

IMPLEMENTATION OF CONTAINERS

Name:Vijay Krishna S

RollNo:2023115075

Aim

To install Docker on Windows Subsystem for Linux (WSL), create and manage containers, and execute a Python environment inside a Docker container.

Objective

- Install Docker Engine in WSL.
- Verify Docker functionality.
- Run lightweight containers.
- Deploy a web server container.
- Execute Python inside a containerized environment.

System Requirements

- Windows 10/11 with WSL enabled
- Ubuntu (WSL distribution)
- Docker Engine
- Internet connectivity

Procedure

Step 1 – Install Docker

Docker packages and dependencies were installed using the apt package manager.

Command used:

`sudo apt update`

`sudo apt install docker.io`

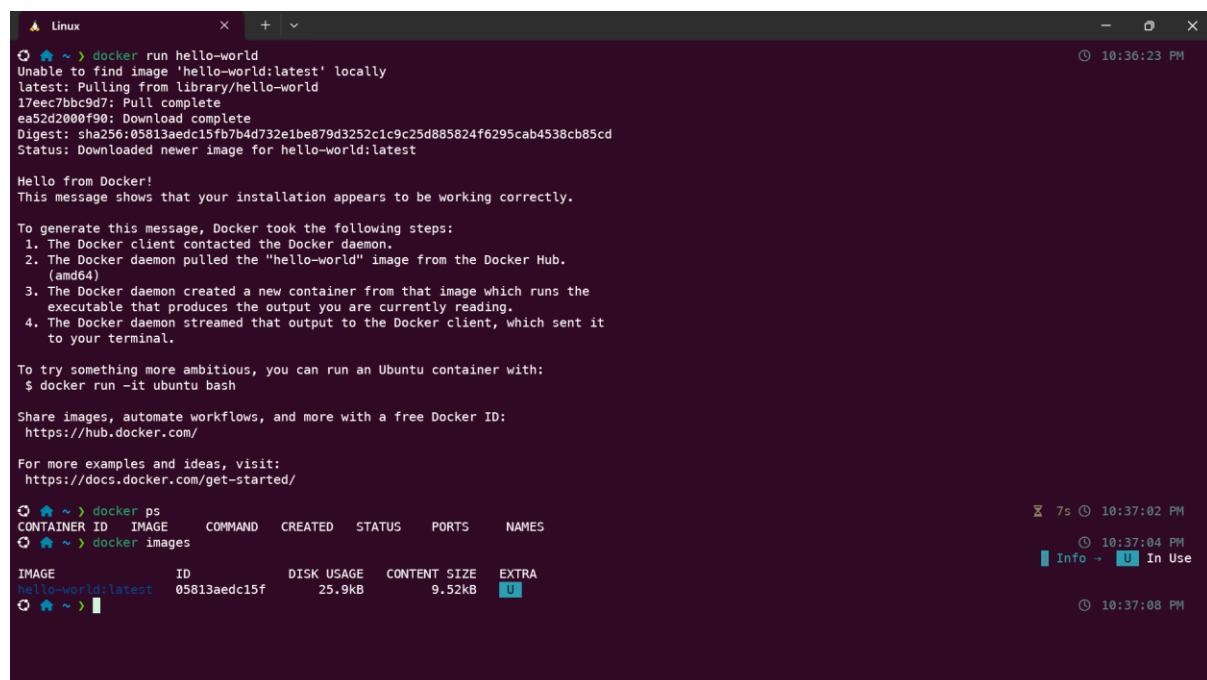
Step 2 – Start Docker Service

The Docker service was started and verified.

`sudo systemctl start docker`

`docker --version`

Step 3 – Create Containers



A screenshot of a terminal window titled "Linux". The window shows the following Docker commands and their outputs:

```
○ ~ > docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17ec7bbc9d7: Pull complete
e052d2000f90: Download complete
Digest: sha256:05813aedc15fb7b4d732e1be879d3252c1c9c25d885824f6295cab4538cb85cd
Status: Downloaded newer image for hello-world:latest

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/
```

At the bottom of the terminal, there are two status bars. The left one shows "CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES" and lists "hello-world:latest" with its details. The right status bar shows "7s 10:37:02 PM", "10:37:04 PM", "Info → In Use", and "10:37:08 PM".

