Exp-2

```
sudo apt update -y
sudo apt install apt-transport-https ca-certificates curl software-properties-common -y
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic
stable" -y
sudo apt update -y
apt-cache policy docker-ce -y
sudo apt install docker-ce -y
#sudo systemctl status docker
sudo chmod 777 /var/run/docker.sock
docker pull ubuntu
docker run -it ubuntu bash
docker ps -a
```

Exp-5

```
hadoop-3.3.6/bin/hdfs namenode -format
export PDSH_RCMD_TYPE=ssh

start-dfs.sh
start-yarn.sh
jps
browse - localhost:9870
hdfs dfs -ls /
hdfs dfs -mkdir /test
hdfs dfs -put /home/hadoop/data.txt   /test
hadoop jar  /home/hadoop/WordCount.jar  com.demo.WordCount /test/data.txt /test2
hadoop fs -cat /test2/part-r-00000
```

Exp-6

sudo snap install microstack --beta

Snap list microstack

sudo microstack init --auto --control

-----goto->10.20.20.1

Lointid-admin

Password-

sudo snap get microstack config.credentials

microstack launch cirros -n MyVM2

Run the docker container from recently created image and run that docker container

Here are the steps to create and run a Docker container from a recently created image:

---

1. Prerequisites

- Ensure Docker is installed and running on your system.
- Confirm you have an image ready for use. List your images using:

bash

Copy code

docker images

Note the image name or ID.

---

2. Steps to Run a Docker Container

Step 1: Create a Docker Image

If you have not already created an image, you can do so using a Dockerfile. Example:

1. Create a Dockerfile:

Dockerfile

Copy code

```
FROM ubuntu:latest

RUN apt-get update && apt-get install -y curl

CMD ["bash"]
```

2. Build the image:

bash

Copy code

```
docker build -t my-custom-image .
```

Here, my-custom-image is the name you assign to the image.

Step 2: Run a Container from the Image

Once the image is created, you can run it as a container using:

bash

Copy code

```
docker run -it --name my-container-name my-custom-image
```

- **Options:**
    - -it: Runs the container in interactive mode with a terminal.
    - --name my-container-name: Assigns a name to the container.
    - my-custom-image: Replace this with your image name or ID.

Step 3: Verify the Running Container

Check if the container is running:

bash

Copy code

```
docker ps
```

Step 4: Access the Running Container

If you want to access the container's terminal:

bash

Copy code

```
docker exec -it my-container-name bash
```

Step 5: Stop the Container

To stop the container:

bash

Copy code

```
docker stop my-container-name
```

---

3. Optional: Save Container State

If you make changes inside the container and want to save them, commit the container to a new image:

bash

Copy code

```
docker commit my-container-name new-image-name
```

Create a Docker image of simple web application from using HTTP web server

---

Steps to Create a Docker Image for a Simple Web Application

1. Prerequisites

- Docker installed and running on your machine.
- A basic understanding of Dockerfile creation.

---

2. Write the Web Application Code

Create a simple HTML file to serve as your web application. For example:

1. Create a directory for your project:

bash

Copy code

```
mkdir simple-web-app
cd simple-web-app
```

2. Create an index.html file:

html

Copy code

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Simple Web App</title>
</head>
<body>
   <h1>Welcome to My Simple Web App!</h1>
</body>
</html>
```

---

3. Create a Dockerfile

Inside the same directory, create a Dockerfile. This will define how the image is built.

Using Apache (httpd):

Dockerfile

Copy code

```
# Use the official Apache image
FROM httpd:latest

# Copy the HTML file to the web server's document root
COPY index.html /usr/local/apache2/htdocs/
```

---

4. Build the Docker Image

Run the following command in the directory containing the Dockerfile:

bash

Copy code

```
docker build -t simple-web-app .
```

- -t simple-web-app: Tags the image with the name simple-web-app.

- .: Specifies the current directory as the build context.

---

5. Run a Container from the Image

Start a container to serve your web application:

bash

Copy code

```
docker run -d -p 8080:80 --name simple-web-container simple-web-app
```

- -d: Runs the container in detached mode.
- -p 8080:80: Maps port 80 inside the container to port 8080 on your host.
- --name simple-web-container: Assigns a name to the container.
- simple-web-app: The name of your image.

---

6. Test the Web Application

1. Open your web browser.
2. Go to http://localhost:8080.
3. You should see the content of your index.html file.

---

7. Stop and Remove the Container (Optional)

To stop the container:

bash

Copy code

```
docker stop simple-web-container
```

To remove the container:

bash

Copy code

```
docker rm simple-web-container
```