

IBM Assignment - 4

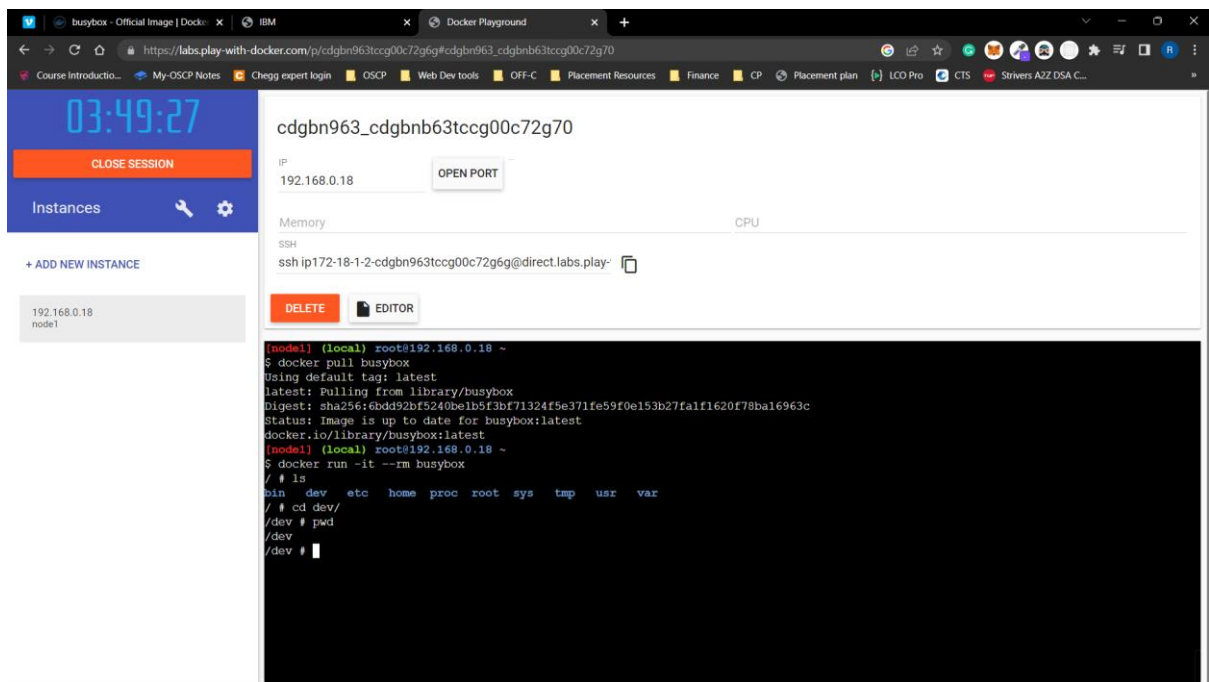
ProjectTitle :Personal Expense Tracker Application

Team ID :PNT2022TMID32657

Name: Nishanthi.R

Registration No :813819104062

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



- 2) Create a docker file for the job portal application and deploy it in Docker desktop application.

Dockerfile

```
FROMpython:3-alpine3.15
WORKDIR/app
COPY./app
RUNpipinstall-rrequirements.txt
EXPOSE5000
CMDpython./app.py
```

```

PS D:\flask\job> docker build -t rohitmaheswaran/job-portal:0.0.1.RELEASE .
[+] Building 1.5s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 318 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/library/python:3-alpine3.15 1.4s
=> [internal] load build context 0.0s
=> => transferring context: 401B 0.0s
=> [1/4] FROM docker.io/library/python:3-alpine3.15@sha256:d89ac9cefd2213b99a792fd8ec4f15c0297d9340a688b00fdae136d0f03cc30e 0.0s
=> CACHED [2/4] WORKDIR /app 0.0s
=> CACHED [3/4] COPY . /app 0.0s
=> CACHED [4/4] RUN pip install -r requirements.txt 0.0s
=> exporting image 0.0s
=> => exporting layers 0.0s
=> writing image sha256:8e89dbefae0271b0e28ed824f5efdcfb8b629cbe92bd63eac1fa9f33a91b68 0.0s
=> naming to docker.io/rohitmaheswaran/job-portal:0.0.1.RELEASE 0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
PS D:\flask\job>

