

Assignment – 4

Cloud Application Development

Assignment Date	19 September 2022
Student Name	KOVI RAHUL
Student Roll Number	211719106037
Maximum Mark	2 Marks

1.Pull an Image from docker hub and run it in docker playground.

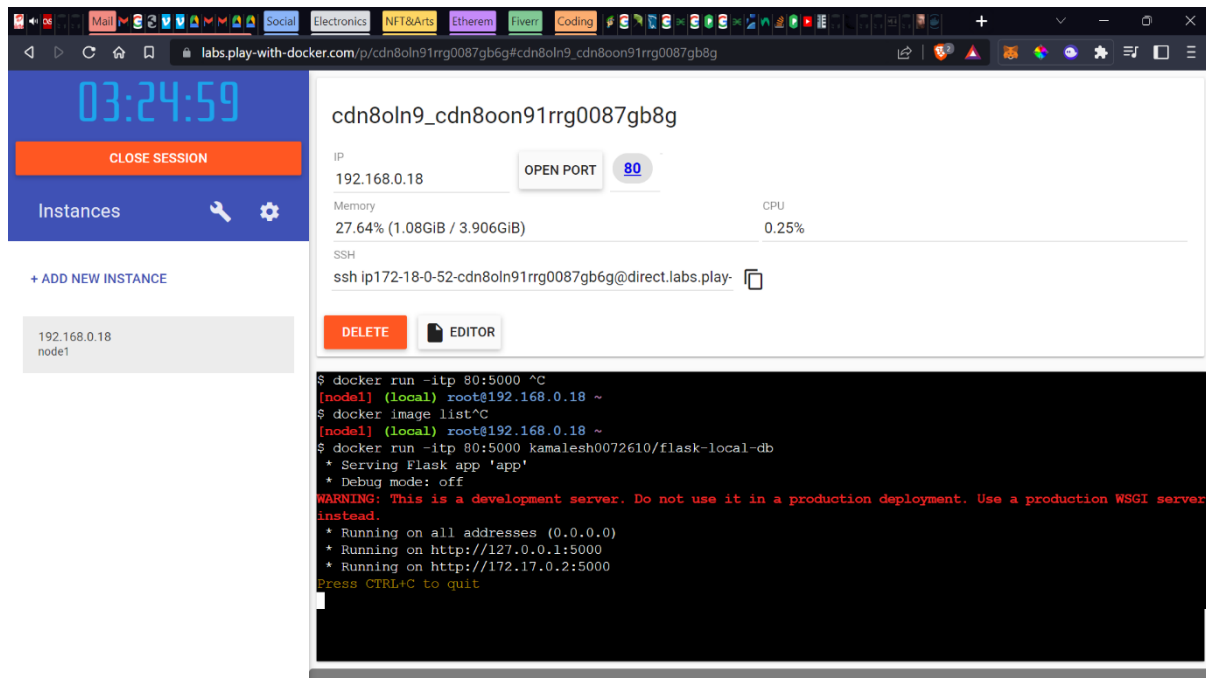
Pushed my own Image to Docker Hub and used that for this assignment.

```
docker pull kamalesh0072610/flask-local-db:latest
docker image list
```

The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a timer at 03:26:56, a 'CLOSE SESSION' button, and a list of instances. One instance, '192.168.0.18 node1', is listed. The main area displays details for the selected instance 'cdn8oln9_cdn8oon91rrg0087gb8g', including its IP address, memory usage (26.79%), CPU usage (0.45%), and an SSH command. Below this, there's a terminal window showing the output of 'docker image list' command, which lists the 'kamalesh0072610/flask-local-db:latest' image.

```
94da21e53a5b: Pull complete
95e3473ba5b7: Pull complete
fc16ec4a0f48: Pull complete
47721b0607f9: Pull complete
bd89fa278679: Pull complete
Digest: sha256:e3d6f139775465b9be3f2e7f7c43e27307c6e10f4fb2382d3217c80830753e16
Status: Downloaded newer image for kamalesh0072610/flask-local-db:latest
docker.io/kamalesh0072610/flask-local-db:latest
(node1) (local) root@192.168.0.18 ~
$ docker image list
REPOSITORY          TAG         IMAGE ID      CREATED      SIZE
kamalesh0072610/flask-local-db  latest     71eb11162295  5 weeks ago  932MB
(node1) (local) root@192.168.0.18 ~
$
```

