

Assignment-4

Cloud App Development

Assignment Date	7 September 2022
Student Name	Ragul R
Student Roll Number	812619104033
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
XcwQkFRc0ZBREFWTVJNd0VRWURWUWFERXdwcmRXSmwKY201bGRHVnpNQjRYRFRJeU1UQXdOVE
UxTWpjME9Gb1hEVE15TVRBd01qRTFNamMwT0Zvd0ZURVRNQkVHQTFVRQpBeE1LYTNWaVpYSnVa
WFJsY3pDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBS3J5CnM2cC9iZT
RBaTlPNmY3VjBuRHFYLORoai9oQmlyWEhIT2dISHJkNWs0V1NjRTRVzeUtWdERYdUx3MVFrUEJlVWkwK
enINWWQ2OEFSRDdnaGxpYzRhUnFJV09vOVNSVEVSaUo0RDJqZWJGWFR0d1BpZGFUEUFIEVm85UjFB
S0hBQXliMQpSZmMvT2FkaEEvb2FHNGxLSFRrQUlKV1JlclFCbThTWFEhTlhmekmtVTV6NElJbE5CNm9Yc
UtmBjGjpbDBxMG1UClkxSndWQ3plVFVXcXo1OUFNvKfVQ01RQkhXcWZ5enQyODVhOVN2YVpyUTF3
VVBqY2Zzc0ZxTUNPeUx3MzI5dkMKVlhienFLMDA5bzVkbWsvZXZJQUxSNmRKL3NNMTg5T2k4bWFFVQ
1NLMzVUU2xjNVlOM2IML3l1UVd4aVFUb0Rwawo3OU9wQjJZQ25DQTdRZWVhR3JFQ0F3RUFBYU5aT
```

UZjd0RnWURWUjBQVVFIL0JBUURBZ0trTUE4R0ExVWRFd0VCCi93UUZNQU1CQWY4d0hRWURWUjB
PQkJZRUZNZHhDVVdRSTVGRmQvU0pwZEd0cUNVL25kMVINQIVHQTFVZEVRUU8KTUF5Q0NtdDFZbV
Z5Ym1WMFpYTXdEUVIKS29aSWH2Y05BUUVMQIFBRGdnRUJBRHIIYnJXa3ZITDU1ZjRQaFI5UwpycER2
UnQwZ1IMZH4RXJSTkJPNFFGSUJXTUhxRDFTMIVITWNzMENnUGpQQ3ptWnUybzNCNkp4eUpkWU
wrRWtjCjJQYkFuWXdpQ3AzaEwxN2NJcXZ1bXNDcng3MVpGdEINNFhtTmd0bjFiTnpWR3QvMzdWaW
NsNjdsTFVydnp0RUcKOFcwMVRoQ2hwVlJiQmptYStOZ295SGNuNFJZY0dJTHZFTVNQa1BZRXLIOVTN5
MmtXRkgwaElMMIZRYmJzUIYxTApJOElVM3VQZEV3ZktLN0dNeW85VGoxc3NsRWVBTnFUeWVUYzg1
MHIZYThYekxVTzloM1kwVWc4a2c1NIFaZVJUCnE1a2lGQ3p6OGRjZi9xN2tHN0RBaENvcFhhc3VQYIV5T
nNvQVRBOTVRdW83MEQzdDNCQ1F2OTIvV2NjNnBKRW0KVUtrPQotLS0tLUVORCBDRVJUSUZJQ0FU
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:

LS0tLS1CRUdJTiBDRVJUSUZJQ0FUR50tLS0tCk1JSURRakNDQWlxZ0F3SUJBZ0lJUS1NR3o4K1IDK013RFF
ZSktdWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVmIaWEp1WlhSbGN6QWVGdzB5TW
pFd01EVXh0VEkzTkRoYUZZ3MHIkV3TUReE16TXdNVFJhTURZeApGekFWQmdOVKBb1REbk41YzNS
bGJUcHRZWE4wWlhiKek1Sc3dHUUVIEVFRREV4SmtiMk5yWlhJdFptOXIMV1JsCmMydDBiM0F3Z2dFaU
1BMEdDU3FHU0liM0RRRUJBUVVBQTRJQkR3QXdnZ0VLQW9JQkFRRRHR2T1VUbElOZmw3MG8KK1hp
