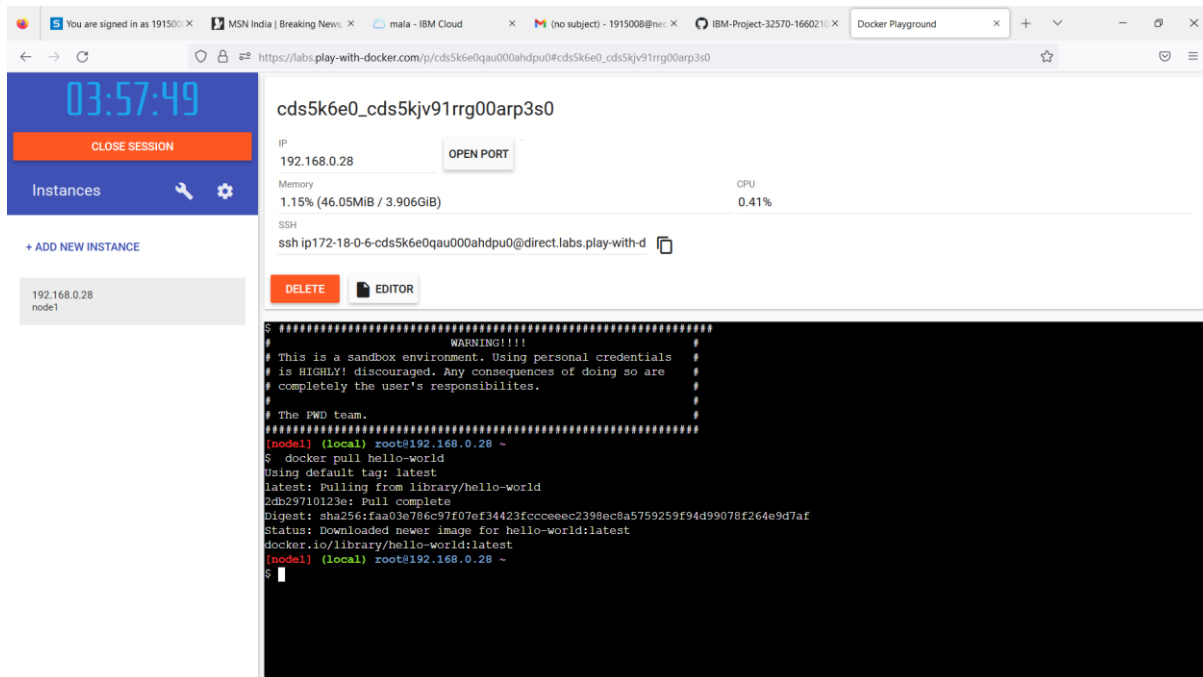
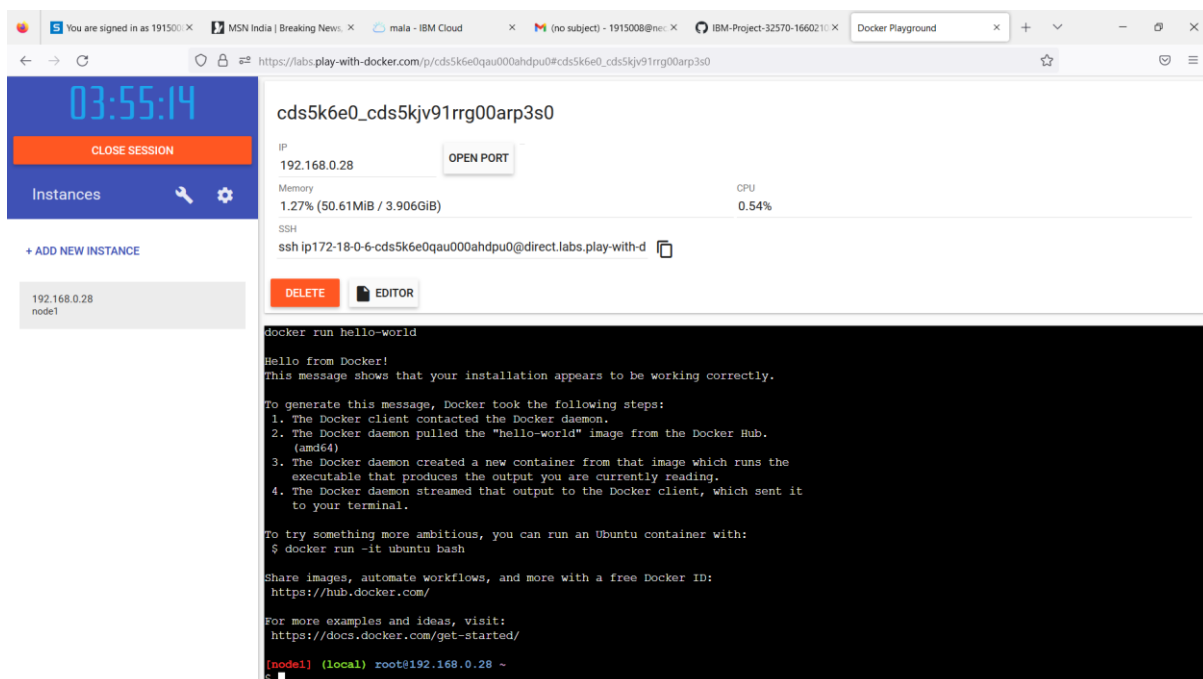


# 1.Pull an image from docker hub and run it in docker playground



The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a clock showing 03:57:49, a 'CLOSE SESSION' button, and an 'Instances' section with a list of instances including '192.168.0.28 node1'. The main area displays the details for a container named 'cds5k6e0\_cds5k91rrg00arp3s0'. It shows the IP address '192.168.0.28', memory usage '1.15% (46.05MiB / 3.906GiB)', and CPU usage '0.41%'. There are buttons for 'OPEN PORT', 'DELETE', and 'EDITOR'. The terminal window shows a warning message and the command 'docker pull hello-world' being executed, resulting in the latest image being pulled from 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@192.168.0.28 ~  
