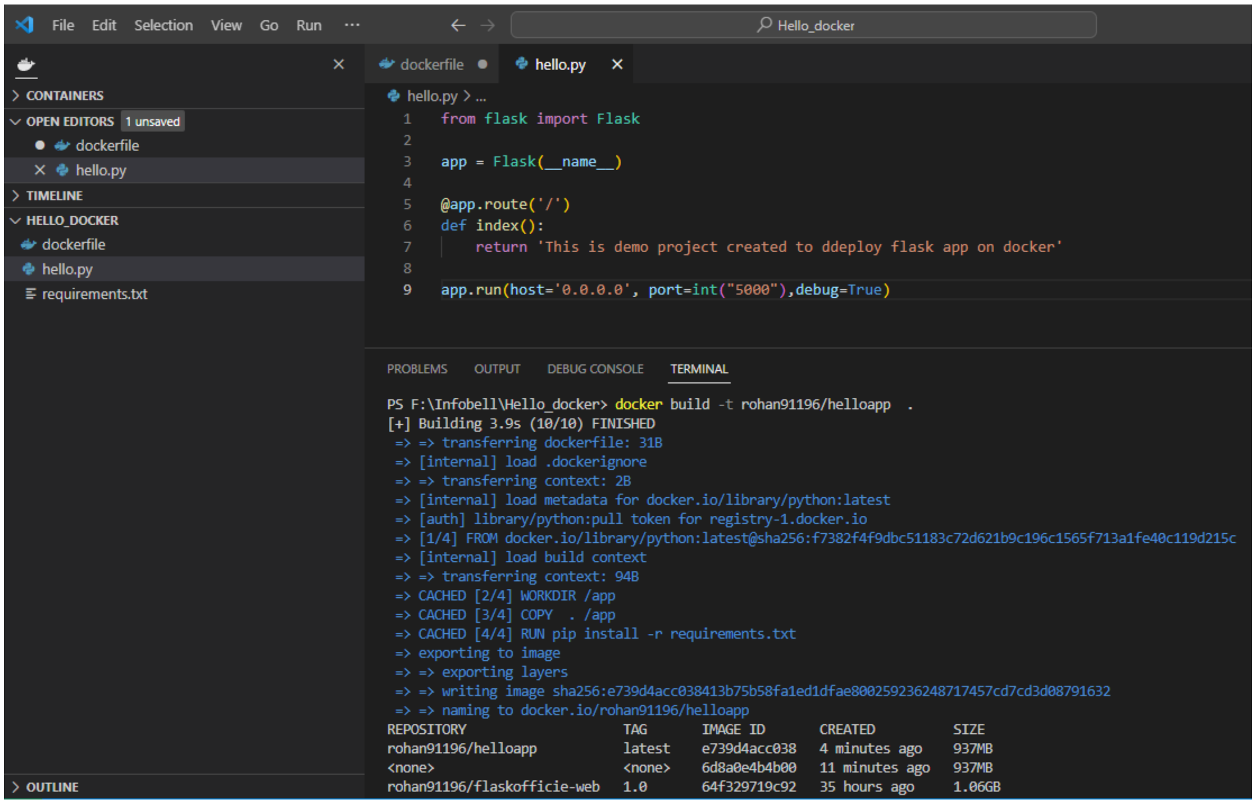
```python
from flask import Flask

app = Flask(__name__)

@app.route('/')
def index():
    return 'This is demo project created to ddeploy flask app on docker'

app.run(host='0.0.0.0', port=int("5000"),debug=True)
```

Step 2: Create docker file



```dockerfile
FROM python:latest
WORKDIR /app
COPY  . /app
RUN pip install -r requirements.txt
EXPOSE 5000
CMD python ./hello.py
```

