## Experiment No. 08

Aim: To understand Docker Architecture and Container Life Cycle, install Docker and execute docker commands to manage images and interact with containers.

Theory: Docker is a containerization technology that allows developers to package their applications and dependencies into lightweight, portable containers. These containers can then be run on any system that supports Docker, making it easier to deploy and manage applications in various environments. Docker Architecture:
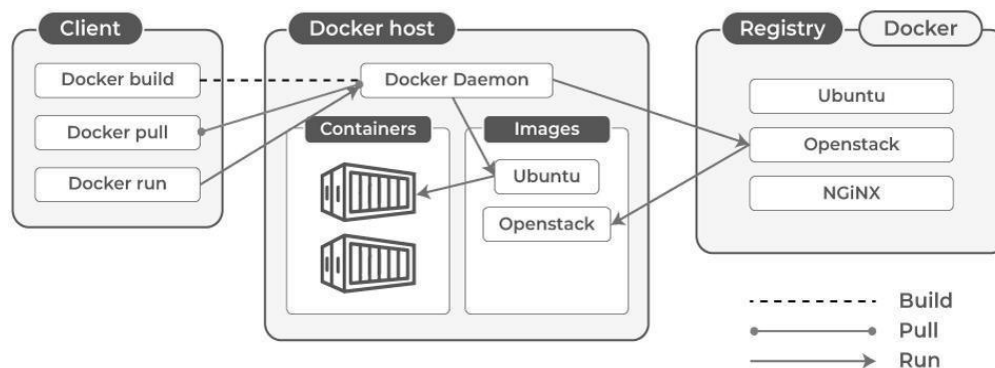
The Docker architecture consists of the following components:

Docker Client: This is the primary interface that developers use to interact with Docker. It sends commands to the Docker daemon and displays the output.

Docker Daemon: This is the background process that runs on the host machine and manages Docker objects such as images, containers, networks, and volumes.

Docker Registry: This is a storage and distribution system for Docker images. It allows developers to share and distribute their images with others.

Docker Image: This is a lightweight, standalone, executable package that includes everything needed to run an application, including the application code, runtime, libraries, dependencies, and system tools.

Docker Container: This is a runtime instance of a Docker image. It is isolated from the host system and other containers, providing a secure and predictable environment for the application to run in.

Container LifeCycle:
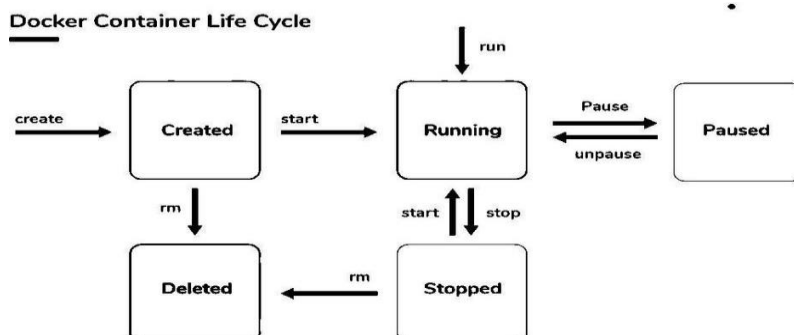
The lifecycle of a Docker container consists of the following stages:

Create: To create a container, you start by creating an image that includes all the necessary components to run the application. This image is then used to create the container.

Start: Once the container is created, you can start it using the docker start command. This launches the container and runs the application inside it.

Run: Once the container is started, you can interact with the application inside the container. You can run commands, access files, and make changes to the application.

Stop: When you're done using the container, you can stop it using the docker stop command. This stops the application inside the container and shuts down the container.

