

## **Assignment-4**

### Cloud App Development

Assignment Date	7 September 2022
Student Name	Mahalakshmi N
Student Roll Number	812619104015
Maximum Marks	2 Marks

### **Questions:**

- 1.Pull an Image from docker hub and run it in docker playground.
- 2.Create a docker file for the jobportal application and deploy it in Docker desktop application.
- 3.Create a IBM container registry and deploy helloworld app or jobportalapp.
- 4.Create a Kubernetes cluster in IBM cloud and deploy helloworld image or jobportal image and also expose the Image and also expose the same app to run in nodeport.

### **Dockerfile**

FROM python:3.10-buster

WORKDIR /app

COPY . .

RUN pip install --no-cache-dir -r requirements.txt

CMD ["gunicorn", "--bind", "0.0.0.0:5000", "app:app"]

## App.py

```
import uuid

from flask import Flask

instance_id = uuid.uuid4().hex

app = Flask(__name__)

@app.route("/")

def get_instance_id():

    return f"<b style='font-size:30px;color:red;'>Instance ID: {instance_id}</b>"

if __name__ == "__main__":

    app.run(port=5000, host="0.0.0.0")
```

## desktop.yml

```
apiVersion: v1

clusters:

- cluster:

    certificate-authority-data:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUMvakNDQWVhZ0F3SUJBZ0lCQURBTklna3Foa2lHO
XcwQkFRc0ZBREFWTVJNd0VRWURWUUVFERXdwcmRXSmwKY201bGRHVnpNQjRYRFRJeU1UQXdOVE
UxTWpjME9Gb1hEVE15TVRBd01qRTFNamMwT0Zvd0ZURVRNQkVHQTFVRQpBeE1LYTNWaVpYSnVa
WFJsY3pDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBERNDQVFvQ2dnRUJBS3J5CnM2cC9iZT
RBaTIPNmY3VjBuRHFYLORoai9oQmlyWEhIT2dlSHJkNWs0V1NjRTVzeUtWdERYdUx3MVFrUEJlVWkwK
enINWWQ2OEFSRDdnaGxpYzRhUnFJV09vOVNSVEVSaUo0RDJqZWJGWFR0d1BpZGFEUFIEVm85UIFB
S0hBQXliMQpSZmMvT2FkaEEvb2FHNGxLSFRrQUlKV1JlclFCbThTWFEhTlhmektVVTv6NElJbE5CNm9Yc
UtmbGJpbDBxMG1UClkxSndWQ3plVFVXcXo1OUFNvKfVQ01RQkhXcWZ5enQyODVkoOVN2YVpyUTF3
VVBqY2Zzc0ZxTUNPeUx3MzI5dkMKVlhienFLMDA5bzVkbWsvZXZJQUxSNmRKL3NNMTg5T2k4bWVQ
1NLMzVUU2xJNVlOM2lML3l1UVd4aVFUb0Rwawo3OU9wQjJZQ25DQTdRZWVhR3JFQ0F3RUFBYU5aT
```

UZjd0RnWURWUjBQQVFIL0JBUURBZ0trTUE4R0ExVWRFd0VCCi93UUZNQU1CQWY4d0hRWURWUjB  
PQkJZRUZNZHhDVVdRSTVGRmQvU0pwZEd0cUNVL25kMVINQIVHQTfVZEVruU8KTUF5Q0NtdDFZbV  
Z5Ym1WMFpYTXdEUVIKS29aSWH2Y05BUUVMQIFBRGdnRUJBRHIIYnJXa3ZITDU1ZjRQaFI5UwpycER2  
UnQwZ1IMZH4RXJSTkJPNFFGSUJXTUhxRDFTMIVITWNzMENnUGpQQ3ptWnUybzNCNkp4eUpkWU  
wrRWtjCjJQYkFuWXdpQ3AzaEwxN2NJcXZ1bXNDcng3MVpGdEINNfhtTmd0bjFiTnpWR3QvMzdWaW  
NsNjdsTFVydnp0RUcKOFcwMVRoQ2hwVlJiQmptYStOZ295SGNuNFJZY0dJTHZFTVNQa1BZRXIOVTN5  
MmtXRkgwaElMMIZRYmJzUIYxTApJOElVM3VQZEV3ZktLN0dNeW85VGoxc3NsRWVBTnFUeWVUYzg1  
MHIZYThYekxVTzloM1kwVWc4a2c1NIFaZVJUcNE1a2lGQ3p6OGRjZi9xN2tHN0RBaENvcFhhc3VQYIV5T  
nNvQVRBOTVRdW83MEQzdDNCQ1F2OTlvV2NjNnBKRW0KVUtrPQotLS0tLUVORCBDRVJUSUZJQ0FU  
RS0tLS0tCg==

server: https://kubernetes.docker.internal:6443

name: docker-desktop

contexts:

- context:

cluster: docker-desktop

user: docker-desktop

name: docker-desktop

current-context: docker-desktop

kind: Config

preferences: {}

users:

- name: docker-desktop

user:

client-certificate-data:

LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSURRakNDQWlxZ0F3SUJBZ0lUJ3INR3o4K1IDK013RFF  
ZSkvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVmIaWEp1WlhSbGN6QWVGdzB5TW  
pFd01EVXh0VEkzTkRoYUZY3MHINekV3TURreE16TXdNVFJhTURZeApGekFWQmdOVKBb1REBk41YzNS  
bGJUcHRZWE4wWlhiKek1Sc3dHUVIEVIFRREV4SmtiMk5yWlhJdFptOXIMV1JsCmMydDBiM0F3Z2dFaU  
1BMEdDU3FHU0liM0RRRUJBUVVBQTRJQkR3QXdnZ0VLQW9JQkFRRRHR2T1VUeElOZmw3MG8KK1hp  
R1o1d1VLbVZ6dnFPOW9ycXB4OE15aXViMUQ0dXkyMHIScUhPTEVwRHkxalpPV0szUHBGTUxGcXQ3Q  
mZDLWpVUGJPN3pGbXBQY2VuVkhmSThqRmRSRUlnQ0U3a2dJNGdqMzNzRlVWbklocHUbIHZQWpqS  
3lIRkYrTFphbyt3Ck5nUTI4d0xiQTkrQlY5ZXJYMKZObHoxRXBnNi9CYWNIMDVaQnJGNGVhR25VdUpQT  
UI3L0NFVHVSajVJRW5KUUnIKUW1YSnBJdVo4TXkzbnJseXVGZzNIYzNaNW91YU10c3B4NEtBSXRyMXhw  
ZUxLU21VTGNISTBIUkhLEZVmm3UwRgp5akNaWEFoZjRlREo2MitJUXZ2UmJ3Y0x3eGFYRytMTIVNMU  
5zdHhpamtOSE9PSUZ1R1JFcIIlSRFlpTHByVk55CjhaMEhCelZGQWdNQkFBR2pkVEJ6TUE0R0ExVWREd0  
VCL3dRRUF3SUZvREFUQmdOVkhTVUVEREFQmdnckJnRUYKQlFjREFqQU1CZ05WSFJNQkFmOEVBakF  
BTUI4R0ExVWRJd1FZTUJhQUZnZ0VdRSTVGRmQvU0pwZEd0cUNVLWpuZDFZTUlwR0ExVWRFUUV  
FXTUJTQ0VtUnZMnRsY2kxbWlzSXRaR1Z6YTNSdmNEQU5CZ2txaGtpRzl3MEJBUXNGCkFBT0NBUEVB  
SCTeLzQ4NXkweWI4cVlxd3U4Q2k3K1phbkIUUVdib1MvaWJObjQxNlpmDDZRRUxhZlVxTHo3S3EKcWh  
FamFGMkhvQ1RrTDAyUVR1WXlPaFhFQlF1eTI0UkFkTml2d2VObW1qc3lyZ2Ric25ua1hCQld6RU80Uz

VySwpCbVvkzOENxUWxFL1djdkTxb0hmVTlvZU1obzYvQlIPc1gxQ1JKVVI3NGFNU2JzNTQybGs0UnQrb3I  
vRmxZVXU1CmJYTWf3QUNhT0xoNUF5ZW9sS0xrOFV0aFg3VHIHTy9scVJWaWN5dlhvcXdPSFZXWENB  
NklzTFR5SUVGUExxNGIKdlRjMFhmcFpyeWVYNDd1QVZyUHY1ODdib1B3WTNtdTaRTtg5RzFwM0Ztd2  
FDRIV1bGJJaVZUNW9mWjRSS1VpOQpFTWWhhK0x4czBibTFFeUtrLzdZY3pJbE8yVjdkb3c9PQotLS0tLUV  
ORCBDRVJUSUZJQ0FURS0tLS0tCg==

client-key-data:

LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUIFb2dJQkFBS0NBUEVBN2J6bEU1U0RYNWU5S  
1BsNghtZWNGQ3BsYzc2anZhSzZxY2ZETW9ybTIRK0xzdHRNCmthaHppeEtROHRZMIRsaXR6NIJUQ3hh  
cmV3WHd2MUQyenU4eFpxVnducDFSM3IQSXhYVJVJDSUFoTzVJQ09JSTkKOTdCVkZaeUlhMTA1MEdB  
STR5c2h4UmZpMldxUHNEWUVQY2NDMndQZmdWZlhxMTloVFpjOVJLWU92d1duQjlpVwpRYXhISG1  
ocDFMaVR6QWUvd2hFN2tZK1NCSnlVYTBKbHlU0xtZkRNdDU2NWNyaFIOeDNOMmVhTG1qTGJLY2  
VDCmdDTGE5Y2FYaXlrcGxDM0J5TKlwUnluUkg5N3RCY293bVZ3SVgrQnd5ZXR2aUVMNzBXOEhDOE1X  
bHh2aXpWRE4KVGJMY1lvNURSemppQmJoa1JLMkVVMklpNmExVGN2R2RCd2MxUIFJREFRQUJBb0lC  
QUhVWnV4aEc2ZUwzSWZTQgoxUzBXOGRrNC9ObmpTRklMU3JpQUxoVENPSHJ6R2xPU0U0Z1htam9  
oZmdZdkgzYTkxWGRxc0pUZ3FMeWpNMjk1CkdPamJCdEZSb2tSdXB6b3JMU1Nkd2pNV3NjZmU1bnN  
oSkiUjQwY080RjklZBVMVdJdFUzb0J2Z0l5cElYZWgKVTd0K1FGalM4bytLVGpBYWZob3k2Ym5idFoyK  
zN0Nnhad3dqelJYcldIRW9DNVZRRm1tbXNKWnFXyVVBBeEZSUGpQdTFCEwPqGQnlvd1MxVVRSMnhIV2  
xtb1J1Y3RGeS9tZDJNcXZ1bngzb2JHSHVdWwXGUDNVdXhrbFERbE5LWDZvCjY3SExCOWRKUDNRNWt1  
WjFROUNwclplNmhCSjZ2NHdSejdzC0luTkD3Q2FraI90NWtUVIR4bXFic0tEU0hXWWcKN2taZHowMEN  
nWUVBNzBldW93MGdSRIRvWG5SS0hYQzM1aUJ2V3VjL2NyeXRFKzZJaEo1c1pQT05OWG01dGNJRQo  
ybEZ2NUJ1MnRtUG5sOS9wcmNDL1IRUklrQldKbCt1ZmpPdFRHaWVWUi81dTdiYwVsOU0xNDBvTUZp  
L2c4YzlkCjJlU3JCdW1oUUVXTXN2U2NmN2hXZDRrSnBDOE5rMnNpemJSa1pWVVV6NWN0UGxYV1lIQ  
0pCMWNDZ1lFQS9sbWcKV3o5UFdqVW95cTlMaWdGYVBiYnd2ajhtaHVkN3RVtMfjWk9rQWgxS05qT  
W9oK0phSmpLZIY4MmVRNIJ6TUxSVgpbRbEV6TSs2N1NqZEpreGxPdGdNR3VPcDd5dUJ5bHM4dVRsK2  
11UVFHYjZFWIA1NkdrY2MvQTZhei9kME1RYVRJCKNRNnNOMDFuSUFQaXRTS2NCYTAvV0QvRkVwdG  
hhRGJLUzIQaGtzTUNnWUJaWm1qalZyWWxrYXU0RHE3YVRSRUkKbERuR3R2dkh2S0xWL1p0WUIYRF  
NUd1VGWtIOMUMyYVF3emlRKzZIS2g0MHgrMkR0b0VSzZFSN2Zab3lnREZMeQpiY0lBMWfVnJN0OU  
d6aEwzR3R6WmJRTVREeFlyaEV3RHk1MnkyWHQ5ekR5RGFFMUlrdkVMZkZlejlvaXdpYWpaClBucjhPV  
XVyTy82Z3VhVjIKczZyEFFLQmdFNFVDVXU1ZFI5RmdRNG5lWk14c3hmNnpwQTRKb1dIRGN5bzdseXIK  
SzgrU1dqT2xKSzRCNkJhTnQ2aG1jRXpuY0l6azdyODYyZkw1Ti9raVc2MlNPVDFhVkhmelV5a0dHV2lrR  
m9McwppqZXVCdHRBUnJnUi9qMk8wdkQxTXNUZ2U0U0pOaVRKdWxLTXRXWmpVVOIXdUtDTDkrZEI  
VVdIUzNPWDIyOGJlCk5obXRBB0dBR3ZPeXR3emRIUGR0TzNscFowS3pibnF6NkQ3SHpHblYyaGdCWU  
NvTDdQaVpHVWMrDhdTOFJwVW4KU29PYmsvOVM1S3Y0NXNITnRjTjd5VS8wVjhpUUFwUmZkVDVSY  
nNjT3ZrUIFSQmo1Vkja0pMVIppeS85QmJDbgpXNU1ZV0NqSi9lWHk0ZGVMdnZFckRJVVHhDRFVYbkN  
WRU1ReUIXcDB4T2Y4b1lla2h6U009Ci0tLS0tRU5EIFJTQSBQUkVQVRFIeTfWS0tLS0tCg==

