Assignment-4

Cloud App Development

Assignment Date	7 September 2022
Student Name	Mohamed Hanifa M
Student Roll Number	812619104018
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:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUMvakNDQWVhZ0F3SUJBZ0lCQURBTkJna3Foa2lHO
XcwQkFRc0ZBREFWTVJNd0VRWURWUVFERXdwcmRXSmwKY201bGRHVnpNQjRYRFRJeU1UQXdOVE
UxTWpjME9Gb1hEVE15TVRBd01qRTFNamMwT0Zvd0ZURVRNQkVHQTFVRQpBeE1LYTNWaVpYSnVa
WFJsY3pDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBS3J5CnM2cC9iZT
RBaTlPNmY3VjBuRHFYL0Roai9oQmlYWEhlT2dlSHJkNWs0V1NjRTVzeUtWdERydUx3MVFrUEJVWkwK
enlNWWQ2OEFSRDdnaGxpYzRhUnFJV09vOVNSVEVSaUo0RDJqZWJGWFR0d1BpZGFEUFlEVm85UlFB
S0hBQXliMQpSZmMvT2FkaEEvb2FHNGxLSFRrQUlKV1JlclFCbThTWFExTlhkemtVVTV6NElJbE5CNm9Yc
UtmbGJpbDBxMG1UClkxSndWQ3plVFVXcXo1OUFNVkFvQ01RQkhXcWZ5enQyODVkOVN2YVpyUTF3
VVBqY2Zzc0ZxTUNPeUx3MzI5dkMKVlhienFLMDA5bzVkbWsvZXZJQUxSNmRKL3NNMTg5T2k4bWFVQ
1NLMzVUU2xJNVI0M2lML3l1UVd4aVFUb0Rwawo3OU9wQjJZQ25DQTdRZWhxR3JFQ0F3RUFBYU5aT
```

