

Aim -

To Study and implement containerization using DOCKER.

Theory -

Docker is a platform and tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all parts it needs, such as libraries and other dependencies, and ship it all out as one package. This ensures that the application will run on any other Linux machine regardless of any customized settings that machine might have that could differ from the machine used for writing and testing the code.

Advantages of Docker:

- Portability: Docker containers encapsulate the application and all its dependencies, making it highly portable across different environments. Developers can build an application once and run it anywhere, whether it's on a developer's laptop, on-premises servers, or in the cloud.
- Consistency: Docker ensures consistency between development, testing, and production environments. Since containers run consistently across different environments, there are fewer chances of "it works on my machine" issues, leading to more reliable deployments.
- Isolation: Docker containers provide lightweight process isolation, ensuring that applications running within containers do not interfere with each other. Each container runs as an isolated process, with its own filesystem, networking, and resources, providing better security and stability.
- Resource Efficiency: Docker containers share the host operating system's kernel, which makes them lightweight and highly efficient in terms of resource utilization. Multiple containers can run on the same host without the overhead of multiple virtual machines, leading to better resource utilization and cost savings.
- Scalability: Docker containers can be easily scaled up or down based on demand. Docker's orchestration tools, such as Docker Swarm and Kubernetes, enable automated container management, deployment, and scaling, making it easy to scale applications horizontally to handle increased workload.

Disadvantages of Docker:

- Learning Curve: Docker has a learning curve, especially for users who are new to containerization and container orchestration concepts. Users need to learn Docker's commands, Dockerfile syntax, container networking, and orchestration tools to effectively use Docker in production environments.
- Complex Networking: Docker's networking can be complex, especially in multi-container or multi-host environments. Users need to understand Docker's

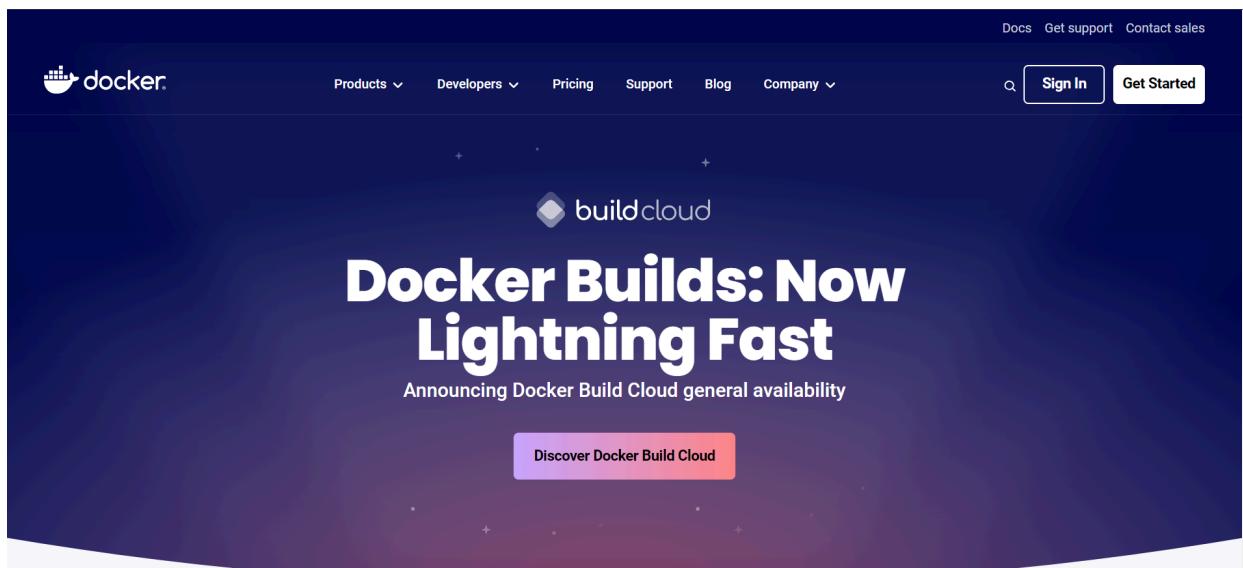
networking modes, bridge networks, overlay networks, and service discovery mechanisms to properly configure networking for containerized applications.

