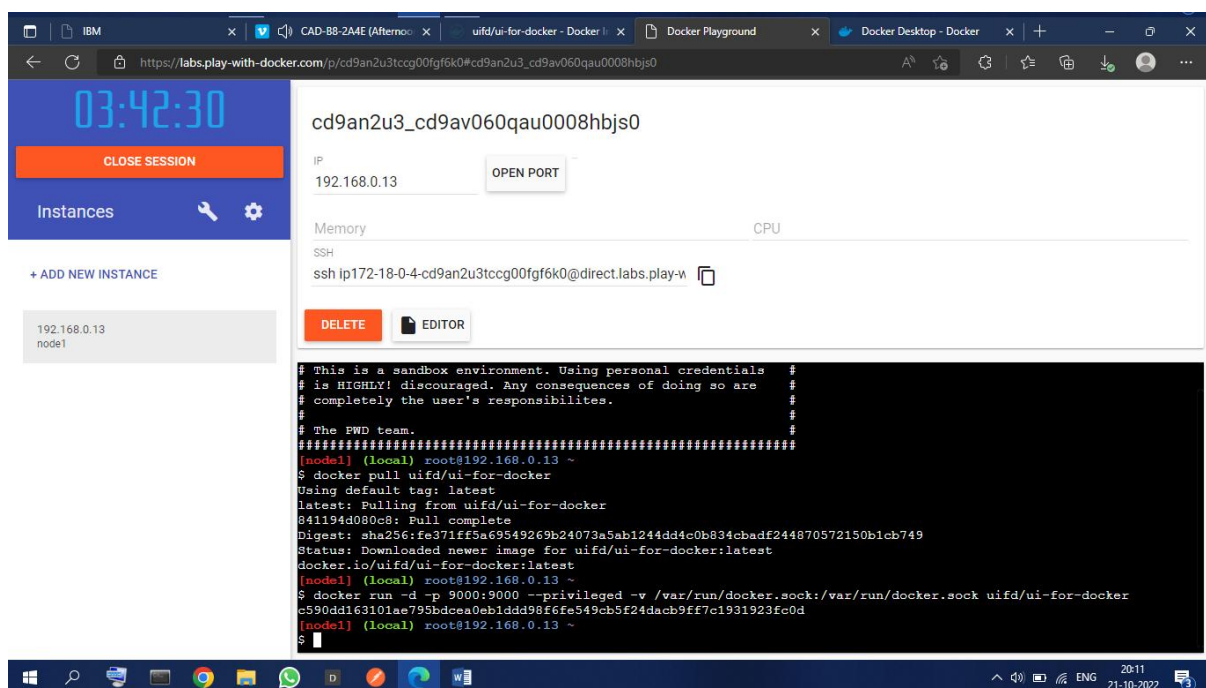
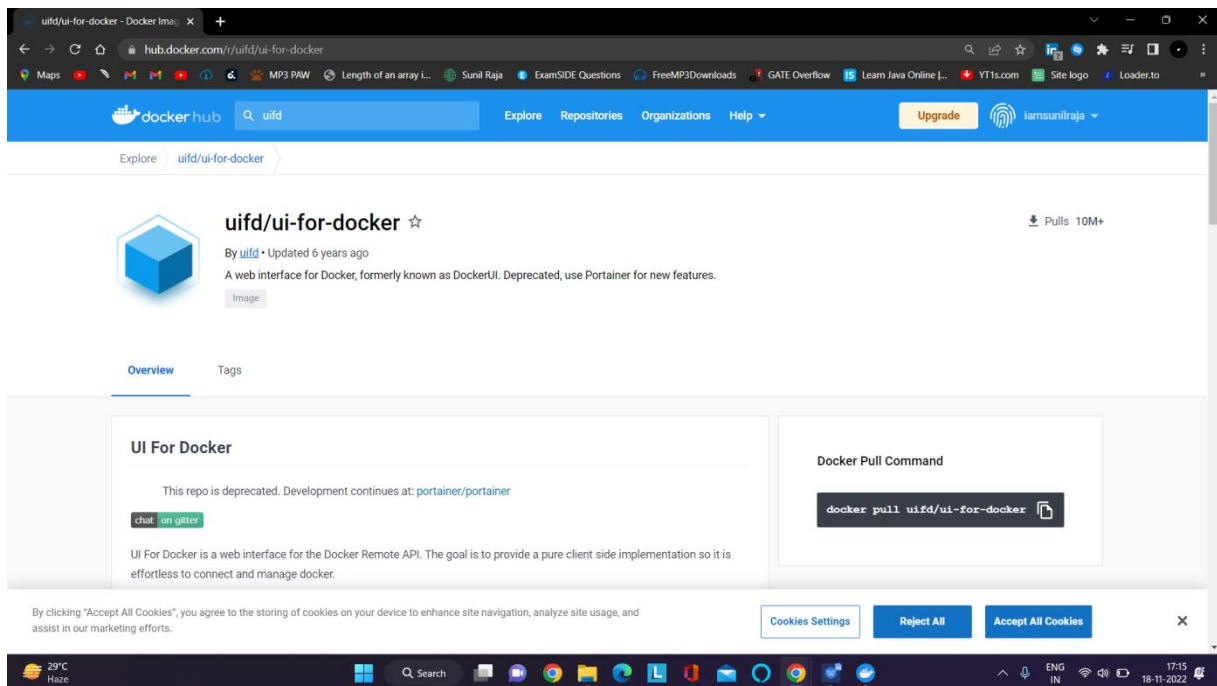


## Assignment -4

### Docker and Kubernetes

Assignment Date	21 October 2022
Team ID	PNT2022TMID53909
Student Name	John Milton S
Student Roll Number	95071912302
Maximum Marks	2 Marks

#### 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

The image displays two screenshots of the 'UI For Docker' web application interface, which is accessed via a browser at the URL `ip172-18-0-4-cd9an2u3tccg00fgf6k0-9000.direct.labs.play-with-docker.com/#/`.