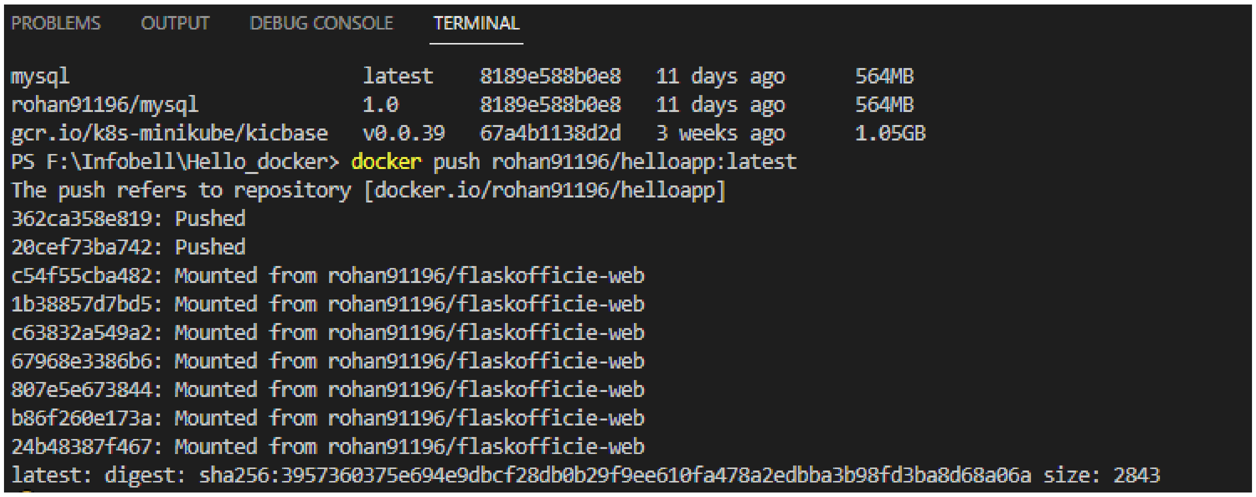
Step 3: Create docker image



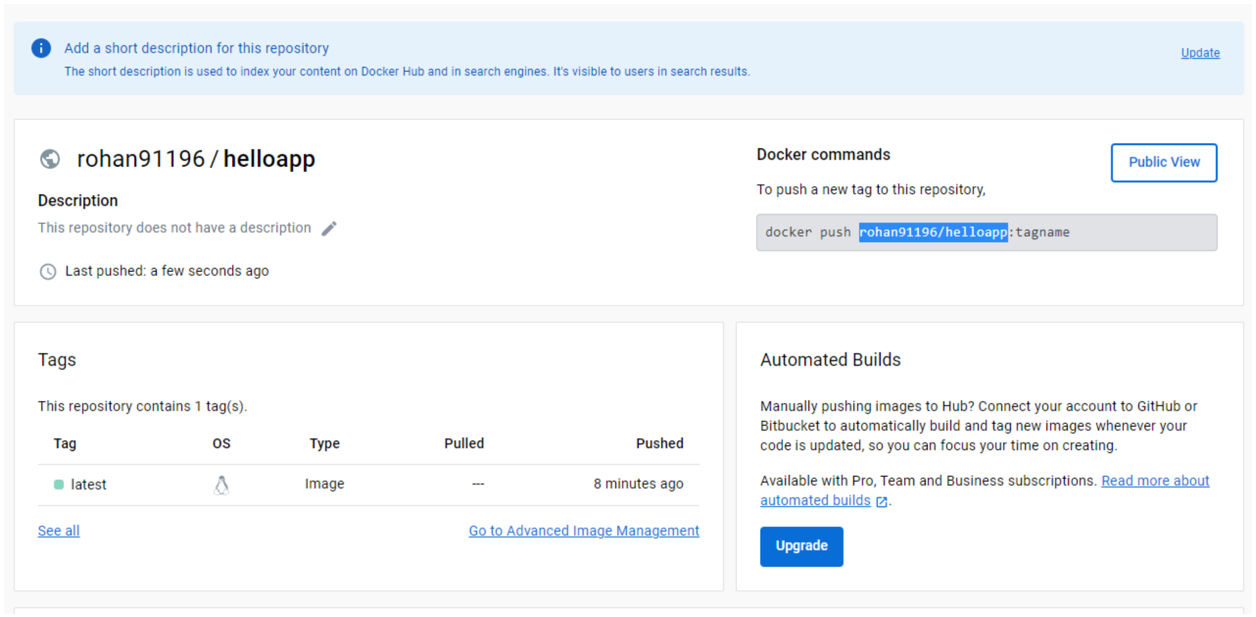
```python
from flask import Flask

app = Flask(__name__)

@app.route('/')
def index():
    return 'This is demo project created to ddeploy flask app on docker'

app.run(host='0.0.0.0', port=int("5000"),debug=True)
```

```
PS F:\Infobell\Hello_docker> docker build -t rohan91196/helloapp  .
[+] Building 3.9s (10/10) FINISHED
 => => transferring dockerfile: 31B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/python:latest
 => [auth] library/python:pull token for registry-1.docker.io
 => [1/4] FROM docker.io/library/python:latest@sha256:f7382f4f9dbc51183c72d621b9c196c1565f713a1fe40c119d215c
 => [internal] load build context
 => => transferring context: 94B
 => CACHED [2/4] WORKDIR /app
 => CACHED [3/4] COPY  . /app
 => CACHED [4/4] RUN pip install -r requirements.txt
 => exporting to image
 => => exporting layers
 => => writing image sha256:e739d4acc038413b75b58fa1ed1dfae800259236248717457cd7cd3d08791632
 => => naming to docker.io/rohan91196/helloapp
REPOSITORY                  TAG       IMAGE ID       CREATED          SIZE
rohan91196/helloapp         latest    e739d4acc038   4 minutes ago    937MB
<none>                      <none>    6d8a0e4b4b00   11 minutes ago   937MB
rohan91196/flaskofficie-web 1.0       64f329719c92   35 hours ago     1.06GB
```