- Security Concerns: Docker containers share the same kernel as the host operating system, which raises security concerns about container escape vulnerabilities. While Docker provides built-in security features such as namespaces and control groups, users need to implement additional security best practices to secure their containerized applications.
- Persistent Storage: Docker containers are designed to be ephemeral by default, meaning that any data written to the container's file system is lost when the container is stopped or deleted. Users need to implement solutions such as Docker volumes or external storage drivers to persist data across container restarts and deployments.
- Tooling Ecosystem: Docker's tooling ecosystem is extensive but fragmented, with multiple tools and libraries available for container management, orchestration, monitoring, and logging. Users need to evaluate and choose the right set of tools to build a comprehensive Docker ecosystem that meets their specific requirements.

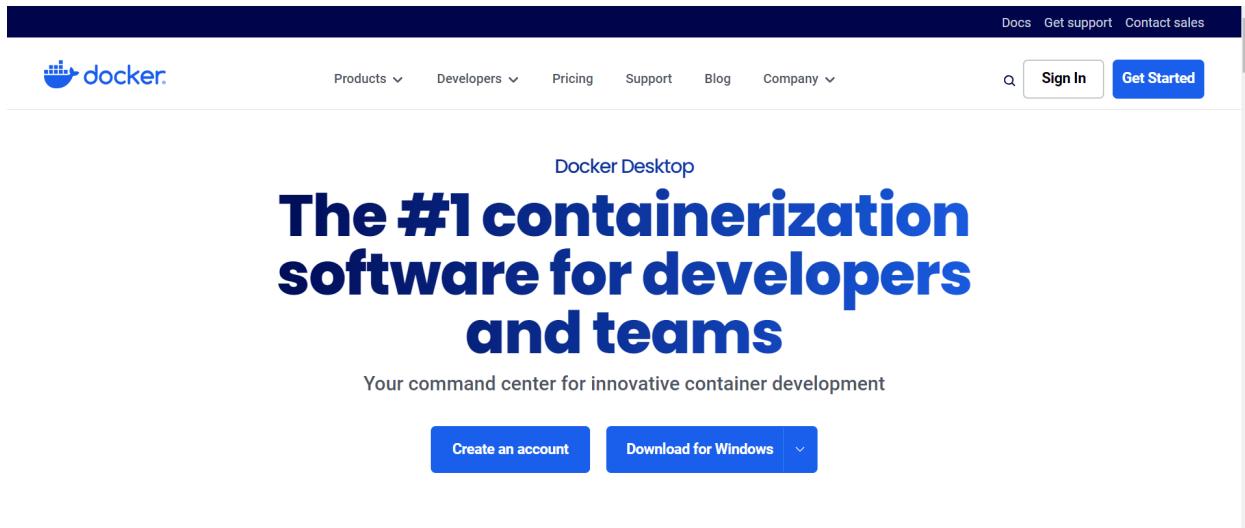
Procedure-

Step 01: Open docker.com

Scroll down, Click on 'Get started for free' tab.