$ 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 after running the 'docker run hello-world' command. The container 'cds5k6e0\_cds5k91rrg00arp3s0' now shows 1.27% memory usage and 0.54% CPU usage. The terminal window displays the output of the command, which is a 'Hello from Docker!' message followed by a list of steps explaining how Docker works.

```
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  
  
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@192.168.0.28 ~  
$
```

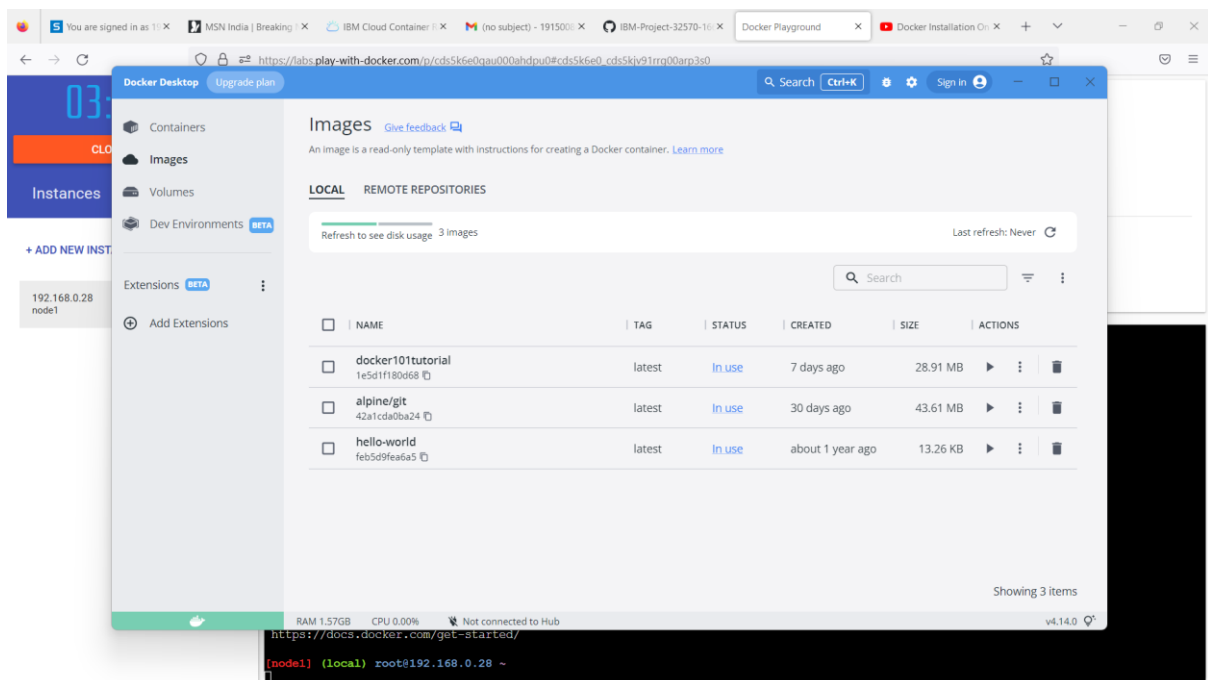
## 2.Create a docker file for the jobportal application and deploy it in Docker desktopapplication.

```
FROM
python:3.7
COPY.