Remove: Finally, when you no longer need the container, you can remove it using the docker rm command. This removes the container from the system and frees up any resources it was using.
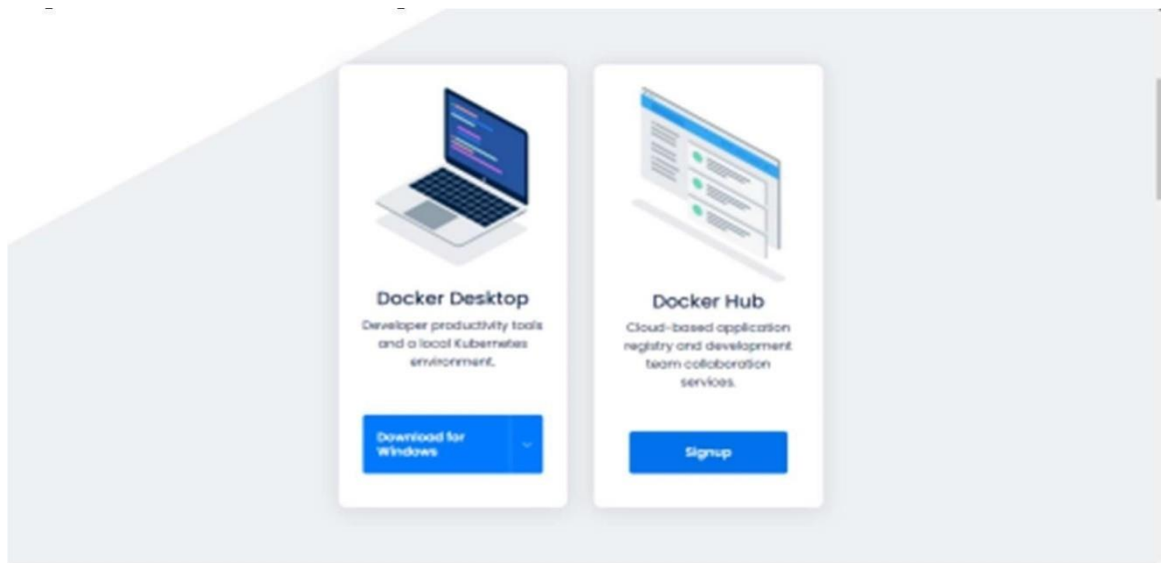
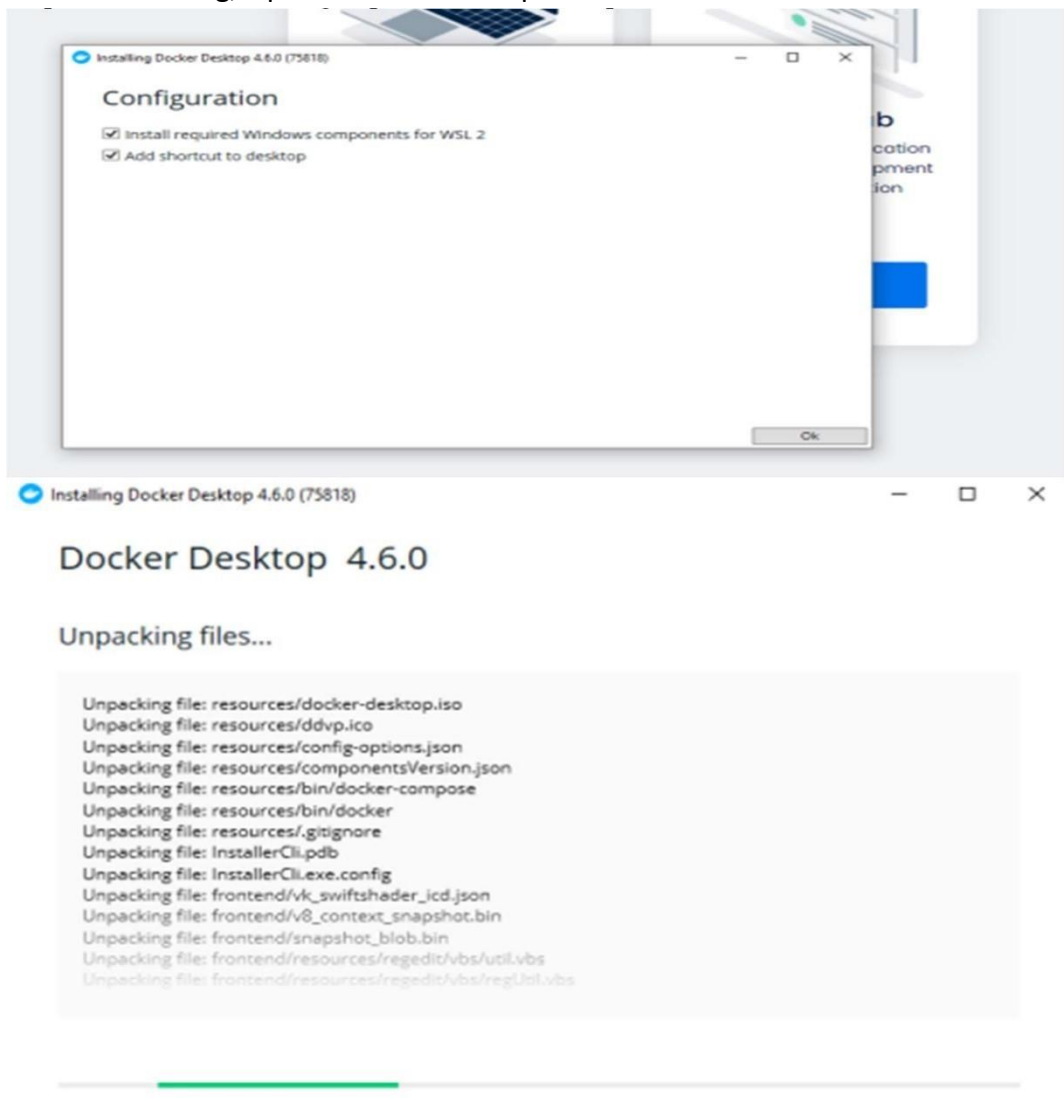**Steps for Installation:**