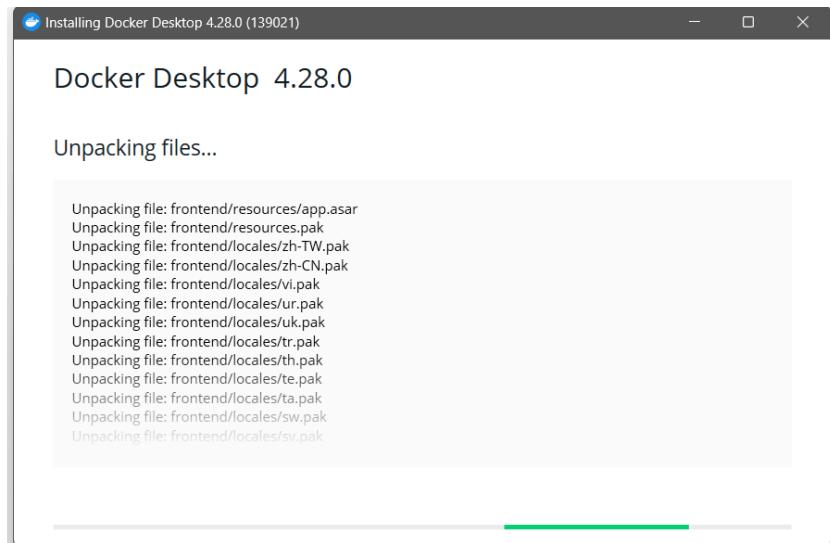
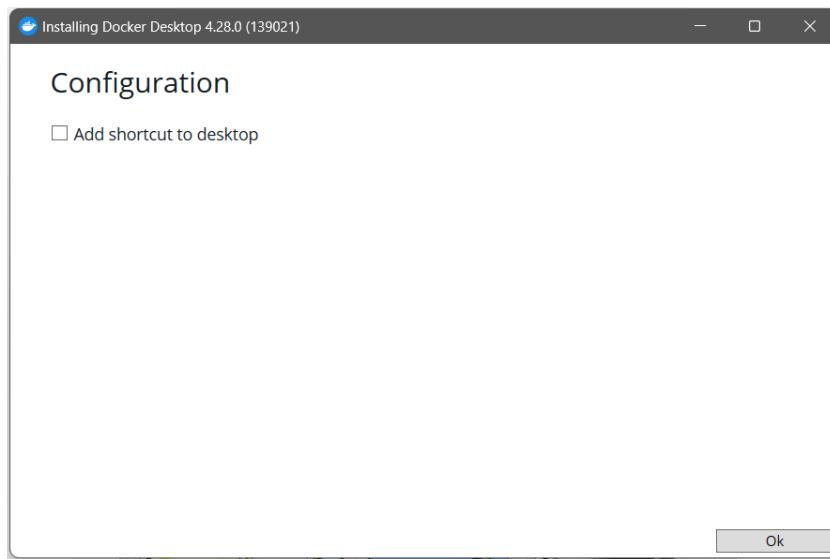


Step 02: Click on Docker Desktop, Download it.

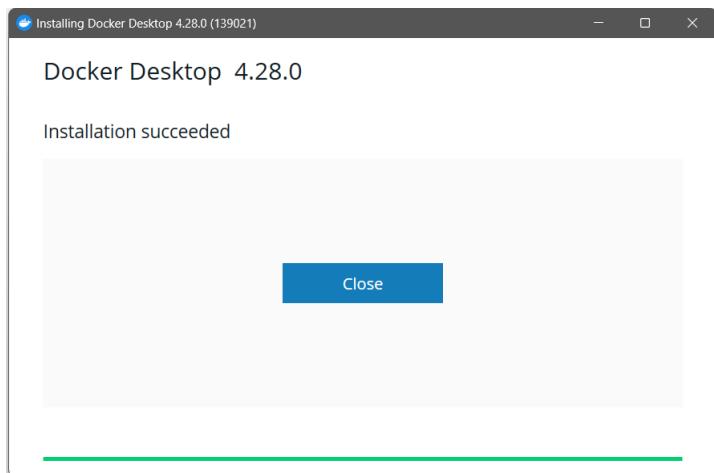


The screenshot shows the Docker Desktop landing page. At the top, there's a dark header with the Docker logo, navigation links for 'Products', 'Developers', 'Pricing', 'Support', 'Blog', and 'Company', and buttons for 'Sign In' and 'Get Started'. Below the header, the text 'Docker Desktop' is followed by a large, bold headline: 'The #1 containerization software for developers and teams'. A subtext below it reads 'Your command center for innovative container development'. At the bottom of the main section are two buttons: 'Create an account' and 'Download for Windows'.