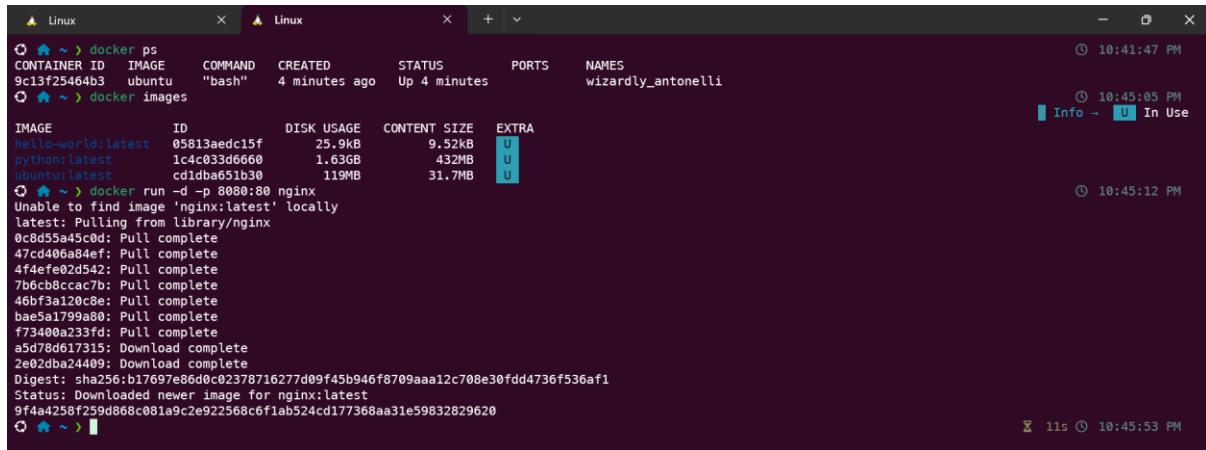


```

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
05813aedc15f hello-world:latest 25.9kB 9.52kB U
1c4c03d6660 python:latest 1.63GB 432MB U

```

docker run -it ubuntu bash
 Unable to find image 'ubuntu:latest' locally
 latest: Pulling from library/ubuntu
 a3629ac5b9f4: Pull complete
 1ba0f5536e37: Download complete
 Digest: sha256:c1dba51b300c3686ecf4e3c4220f026b521fb7697881737d24f20082b2b
 Status: Downloaded newer image for ubuntu:latest
 root@9c13f25464b3:/# ls
 bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
 root@9c13f25464b3:/# ^C

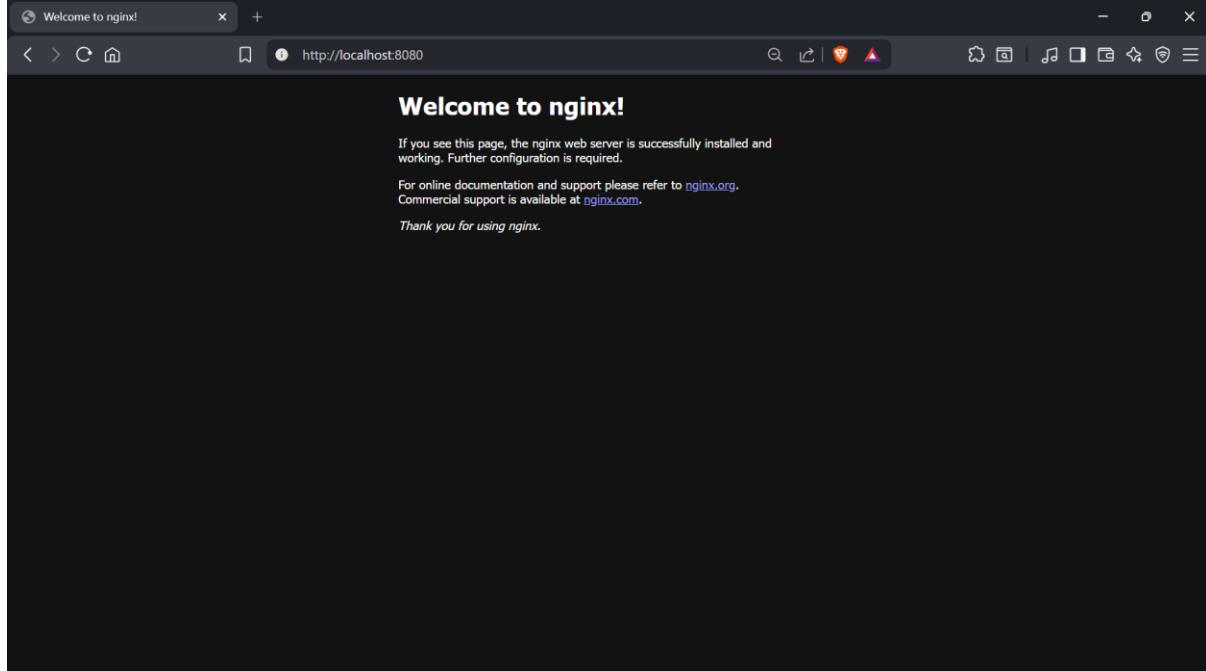


```

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
9c13f25464b3 ubuntu "bash" 4 minutes ago Up 4 minutes wizardly_antonelli

