

# Assignment-Week 4

## Task 2 - Docker: Installation, Basic Container Operations & Image Building using Dockerfile

### Introduction to Docker

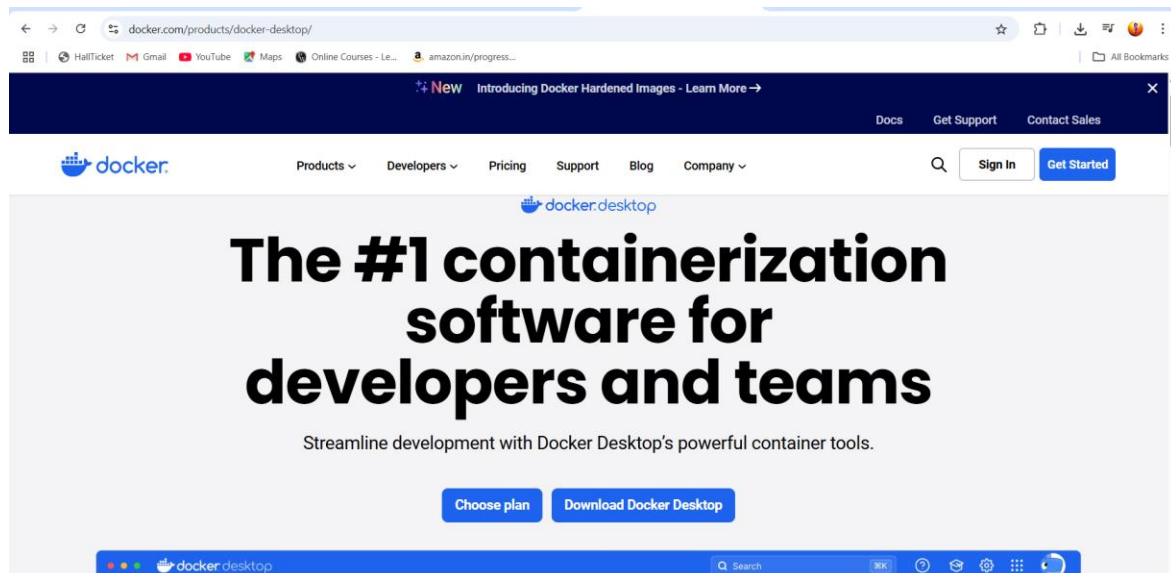
Docker is an open-source platform that simplifies application development and deployment by using containerization. A container is a lightweight, standalone, executable package of software that includes everything needed to run an application—code, runtime, libraries, and system tools.

Unlike traditional virtual machines, containers are more efficient because they share the same OS kernel and consume fewer resources.

### Docker Installation

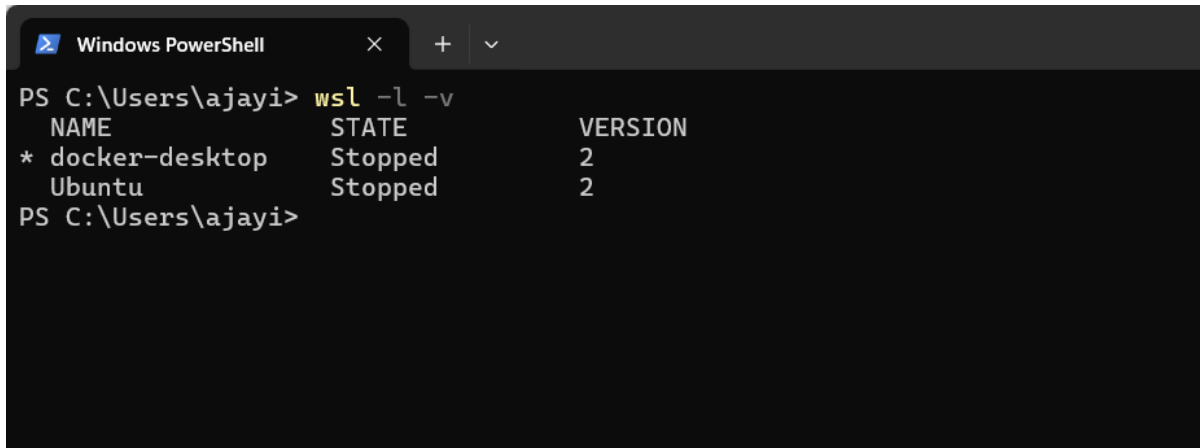
#### On Windows (Using Docker Desktop)

1. Download Docker Desktop from the official website:  
<https://www.docker.com/products/docker-desktop>



2. Run the installer and follow the setup steps.

3. Enable WSL 2 (Windows Subsystem for Linux) when prompted.

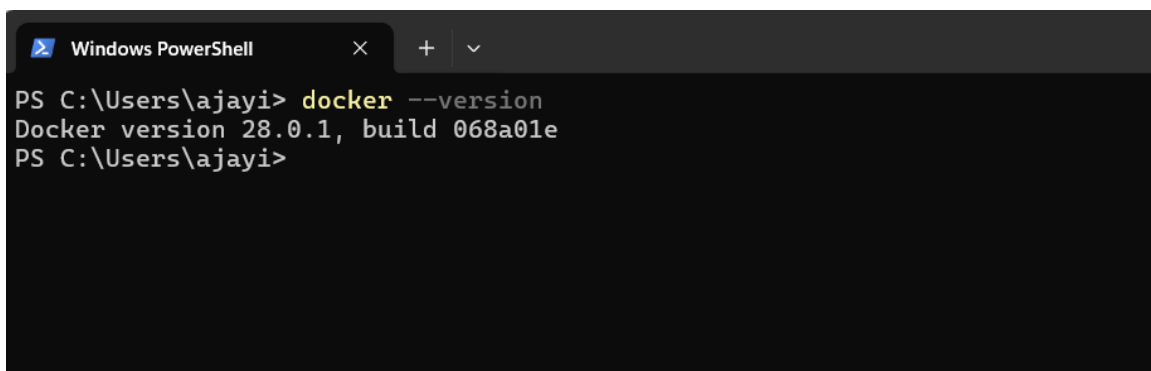


```
Windows PowerShell
PS C:\Users\ajayi> wsl -l -v
  NAME                STATE      VERSION
* docker-desktop      Stopped    2
  Ubuntu              Stopped    2
PS C:\Users\ajayi>
```

4. Restart your system after installation.

5. Open a terminal (Command Prompt or PowerShell) and verify Docker installation:

*\$ docker --version*



```
Windows PowerShell
PS C:\Users\ajayi> docker --version
Docker version 28.0.1, build 068a01e
PS C:\Users\ajayi>
```

## On Linux (Ubuntu)

1. Update your system:

*\$ sudo apt update && sudo apt upgrade*

2. Install Docker:

*\$ sudo apt install docker.io*

3. Enable and start Docker service:

*\$ sudo systemctl enable docker*

*\$ sudo systemctl start docker*

#### 4. Verify installation:

```
$ docker --version
```

## Basic Docker Container Operations

Once Docker is installed, you can start using it to run and manage containers.

### Pulling an Image

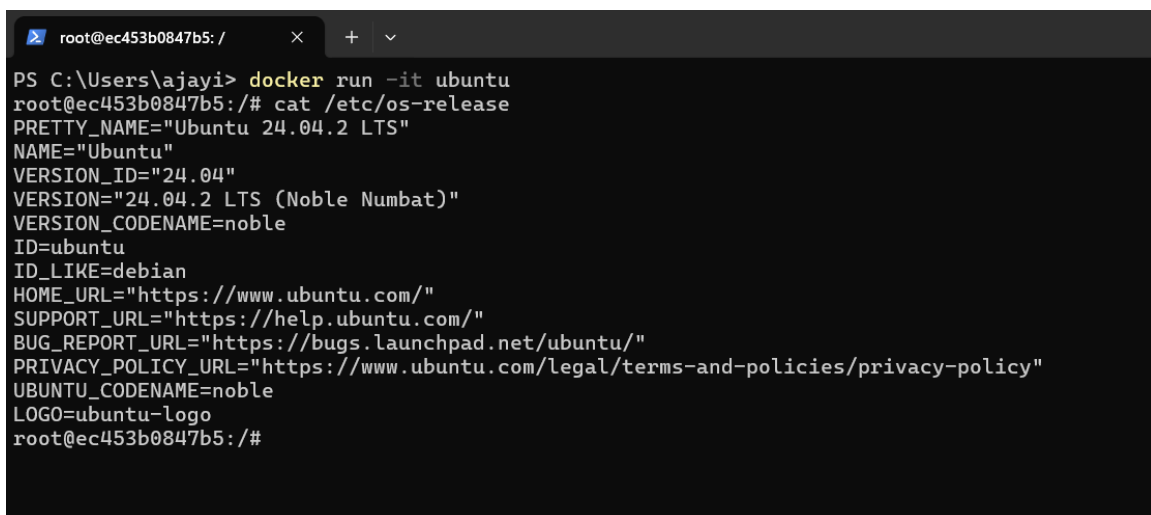
```
$ docker pull ubuntu
```



```
Windows PowerShell
PS C:\Users\ajayi> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
d9d352c11bbd: Already exists
Digest: sha256:b59d21599a2b151e23eea5f6602f4af4d7d31c4e236d22bf0b62b86d2e386b8f
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
PS C:\Users\ajayi>
```

### Running a Container

```
$ docker run -it ubuntu
```



```
root@ec453b0847b5: /
PS C:\Users\ajayi> docker run -it ubuntu
root@ec453b0847b5: /# cat /etc/os-release
PRETTY_NAME="Ubuntu 24.04.2 LTS"
NAME="Ubuntu"
VERSION_ID="24.04"
VERSION="24.04.2 LTS (Noble Numbat)"
VERSION_CODENAME=noble
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=noble
LOGO=ubuntu-logo
root@ec453b0847b5: /#
```

### Listing Running Containers

```
$ docker ps
```

### Listing All Containers

```
$ docker ps -a
```

```
Windows PowerShell
PS C:\Users\ajayi> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS          NAMES
PS C:\Users\ajayi> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS          NAMES
ec453b0847b5   ubuntu    "/bin/bash"             About a minute ago    Exited (0)    16 seconds ago
23b088a31810   ubuntu    "/bin/bash"             4 minutes ago        Exited (0)    About a minute ago
e847a94c43df   ubuntu    "/bin/bash"             4 minutes ago        Exited (0)    4 minutes ago
d1463d4dfc1f   mongo-express "/sbin/tini -- /dock... 46 hours ago        Exited (255)  7 minutes ago    0.0.0.0:8
081->8081/tcp   projects-mongo-express-1
7595d9d4e41e   mongo     "docker-entrypoint.s..." 46 hours ago        Exited (255)  7 minutes ago    0.0.0.0:2
7017->27017/tcp projects-mongodb-1
7d8b88e7dcee   projects-my-app "docker-entrypoint.s..." 46 hours ago        Exited (255)  7 minutes ago    0.0.0.0:3
000->3000/tcp   projects-my-app-1
PS C:\Users\ajayi>
```

## Stopping a Container

```
$ docker stop <container_id>
```

## Removing a Container

```
$ docker rm <container_id>
```

## Removing an Image

```
$ docker rmi <image_name>
```

## Running a Web App

```
$ docker run -d -p 8080:80 nginx
```

- This runs the Nginx web server in a container and maps port 8080 on the host to port 80 in the container.

## Creating and Building Docker Images Using a Dockerfile

A Dockerfile is a script with instructions to build a custom Docker image.

### Sample Dockerfile

```
FROM ubuntu:20.04
```

```
LABEL maintainer="ajayingle3878@gmail.com"
```

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

```
CMD ["echo", "Hello from my first Docker image!"]
```

```
~
FROM ubuntu:20.04

LABEL maintainer="ajaying1e3878@gmail.com"

RUN apt update && apt install -y curl

CMD ["echo", "Hello from my first Docker image!"]
```

```
~
~
~
~
~
~
~
~
~
~
~
~
```

Dockerfile [unix] (14:25 27/06/2025)

8,0-1 All

## Steps to Build an Image

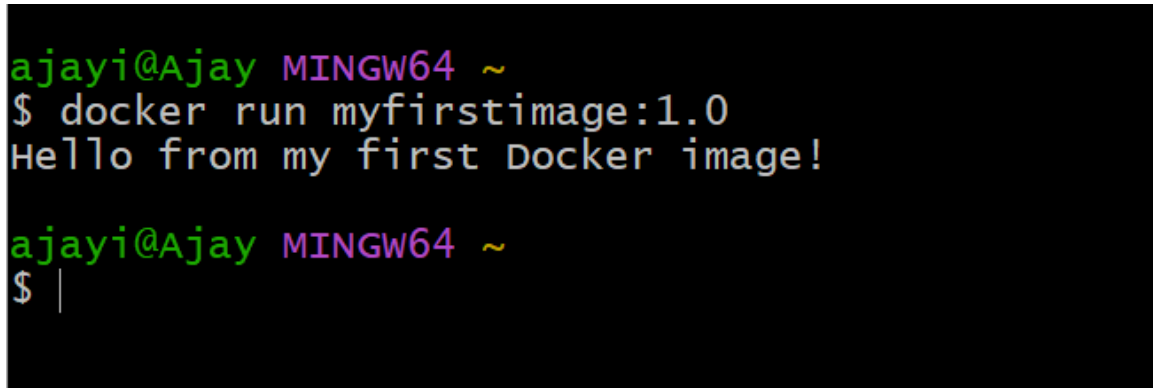
1. Create a file named Dockerfile (without any extension).
2. Place it in a working directory.
3. Build the image:

```
$ docker build -t myfirstimage:1.0 .
```

```
ajay1@Ajay MINGW64 ~
$ docker build -t myfirstimage:1.0 .
[+] Building 52.0s (7/7) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 193B                             0.0s
=> [internal] load metadata for docker.io/library/ubuntu:20.04  5.1s
=> [auth] library/ubuntu:pull token for registry-1.docker.io    0.0s
=> [internal] load .dockerignore                                 0.0s
=> => transferring context: 2B                                    0.0s
=> [1/2] FROM docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8f 10.8s
=> => resolve docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8f 0.0s
=> => sha256:8feb4d8ca5354def3d8fce243717141ce31e2c42870 6.69kB / 6.69kB 0.0s
=> => sha256:c664f8f86ed5a386b0a340d981b8f81714e21a8b9c73f65 424B / 424B 0.0s
=> => sha256:b7bab04fd9aa0c771e5720bf0cc7cbf993fd6946645 2.30kB / 2.30kB 0.0s
=> => sha256:13b7e930469f6d3575a320709035c6acf6f5485a7 27.51MB / 27.51MB 8.8s
=> => extracting sha256:13b7e930469f6d3575a320709035c6acf6f5485a76abcf03 1.7s
=> [2/2] RUN apt update && apt install -y curl                  35.8s
=> exporting to image                                           0.3s
=> => exporting layers                                           0.2s
=> => writing image sha256:1170a9fbcbec1b03363285f39fec82b5c76e4e9bf350b 0.0s
=> => naming to docker.io/library/myfirstimage:1.0             0.0s
```

#### 4. Run the image:

```
$ docker run myfirstimage:1.0
```

A terminal window with a black background and colorful text. The prompt is 'ajayi@Ajay MINGW64 ~'. The command '\$ docker run myfirstimage:1.0' is entered, and the output 'Hello from my first Docker image!' is displayed. The prompt is then shown again with a cursor after the dollar sign.

```
ajayi@Ajay MINGW64 ~  
$ docker run myfirstimage:1.0  
Hello from my first Docker image!  
  
ajayi@Ajay MINGW64 ~  
$ |
```

#### Explanation

- FROM: Base image
- RUN: Execute commands during image build
- CMD: Default command to run when container starts
- -t: Tags the image with a name and version

#### Benefits of Docker

- Lightweight and fast
- Portable across environments
- Simplifies CI/CD
- Isolates applications
- Works with microservices architecture

#### Conclusion

Docker has revolutionized the way we build, ship, and run applications. With simple installation steps, powerful container commands, and the ability to create custom images using Dockerfiles, it is an essential tool in modern DevOps and development pipelines.

By mastering Docker basics, you can efficiently manage application environments, reduce dependency conflicts, and enhance deployment consistency across different stages of software development.