```
UZjd0RnWURWUjBQQVFIL0JBUURBZ0trTUE4R0ExVWRFd0VCCi93UUZNQU1CQWY4d0hRWURWUjB
PQkJZRUZNZHhDVVdRSTVGRmQvU0pwZEd0cUNVL25kMVlNQlVHQTFVZEVRUU8KTUF5Q0NtdDFZbV
Z5Ym1WMFpYTXdEUVlKS29aSWh2Y05BUUVMQlFBRGdnRUJBRHllYnJXa3ZITDU1ZjRQaFI5UwpycER2
UnQwZ1lMZHZ4RXJSTkJPNFFGSUJXTUhxRDFTMlVITWNzMENnUGpQQ3ptWnUybzNCNkp4eUpkWU
wrRWtjCjJQYkFuWXdpQ3AzaEwxN2NJcXZ1bXNDcng3MVpGdElNNFhtTmd0bjFiTnpWR3QvMzdWaW
NsNjdsTFVydnp0RUcKOFcwMVRoQ2hwVlJiQmptYStOZ295SGNuNFJZY0dJTHZFTVNQa1BZRXlOVTN5
MmtXRkgwaElMMlZRYmJzUlYxTApJOElVM3VQZEV3ZktLN0dNeW85VGoxc3NsRWVBTnFUeWVUYzg1
MHlZYThYekxVTzloM1kwVWc4a2c1NlFaZVJUCnE1a2lGQ3p6OGRjZi9xN2tHN0RBaENvcFhhc3VQYlV5T
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:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURRakNDQWlxZ0F3SUJBZ0lJU3lNR3o4K1lDK013RFF
ZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVmlaWEp1WlhSbGN6QWVGdzB5TW
pFd01EVXhOVEkzTkRoYUZ3MHlNekV3TURreE16TXdNVFJhTURZeApGekFWQmdOVkJBb1REbk41YzNS
bGJUcHRZWE4wWlhKek1Sc3dHUVlEVlFRREV4SmtiMk5yWlhJdFptOXlMV1JsCmMydDBiM0F3Z2dFaU
1BMEdDU3FHU0liM0RRRUJBUVVBQTRJQkR3QXdnZ0VLQW9JQkFRRHR2T1VUbElOZmw3MG8KK1hp
R1o1d1VLbVZ6dnFPOW9ycXB4OE15aXViMUQ0dXkyMHlScUhPTEVwRHkxalpPV0szUHBGTUxGcXQ3Q
mZDLwpVUGJPN3pGbXBYQ2VuVkhmSThqRmRSRUlnQ0U3a2dJNGdqMzNzRlVWbkloclhUblFZQWpqS
3lIRkYrTFphbyt3Ck5nUTl4d0xiQTkrQlY5ZXJYMkZObHoxRXBnNi9CYWNIMDVaQnJGNGVhR25VdUpQT
UI3L0NFVHVSajVJRW5KUnlKUW1YSnBJdVo4TXkzbnJseXVGZzNIYzNaNW91YU10c3B4NEtBSXRyMXhw
ZUxLU21VTGNISTBIUkhLZEVmM3UwRgp5akNaWEFoZjRIREo2MitJUXZ2UmJ3Y0x3eGFYRytMTlVNMU
5zdHhpamtOSE9PSUZ1R1JFcllSRFlpTHByVk55CjhaMEhCelZGQWdNQkFBR2pkVEJ6TUE0R0ExVWREd0
VCL3dRRUF3SUZvREFUQmdOVkhTVUVEREFLQmdnckJnRUYKQlFjREFqQU1CZ05WSFJNQkFmOEVBakF
BTUI4R0ExVWRJd1FZTUJhQUZNZHhDVVdRSTVGRmQvU0pwZEd0cUNVLwpuZDFZTUIwR0ExVWRFUV
FXTUJTQ0VtUnZZMnRsY2kxbWIzSXRaR1Z6YTNSdmNEQU5CZ2txaGtpRzl3MEJBUXNGCkFBT0NBUUVB
SCtELzQ4NXkweWl4cVIxd3U4Q2k3K1phbklUUVdIb1MvaWJObjQxNlpMdDZRRUxhZlVxTHo3S3EKcWh
FamFGMkhvQ1RrTDAyUVR1WXlPaFhFQlF1eTI0UkFkTml2d2VObW1qc3IyZ2Ric25ua1hCQld6RU80Uz
```

VySwpCbVkzOENxUWxFL1djdktXb0hmVTIvZU1obzYvQllPc1gxQ1JKVVl3NGFNU2JzNTQybGs0UnQrb3l
vRmxZVXU1CmJYTWF3QUNhT0xoNUF5ZW9sS0xrOFV0aFg3VHlHTy9scVJWaWN5dlhvcXdPSFZXWENB
NklzTFR5SUVGUExxNGIKdlRjMFhmcFpyeWVYNDd1QVZyUHY1ODdib1B3WTNtdTArTTg5RzFwM0Ztd2
FDRlV1bGJJaVZUNW9mWjRSS1VpOQpFTWhhK0x4czBibTFFeUtrLzdZY3pJbE8yVjdkb3c9PQotLS0tLUV
ORCBDRVJUSUZJQ0FURS0tLS0tCg==

client-key-data:

LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUlFb2dJQkFBS0NBUUVBN2J6bEU1U0RYNWU5S
1BsNGhtZWNGQ3BsYzc2anZhSzZxY2ZETW9ybTlRK0xzdHRNCmthaHppeEtROHRZMlRsaXR6NlJUQ3hh
cmV3WHd2MUQyenU4eFpxVnducDFSM3lQSXhYVVJDSUFoTzVJQ09JSTkKOTdCVkZaeUlhMTA1MEdB
STR5c2h4UmZpMldxUHNEWUVQY2NDMndQZmdWZlhxMTloVFpjOVJLWU92d1duQjlPVwpRYXhlSG1
ocDFMaVR6QWUvd2hFN2tZK1NCSnlVYTBKbHlhU0xtZkRNdDU2NWNyaFlOeDNOMmVhTG1qTGJLY2
VDCmdDTGE5Y2FYaXlrcGxDM0J5TkIwUnluUkg5N3RCY293bVZ3SVgrQnd5ZXR2aUVMNzBXOEhDOE1X
bHh2aXpWRE4KVGJMY1lvNURSemppQmJoa1JLMkVRMklpNmExVGN2R2RCd2MxUlFJREFRQUJBb0lC
QUhVWnV4aEc2ZUwzSWZTQgoxUzBXOGRrNC9ObmpTRklMU3JpQUxoVENPSHJ6R2xPU0U0Z1htam9
oZmdZdkgzYTkxWGRXc0pUZ3FMeWpNMjk1CkdPamJCdEZSb2tSdXB6b3JMU1Nkd2pNV3NjZmU1bnN
oSkJiUjQwY080RjlKLzBVMVdJdFUzb0J2Z0l5cElYZWgKVTd0K1FGalM4bytLVGpBYWZob3k2Ym5idFoyK
zN0Nnhad3dqelJYcldIRW9DNVZRRm1tbXNKWnFXYVVBeEZSUgpQdTFCeWpGQnIvd1MxVVRSMnhIV2
xtb1J1Y3RGeS9tZDJNcXZ1bngzb2JHSHVDaWxGUDNVdXhrbFErbE5LWDZvCjY3SExCOWRKUDNrNWt1
WjFROUNwclplNmhCSjZ2NHdSejdzc0luTkd3Q2Frai90NWtUVlR4bXFIc0tEU0hXWWcKN2taZHowMEN
nWUVBNzBldW93MGdSRlRvWG5SS0hYQzM1aUJ2V3VjL2NyeXRFKzZJaEo1c1pQT05OWG01dGNJRQo
ybEZ2NUJ1MnRtUG5sOS9wcmNDL1lRUklrQldKbCt1ZmpPdFRHaWVWUi81dTdIYWVsOU0xNDBvTUZp
L2c4YzlqCjJJU3JCdW1oUlVXTXN2U2NmN2hXZDRrSnBDOE5rMnNpemJSa1pWVVV6NWNOUGxYV1lIQ
0pCMWNDZ1lFQS9sbWcKV3o5UFdqVW95cTlMaWdGYVBiYnd2ajhtaHVkN3RVTmFjWk9rQWgxS05qT
W9oK0phSmpLZlY4MmVRNlJ6TUxSVgpRbEV6TSs2N1NqZEpreGxPdGdNR3VPcDd5dUJ5bHM4dVRsK2
11UVFHYjZFWlA1NkdrY2MvQTZhei9kME1RYVRJCkNRNnNOMDFuSUFQaXRTS2NCYTAvV0QvRkVwdG
hhRGJLUzlQaGtzTUNnWUJaWm1qalZyWWxrYXU0RHE3YVRSUkKbERuR3R2dkh2S0xWL1p0WUlYRF
NUd1VGWTlOMUMyYVF3emlRKzZIS2g0MHgrMkR0b0VSZzFSN2Zab3lnREZMeQpiY0lBMWFVNjN0OU
d6aEwzR3R6WmJRTVREeEFIyaEV3RHk1MnkyWHQ5ekR5RGFFMUlrdkVMZkZIejIvaXdpYWpaClBucjhPV
XVyTy82Z3VhVjlKczZyeFFLQmdFNFVDVXU1ZFl5RmdRNG5IWk14c3hmNnpwQTRKb1dIRGN5bzdseXIK
SzgrU1dqT2xKSzRCNkJhTnQ2aG1jRXpuY0l6azdyODYyZkw1Ti9raVc2MlNPVDFhVkhmelV5a0dHV2IrR
m9McwpqZXVCdHRBUnJnUi9qMk8wdkQxTXNUZ2U0U0pOaVRKdWxLTXRXWmpVV0lXdUtDTDkrZEll
VVdIUzNPWDIyOGJICk5obXRBb0dBR3ZPeXR3emRlUGR0TzNscFowS3pibnF6NkQ3SHpHblYyaGdCWU
NvTDdQaVpHVWMrdHdtOFJwVW4KU29PYmsvOVM1S3Y0NXNlTnRjTjd5VS8wVjhpUUFwUmZkVDVSY
nNjT3ZrUlFSQmo1VkJxa0pMVlppeS85QmJDbgpXNU1ZV0NqSi9lWHk0ZGVMdnZFckRJVHhDRFVYbkN
WRU1ReUlXcDB4T2Y4b1lla2h6U009Ci0tLS0tRU5EIFJTQSBQUklWQVRFIEtFWS0tLS0tCg==

razorops.yaml

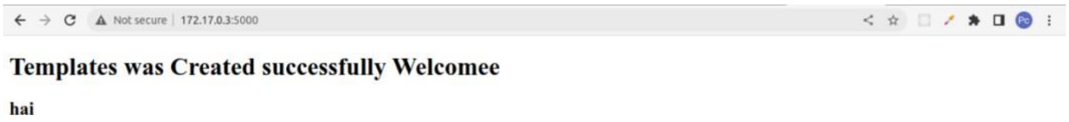
```
version: v2