```

Docker Desktop

Upgrade plan

rohitmaheswaran

Containers

Images

Volumes

Dev Environments BETA

Extensions BETA

Add Extensions

Containers Give feedback

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)

Only show running containers

Search

	NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
	wizardly_cartwright f5f7a9badf5c	au.icr.io/ibm-cr-flask/ibm-assig	Exited	5000:5000		
	wizardly_leavitt 17d6b0983942	au.icr.io/ibm-cr-flask/ibm-assig	Exited	5000:5000		
	docker-tutorial cab713ba7924	docker101tutorial:latest	Exited	80:80		
	repo df92bf4336c1	alpine/git:latest	Exited			
	elegant_cannon fa213af71581	rohitmaheswaran/job-portal:0	Running	5000:5000	46 seconds ago	

Showing 5 items

RAM 1.64GB

CPU 0.22%

Connected to Hub

v4.13.1

mycluster-free - IBM Cloud

IBM Cloud Shell

Register

Register

[http://localhost:5000/login](#)

Course Introductio...

My-OSCP Notes

Chegg expert login

OSCP

Web Dev tools

OFF-C

Placement Resources

Finance

CP

Placement plan

LCO Pro

CTS

Strivers AZZ DSA C...

Jobs.com

Register

Login to your Account

Email

Password

Login

```
C:\Windows\System32\cmd.exe
C:\Users\91877\Downloads\hello-flask-main>docker build -t nishanth5302/hello-flask:0.0.1.RELEASE .
[*] Building 754.1s (9/9) FINISHED
-> [internal] load build definition from Dockerfile
0.1s
-> > transferring dockerfile: 157B
0.0s
-> [internal] load dockerignore
0.1s
-> > transferring context: 2B
0.0s
-> [internal] load metadata for docker.io/library/python:alpine3.15
16.5s
-> [internal] load build context
0.0s
-> > transferring context: 401B
0.0s