```

docker run -d -p 8080:80 nginx
 Unable to find image 'nginx:latest' locally
 latest: Pulling from library/nginx
 0c8d55a45c0d: Pull complete
 47cd406a84ef: Pull complete
 4f4fe082d542: Pull complete
 7b6ccb8cac7b: Pull complete
 46bf3a120c8e: Pull complete
 bae5a1799a80: Pull complete
 f73400a233fd: Pull complete
 a5d78d617315: Download complete
 2e02db424409: Download complete
 Digest: sha256:b17697e86d0c02378716277d09f45b946f8709aaa12c708e30ffd4736f536af1
 Status: Downloaded newer image for nginx:latest
 9f4a4258f259d868c81a9c2e922568c6f1ab524cd177368aa31e59832829620



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.

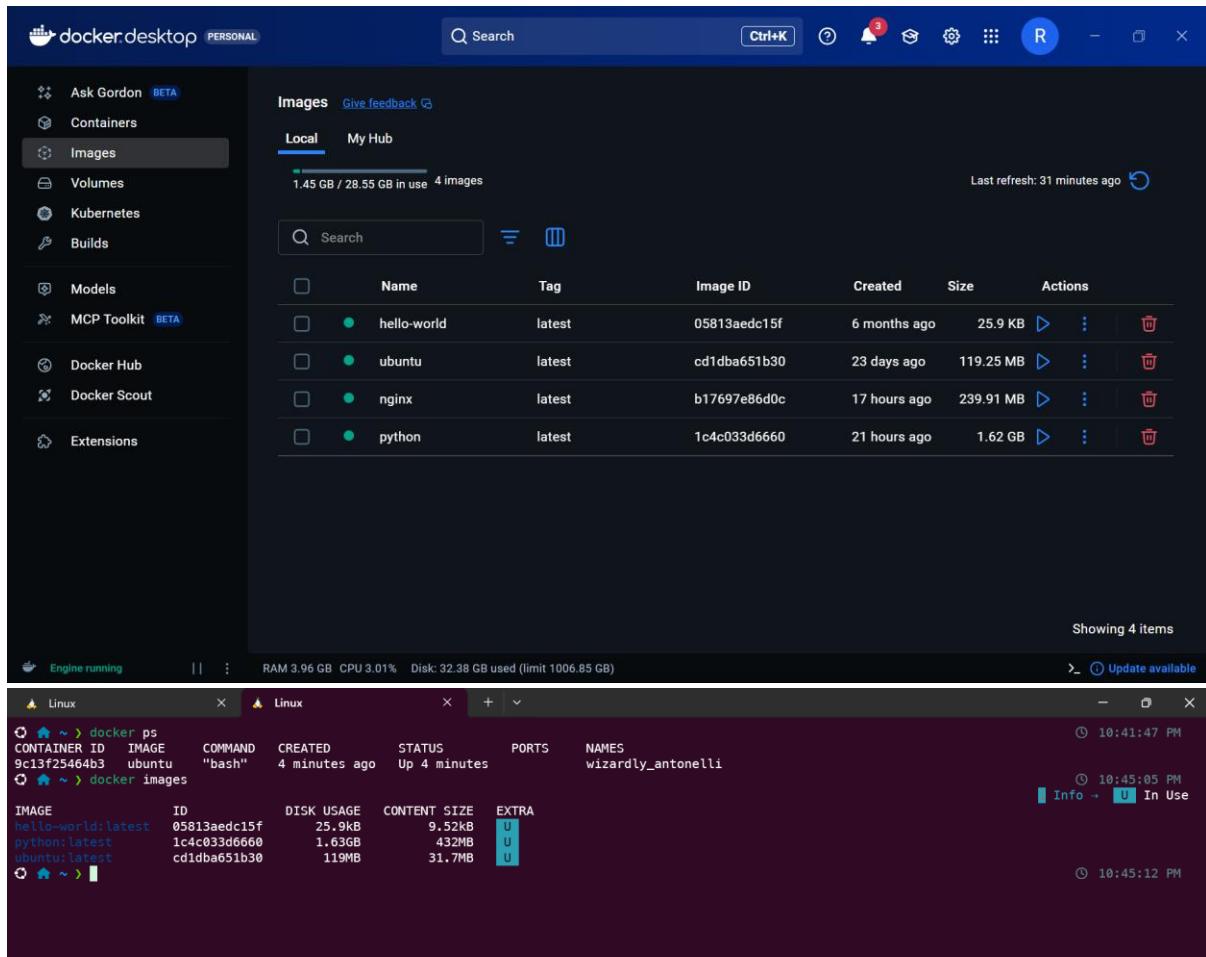
Thank you for using nginx.