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

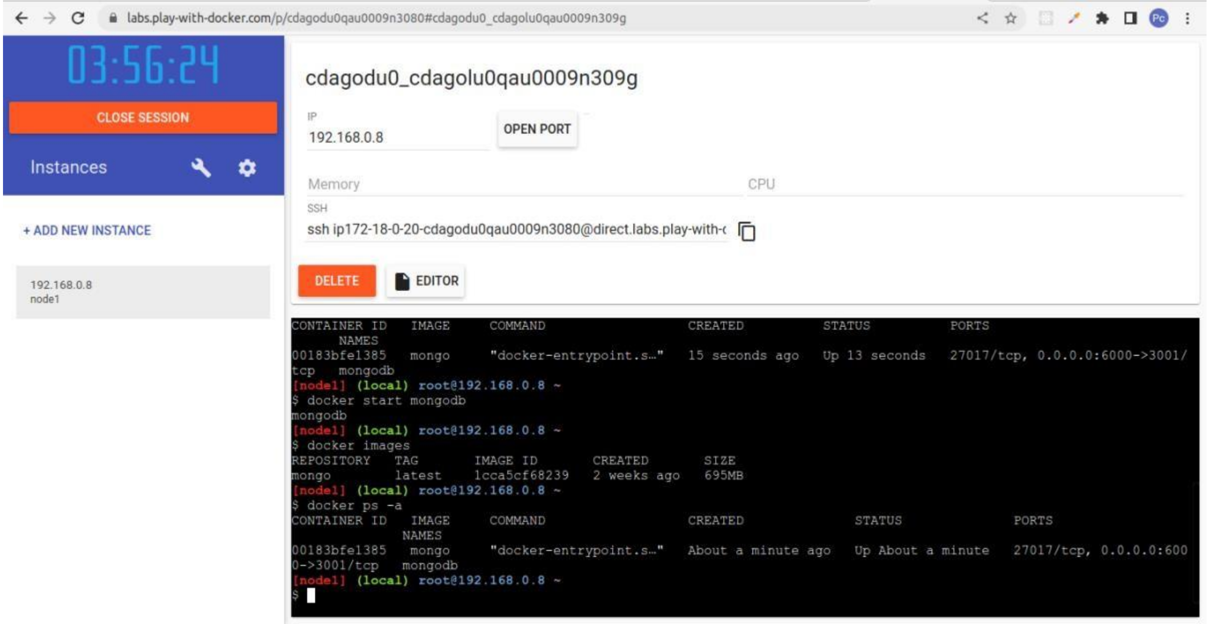
```
flask
gunicorn
```

Outputs:



Templates was Created successfully Welcomee

hai

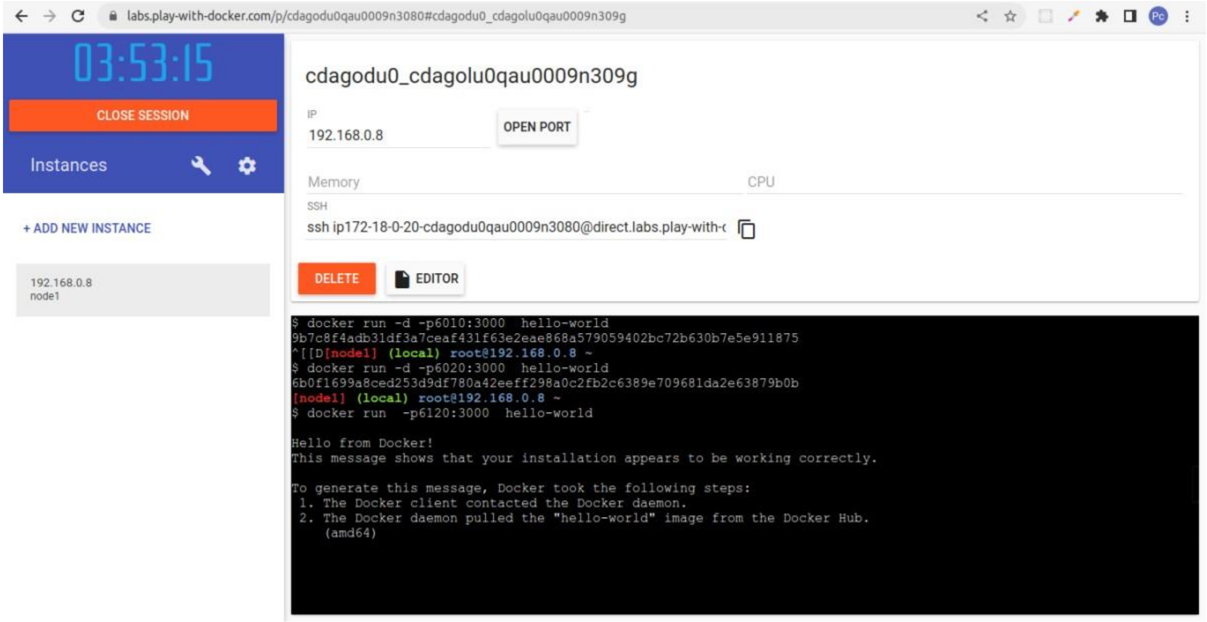


cdagodu0_cdagolu0qau0009n309g

IP
192.168.0.8

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

```
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS
     NAMES
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
              NAMES
00183bfe1385   mongo     "docker-entrypoint.s…"   About a minute ago   Up About a minute   27017/tcp, 0.0.0.0:600
0->3001/tcp   mongodb
[node1] (local) root@192.168.0.8 ~
$ 
```