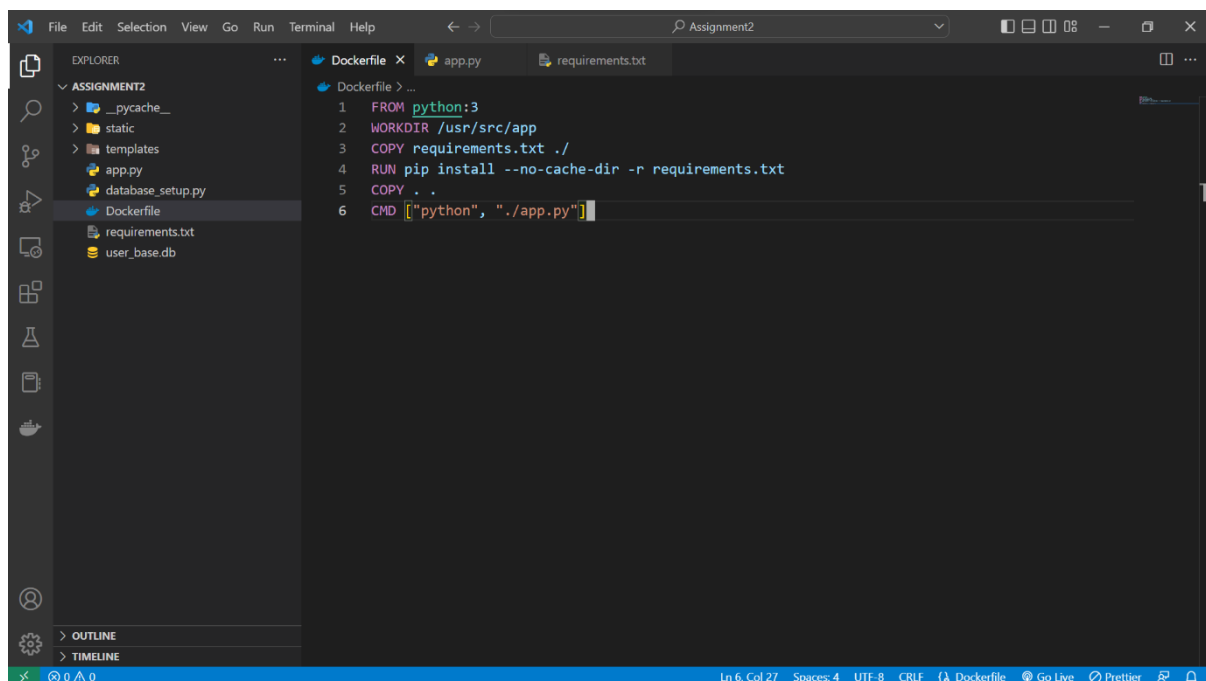
`docker run -itp 80:5000 kamalesh0072610/flask-local-db` – run in interactive mode.



The screenshot shows the Play-with-Docker interface. On the left, there's a sidebar with a clock showing 03:24:59, a 'CLOSE SESSION' button, and a list of instances. The main area displays details for a container named 'cdn8oln9_cdn8oon91rrg0087gb8g' with IP 192.168.0.18. Below this, there's a terminal window showing the command `docker run -itp 80:5000 kamalesh0072610/flask-local-db` and the output of the Flask application, which includes a warning about using a development server and the Flask version.

2. Create a dockerfile for the job portal / flask application and deploy it in Docker desktop application.

I've used the flask application used for assignment 2 for this assignment.



The screenshot shows the Visual Studio Code editor with a project named 'Assignment2'. The Explorer sidebar on the left shows the file structure, including 'Dockerfile', 'app.py', 'requirements.txt', 'database_setup.py', and 'user_base.db'. The Dockerfile content is displayed in the main editor area, showing the following instructions:

```
1 FROM python:3
2 WORKDIR /usr/src/app
3 COPY requirements.txt ./
4 RUN pip install --no-cache-dir -r requirements.txt
5 COPY . .
6 CMD ["python", "./app.py"]
```

