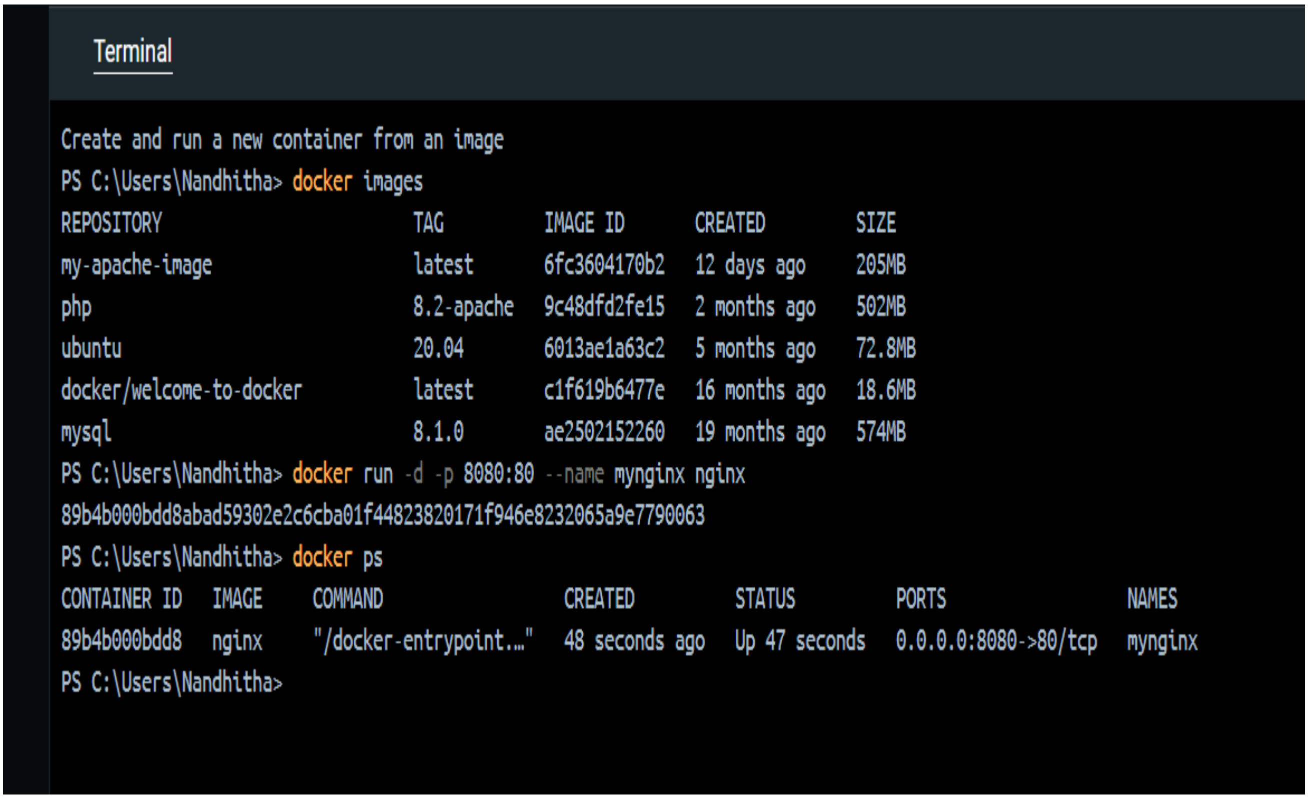
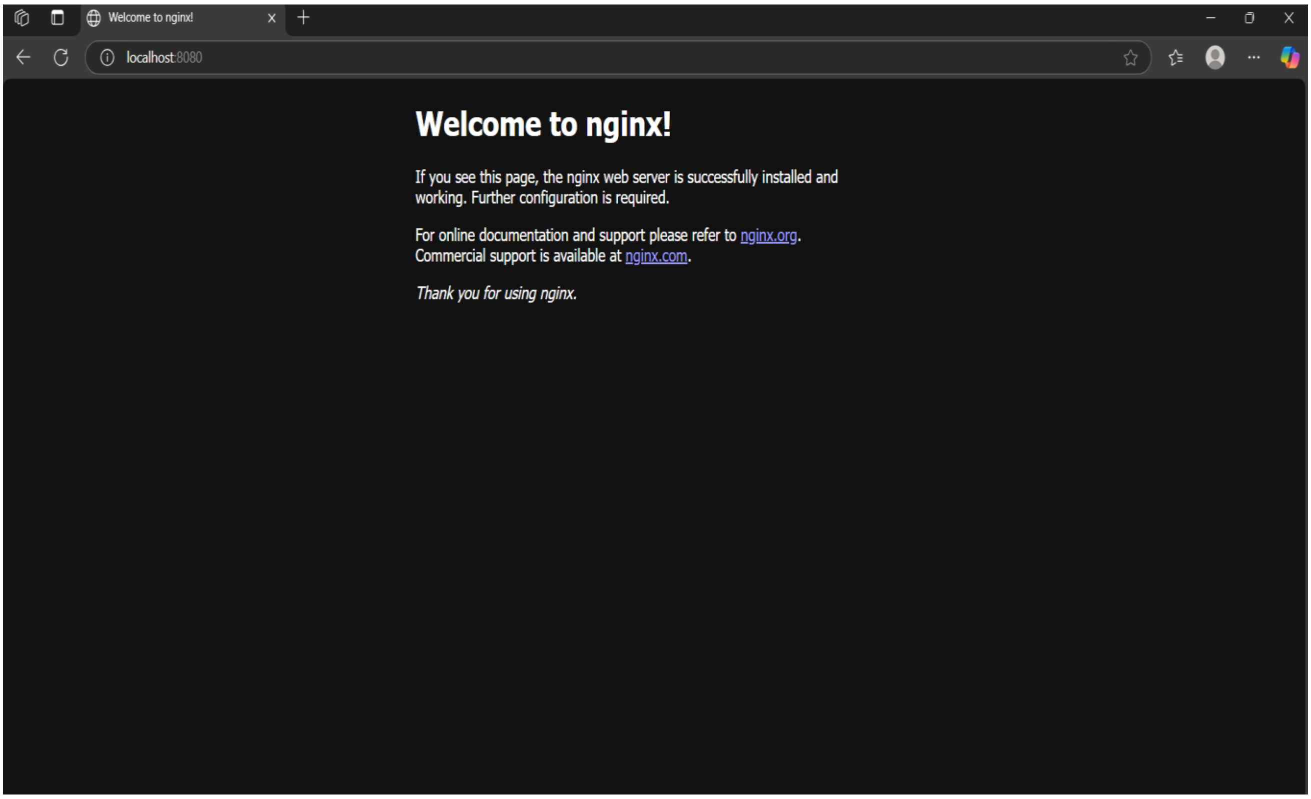


```
Create and run a new container from an image
PS C:\Users\Nandhitha> docker images
REPOSITORY                   TAG          IMAGE ID       CREATED          SIZE
my-apache-image              latest       6fc3604170b2   12 days ago      205MB
php                          8.2-apache   9c48dfd2fe15   2 months ago     502MB
ubuntu                       20.04        6013ae1a63c2   5 months ago     72.8MB
docker/welcome-to-docker     latest       c1f619b6477e   16 months ago    18.6MB
mysql                        8.1.0        ae2502152260   19 months ago    574MB
PS C:\Users\Nandhitha> docker run -d -p 8080:80 --name mynginx nginx
89b4b000bdd8abad59302e2c6cba01f44823820171f946e8232065a9e7790063
PS C:\Users\Nandhitha> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                  NAMES
89b4b000bdd8   nginx     "/docker-entrypoint.…"   48 seconds ago   Up 47 seconds   0.0.0.0:8080->80/tcp   mynginx
PS C:\Users\Nandhitha>
```