Step 4: Push image to docker hub repository



```
mysql                      latest    8189e588b0e8   11 days ago    564MB
rohan91196/mysql           1.0       8189e588b0e8   11 days ago    564MB
gcr.io/k8s-minikube/kicbase  v0.0.39   67a4b1138d2d   3 weeks ago    1.05GB
PS F:\Infobell\Hello_docker> docker push rohan91196/helloapp:latest
The push refers to repository [docker.io/rohan91196/helloapp]
362ca358e819: Pushed
20cef73ba742: Pushed
c54f55cba482: Mounted from rohan91196/flaskofficie-web
1b38857d7bd5: Mounted from rohan91196/flaskofficie-web
c63832a549a2: Mounted from rohan91196/flaskofficie-web
67968e3386b6: Mounted from rohan91196/flaskofficie-web
807e5e673844: Mounted from rohan91196/flaskofficie-web
b86f260e173a: Mounted from rohan91196/flaskofficie-web
24b48387f467: Mounted from rohan91196/flaskofficie-web
latest: digest: sha256:3957360375e694e9dbcf28db0b29f9ee610fa478a2edbba3b98fd3ba8d68a06a size: 2843
```



Add a short description for this repository

The short description is used to index your content on Docker Hub and in search results. It's visible to users in search results.