docker build -t flask-app-assi4 . - build the image

```
C:\WINDOWS\system32\cmd. X + v

E:\KAMALESH\IBM_trying\Assignment2>docker build -t flask-app-assi4 .
[+] Building 4.4s (11/11) FINISHED
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 198B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/library/python:3 4.2s
=> [auth] library/python:pull token for registry-1.docker.io 0.0s
=> [1/5] FROM docker.io/library/python:3@sha256:b941b836b18734f4992a168b579b7c16ff4c3b544782953eeab3a5 0.0s
=> => resolve docker.io/library/python:3@sha256:b941b836b18734f4992a168b579b7c16ff4c3b544782953eeab3a5 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 135.83kB 0.0s
=> CACHED [2/5] WORKDIR /usr/src/app 0.0s
=> CACHED [3/5] COPY requirements.txt / 0.0s
=> CACHED [4/5] RUN pip install --no-cache-dir -r requirements.txt 0.0s
=> [5/5] COPY . 0.0s
=> => exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:5fed83284be3857af98b40fda3e74ef8765581f9cf21edf6257a8d8c78d1325d 0.0s
=> => naming to docker.io/library/flask-app-assi4 0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
E:\KAMALESH\IBM_trying\Assignment2>
```

```
C:\WINDOWS\system32\cmd. X + v

E:\KAMALESH\IBM_trying\Assignment2>docker build -t flask-app-assi4 .
[+] Building 4.4s (11/11) FINISHED
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 198B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/library/python:3 4.2s
=> [auth] library/python:pull token for registry-1.docker.io 0.0s
=> [1/5] FROM docker.io/library/python:3@sha256:b941b836b18734f4992a168b579b7c16ff4c3b544782953eeab3a5 0.0s
=> => resolve docker.io/library/python:3@sha256:b941b836b18734f4992a168b579b7c16ff4c3b544782953eeab3a5 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 135.83kB 0.0s
=> CACHED [2/5] WORKDIR /usr/src/app 0.0s
=> CACHED [3/5] COPY requirements.txt / 0.0s
=> CACHED [4/5] RUN pip install --no-cache-dir -r requirements.txt 0.0s
=> [5/5] COPY . 0.0s
=> => exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:5fed83284be3857af98b40fda3e74ef8765581f9cf21edf6257a8d8c78d1325d 0.0s
=> => naming to docker.io/library/flask-app-assi4 0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
E:\KAMALESH\IBM_trying\Assignment2>docker image list
REPOSITORY          IMAGE ID      CREATED      TAG
flask-app-assi4      5fed83284be3 49 seconds ago  latest
flask-app-testing    2d8f454de374 11 hours ago  latest
flask-testing-app    78a4955b95b2 10 days ago   latest
jp.icr.io/training/flask-local-db 71eb11162295 5 weeks ago   latest
kamalesh0072610/flask-local-db 71eb11162295 5 weeks ago   latest
flask-local-db       71eb11162295 5 weeks ago   latest
registry.k8s.io/ingress-nginx/controller d681a4ce3c50 6 weeks ago   <none>
```

Running the docker application locally.

The screenshot shows Docker Desktop on the left and a web browser on the right. In Docker Desktop, the 'Containers' tab is active, showing a list of running containers. The container 'lucid_hodgkin' is highlighted with a red box. It is running the 'flask-app-a' image on port 5000. The web browser shows the 'Flask App' running on 'localhost:5000'. The page displays a 'Registered User List' with the following entries:

Name	Path
Kamalesh	Pathy
Mahesh D	

The footer of the web page shows '© Kamaleshpathy VA' and '+91 8056117670'.

3. Create a IBM container registry and push docker image of flask application or job portal app.

Pushed the image to ibm container registry.

```
ibmcloud login
```

```
ibmcloud plugin install container-registry -r "IBM Cloud"
```

```
ibmcloud cr namespace-add training
```

```
ibmcloud cr login
```

```
docker tag flask-app-assig4 jp.icr.io/training/flask-app-assi4:latest
```

