

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 03:07:25, a 'CLOSE SESSION' button, and an 'Instances' section with a '+ ADD NEW INSTANCE' button. Below this, a list of instances shows '192.168.0.7 node1'. The main area displays the instance details for 'cdobdh60_cdoc5sm0qau000ccntlg', including IP (192.168.0.7), Memory (1.09% (43.76MiB / 3.906GiB)), CPU (0.15%), and an SSH command: 'ssh ip172-18-0-78-cdobdh60qau000ccnsgg@direct.labs.pla'. Below this, there are 'DELETE' and 'EDITOR' buttons. The terminal window shows a warning message and the command 'docker pull hello-world' being executed. The output shows the image being pulled from the Docker Hub.

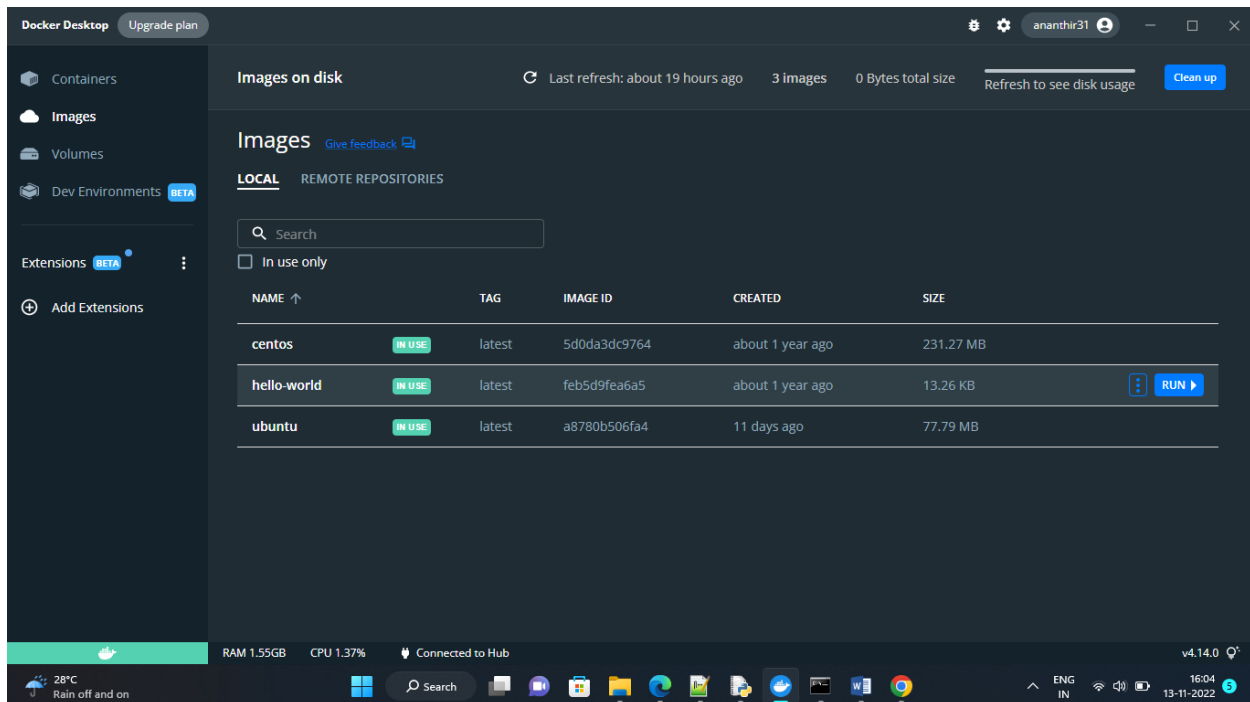
```
#####  
# WARNING!!!!  
# This is a sandbox environment. Using personal credentials  
# is HIGHLY! discouraged. Any consequences of doing so are  
# completely the user's responsibilities.  
#  
# The FWD team.  
#####  
[node1] (local) root@192.168.0.7 ~  
$ docker pull hello-world  
Using default tag: latest  
latest: Pulling from library/hello-world  
2db29710123e: Pull complete  
Digest: sha256:faa03e786c97f07ef34423fccceec2398ec8a5759259f94d99078f264e9d7af  
Status: Downloaded newer image for hello-world:latest  
docker.io/library/hello-world:latest  
[node1] (local) root@192.168.0.7 ~  
$
```