[1/4] FROM docker.io/library/python:alpine3.15@sha256:d8bac9cf4d2213b99a792f0bec4f15cd297d0340a88b00fdae1370415
0.0s
-> resolve docker.io/library/python:alpine3.15@sha256:d8bac9cf4d2213b99a792f0bec4f15cd297d0340a88b00fdae1370415
0.0s
-> sha256:d8bac9cf4d2213b99a792f0bec4f15cd297d0340a88b00fdae1370415 / 1.45MB / 1.45MB
0.0s
-> sha256:080d223179e0453e08011ab2012e3600c351643747d1c9b6eac0f13bc617 1.27MB / 1.27MB
0.0s
-> sha256:c0bc56b75607560729087c1101361006a2a763bb767a719c8147f74346a806d4 7.01MB / 7.01MB
0.0s
-> sha256:196211af6e04053b2f96bf34fc7f7a60712247c01cab483c5fa7132cf782ce 2.32MB / 2.32MB
173.1s
-> sha256:7dcb1581fe1d2ed60f136099a2d780245d81b3917075ed34786a0ae2fc70 673.18MB / 673.18MB
34.6s
-> sha256:292f7bce2a772c05ae5f110949191cb3feadcd42d8d4d6a47105a5161d15 3.00MB / 3.00MB
301.1s
-> sha256:429c086d42bc04771766d4f122c7e03023b08a40f799bc3629c623613c4 23MB / 23MB
46.6s
-> sha256:8652f1fe7bce2a772c05ae5f110949191cb3feadcd42d8d4d6a47105a5161d15 3.00MB / 3.00MB
322.4s
-> extracting sha256:96211af6e04053b2f96bf34fc7f7a60712247c01cab483c5fa7132cf782ce
0.3s
-> extracting sha256:7dcb1581fe1d2ed60f136099a2d780245d81b3917075ed34786a0ae2fc70
0.2s
-> extracting sha256:c0bc56b75607560729087c1101361006a2a763bb767a719c8147f74346a806d4
0.2s
-> extracting sha256:429c086d42bc04771766d4f122c7e03023b08a40f799bc3629c623613c4
0.5s
-> extracting sha256:8652f1fe7bce2a772c05ae5f110949191cb3feadcd42d8d4d6a47105a5161d15
0.2s
[2/6] WORKDIR /app
0.1s
[3/6] COPY . /app
0.0s
[4/6] RUN pip install -r requirements.txt
31.0s
-> exporting to image
0.2s
-> exporting layers
0.2s
-> writing image sha256:7c0eeef1709e76c2affe770a70ef813627aff5c8f61a2f2885ef050bdc04b3b