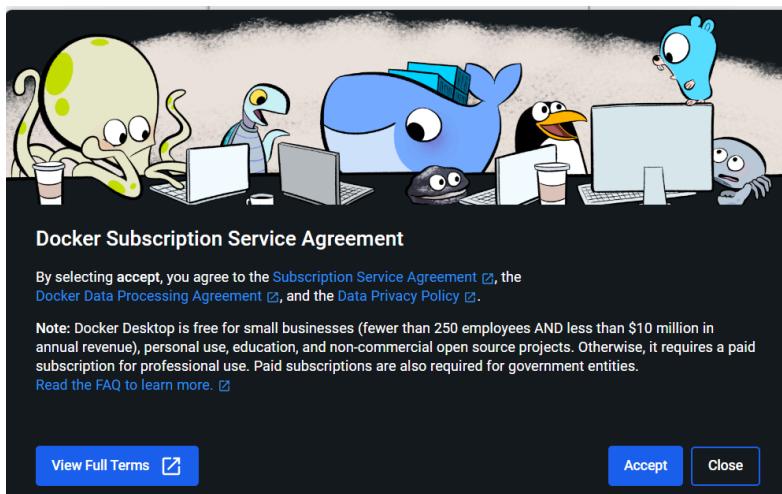
Step 03: After downloading, Open 'Docker Desktop Installer' & start installation.



Step 04: After Installation, Restart your device.



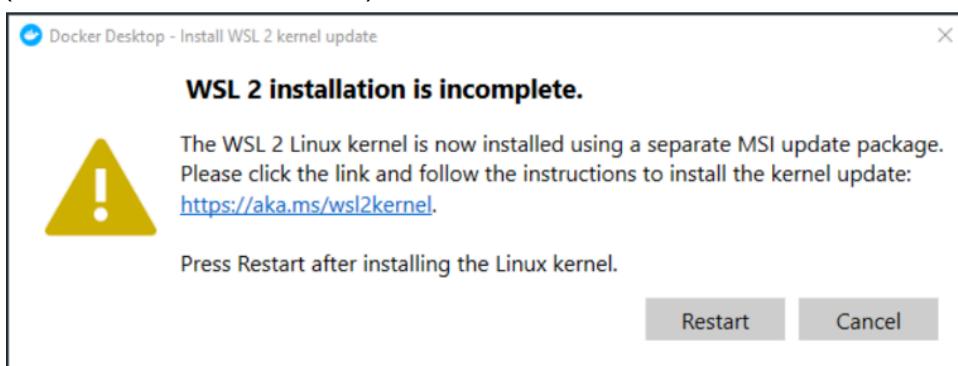
Step 05: Accept the terms and conditions, Click on Accept.



The following window should pop up.

Click on the link - <https://aka.ms/wsl2kernel>.

(Do not close this window).

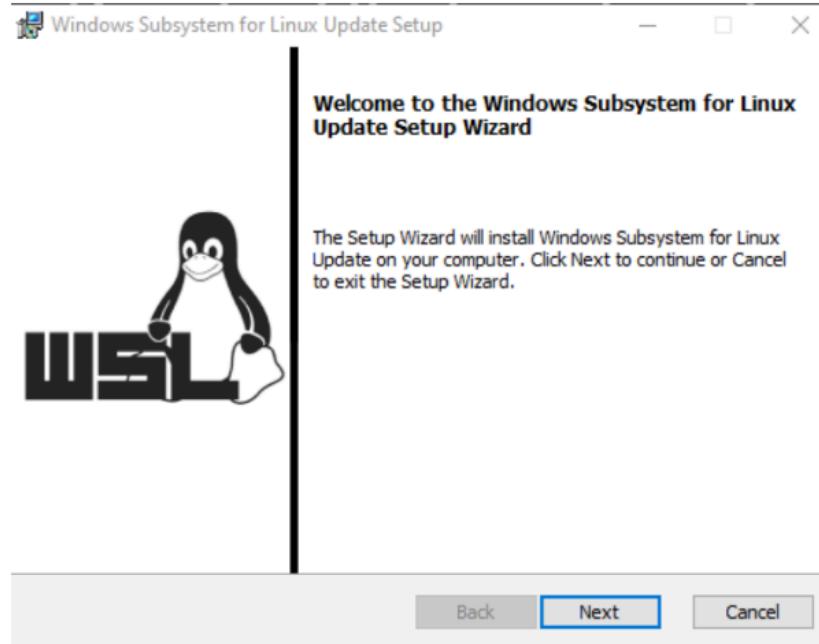


Download the WSL2 Linux kernel update package for x64 machines.