Step 01: Open docker.com

Scroll down, Click on 'Get started for free' tab.
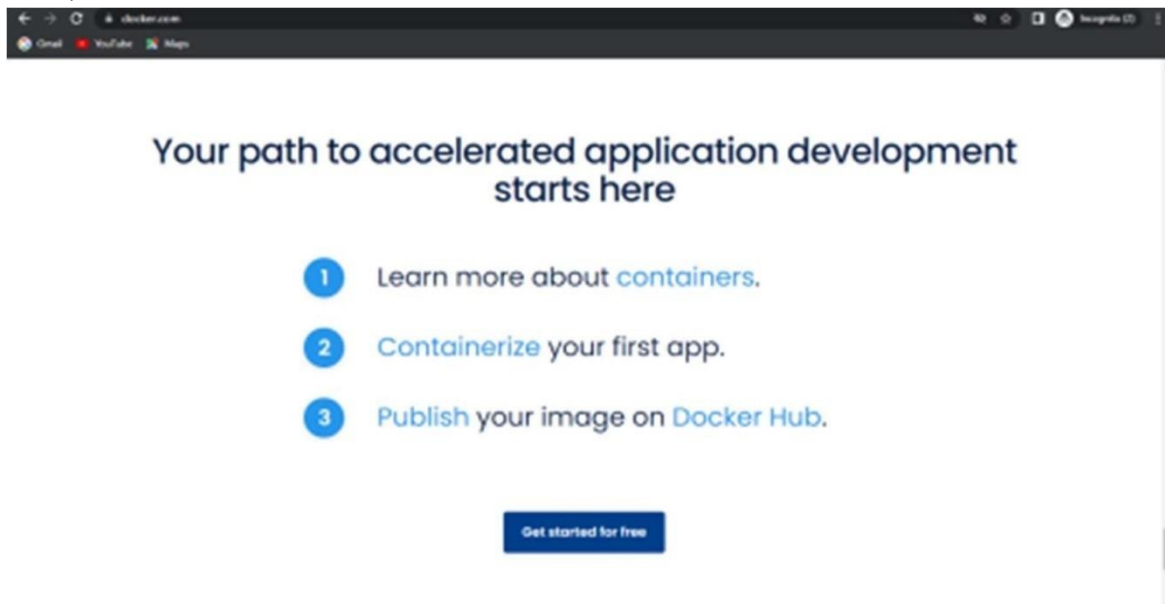
Step 02: Click on Docker Desktop, Download it.