## razorops.yaml

version: v2

tasks:

build:

runner: razorci/python:3.9

steps:

- checkout
- cache/pull: pip-deps-{{ checksum "requirements.txt" }}
- commands:
  - python -c 'import site; print(site.getsitepackages())'
  - pip install -r requirements.txt
- commands:
  - python test.py
- cache/push:
  - key: pip-deps-{{ checksum "requirements.txt" }}
- paths:
  - /usr/local/lib/python3.9/site-packages
- reports/junit: test-reports/\*.xml

## Requirements.txt

flask

gunicorn

## Outputs:



03:56:24

CLOSE SESSION

Instances

+ ADD NEW INSTANCE

192.168.0.8  
node1

cdagodu0\_cdagolu0qau0009n309g

IP  
192.168.0.8

OPEN PORT

MemoryCPU

SSH  
ssh ip172-18-0-20-cdagodu0qau0009n3080@direct.labs.play-with-d

DELETEEDITOR

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
00183bfe1385	mongo	"docker-entrypoint.s..."	15 seconds ago	Up 13 seconds	27017/tcp, 0.0.0.0:6000->3001/

```
[node1] (local) root@192.168.0.8 ~  
$ docker start mongodb  
mongodb  
[node1] (local) root@192.168.0.8 ~  
$ docker images  
REPOSITORY TAG IMAGE ID CREATED SIZE  
mongo latest 1cca5cf68239 2 weeks ago 695MB  
[node1] (local) root@192.168.0.8 ~  
$ docker ps -a  


| CONTAINER ID | IMAGE | COMMAND                  | CREATED            | STATUS            | PORTS                             |
|--------------|-------|--------------------------|--------------------|-------------------|-----------------------------------|
| 00183bfe1385 | mongo | "docker-entrypoint.s..." | About a minute ago | Up About a minute | 27017/tcp, 0.0.0.0:6000->3001/tcp |

[node1] (local) root@192.168.0.8 ~  
$
```

