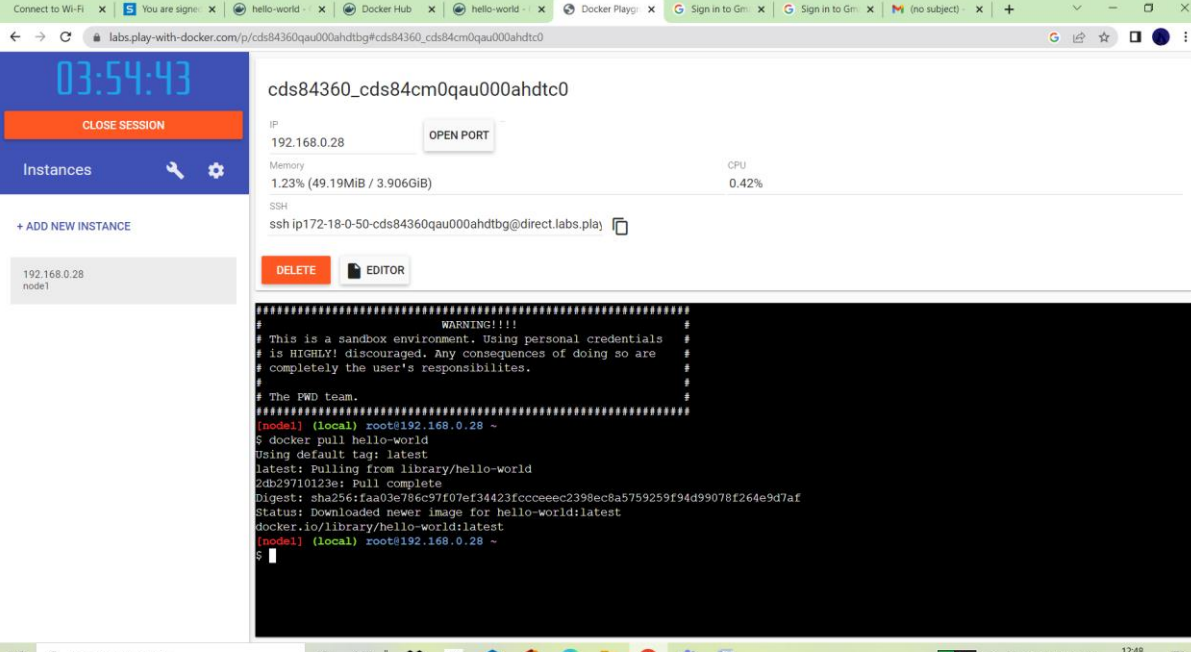


Team ID :PNT2022TMID20692

Project Name : News Tracker Application

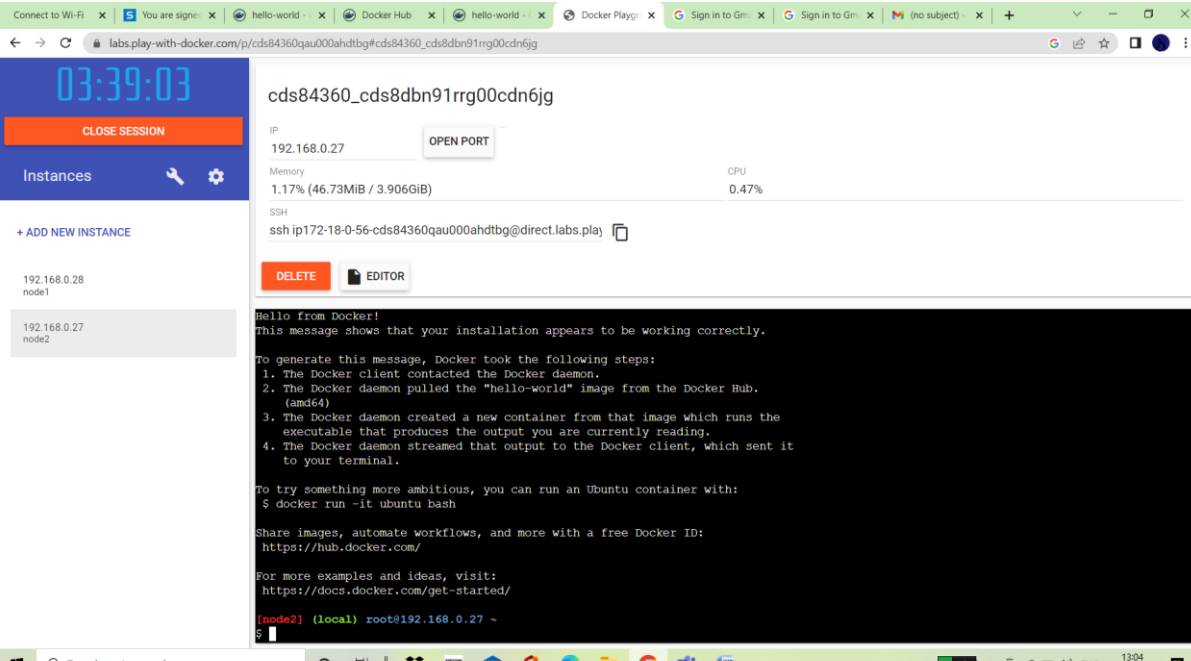
Name : Abdul Nazer M

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



The screenshot shows the Docker Playground interface in a web browser. The top bar displays the URL: `labs.play-with-docker.com/p/cds84360qau000ahdtbg#cds84360_cds84cm0qau000ahdtc0`. The left sidebar shows a timer at 03:54:43, a 'CLOSE SESSION' button, and a list of instances with 'node1' at IP 192.168.0.28. The main panel shows the instance details for 'cds84360_cds84cm0qau000ahdtc0' with IP 192.168.0.28, 1.23% memory usage, and 0.42% CPU usage. The terminal window displays 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 FWD team. #
#####
(node1) (local) root@192.168.0.28 ~
$ docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:faa03e786c97f07ef34423fcc0eeec2398ec8a5759259f94d99078f264e9d7af
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
(node1) (local) root@192.168.0.28 ~
$
```



The screenshot shows the Docker Playground interface in a web browser. The top bar displays the URL: `labs.play-with-docker.com/p/cds84360qau000ahdtbg#cds84360_cds8dbn91rrg00cdn6jg`. The left sidebar shows a timer at 03:39:03, a 'CLOSE SESSION' button, and a list of instances with 'node1' at IP 192.168.0.28 and 'node2' at IP 192.168.0.27. The main panel shows the instance details for 'cds84360_cds8dbn91rrg00cdn6jg' with IP 192.168.0.27, 1.17% memory usage, and 0.47% CPU usage. The terminal window displays the following output:

```
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.

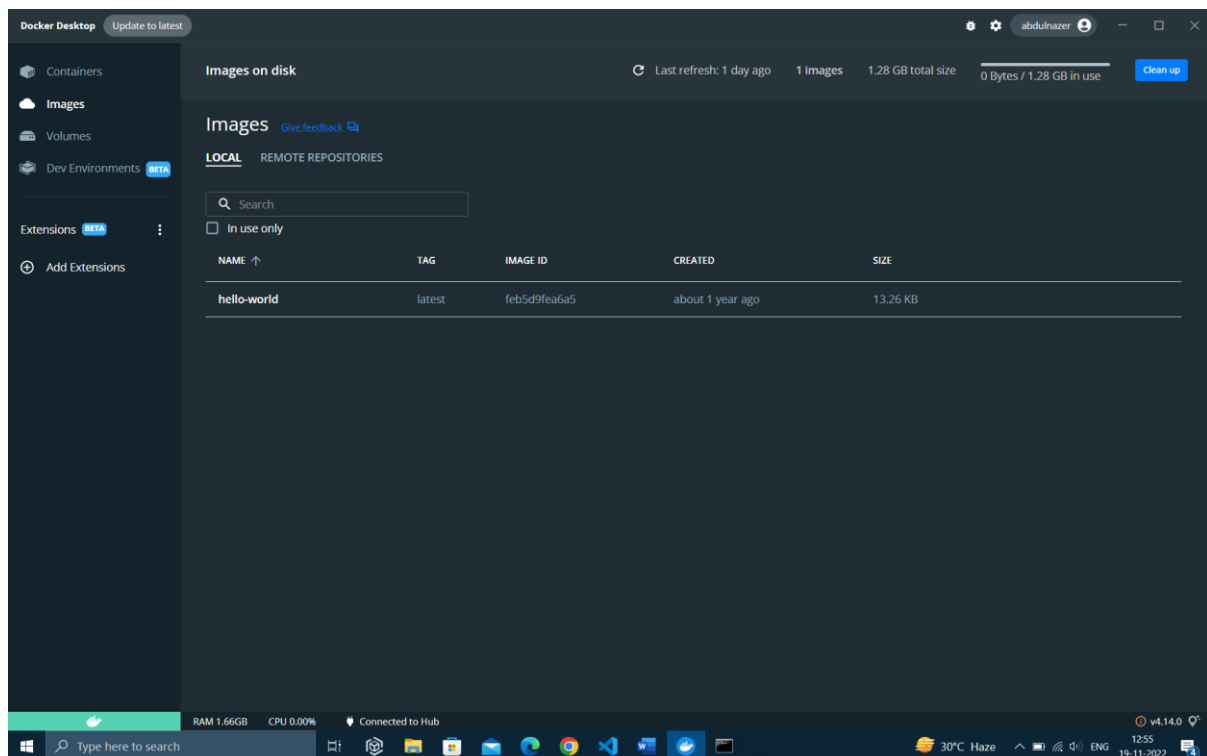
To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

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/
(node2) (local) root@192.168.0.27 ~
$
```

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

```
FROM python:3.10
WORKDIR /app
ADD . /app
COPY requirements.txt /app
RUN python -m pip install -r requirements.txt
RUN python -m pip install ibm_db
EXPOSE 5001
ENTRYPOINT [ "python" ]
CMD [ "app.py" ]
```



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

The screenshot shows the IBM Cloud Container Registry interface. The left sidebar contains the 'Container Registry' menu with options: Quick start, Namespaces (1), Repositories (0), Images (0), Trash (0), and Settings. The main area is titled 'Namespaces' and shows a table with one namespace: 'assi-hello-world' in the 'Default' resource group, with 0 repository and 0 image counts. A 'Create' button is visible in the top right of the namespace list.

Name	Resource group	Repository count	Image count	Retention policy
assi-hello-world	Default	0	0	

Items per page: 25 | 1-1 of 1 item | 1 of 1 page