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

DOCKER

Docker is a platform that provides virtual containers on which an application can be deployed independent of the underlying OS of the server.

Further the container can be created from a replica called docker image which contains all the dependencies and can run on any OS that has docker engine, with similar results.

VIRTUALIZATION:

Virtualization is the process of sharing hardware resources across several virtually isolated and mutually independent systems.

It is achieved by using a hypervisor which acts as a bridge between the Operating System of each of the virtual machines and the underlying hardware.

Applications in virtual environments run on a host operating system on top of the hypervisor.

BASIC DOCKER COMMANDS

- Display docker images available in our machine

```
$ docker images
```

- Download docker image.

```
$ docker pull <image-name / image-id>
```

- Run docker image.

```
$ docker run <image-name / image-id>
```

- Delete docker image.

```
$ docker rmi <image-name / image-id>
```

- Display all running docker containers.

```
$ docker ps
```

- Display all running and stopped containers.

```
$ docker ps -a
```

- Delete docker container.

```
$ docker rm <container-id>
```

- Delete docker image forcefully.

```
$ docker rmi  -f <image-id>
```

- Stop Docker container.

```
$ docker stop <container-id>
```

#DOCKER COMMANDS FOR UBUNTU

```
$ sudo apt update -y

$ sudo apt install docker -y

$ sudo service docker start (or) sudo systemctl start docker

$ sudo service docker enable (or) sudo systemctl enable docker
```

DOCKER COMPOSE

Docker Compose is a tool that allows you to define and manage multi-container Docker applications. It simplifies the process of running multiple containers, their configurations, and their interdependencies. Compose uses a YAML file to define the services, networks, and volumes required for your application.

	Docker Compose is a tool which is used to manage multi container-based applications.

	Using Docker Compose we can easily setup & deploy multi container-based applications.

	We will give containers information to Docker Compose using YML file (docker-compose.yml)

	Docker Compose YML should have all the information related to containers creation.

	Docker Compose YML File Looks Like: