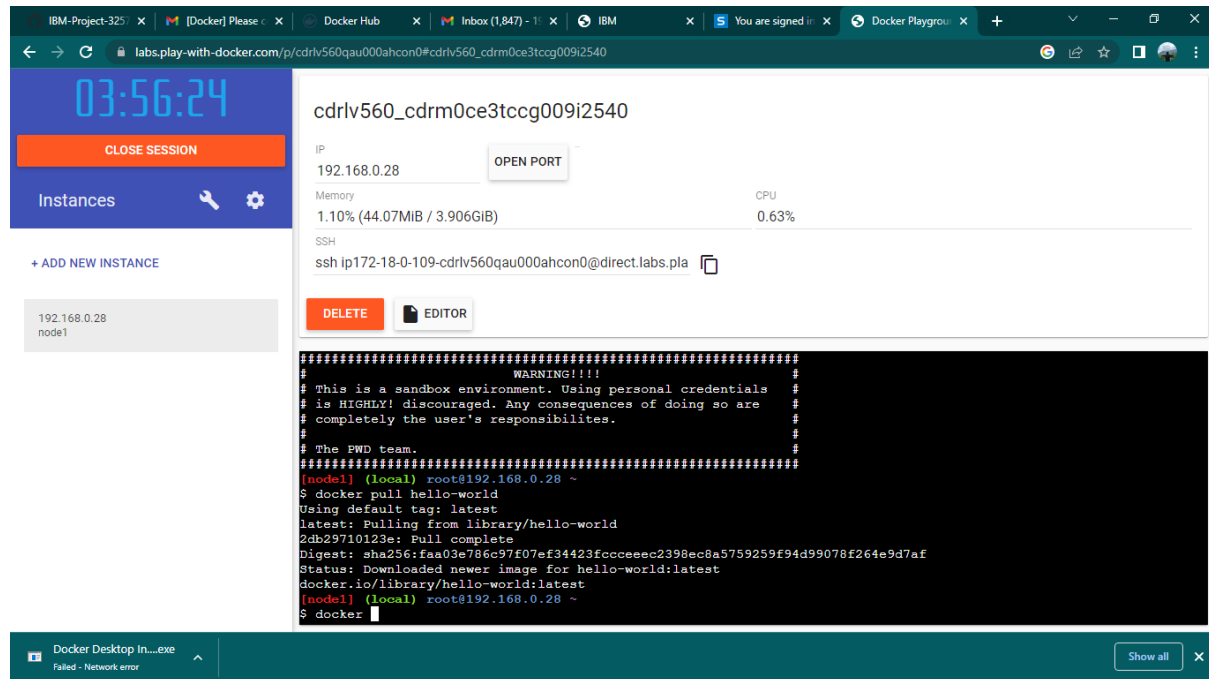


PERSONAL EXPENSE TRACKER-CLOUD APPLICATION DEVELOPMENT

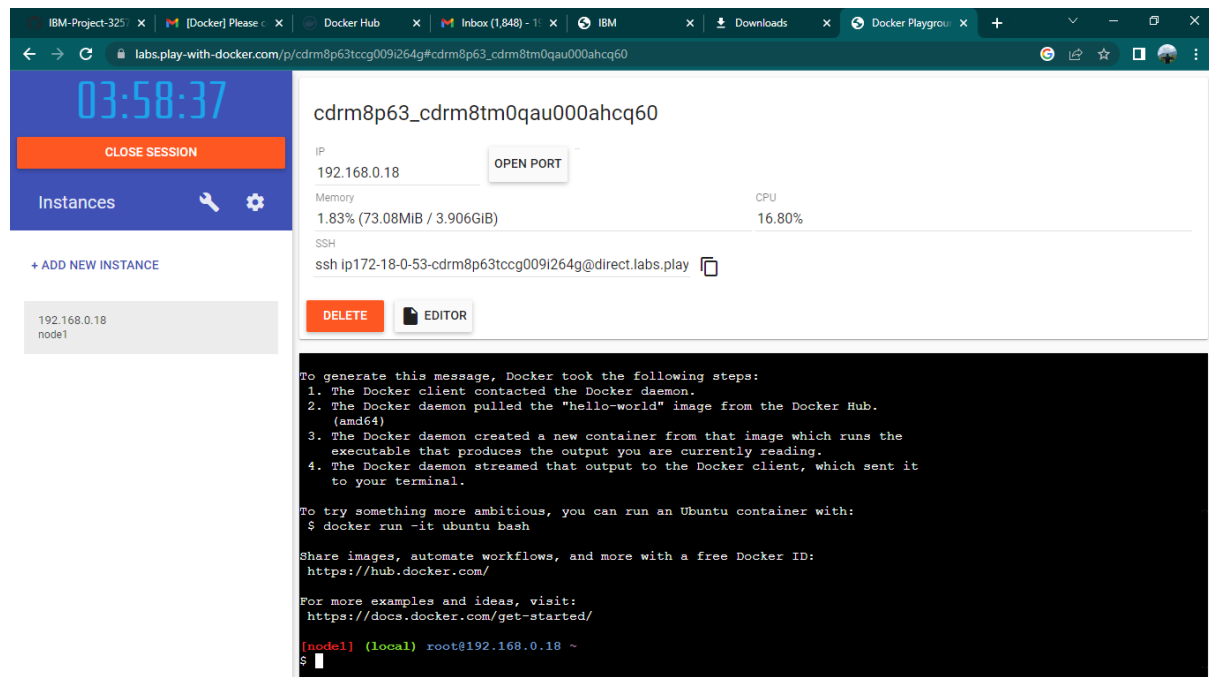
Assignment 4

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



The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a clock showing 03:56:24, a 'CLOSE SESSION' button, and a list of instances. The main area displays details for a container named 'cdrv560_cdrm0ce3tccg009i2540'. It shows the IP address 192.168.0.28, memory usage (1.10% / 44.07MiB / 3.906GiB), and CPU usage (0.63%). Below this, there's an SSH link and buttons for 'DELETE' and 'EDITOR'. The terminal window shows the following 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 PWD team.  
#####  
[node1] (local) root@192.168.0.28 ~  
$ docker pull hello-world  
Using default tag: latest  
latest: Pulling from library/hello-world  
2db29710123e: Full complete  
Digest: sha256:f5a03e786c97f07ef34423fccceec2398ec8a5759259f94d99078f264e9d7af  