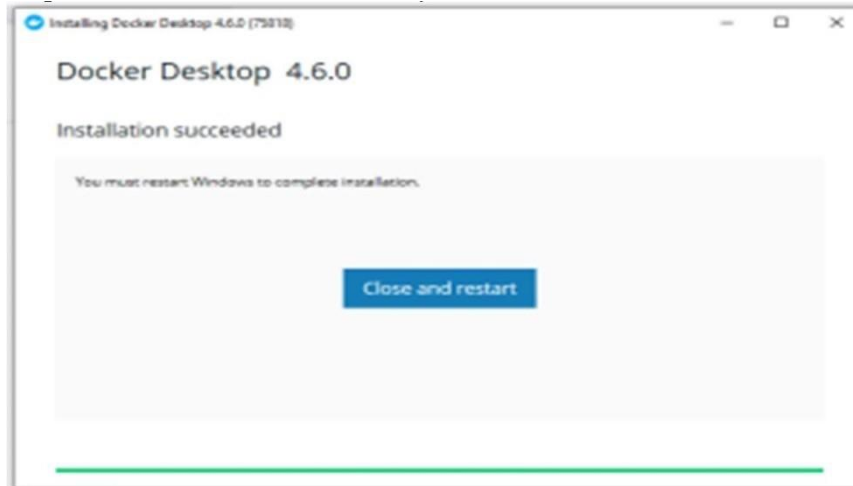Step 03: After downloading, Open 'Docker Desktop Installer' & start installation.
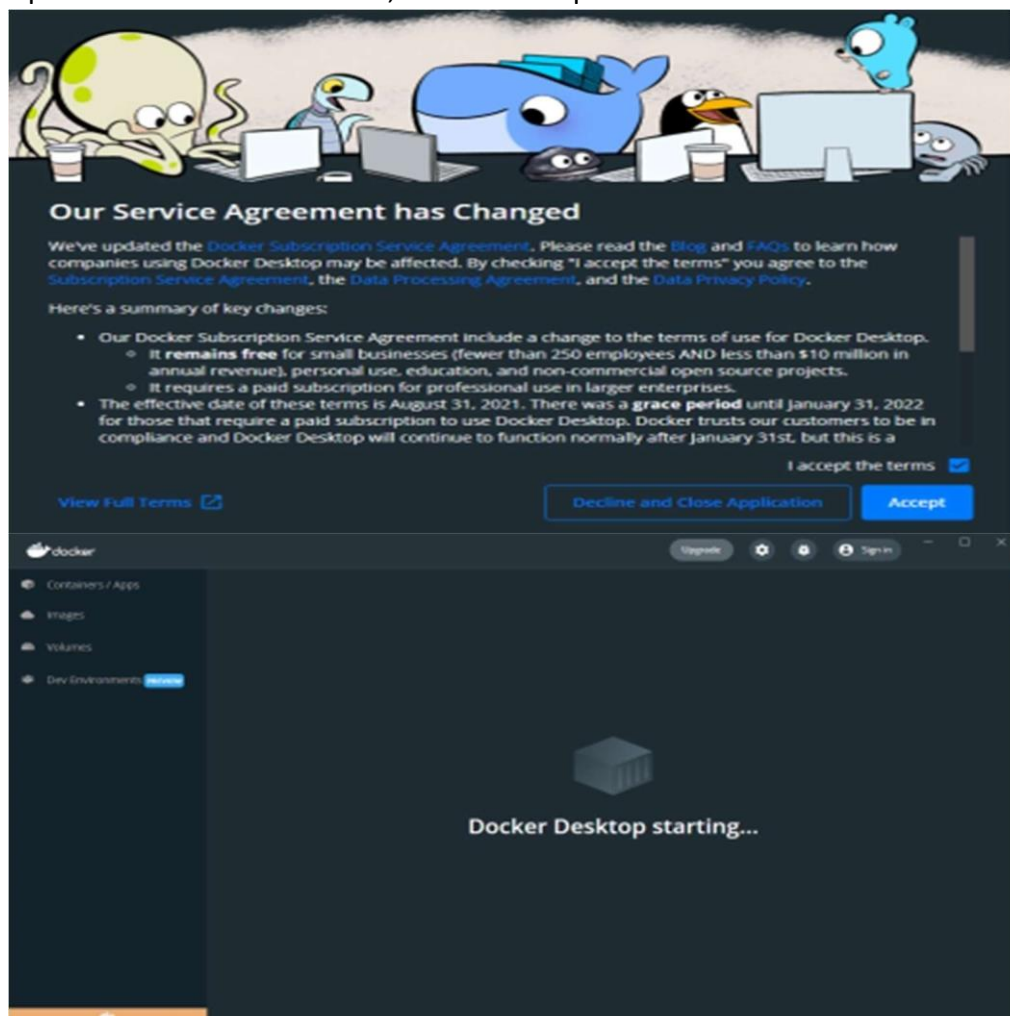
Step 04: After Installation, Restart your device.