0.0s
-> naming to docker.io/nishanth5302/hello-flask:0.0.1.RELEASE
0.0s

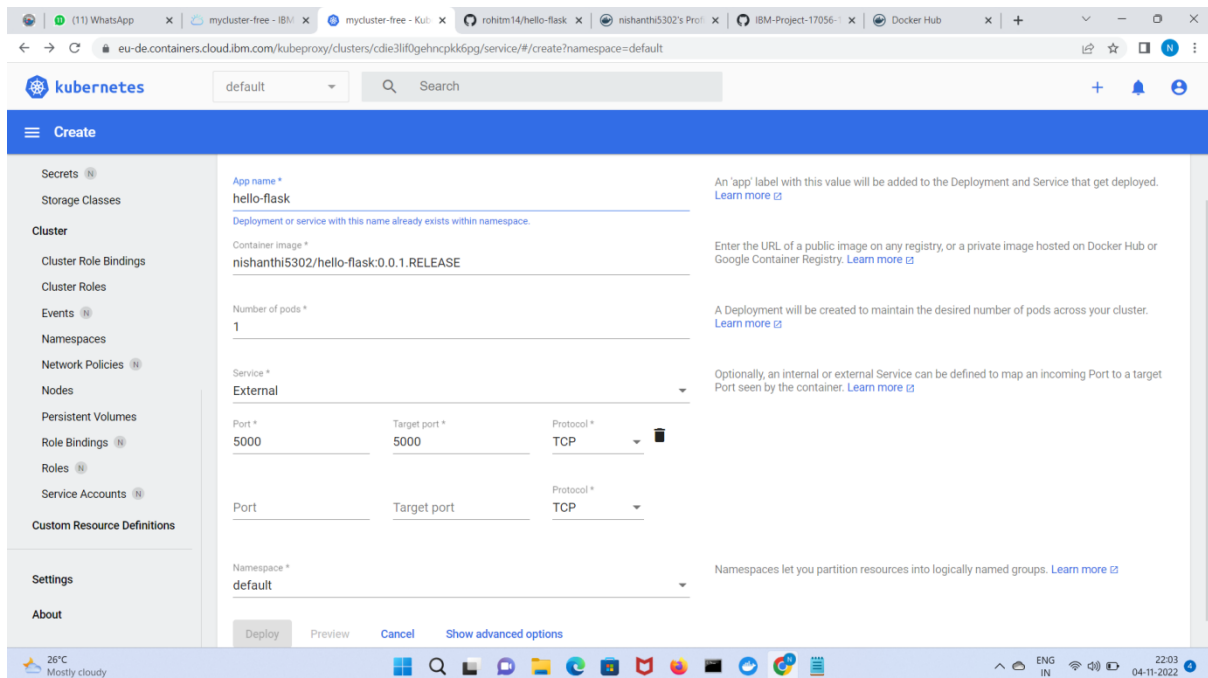
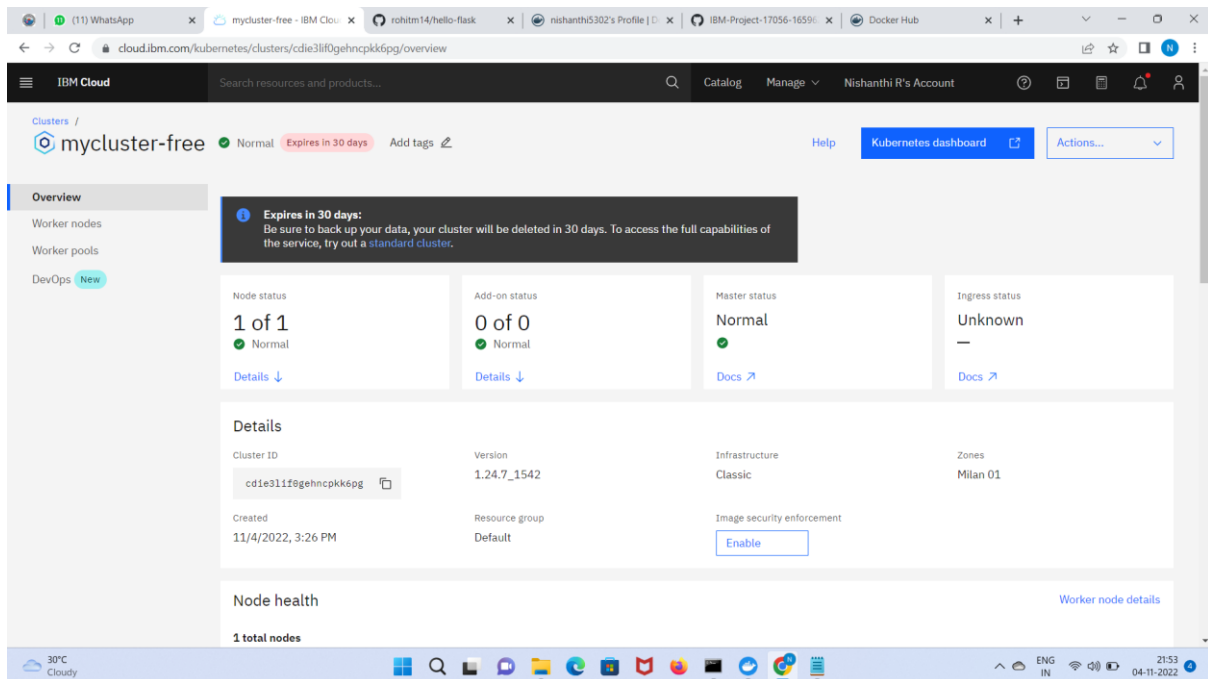
Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

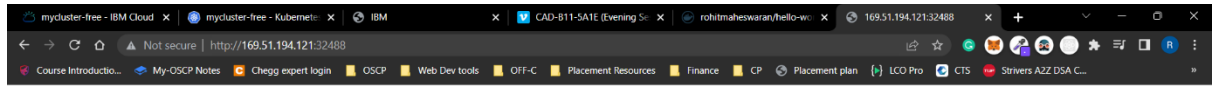
C:\Users\91877\Downloads\hello-flask-main>docker build -t nishanth5302/hello-flask:0.0.1.RELEASE
```

```
C:\Users\91877\Downloads\hello-flask-main>docker push icr.io/ibm-nishanthi/hello-flask:hello-flask
The push refers to repository [icr.io/ibm-nishanthi/hello-flask]
043b529782f9: Pushed
4c853c78241c: Pushed
b40222338aba: Pushed
f9a01ea63d59: Pushed
eb71c8b7b3b7: Pushed
76d682e14461: Pushed
d59c8eb0f9e4: Pushed
34d5ebaa5410: Pushed
hello-flask: digest: sha256:e3e7a95ff8007fa88fd47e339984257f4841e5a2529e6891e455daf5eafe1f11 size: 1993
```

The screenshot displays the IBM Cloud Container Registry interface. The left sidebar contains navigation links: 'Container Registry', 'Quick start', 'Namespaces' (selected), 'Repositories', 'Images', 'Trash', and 'Settings'. The main content area is titled 'Namespaces' and shows a table of namespaces under the 'Global' location. The table has columns for 'Name', 'Resource group', 'Repository count', 'Image count', and 'Retention policy'. One namespace, 'ibm-nishanthi', is listed with 1 repository and 1 image. The bottom of the page shows a status bar with '30°C Cloudy' and system icons.

- 4) Create a Kubernetes cluster in IBM cloud and deploy hello world image or job portal image and also expose the same app to run in node port.





Welcome to Flask App!! :)