

Assignment -4

Student Name	Sakthi Oviya . A
Student Roll Number	820419205047
Maximum Marks	2 Marks

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

The screenshot shows the Docker Playground interface for an instance named 'cdqghpv9_cdqhb3f91rrg00acd340'. The instance is running on IP 192.168.0.28 with port 9000 open. The terminal output shows the process of pulling the 'latest' image from 'library/docker' and attempting to run it. The output indicates that the image was pulled successfully, but there was an error during the connection to the Docker daemon on the host (8.8.8.53), stating 'dial tcp: lookup docker on 8.8.8.53: no such host'. The user then attempts to run the container with the command: `docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uid/ui-for-docker`. The output shows that the container failed to start due to a driver error: 'driver failed programming external connectivity on endpoint busy_bardeen (43b924e48b19f5ce51d6883fd ea22cefabefb1567211040fa6fb48dba8441f2e): Bind for 0.0.0.0:9000 failed: port is already allocated.'

The screenshot shows the 'UI For Docker' interface, specifically the 'Images' tab. The interface displays a list of images pulled from Docker Hub. The table below shows the details of the pulled images:

Select	Id	Repository	VirtualSize	Created
<input type="checkbox"/>	sha256:adc767c402...	docker:latest	143.8 MB	2022-10-26
<input type="checkbox"/>	sha256:965940f98f...	uid/ui-for-docker:latest	7.7 MB	2016-09-08

The interface also shows the Docker API Version: 1.41 and UI Version: v0.11.0.

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

The screenshot shows the Docker Desktop interface. On the left, a sidebar displays a timer at 03:41:57, a 'CLOSE SESSION' button, and a list of instances including '192.168.0.28 node1'. The main panel shows details for instance 'cdbuqoe0_cdbv24u3tccg009se10g', including its IP (192.168.0.28) and an 'OPEN PORT' button. Below this, a terminal window shows the execution of Docker commands:

```
$ docker pull docker/getting-started
Using default tag: latest
latest: Pulling from docker/getting-started
df9b9388f04a: Pull complete
5867cba5fcbd: Pull complete
4b639e65cb3b: Pull complete
061ed9e2b976: Pull complete
bc19f3e8eeb1: Pull complete
4071be97c256: Pull complete
79b586f1a54b: Pull complete
0c9732f525d6: Pull complete
Digest: sha256:b558be874169471bd4e65bd6eac8c303b271a7ee8553ba47481b73b2bf597aee
Status: Downloaded newer image for docker/getting-started:latest
[rodel] (local) root@192.168.0.28 ~
$ docker run -dp 80:80 docker/getting-started
ab2bf1a20f94991390dca336a2a087e7280f31c25ef2f0e3f9c9812a6fee0706
[rodel] (local) root@192.168.0.28 ~
$
```

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

The screenshot shows the IBM Cloud Container Registry console. The left sidebar contains navigation links for 'Container Registry', 'Quick start', 'Namespaces', 'Repositories', 'Images', 'Trash', and 'Settings'. The main area is titled 'Namespaces' and shows a table of existing namespaces. A 'Create namespace' notification is visible in the top right. The table lists namespaces 'helloworld11' and 'test112001', both in the 'Default' resource group with 0 repository and image counts.

Name	Resource group	Repository count	Image count	Retention policy
helloworld11	Default	0	0	
test112001	Default	0	0	

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

