Placement Empowerment Program

Cloud Computing and DevOps Centre

Installing Docker and Running Your First Container on Windows

Name: Hema S

Department : ECE

Introduction and Overview

Docker is a powerful tool that allows developers to create, deploy, and run applications in isolated environments called containers. These containers ensure that applications run consistently across different systems, eliminating compatibility issues.

In this guide, we will go through the step-by-step process of installing Docker on Windows, setting up a basic Nginx web server inside a container, and accessing it through a browser.

Objective

- To install Docker on Windows and verify its functionality.
- To learn how to pull and run a basic Nginx container.
- To understand how to access a containerized web application using a browser.
- To gain hands-on experience with containerization and Docker commands.

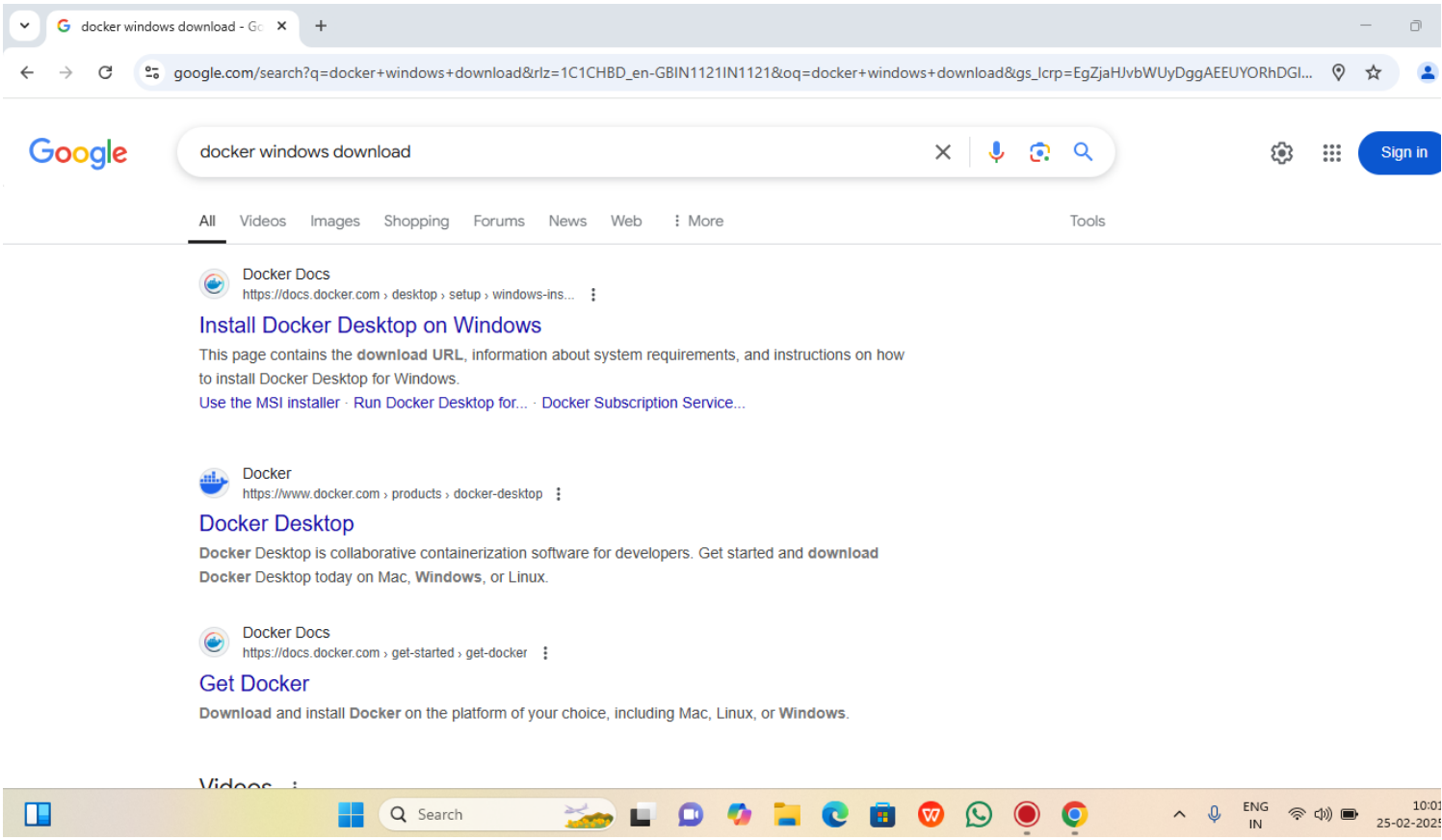
Importance

- Simplifies Deployment – Containers eliminate compatibility issues by packaging applications with all their dependencies.
- Improves Efficiency – Docker containers are lightweight and use system resources more effectively than virtual machines.
- Enhances Portability – Applications run consistently across different environments (local, cloud, or server).
- Boosts Scalability – Docker makes it easier to scale applications up or down based on demand.
- Speeds Up Development – Developers can quickly create isolated environments for testing and debugging**.**