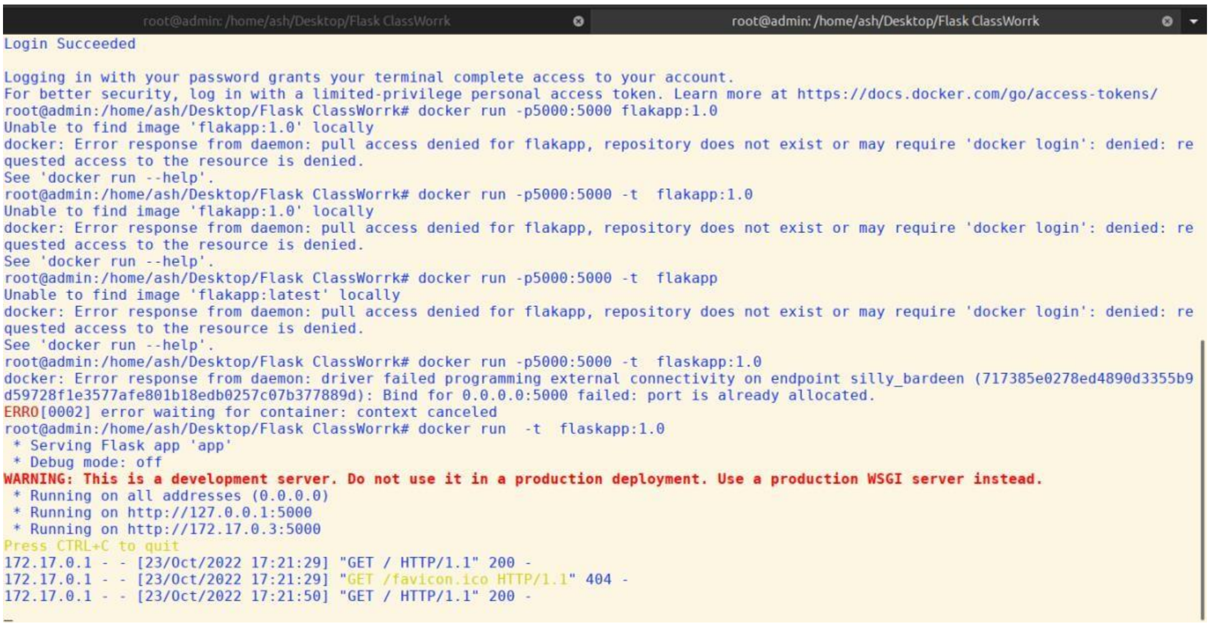


cdagodu0_cdagolu0qau0009n309g

IP
192.168.0.8

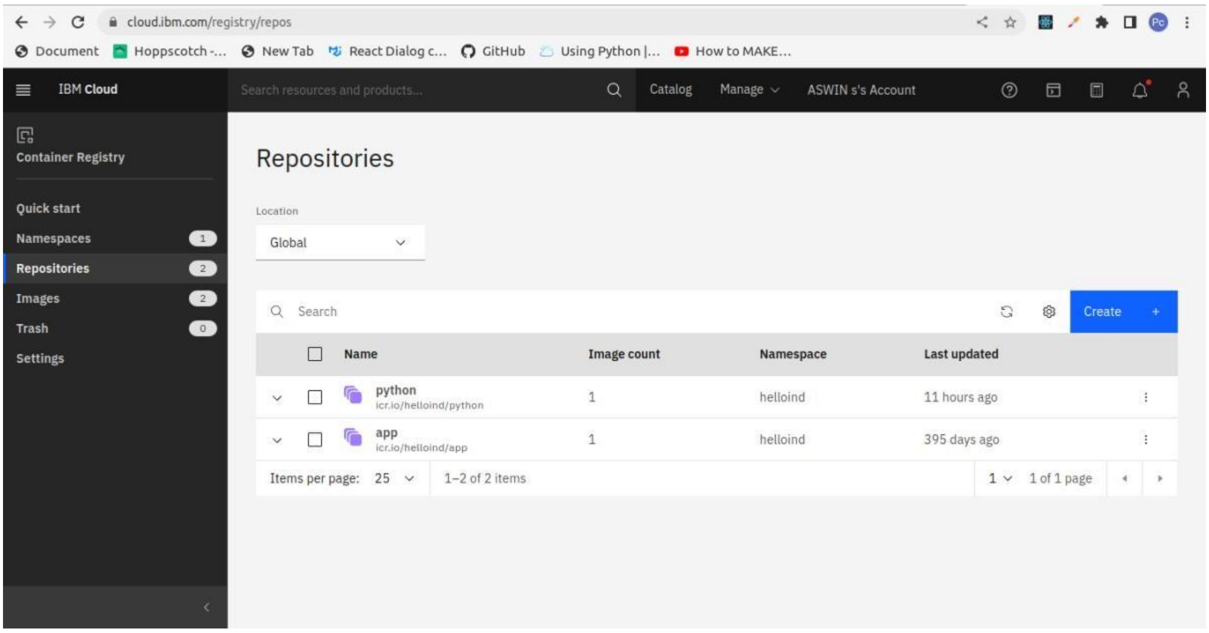
SSH
ssh ip172-18-0-20-cdagodu0qau0009n3080@direct.labs.play-with-(

```
$ 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
root@admin: /home/ash/Desktop/Flask ClassWorrk

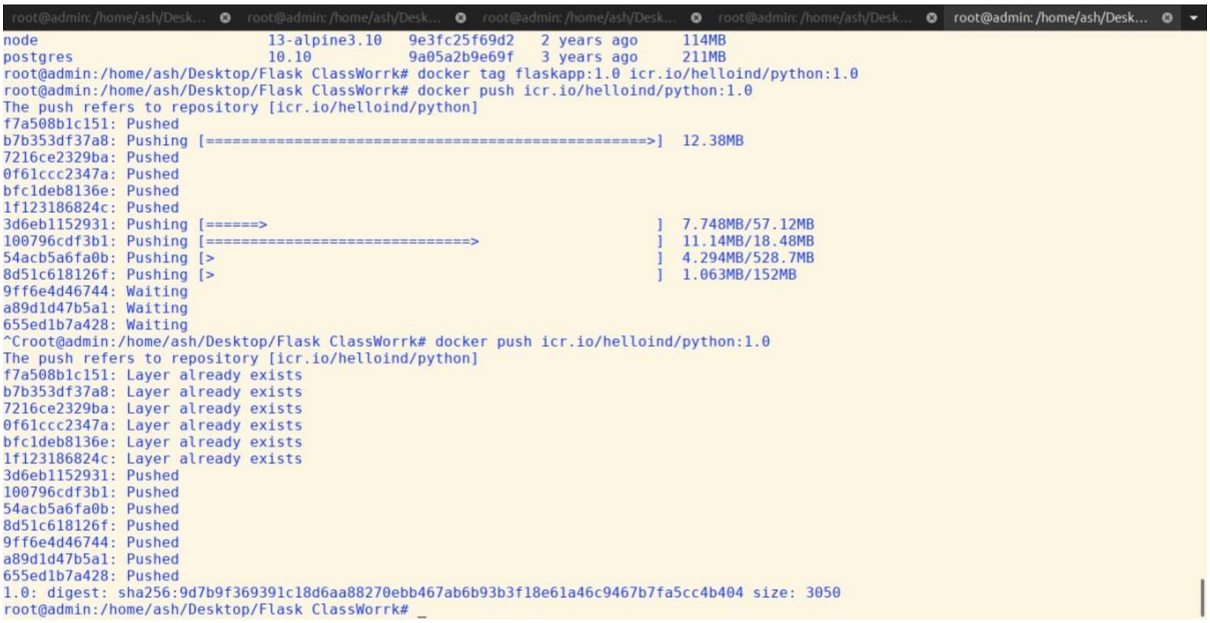
```
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/
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
d59728f1e3577afe801b18edb0257c07b377889d): 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 -
_
```



Repositories

	Name	Image count	Namespace	Last updated
□	python icr.io/helloind/python	1	helloind	11 hours ago
□	app icr.io/helloind/app	1	helloind	395 days ago

Items per page: 25　1–2 of 2 items　1　1 of 1 page



root@admin: /home/ash/Desk...
root@admin: /home/ash/Desk...
root@admin: /home/ash/Desk...
root@admin: /home/ash/Desk...
root@admin: /home/ash/Desk...

```
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
```