Update

rohan91196 / helloapp

Description

This repository does not have a description

Last pushed: a few seconds ago

Docker commands

Public View

To push a new tag to this repository,

```
docker push rohan91196/helloapp:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	---	8 minutes ago

See all

Go to Advanced Image Management

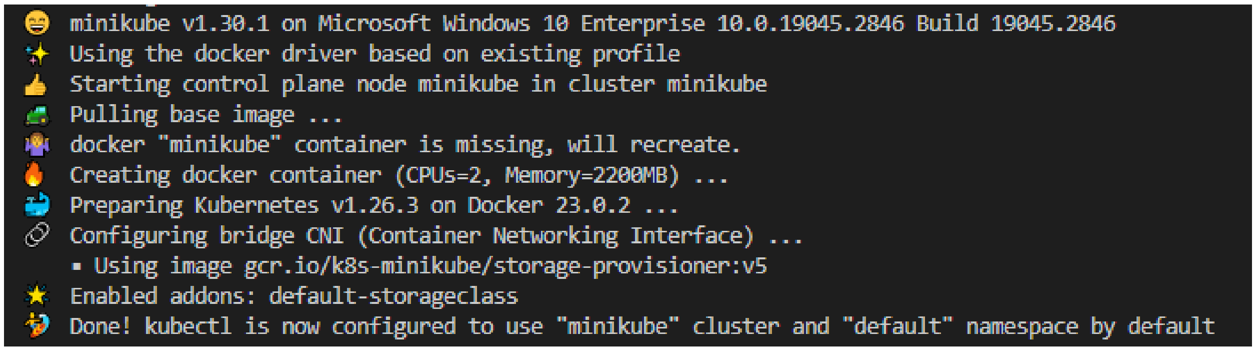
Automated Builds

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.

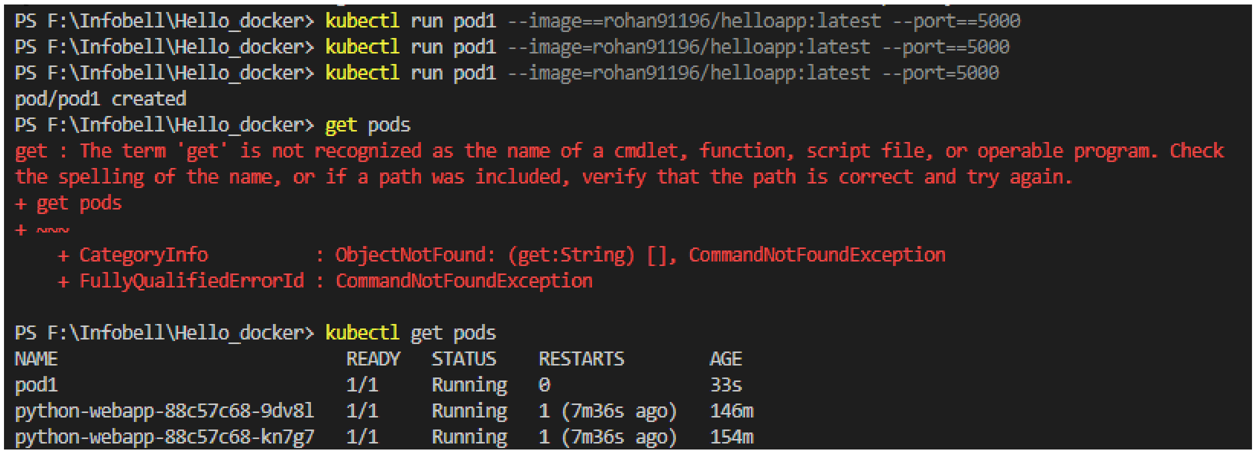
Upgrade

Step 5: Start minikube



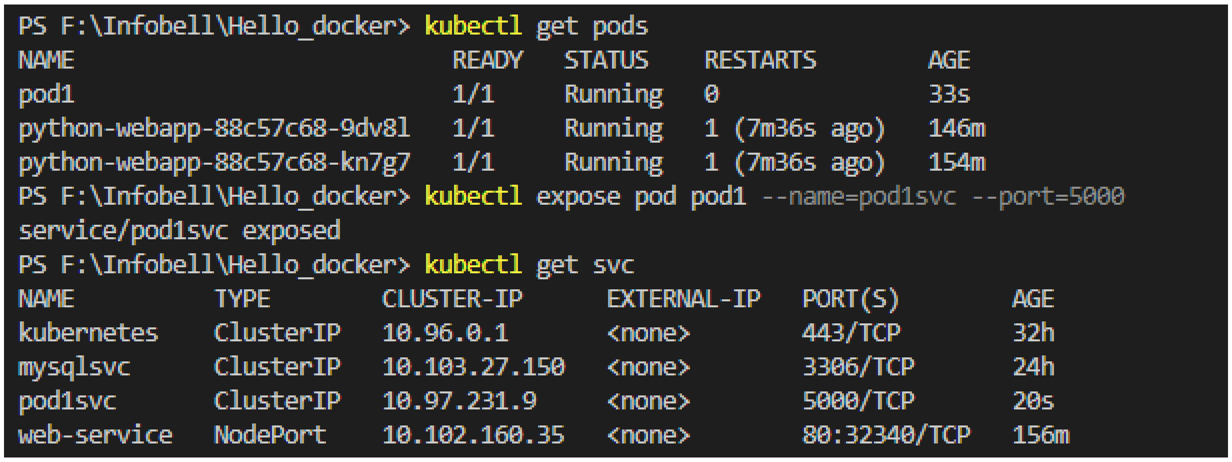
```
😄  minikube v1.30.1 on Microsoft Windows 10 Enterprise 10.0.19045.2846 Build 19045.2846
✨  Using the docker driver based on existing profile
👍  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
🤷  docker "minikube" container is missing, will recreate.
🔥  Creating docker container (CPUs=2, Memory=2200MB) ...
🐳  Preparing Kubernetes v1.26.3 on Docker 23.0.2 ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

Step 6: Create pod with image

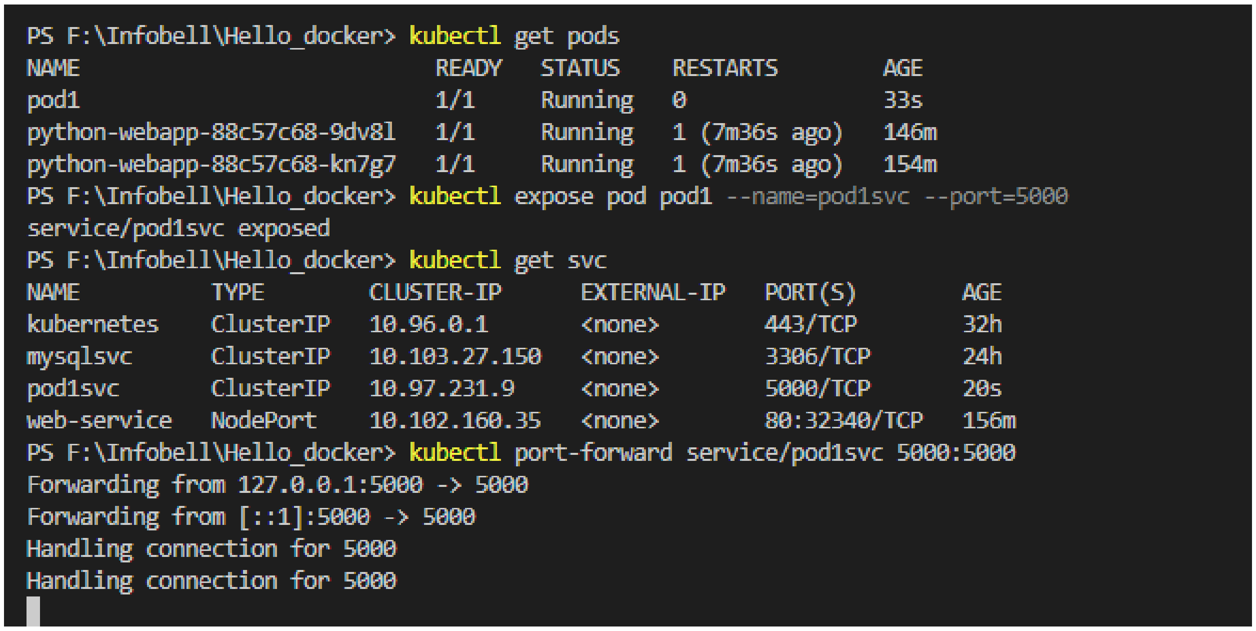


```
PS F:\Infobell\Hello_docker> kubectl run pod1 --image==rohan91196/helloapp:latest --port==5000
PS F:\Infobell\Hello_docker> kubectl run pod1 --image=rohan91196/helloapp:latest --port==5000
PS F:\Infobell\Hello_docker> kubectl run pod1 --image=rohan91196/helloapp:latest --port=5000
pod/pod1 created
PS F:\Infobell\Hello_docker> get pods
get : The term 'get' is not recognized as the name of a cmdlet, function, script file, or operable program. Check
the spelling of the name, or if a path was included, verify that the path is correct and try again.
+ get pods
+ ~~~
    + CategoryInfo          : ObjectNotFound: (get:String) [], CommandNotFoundException
    + FullyQualifiedErrorId : CommandNotFoundException

PS F:\Infobell\Hello_docker> kubectl get pods
NAME                          READY   STATUS    RESTARTS        AGE
pod1                          1/1     Running   0               33s
python-webapp-88c57c68-9dv8l  1/1     Running   1 (7m36s ago)   146m
python-webapp-88c57c68-kn7g7  1/1     Running   1 (7m36s ago)   154m
```

Step 7: Create service


```
PS F:\Infobell\Hello_docker> kubectl get pods
NAME                            READY   STATUS    RESTARTS        AGE
pod1                            1/1     Running   0               33s
python-webapp-88c57c68-9dv8l    1/1     Running   1 (7m36s ago)   146m
python-webapp-88c57c68-kn7g7    1/1     Running   1 (7m36s ago)   154m
PS F:\Infobell\Hello_docker> kubectl expose pod pod1 --name=pod1svc --port=5000
service/pod1svc exposed
PS F:\Infobell\Hello_docker> kubectl get svc
NAME          TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes    ClusterIP   10.96.0.1       <none>        443/TCP        32h
mysqlsvc      ClusterIP   10.103.27.150   <none>        3306/TCP       24h
pod1svc       ClusterIP   10.97.231.9     <none>        5000/TCP       20s
web-service   NodePort    10.102.160.35   <none>        80:32340/TCP   156m
```

Step 8: Create port forwarding


```
PS F:\Infobell\Hello_docker> kubectl get pods
NAME                            READY   STATUS    RESTARTS        AGE
pod1                            1/1     Running   0               33s
python-webapp-88c57c68-9dv8l    1/1     Running   1 (7m36s ago)   146m
python-webapp-88c57c68-kn7g7    1/1     Running   1 (7m36s ago)   154m
PS F:\Infobell\Hello_docker> kubectl expose pod pod1 --name=pod1svc --port=5000
service/pod1svc exposed
PS F:\Infobell\Hello_docker> kubectl get svc
NAME          TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes    ClusterIP   10.96.0.1       <none>        443/TCP        32h
mysqlsvc      ClusterIP   10.103.27.150   <none>        3306/TCP       24h
pod1svc       ClusterIP   10.97.231.9     <none>        5000/TCP       20s
web-service   NodePort    10.102.160.35   <none>        80:32340/TCP   156m
PS F:\Infobell\Hello_docker> kubectl port-forward service/pod1svc 5000:5000
Forwarding from 127.0.0.1:5000 -> 5000
Forwarding from [::1]:5000 -> 5000
Handling connection for 5000
Handling connection for 5000
```