Status: Downloaded newer image for hello-world:latest  
docker.io/library/hello-world:latest  
[node1] (local) root@192.168.0.28 ~  
$ docker
```

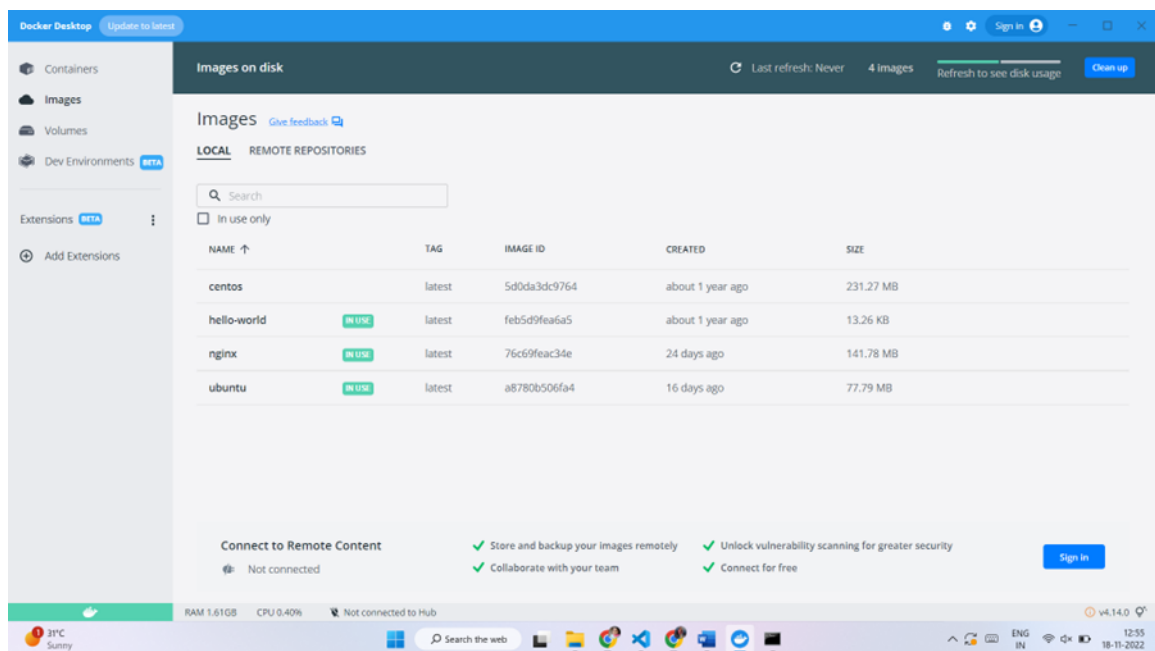


The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a clock showing 03:58:37, a 'CLOSE SESSION' button, and a list of instances. The main area displays details for a container named 'cdrm8p63_cdrm8tm0qau000ahcq60'. It shows the IP address 192.168.0.18, memory usage (1.83% / 73.08MiB / 3.906GiB), and CPU usage (16.80%). Below this, there's an SSH link and buttons for 'DELETE' and 'EDITOR'. The terminal window shows the following output:

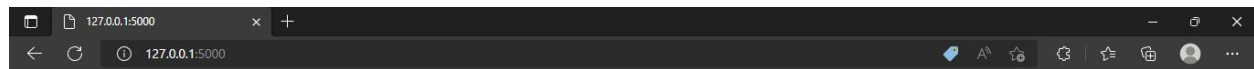
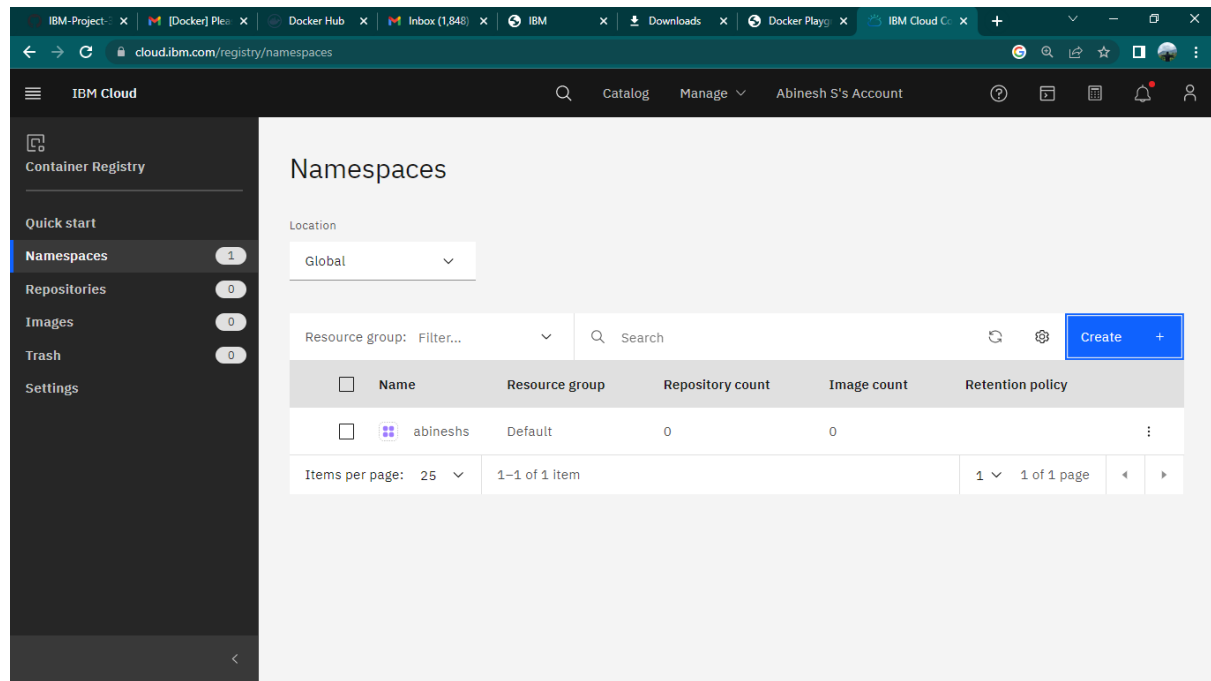
```
To generate this message, Docker took the following steps:  
1. The Docker client contacted the Docker daemon.  
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
   (amd64)  
3. The Docker daemon created a new container from that image which runs the  
   executable that produces the output you are currently reading.  
4. The Docker daemon streamed that output to the Docker client, which sent it  
   to your terminal.  
  
To try something more ambitious, you can run an Ubuntu container with:  
$ docker run -it ubuntu bash  
  
Share images, automate workflows, and more with a free Docker ID:  
https://hub.docker.com/  
  
For more examples and ideas, visit:  
https://docs.docker.com/get-started/  
[node1] (local) root@192.168.0.18 ~  
$
```

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

```
FROM python:3.7
COPY ./app
WORKDIR /app
COPY requirements.txt /app
RUN python -m pip install -r requirements.txt
EXPOSE 5001
ENTRYPOINT [ "python" ]
CMD [ "app.py" ]
```



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



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.