Step 05: Accept the terms and conditions, Click on Accept.

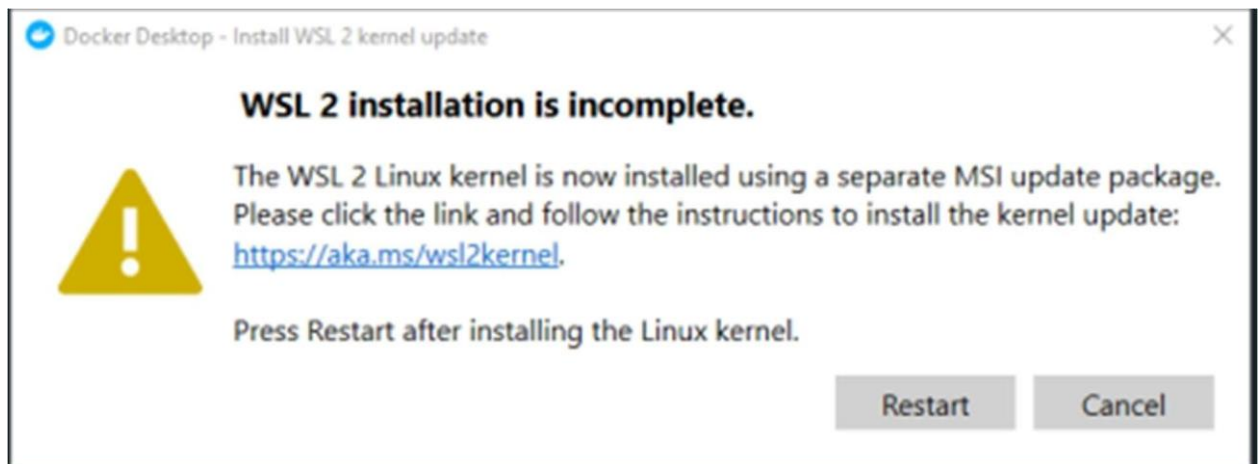The following window should pop up. Click
on the link - https://aka.ms/wsl2kernel.
(Do not close this window).

Download the WSL2 Linux kernel update package for x64 machines.
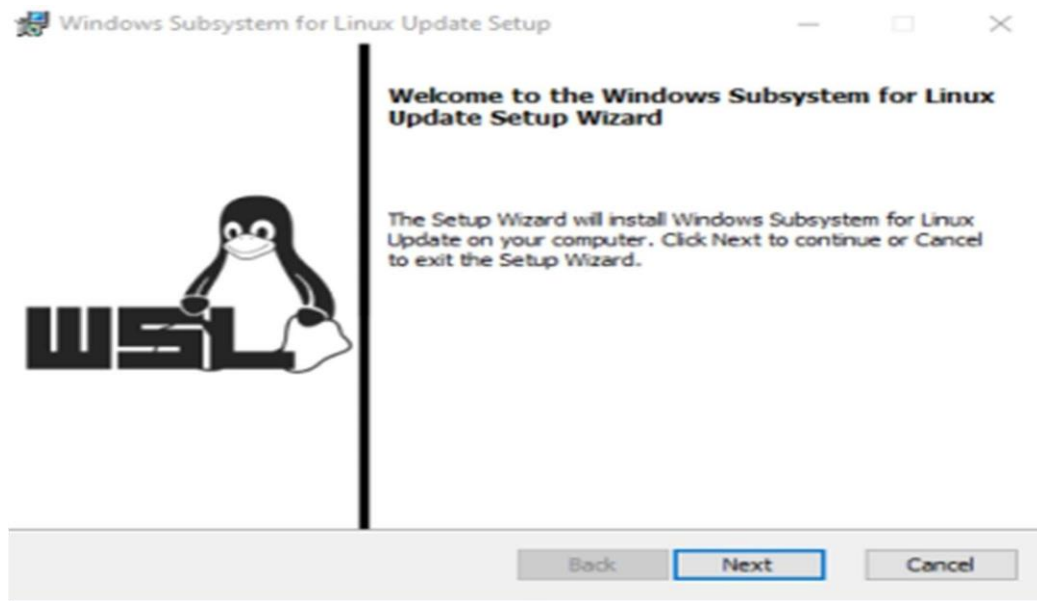