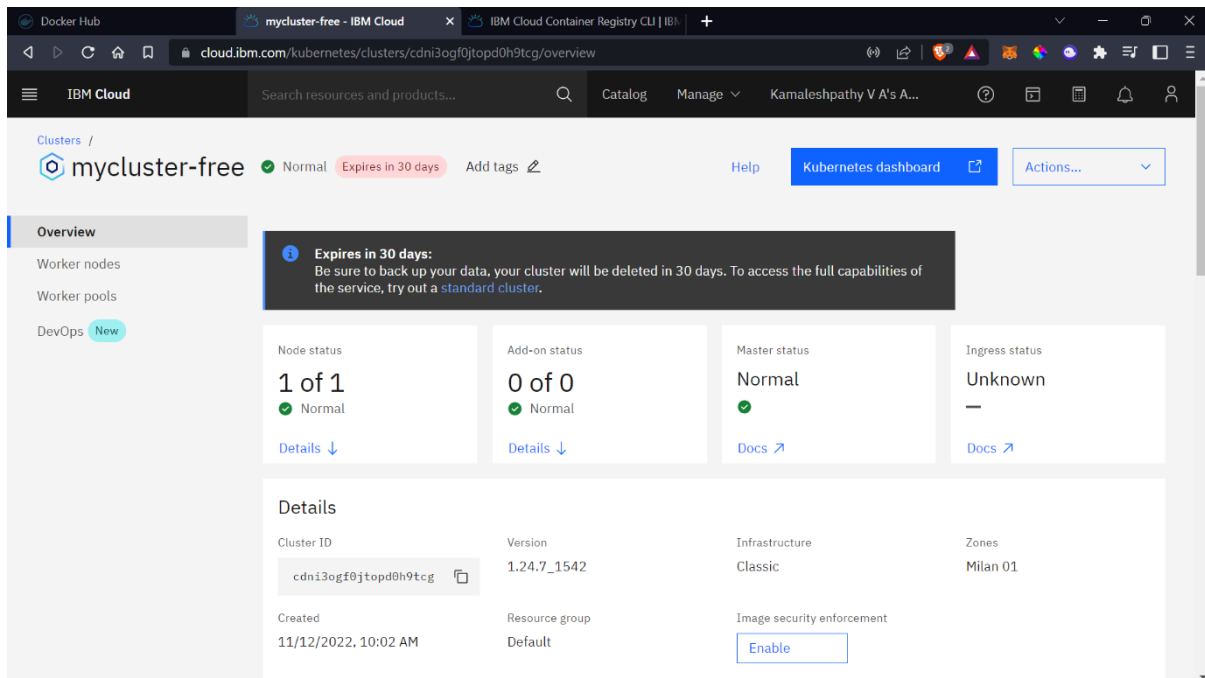
```
docker push jp.icr.io/training/flask-app-assi4:latest
```

The screenshot shows the IBM Cloud Container Registry interface. The 'Images' section is active, displaying a list of images. The image 'training/flask-app-assi4@sha256:7deb95a2bd91...' is highlighted. The interface shows the following details for the image:

Repository@digest	Tags	Manifest type	Created	Size	Security status
training/flask-app-assi4@sha256:7deb95a2bd91...	latest	Docker	7 hours ago	362 MB	6 issues

The interface also shows a sidebar with 'Container Registry' and 'Images' sections, and a top navigation bar with 'Search resources and products...' and 'Catalog'.

