

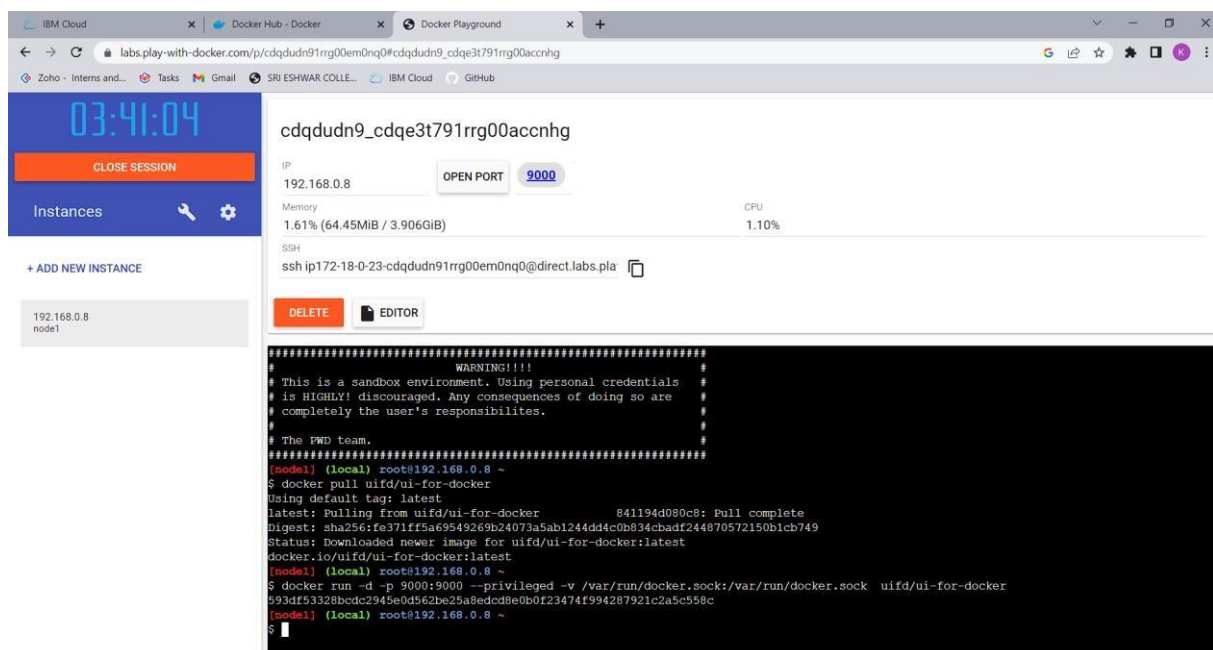
Assignment -4

Docker and Kubernetes

Assignment Date	21 October 2022
Student Name	Navaneethakrishnan.M
Student Roll Number	722819104085
Maximum Marks	2 Marks

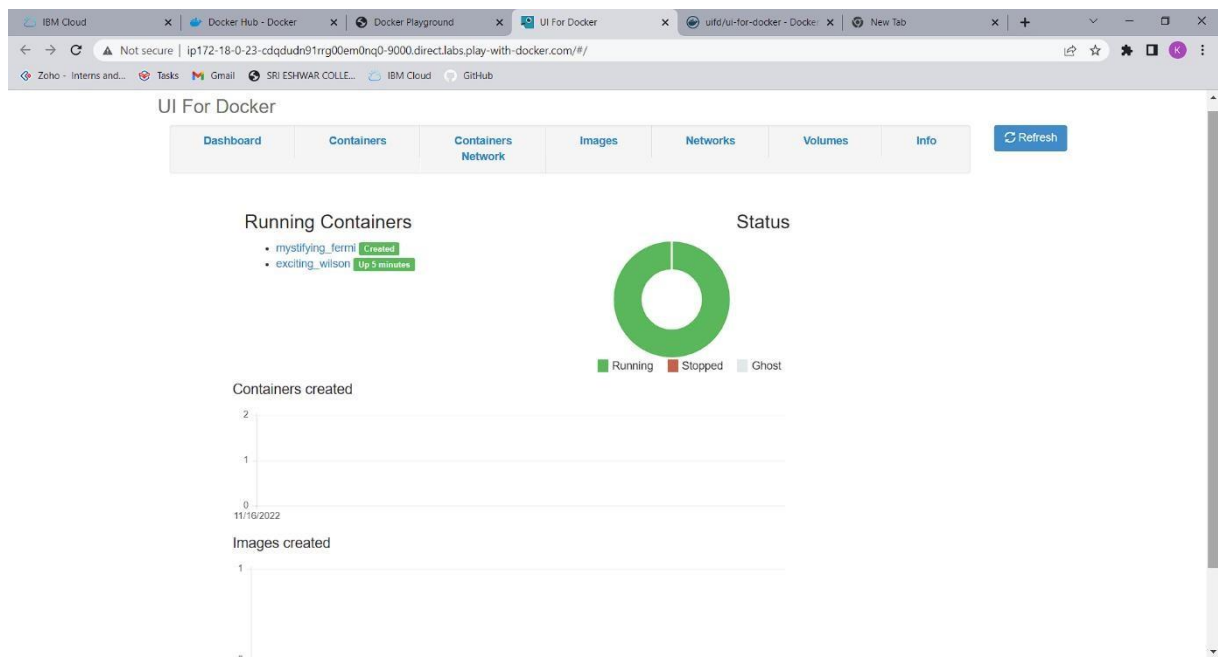
Question-1:

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



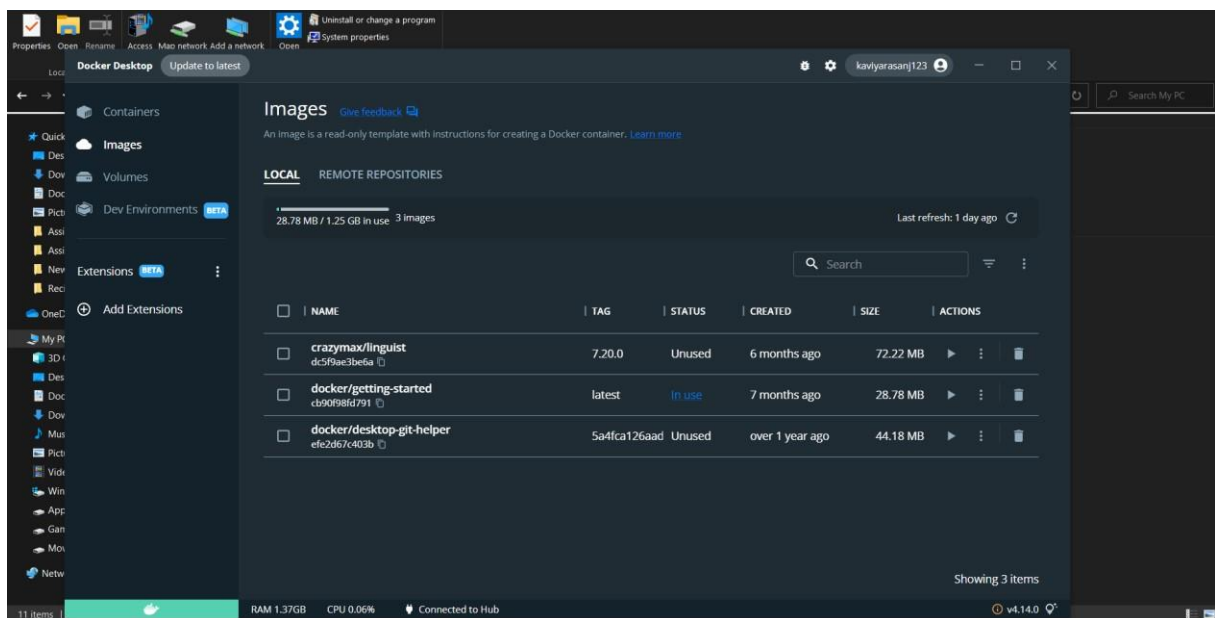
The screenshot displays the Docker Playground interface in a web browser. The top navigation bar includes links for IBM Cloud, Docker Hub, and Docker Playground. The main content area shows the instance details for 'cdqdu9n_cdqe3t791rrg00accnhg'. The instance is running on IP 192.168.0.8 with 1.61% memory usage (64.45MiB / 3.906GiB) and 1.10% CPU usage. The SSH command is 'ssh ip172-18-0-23-cdqdu9n1rrg00em0nq0@direct.labs.pla'. The terminal window shows the following commands and output:

```
##### WARNING!!!! #####
# This is a sandbox environment. Using personal credentials #
# is HIGHLY discouraged. Any consequences of doing so are #
# completely the user's responsibilities. #
# The FWD team. #
#####
[node1] (local) root@192.168.0.8 ~
$ docker pull uifd/ui-for-docker
Using default tag: latest
latest: Pulling from uifd/ui-for-docker
Digest: sha256:fe371ff5a69549269b24073a5ab1244dd4c0b834cbadf244870572150bcb749
Status: Downloaded newer image for uifd/ui-for-docker:latest
docker.io/uifd/ui-for-docker:latest
[node1] (local) root@192.168.0.8 ~
$ docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker
993df53328bcd2945e0d562be25a8edcd8e0b0f23474f994287921c2a5c558c
[node1] (local) root@192.168.0.8 ~
$
```



Question 2:

Create a docker file for the jobportal application and deploy it in Docker desktop application.



```
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:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc
-> resolve docker.io/library/python:3.6@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc
-> sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc 1.86kB / 1.86kB
-> sha256:d897a4907a8ec079df5ac31872359c2de510f82214c0448e926393b376d3b60d 2.22kB / 2.22kB
-> sha256:54208638d07c5e3ad24c6e21fc889abbc8486a27634c0892086ff71f3f44b104 9.27kB / 9.27kB
-> sha256:8e29546541cbbd389281d21f73a9d1b78665c1b95074f32089ee0776e1a7 54.97MB / 54.97MB
-> sha256:9b829c73b2b097d5c07a54f0ef3e921995a296c714b53a3aa67010231fcd 5.15MB / 5.15MB
-> sha256:c5b7bae36172f070eca53f35823e21baa85d61d5d95c5a95ab53d748cd056 10.87MB / 10.87MB
-> sha256:6494a4811622b31c027ccac322ca463937f0805f569a93e6f15c01aade718793 54.57MB / 54.57MB
-> sha256:6f9f74896dfa93fe0172f594fabae5e0b4a8a041a0fef09112efc7e4d3c78f7 196.53MB / 196.53MB
-> sha256:5e3b1213efc56598e78bd602983945c164de2a37205e06a62dada823124dc743 6.20MB / 6.20MB
-> extracting sha256:0e29546541cbbd389281d21f73a9d1b78665c1b95074f32089ee0776e1a7
-> sha256:9fddfd56334f2a6efad7e241bf5e7459c40ed105c5478676f41c1244bd96752 14.21MB / 14.21MB
-> extracting sha256:9b829c73b2b097d5c07a54f0ef3e921995a296c714b53a3aa67010231fcd 2.38kB / 2.38kB
-> extracting sha256:c5b7bae36172f070eca53f35823e21baa85d61d5d95c5a95ab53d748cd056 4.06kB / 4.06kB
-> sha256:404f0204bac0432ca522cbb9f254b1c91fca6800bfeaf0be0b743b2f31bab7 235B / 235B
-> sha256:c4f42be2be53b90ebffce04e10f13de538434ccc5f5d954a56848a6169a3a3f 2.21MB / 2.21MB
-> extracting sha256:6494a4811622b31c027ccac322ca463937f0805f569a93e6f15c01aade718793 27.38kB / 27.38kB
-> extracting sha256:6f9f74896dfa93fe0172f594fabae5e0b4a8a041a0fef09112efc7e4d3c78f7 131.46kB / 131.46kB
-> extracting sha256:5e3b1213efc56598e78bd602983945c164de2a37205e06a62dada823124dc743 8.25kB / 8.25kB
-> extracting sha256:9fddfd56334f2a6efad7e241bf5e7459c40ed105c5478676f41c1244bd96752 11.38kB / 11.38kB
-> extracting sha256:404f0204bac0432ca522cbb9f254b1c91fca6800bfeaf0be0b743b2f31bab7 0.06kB / 0.06kB
-> extracting sha256:c4f42be2be53b90ebffce04e10f13de538434ccc5f5d954a56848a6169a3a3f 2.25kB / 2.25kB
-> [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 sha256:1756719486df002fed5dee305c5221513f2ff2d1b49a8d242b22a28af0379f19
-> 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>
```

Question 3:

Create a IBM container registry and deploy helloworld app or jobportalapp.

```
Administrator: Windows PowerShell (800)
C:\Windows\system32> docker tag hello-world icr.io/1212ins/hello-world
PS C:\Windows\system32> docker push icr.io/1212ins/hello-world
Using default tag: latest
The push refers to repository [icr.io/1212ins/hello-world]
e0f0e1baac5f1: mounted from 8000ins/hello-world
latest: digest: sha256:c4f42be2be53b90ebffce04e10f13de538434ccc5f5d954a56848a6169a3a3f size: 525
PS C:\Windows\system32>
```

Question 4:

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

The screenshot shows the IBM Cloud console interface for a Kubernetes cluster named 'mycluster-kmnr'. The 'Worker nodes' tab is active, showing a table of worker pools. There is one worker pool named '00000009' in the 'Milan 01' zone. The node is in a 'Normal' status and is running version '1.24.7_1543'. The table also lists the private IP (10.144.212.250) and public IP (169.51.203.187) for the node. The console also shows the cluster's overview, including its status (Normal) and expiration date (Expires in 24 days).

Name	Status	Worker pool	Zone	Private IP	Public IP	Version
00000009	Normal	default	Milan 01	10.144.212.250	169.51.203.187	1.24.7_1543