After Download is complete, Run the .msipackage.
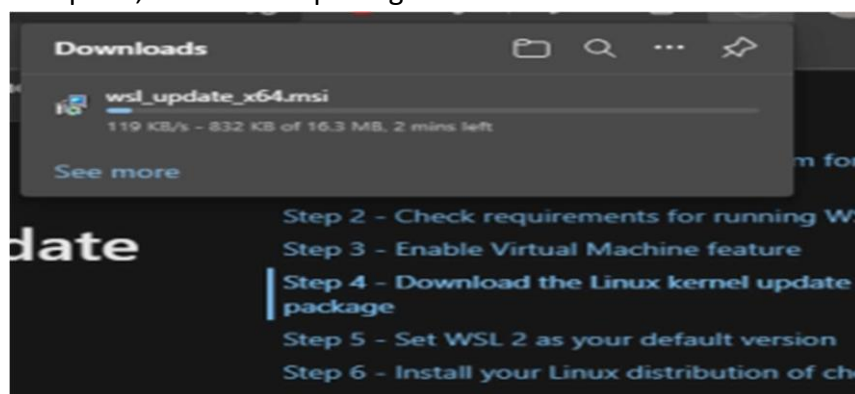
Click on next.

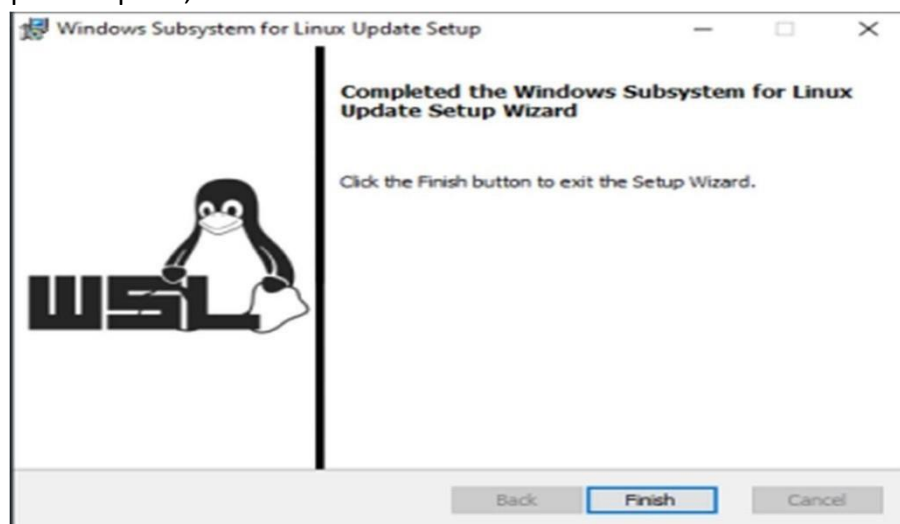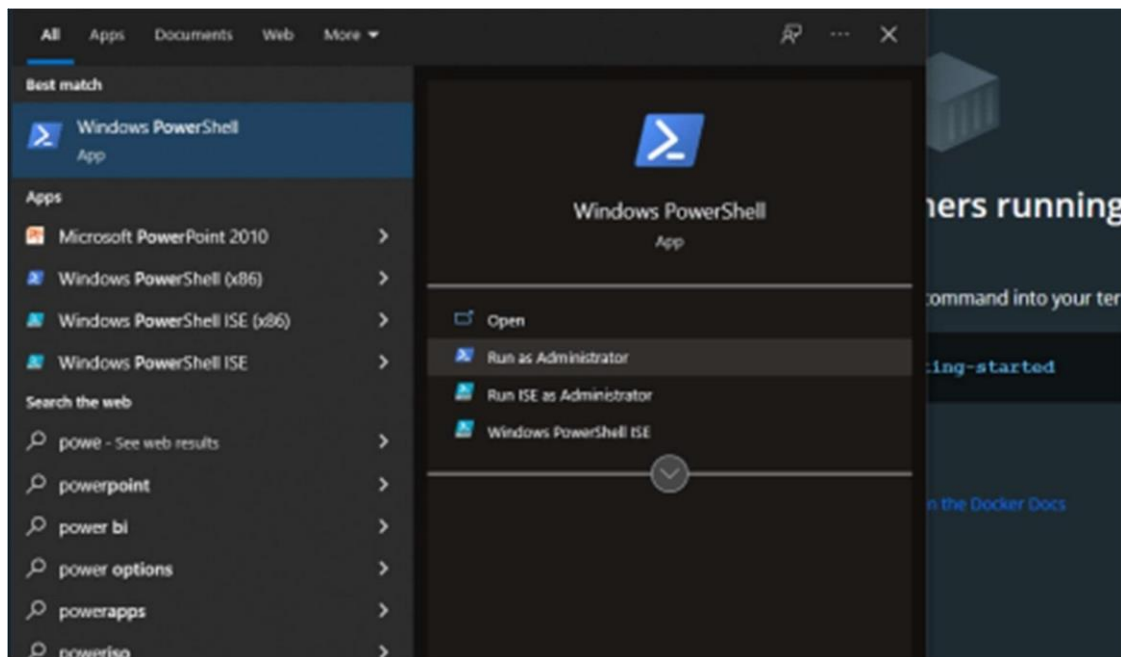After, the setup is complete, Click on finish.