**Top Screenshot:** The interface shows the 'UI For Docker' title and a navigation bar with tabs: Dashboard, Containers, Containers Network, Images, Networks, Volumes, and Info. A 'Refresh' button is located on the right. The main content area features a large 'UI For Docker' header with the subtitle 'The UI for Docker container engine' and a 'Learn more.' button. Below this, the 'Running Containers' section lists a single container named 'beautiful\_goldwasser' with a status of 'Up About a minute'. The 'Status' section displays a green donut chart representing the system's health.

**Bottom Screenshot:** This screenshot shows the same interface but with additional sections visible. The 'Running Containers' section remains the same. The 'Status' section now includes a legend for the donut chart: 'Running' (green), 'Stopped' (red), and 'Ghost' (grey). Below the status section, there are two line graphs: 'Containers created' and 'Images created', both showing a count of 1 on the y-axis against a date of 21/10/2022 on the x-axis.

### 3. Create a Kubernetes cluster in IBM cloud and deploy helloworld image or jobportal image and also expose the same app to run in nodeport.

```
C:\Windows\System32\cmd.exe
-> [internal] load build definition from Dockerfile
-> transferring dockerfile: 32B
-> [internal] load .dockerignore
-> transferring context: 2B
-> [internal] load metadata for docker.io/library/python:3.6
-> [auth] library/python:pull token for registry-1.docker.io
-> [internal] load build context
-> transferring context: 687B
-> [1/6] FROM docker.io/library/python:3.6@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f308af6fc
-> resolve docker.io/library/python:3.6@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f308af6fc
-> sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f308af6fc 1.86kB / 1.86kB
-> sha256:d097a4907a8ec079df5ac31872359c2de510f82214c0448e926393b376d3b60d 2.22kB / 2.22kB
-> sha256:5420663807c5e3ad24c6e21fc889abbcb486a27634c0892086ff71f3f44b104 9.27kB / 9.27kB
-> sha256:0e29546d541cddb309281d21a73a9d1db78665c1b95b74f32b09e0b77a6e1e3 54.92MB / 54.92MB
-> sha256:9b829c73b52b92b97d5c07a54fb0f3e921995a296c714b53a32ae67d19231fcd 5.15MB / 5.15MB
-> sha256:cb5b7ae3b1722f070eca53f35823ed21baa85d61d5d95cd5a95ab53d740cdd56 10.87MB / 10.87MB
-> sha256:6494e4811622b31c027ccac322ca463937fd805f569a93e6f15c01aade718793 54.57MB / 54.57MB
-> sha256:df9774896df93f0712f594fba950b48a481a0fef09112efc7a4d3c7077 196.51MB / 196.51MB
-> sha256:5e3b1213efc55080e78b0d02883945c164d0237205e06a62dada823124dc7a3 6.29MB / 6.29MB
-> extracting sha256:0e29546d541cddb309281d21a73a9d1db78665c1b95b74f32b09e0b77a6e1e3
-> sha256:9fd4dfc56334f2e6efad7e241bf5e7459c40ed185c5478676f41c1244bd96752 14.21MB / 14.21MB
-> extracting sha256:9b829c73b52b92b97d5c07a54fb0f3e921995a296c714b53a32ae67d19231fcd
-> extracting sha256:cb5b7ae3b1722f070eca53f35823ed21baa85d61d5d95cd5a95ab53d740cdd56
-> sha256:404f02044bac0432ca522cbb9f254b1c91fcea6806bfeef0be0b243b2f31bab7 235B / 235B
-> sha256:c4f42be2b53b900ebffcc040c1df13de538434ccc5f5d954a56848a6169a3a3f 2.21MB / 2.21MB
-> extracting sha256:6494e4811622b31c027ccac322ca463937fd805f569a93e6f15c01aade718793
-> extracting sha256:df9774896df93f0712f594fba950b48a481a0fef09112efc7a4d3c7077
-> extracting sha256:5e3b1213efc55080e78b0d02883945c164d0237205e06a62dada823124dc7a3
-> extracting sha256:9fd4dfc56334f2e6efad7e241bf5e7459c40ed185c5478676f41c1244bd96752
-> extracting sha256:404f02044bac0432ca522cbb9f254b1c91fcea6806bfeef0be0b243b2f31bab7
-> extracting sha256:c4f42be2b53b900ebffcc040c1df13de538434ccc5f5d954a56848a6169a3a3f
-> [2/6] WORKDIR /app
-> [3/6] ADD . /app
-> [4/6] COPY requirements.txt /app
-> [5/6] RUN python3 -m pip install -r requirements.txt
-> [6/6] RUN python3 -m pip install ibm_db
-> exporting to image
-> exporting layers
-> writing image sha256:1756719486df002fad5dae305c5221513f2ff2d1b49a8d32b27a28af0379f19
-> naming to docker.io/library/job-portal-main

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
c:\Users\VK-PC\Desktop\job-portal-main>
```

Docker Desktop Upgrade plan

Containers

Images

Volumes

Dev Environments BETA

Extensions BETA

Add Extensions

Images on disk Last refresh: about 1 hour ago 1 Images 0 Bytes total size Refresh to see disk usage Clean up

Images Give feedback

LOCAL REMOTE REPOSITORIES

Search

☐ In use only

NAME	TAG	IMAGE ID	CREATED	SIZE
job-portal-main	latest	1756719486df	less than a minute ago	1.08 GB

RAM 2.53GB CPU 1.56% Connected to Hub v4.13.0