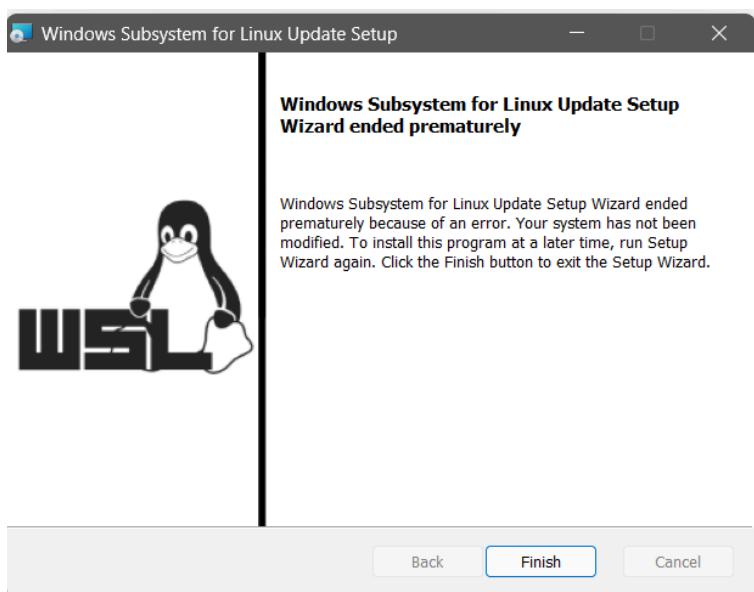
The screenshot shows a Microsoft Docs page titled "Step 4 - Download the Linux kernel update package". The page includes a sidebar with links like "WSL Documentation", "Overview", "Install", "Install WSL", "Manual install steps for older versions", "Install on Windows Server", and "Tutorials". The main content area describes the Linux kernel update package and provides a link to download it. A sidebar on the right promotes AI disruption and training modules.

After Download is complete, run the .msi package.

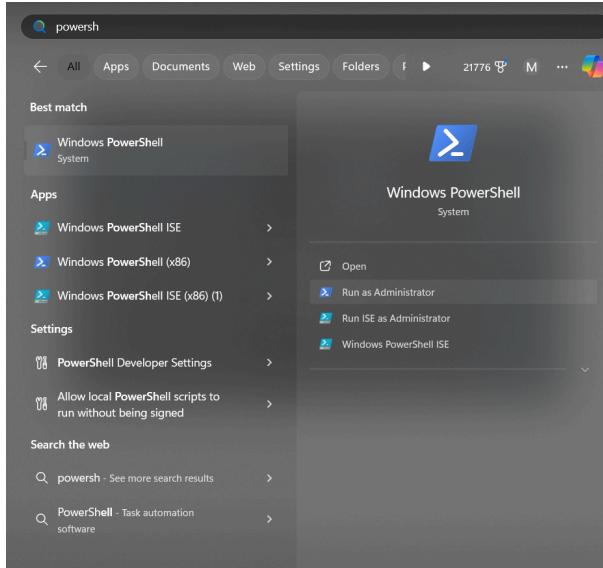
Click on next.



After the setup is complete, Click on finish.



Open Powershell as an Administrator.



Run the following Command:

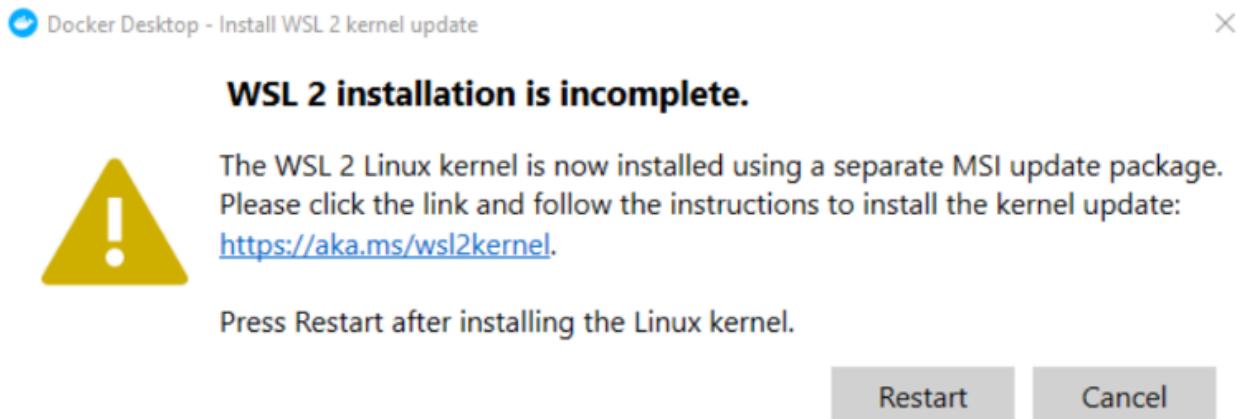
```
wsl --set-default-version 2
```

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> wsl --set-default-version 2
For information on key differences with WSL 2 please visit https://aka.ms/wsl2
The operation completed successfully.
PS C:\WINDOWS\system32>
```

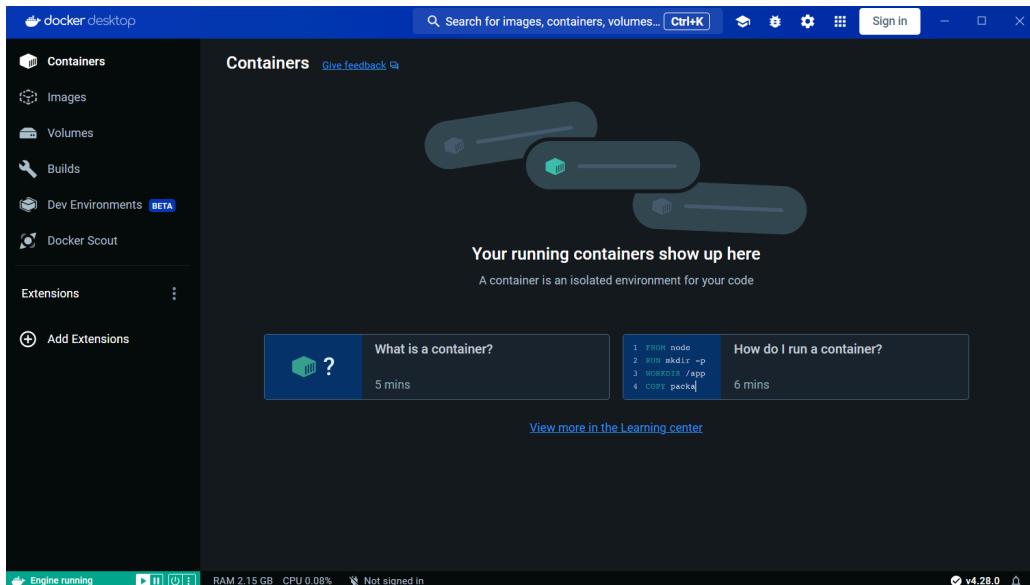
Now, Click on Restart.



Docker should now restart.

The following window should pop up.

This means, Installation is now complete.



Open Command Prompt, run the following commands:

1) To check the version of Docker:

docker --version

2) To install image of ubuntu

docker pull ubuntu

```
PS C:\WINDOWS\system32> docker --version
Docker version 25.0.3, build 4debf41
PS C:\WINDOWS\system32> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
bccd10f490ab: Pull complete
Digest: sha256:77906da86b60585ce12215807090eb327e7386c8fafb5402369e421f44eff17e
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview ubuntu
PS C:\WINDOWS\system32>
```

3) Check downloaded images,

docker images

```
View a summary of image vulnerabilities and recommendations → docker scout quickview ubuntu
PS C:\WINDOWS\system32> docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
ubuntu          latest        ca2b0f26964c   2 weeks ago   77.9MB
PS C:\WINDOWS\system32> -
```

4) Run ubuntu OS

docker run -it ubuntu /bin/bash