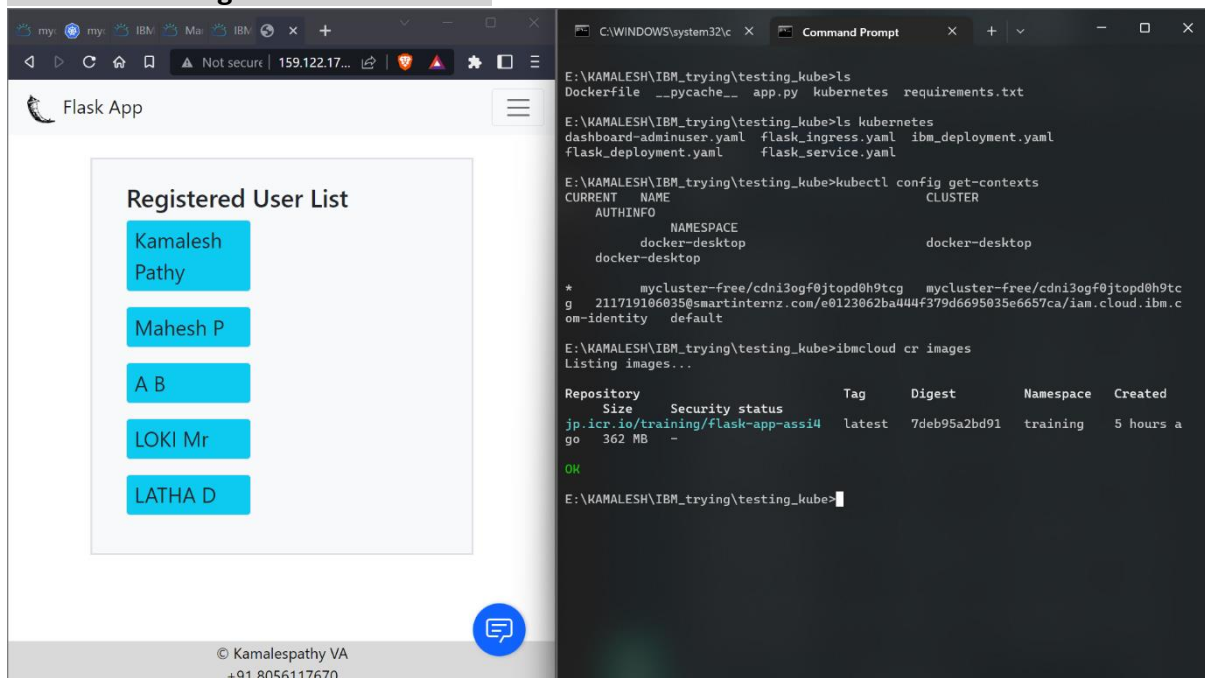
4. Create a Kubernetes cluster in IBM cloud and deploy flask application image or job portal image and also expose the same app to run in nodeport.



ibmcloud plugin install container-service

ibmcloud ks cluster config --cluster cdni3ogf0jtopd0h9tcg

kubect1 config current-context



ibm_deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: flask-app
```

```
spec:
  replicas: 5
  selector:
    matchLabels:
      app: flask-app
  template:
    metadata:
      labels:
        app: flask-app
```

```
    spec:
      containers:
        - name: flask-app-container
          image: jp.icr.io/training/flask-app-assi4
          imagePullPolicy: Always
          ports:
            - containerPort: 5000
              protocol: TCP
```

flask_service.yaml

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

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

```

```

kubectl apply -f kubernetes/ibm_deployment.yaml
kubectl apply -f kubernetes/flask_service.yaml
kubectl apply -f kubernetes/flask_ingress.yaml
kubectl expose deployment flask-app --type=NodePort --name=flask-app

```

The screenshot shows a web browser window displaying the 'Flask App' interface. The interface has a header 'Flask App' and a sidebar with a 'Registered User List' containing five entries: 'Kamalesh Pathy', 'Mahesh P', 'A B', 'LOKI Mr', and 'LATHA D'. The footer of the browser window shows '© Kamalespathy VA +91 8056117670' and a chat icon. To the right of the browser window is a Windows Command Prompt window. The Command Prompt shows the following commands and output:

```

E:\KAMALESH\IBM_trying\testing_kube>ibmcloud cr images
Listing images...

Repository      Tag      Digest      Namespace      Created
Size      Security status
jp.icr.io/training/flask-app-ssi4  latest  7deb95a2bd91  training  5 hours a
go 362 MB  -

OK

E:\KAMALESH\IBM_trying\testing_kube>kubectl get all
NAME                 READY   STATUS    RESTARTS   AGE
pod/flask-app-69dfc957b4-hfdmg  1/1     Running   0           103m

NAME                 AGE      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)
service/flask-app      87m     NodePort   172.21.6.6       <none>            5000:31356/TCP
service/flask-app-service  97m     ClusterIP  172.21.241.192   <none>            5000/TCP
service/kubernetes     5h20m   ClusterIP  172.21.0.1       <none>            443/TCP

NAME                 READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/flask-app  1/1     1             1           114m

NAME                 DESIRED   CURRENT   READY   AGE
replicaset.apps/flask-app-67ff589dd4  0         0         0       114m
replicaset.apps/flask-app-69dfc957b4  1         1         1       104m

```

The screenshot shows the 'Registered User List' section of the Flask App interface. It contains five entries, each in a blue button: 'Kamalesh Pathy', 'Mahesh P', 'A B', 'LOKI Mr', and 'LATHA D'.