/app
WORKDI
R /app

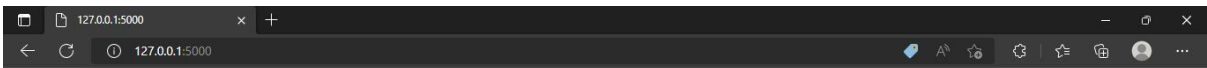
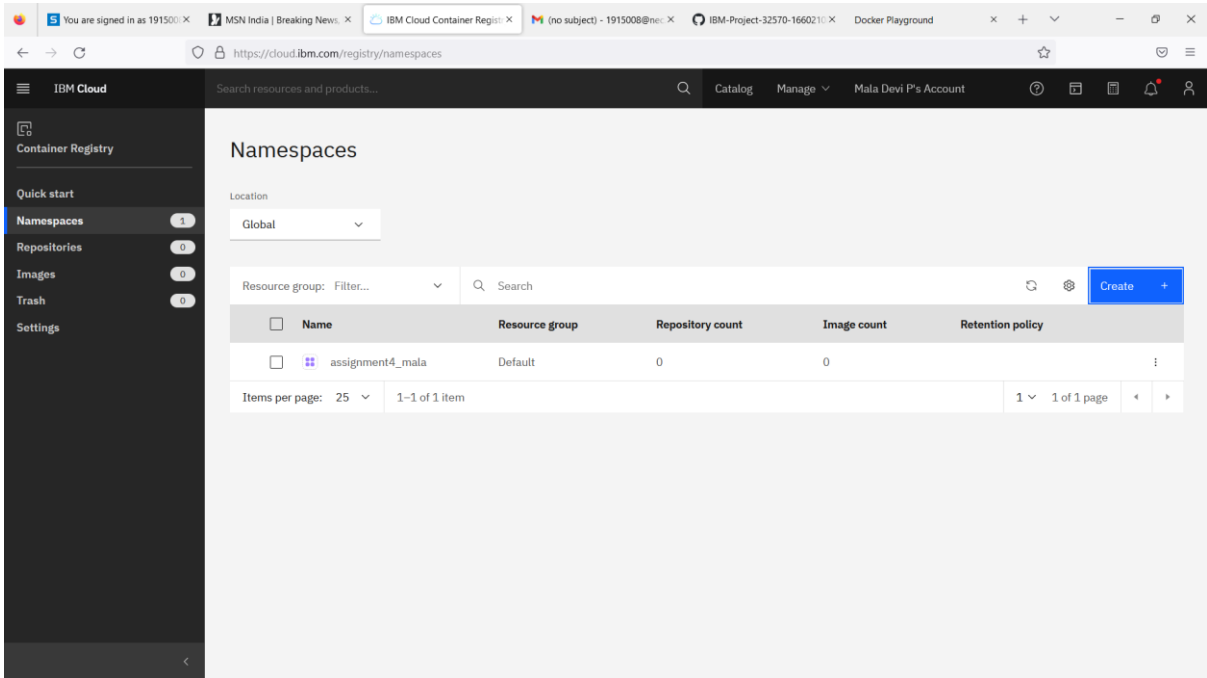
COPY requirements.txt /app

RUN python -m pip install -r
requirements.txtEXPOSE 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.

The screenshot displays the IBM Cloud Kubernetes dashboard for a cluster named 'mala'. The interface includes a top navigation bar with the IBM Cloud logo, a search bar, and user account information. A left sidebar lists navigation options: Overview, Worker nodes, Worker pools, and DevOps (marked as 'New'). The main content area shows the cluster's overview, including a warning banner about a 30-day expiration, status cards for Node status (1 of 1 Normal), Add-on status (0 of 0 Normal), Master status (Normal), and Ingress status (Healthy). Below these is a 'Details' section with metadata such as Cluster ID, Version (1.24.8\_1544), Infrastructure (Classic), Zones (Milan 01), Created time, Resource group, and Image security enforcement (Enabled). At the bottom, the 'Node health' section shows 1 total nodes in a healthy state, represented by a green bar.

Node status	Add-on status	Master status	Ingress status
1 of 1 Normal	0 of 0 Normal	Normal	Healthy

Cluster ID	Version	Infrastructure	Zones
cds56p0f01ate880au70	1.24.8_1544	Classic	Milan 01

Created	Resource group	Image security enforcement
11/19/2022, 9:24 AM	Default	Enable

Node health: 1 total nodes