```
PS C:\WINDOWS\system32> docker run -it ubuntu /bin/bash
root@03d162e4b547:/# -
```

5) Open another Command Prompt and follow the steps shown below.

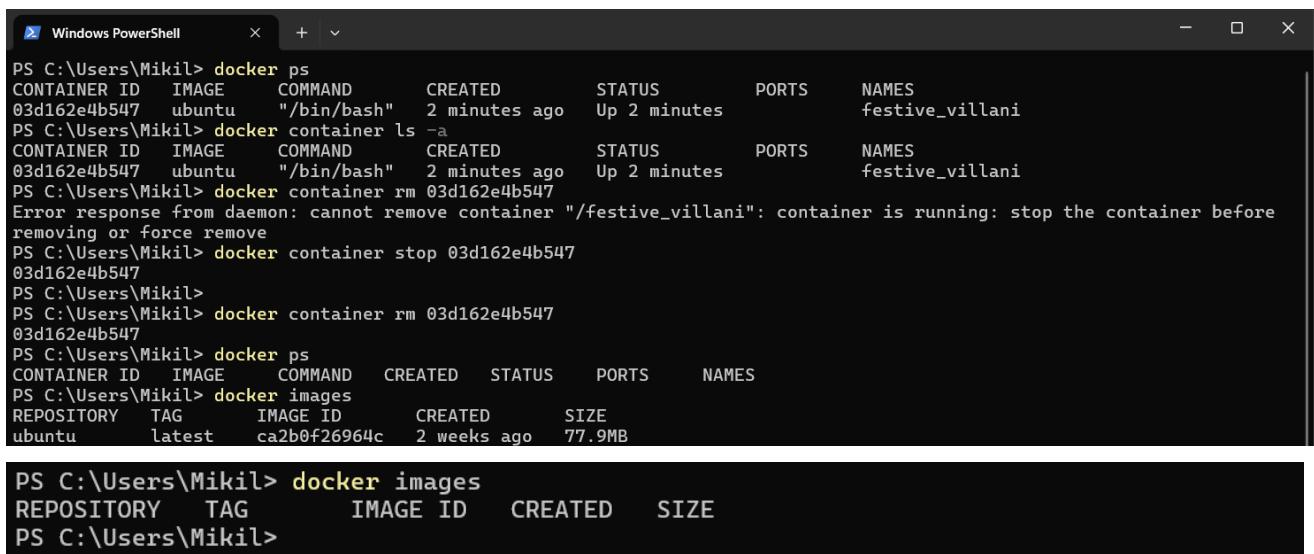
-docker ps

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

- docker container stop 03d162e4b547
- docker container rm 03d162e4b547
- docker ps
- docker //list all docker commands
- docker images
- docker image rm ca2b0f276964c // copy image id from previous output (Use your image ID in the above command)
- docker run -it ubuntu /bin/bash //check output



```
PS C:\Users\Mikil> docker ps
CONTAINER ID   IMAGE     COMMAND      CREATED     STATUS      PORTS     NAMES
03d162e4b547   ubuntu    "/bin/bash"  2 minutes ago  Up 2 minutes          festive_villani
PS C:\Users\Mikil> docker container ls -a
CONTAINER ID   IMAGE     COMMAND      CREATED     STATUS      PORTS     NAMES
03d162e4b547   ubuntu    "/bin/bash"  2 minutes ago  Up 2 minutes          festive_villani
PS C:\Users\Mikil> docker container rm 03d162e4b547
Error response from daemon: cannot remove container "/festive_villani": container is running: stop the container before removing or force remove
PS C:\Users\Mikil> docker container stop 03d162e4b547
03d162e4b547
PS C:\Users\Mikil>
PS C:\Users\Mikil> docker container rm 03d162e4b547
03d162e4b547
PS C:\Users\Mikil> docker ps
CONTAINER ID   IMAGE     COMMAND      CREATED     STATUS      PORTS     NAMES
PS C:\Users\Mikil> docker images
REPOSITORY     TAG      IMAGE ID      CREATED     SIZE
ubuntu         latest   ca2b0f276964c  2 weeks ago  77.9MB

PS C:\Users\Mikil> docker images
REPOSITORY     TAG      IMAGE ID      CREATED     SIZE
PS C:\Users\Mikil>
```

Conclusion -

Thus, we successfully studied and implemented containerization using DOCKER.