Open Powershell as an Administrator.

Run the following Command:

wsl --setdefault-version 2

Now, Click on Restart
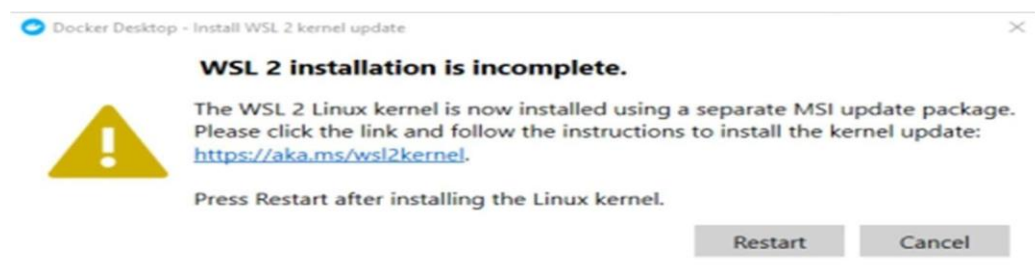
Docker should now restart. Click
on Start.
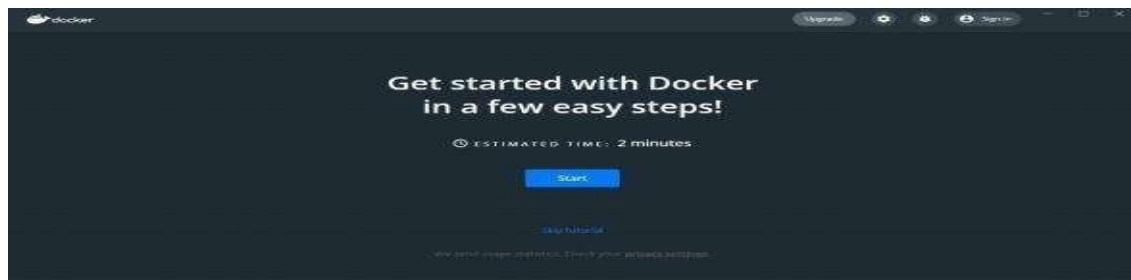
The following window should pop up.
This means, Installation is now
complete.