R1o1d1VLbVZ6dnFPOW9ycXB4OE15aXViMUQ0dXkyMHlScUhPTEVwRHkxalpPV0szUHBGTUxGcXQ3Q
mZDLwpVUGJPN3pGbXBYQ2VuVkhmSThqRmRSRUlnQ0U3a2dJNGdqMzNzRlVWbklocHUbIHZQWpqS
3lIRkYrTFphbyt3Ck5nUTI4d0xiQTkrQlY5ZXJYMKZObHoxRXBnNi9CYWNIMDVaQnJGNGVhR25VdUpQT
UI3L0NFVHVSajVJRW5KUUnIKUW1YSnBJdVo4TXkzbnJseXVGZzNiYzNaNW91YU10c3B4NEtBSXRyMXhw
ZUxLU21VTGNISTBIUkhLEZVmm3UwRgp5akNaWEFoZjRIRiEo2MitJUXZ2UmJ3Y0x3eGFYRytMTIVNMU
5zdHhpamtOSE9PSUZ1R1JFcllSRFlpTHByVk55CjhaMEhCelZGQWdNQkFBR2pkVEJ6TUE0R0ExVWREd0
VCL3dRRUF3SUZvREFUQmdOVkhTVUVEREFLQmdnckJnRUyKQlFjREFqQU1CZ05WSFJNQkFmOEVBakF
BTUI4R0ExVWRJd1FZTUJhQUZnZ0VdRSTVGRmQvU0pwZEd0cUNVLWpuZDFZTUlwR0ExVWRFUUV
FXTUJQT0VtUnZMnRsY2kxbWlzSXRaR1Z6YTNSdmNEQU5CZ2txaGtpRzl3MEJBUXNGCkFBT0NBUEVB
SCElZQ4NXkweWI4cVlxd3U4Q2k3K1phbkliUUVdlb1MvaWJObjQxNlpMdDZRRUxhZlVxTHo3S3EKcWh
FamFGMkhvQ1RrTDAyUVR1WXlPaFhFQlF1eTI0UkFkTml2d2VObW1qc3lyZ2Ric25ua1hCQld6RU80Uz

runner: razorci/python:3.9

steps:

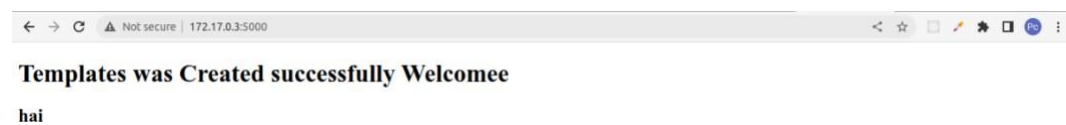
- checkout
- cache/pull: pip-deps-{{ checksum "requirements.txt" }}
- commands:
 - python -c 'import site; print(site.getsitepackages())'
 - pip install -r requirements.txt
- commands:
 - python test.py
- cache/push:
 - key: pip-deps-{{ checksum "requirements.txt" }}
- paths:
 - /usr/local/lib/python3.9/site-packages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

Memory

CPU

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

DELETE

EDITOR

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/
tcp mongodb					
{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, 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

Memory

CPU

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

DELETE

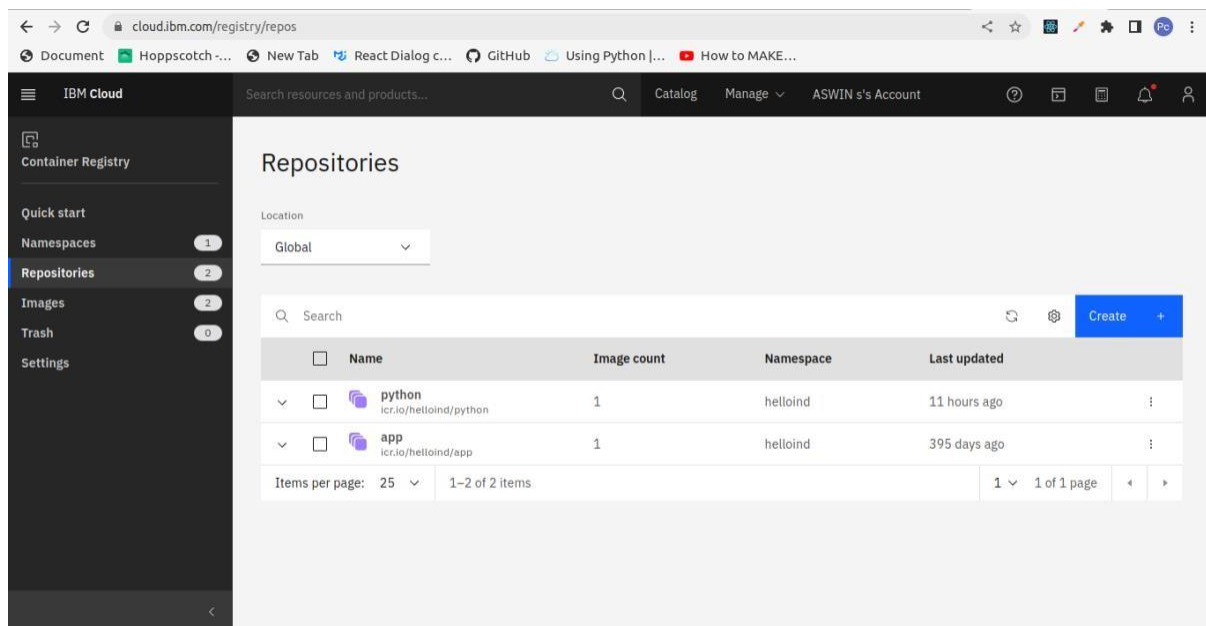
EDITOR

```
$ 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: requested 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: requested 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: requested 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 (717385e0278ed4890d3355b9d59728f1e3577afe801b18edb025c07b377889d): 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