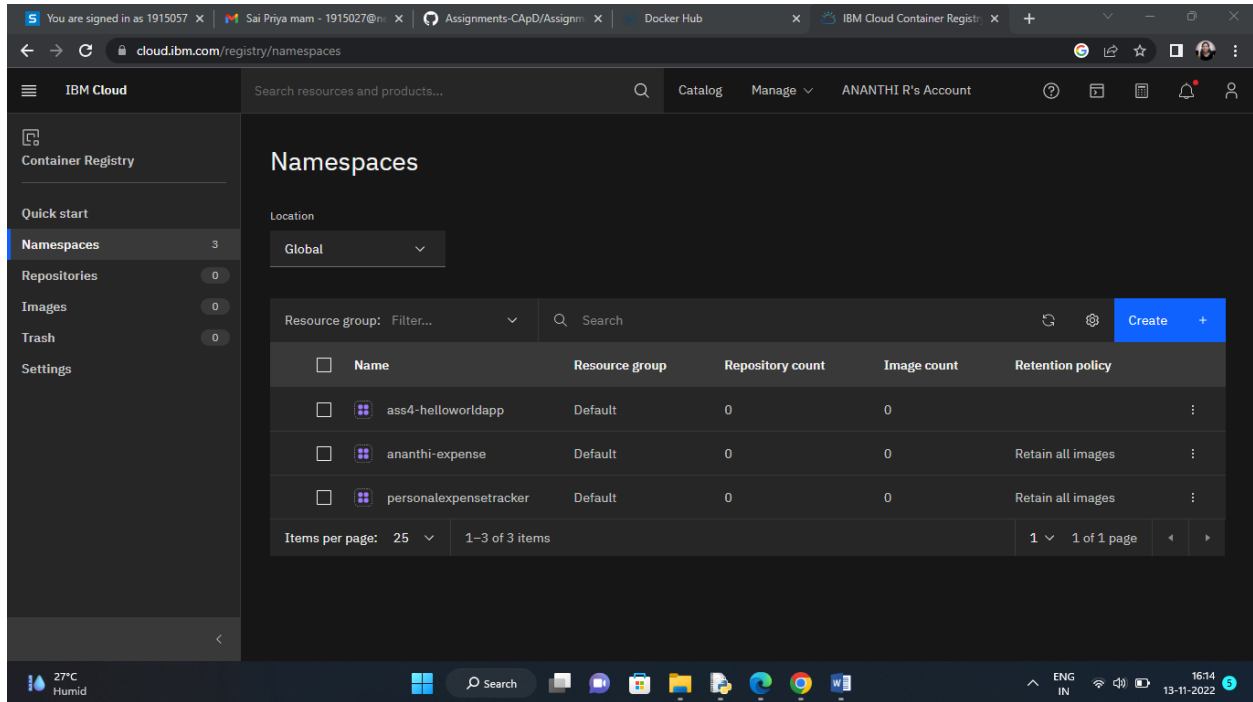
The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a clock showing 03:06:10, a 'CLOSE SESSION' button, and an 'Instances' section with a '+ ADD NEW INSTANCE' button. Below this, a list of instances shows '192.168.0.7 node1'. The main area displays the instance details for 'cdobdh60_cdoc5sm0qau000ccntlg', including IP (192.168.0.7), Memory (1.21% (48.28MiB / 3.906GiB)), CPU (0.56%), and an SSH command: 'ssh ip172-18-0-78-cdobdh60qau000ccnsgg@direct.labs.pla'. Below this, there are 'DELETE' and 'EDITOR' buttons. The terminal window shows the command 'docker run hello-world' being executed. The output shows the 'Hello from Docker!' message and a list of steps explaining the process.

```
[node1] (local) root@192.168.0.7 ~  
$ docker run hello-world  
  
Hello from Docker!  
This message shows that your installation appears to be working correctly.  
  
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
```

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.**



Hello World



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.