03:53:15

CLOSE SESSION

Instances

+ ADD NEW INSTANCE

192.168.0.8  
node1

cdagodu0\_cdagolu0qau0009n309g

IP  
192.168.0.8

OPEN PORT

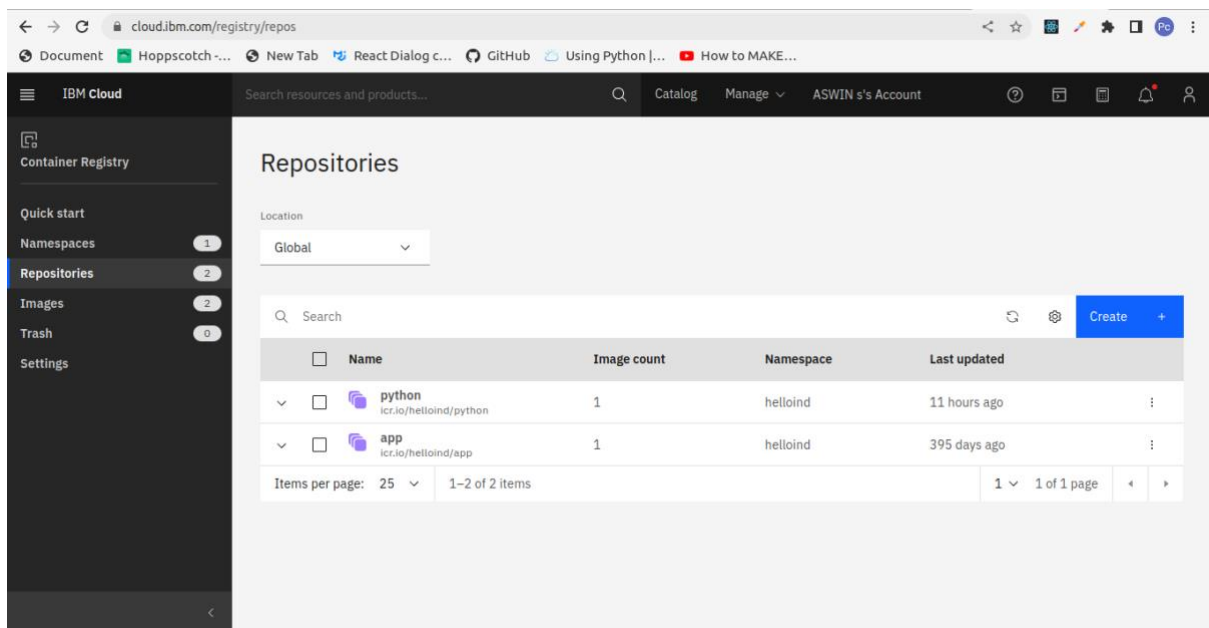
MemoryCPU

SSH  
ssh ip172-18-0-20-cdagodu0qau0009n3080@direct.labs.play-with-d

DELETEEDITOR

```
$ docker run -d -p6010:3000 hello-world  
9b7c8f4adb31df3a7ceaf431f63e2eae868a579059402bc72b630b7e5e911875  
^[[D[node1] (local) root@192.168.0.8 ~  
$ docker run -d -p6020:3000 hello-world  
6b0f1699a8ced253d9df780a42eeff298a0c2fb2c6389e709681da2e63879b0b  
[node1] (local) root@192.168.0.8 ~  
$ docker run -p6120:3000 hello-world  
  
Hello from Docker!  
This message shows that your installation appears to be working correctly.  
  
To generate this message, Docker took the following steps:  
1. The Docker client contacted the Docker daemon.  
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
(amd64)
```

```
root@admin:/home/ash/Desktop/Flask ClassWorrk# docker run -p5000:5000 flakapp:1.0
Unable to find image 'flakapp:1.0' locally
docker: Error response from daemon: pull access denied for flakapp, repository does not exist or may require 'docker login': denied: re
quested access to the resource is denied.
See 'docker run --help'.
root@admin:/home/ash/Desktop/Flask ClassWorrk# docker run -p5000:5000 -t flakapp:1.0
Unable to find image 'flakapp:1.0' locally
docker: Error response from daemon: pull access denied for flakapp, repository does not exist or may require 'docker login': denied: re
quested access to the resource is denied.
See 'docker run --help'.
root@admin:/home/ash/Desktop/Flask ClassWorrk# docker run -p5000:5000 -t flakapp
Unable to find image 'flakapp:latest' locally
docker: Error response from daemon: pull access denied for flakapp, repository does not exist or may require 'docker login': denied: re
quested access to the resource is denied.
See 'docker run --help'.
root@admin:/home/ash/Desktop/Flask ClassWorrk# docker run -p5000:5000 -t flaskapp:1.0
docker: Error response from daemon: driver failed programming external connectivity on endpoint silly_bardeen (717385e0278ed4890d3355b9
d59728f1e3577afe801b18edb025c07b377889d): Bind for 0.0.0.0:5000 failed: port is already allocated.
ERRO[0002] error waiting for container: context canceled
root@admin:/home/ash/Desktop/Flask ClassWorrk# docker run -t flaskapp:1.0
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.3:5000
Press CTRL+C to quit
172.17.0.1 - - [23/Oct/2022 17:21:29] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [23/Oct/2022 17:21:29] "GET /favicon.ico HTTP/1.1" 404 -
172.17.0.1 - - [23/Oct/2022 17:21:50] "GET / HTTP/1.1" 200 -
```



```
root@admin:/home/ash/Desktop... root@admin:/home/ash/Desktop... root@admin:/home/ash/Desktop... root@admin:/home/ash/Desktop... root@admin:/home/ash/Desktop...
node 13-alpine3.10 9e3fc25f69d2 2 years ago 114MB
postgres 10.10 9a05a2b9e69f 3 years ago 211MB
root@admin:/home/ash/Desktop/Flask ClassWorrk# docker tag flaskapp:1.0 icr.io/helloind/python:1.0
root@admin:/home/ash/Desktop/Flask ClassWorrk# docker push icr.io/helloind/python:1.0
The push refers to repository [icr.io/helloind/python]
f7a508b1c151: Pushed
b7b353df37a8: Pushing [=====] 12.38MB
7216ce2329ba: Pushed
0f61ccc2347a: Pushed
bfc1deb8136e: Pushed
1f123186824c: Pushed
3d6eb1152931: Pushing [=====>] 7.748MB/57.12MB
100796cdf3b1: Pushing [=====] 11.14MB/18.48MB
54acb5a6fa0b: Pushing [>] 4.294MB/528.7MB
8d51c618126f: Pushing [>] 1.063MB/152MB
9ff6e4d46744: Waiting
a89d1d47b5a1: Waiting
655ed1b7a428: Waiting
^Croot@admin:/home/ash/Desktop/Flask ClassWorrk# docker push icr.io/helloind/python:1.0
The push refers to repository [icr.io/helloind/python]
f7a508b1c151: Layer already exists
b7b353df37a8: Layer already exists
7216ce2329ba: Layer already exists
0f61ccc2347a: Layer already exists
bfc1deb8136e: Layer already exists
1f123186824c: Layer already exists
3d6eb1152931: Pushed
100796cdf3b1: Pushed
54acb5a6fa0b: Pushed
8d51c618126f: Pushed
9ff6e4d46744: Pushed
a89d1d47b5a1: Pushed
655ed1b7a428: Pushed
1.0: digest: sha256:9d7b9f369391c18d6aa88270ebb467ab6b93b3f18e61a46c9467b7fa5cc4b404 size: 3050
root@admin:/home/ash/Desktop/Flask ClassWorrk# _
```

## Kubernetes

### Dashboard-adminuser.yaml

apiVersion: v1

kind: ServiceAccount

metadata:

name: admin-user

namespace: kubernetes-dashboard

---

apiVersion: v1

kind: Secret

metadata:

name: admin-user-token

namespace: kubernetes-dashboard

annotations:

kubernetes.io/service-account.name: admin-user

type: kubernetes.io/service-account-token



## **flask\_deployment.yaml**

apiVersion: apps/v1

kind: Deployment

metadata:

name: flask-app

spec:

replicas: 3

selector:

matchLabels:

app: flask-app

template:

metadata:

labels:

app: flask-app

spec:

containers:

- name: flask-app-container

image: flask-app-testing

imagePullPolicy: Never

ports:

- containerPort: 5000

protocol: TCP

## **flask\_ingress.yaml**

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

```
name: flask-app-ingress

annotations:

  kubernetes.io/ingress.class: nginx
  nginx.ingress.kubernetes.io/ssl-redirect: "false"
```

```
spec:

  # ingressClassName: nginx

  rules:

    - http:

      paths:

        - backend:

            service:

              name: flask-app-service

              port:

                number: 5000

            path: /

            pathType: Prefix
```

## **flask\_service.yaml**

```
apiVersion: v1
kind: Service
metadata:
  name: flask-app-service
spec:
  type: ClusterIP
  ports:
    - port: 5000
  selector:
    app: flask-app
```

---

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

name: admin-user

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: ClusterRole

name: cluster-admin

subjects:

- kind: ServiceAccount

name: admin-user

namespace: kubernetes-dashboard