Step 4 – View Docker Containers

All containers were listed to verify their status.

`docker ps -a`

This displayed running and stopped containers.



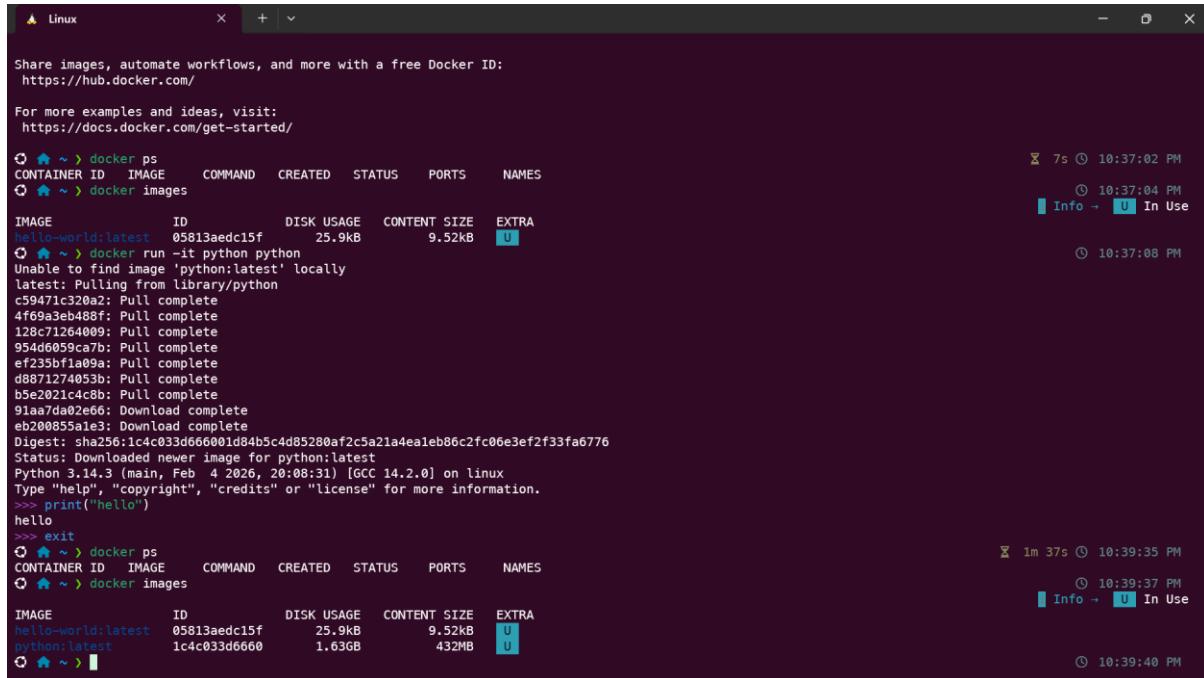
Step 5 – Run Python Container

A Python development container was launched.

`docker run -it python:3.11-slim bash`

`python3`

Python executed successfully inside the container.



The screenshot shows a terminal window titled "Linux" with a dark theme. It displays a sequence of Docker commands and their outputs:

```
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/  
  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
05813aedc15f hello-world:latest 25.9kB 9.52kB U  
  
docker run -it python python  
Unable to find image 'python:latest' locally  
latest: Pulling from library/python  
c59471c320a2: Pull complete  
4f69a3eb488f: Pull complete  
128c71264009: Pull complete  
954d6059ca7b: Pull complete  
ef235bf1a09a: Pull complete  
d8871274053b: Pull complete  
b5e2021c4cbb: Pull complete  
91a7ada02e66: Download complete  
eb200855a1e3: Download complete  
Digest: sha256:1c4c033d666001d84b5c4d85280af2c5a21a4ea1eb86c2fc06e3ef2f33fa6776  
Status: Downloaded newer image for python:latest  
Python 3.14.3 (main, Feb 4 2026, 20:08:31) [GCC 14.2.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print("Hello")  
Hello  
>>> exit  
  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
05813aedc15f hello-world:latest 25.9kB 9.52kB U  
1c4c033d6660 python:latest 1.63GB 432MB U
```

Output

- Docker Engine installed successfully.
- Containers were created and executed without errors.
- Python environment functioned correctly inside Docker.

Result

Docker was successfully installed and configured in WSL. Multiple containers were created and managed, and a Python runtime environment was executed inside a Docker container, demonstrating effective containerization.