Output:

Open Command Prompt, run the following commands:

- To check the version of
Docker: docker --version
- To install image of
ubuntu docker pull
ubuntu


```
PS C:\Users\Lenovo> docker --version
Docker version 20.10.13, build a224086
PS C:\Users\Lenovo> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
4d32b49e2995: Pull complete
Digest: sha256:bea6d19168bbfd6af8d77c2cc3c572114eb5d113e6f422573c93cb605a0e2ffb
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
PS C:\Users\Lenovo>
```

Check downloaded
images, docker images

```
PS C:\Users\Lenovo> docker images
REPOSITORY    TAG       IMAGE ID       CREATED      SIZE
ubuntu        latest    ff0fea8310f3   4 days ago   72.8MB
PS C:\Users\Lenovo>
```

- Run ubuntu OS

docker run -it ubuntu /bin/bash

```
PS C:\Users\Lenovo> docker run -it ubuntu /bin/bash
root@f45775828da6:/#
```
-

docker ps docker

container ls –a

docker container rm b71e3e6b1118 //copy docker id for remove but first (Use your container ID in the above command) stop your docker

1) docker container stop b71e3e6b1118
2) docker container rm b71e3e6b1118
3) docker ps
4) docker //list all docker commands
5) docker images
6) docker image rm ff0fea8310f3 // copy image id from previous output (Use your image ID in the above command)
7) docker run -it ubuntu /bin/bash //check output

```
PS C:\Users\Lenovo> docker ps
CONTAINER ID   IMAGE    COMMAND       CREATED           STATUS             PORTS    NAMES
f45775828da6   ubuntu   "/bin/bash"   About a minute ago   Up About a minute        nostalgic_elion
PS C:\Users\Lenovo> docker container ls -a
CONTAINER ID   IMAGE    COMMAND       CREATED           STATUS             PORTS    NAMES
f45775828da6   ubuntu   "/bin/bash"   About a minute ago   Up About a minute        nostalgic_elion
PS C:\Users\Lenovo> docker container rm f45775828da6
Error response from daemon: You cannot remove a running container f45775828da6297e793470cd07835cf764532a3d5eded8e4094ffc
bc0f687858. Stop the container before attempting removal or force remove
PS C:\Users\Lenovo> docker container stop f45775828da6
f45775828da6
PS C:\Users\Lenovo> docker container rm f45775828da6
f45775828da6
PS C:\Users\Lenovo> docker ps
CONTAINER ID   IMAGE    COMMAND    CREATED    STATUS    PORTS    NAMES
PS C:\Users\Lenovo> docker images
REPOSITORY    TAG       IMAGE ID       CREATED      SIZE
ubuntu        latest    ff0fea8310f3   4 days ago   72.8MB
PS C:\Users\Lenovo> docker image rm f45775828da6
Error: No such image: f45775828da6
PS C:\Users\Lenovo> docker image rm ff0fea8310f3
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:bea6d19168bbfd6af8d77c2cc3c572114eb5d113e6f422573c93cb605a0e2ffb
Deleted: sha256:ff0fea8310f3957d9b1e6ba494f3e4b63cb348c76160c6c15578e65995ffaa87
Deleted: sha256:867d0767a47c392f80acb51572851923d6d3e55289828b0cd84a96ba342660c7
PS C:\Users\Lenovo> docker images
REPOSITORY    TAG       IMAGE ID   CREATED   SIZE
PS C:\Users\Lenovo>
```

Conclusion:

Q1. What is the difference between containerization and virtualization?

- Containerization involves packaging an application and its dependencies into a container that can run on any system, without the need for a separate operating system. It provides lightweight isolation and shares the host system's kernel.
- Virtualization, on the other hand, involves creating virtual machines (VMs) that run a separate operating system on top of a hypervisor. Each VM requires its own operating system, resulting in higher resource overhead compared to containers.

Q2. What is Docker Daemon?

The Docker Daemon is a background process that runs on the host machine and manages Docker objects such as images, containers, networks, and volumes. It listens for Docker API requests from the Docker Client and performs the requested actions, such as pulling images, creating containers, and starting/stopping containers.