Step-by-Step Overview

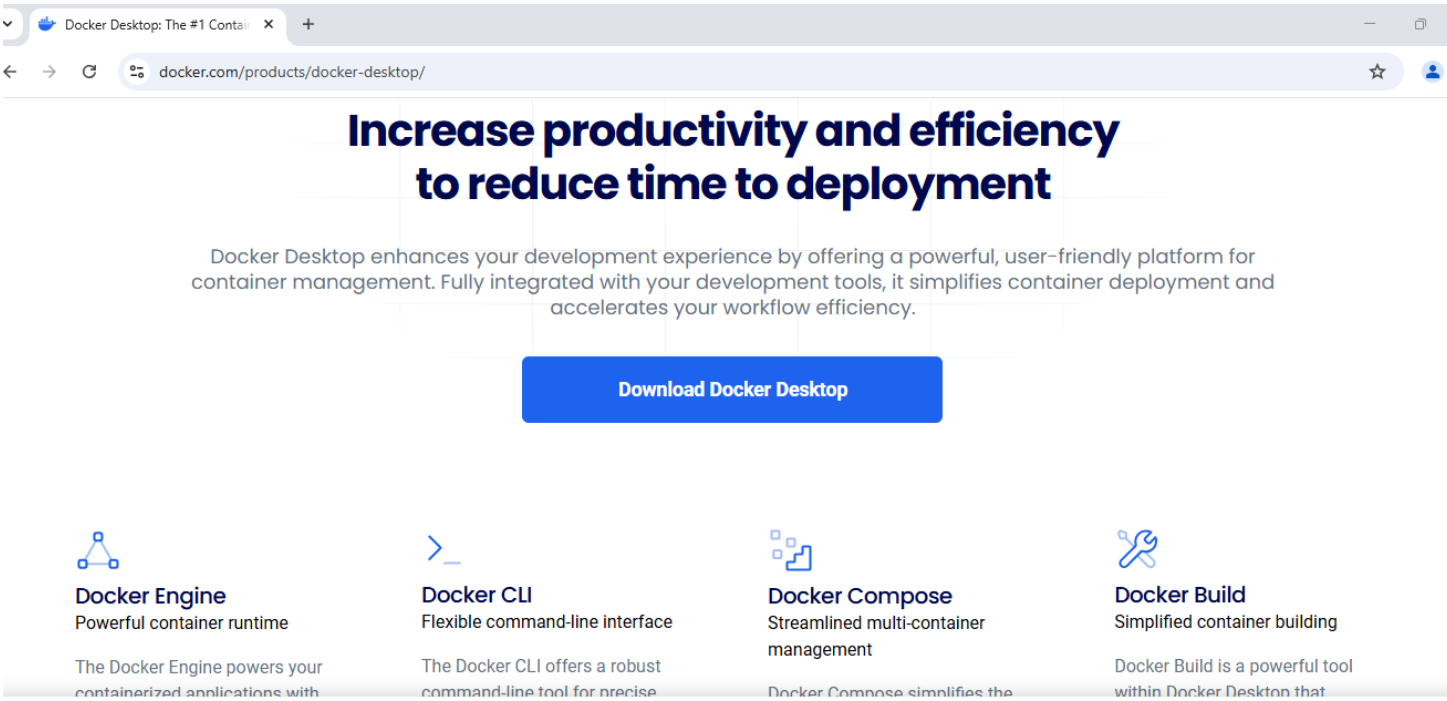
Step 1: Install Docker Desktop.

1. Download **Docker Desktop for Windows** from the official website: https://www.docker.com/products/docker-desktop/
2. Run the installer and follow the on-screen instructions.
3. Ensure **WSL 2** is enabled (Docker requires this for Windows).
4. Restart your PC and launch **Docker Desktop**.

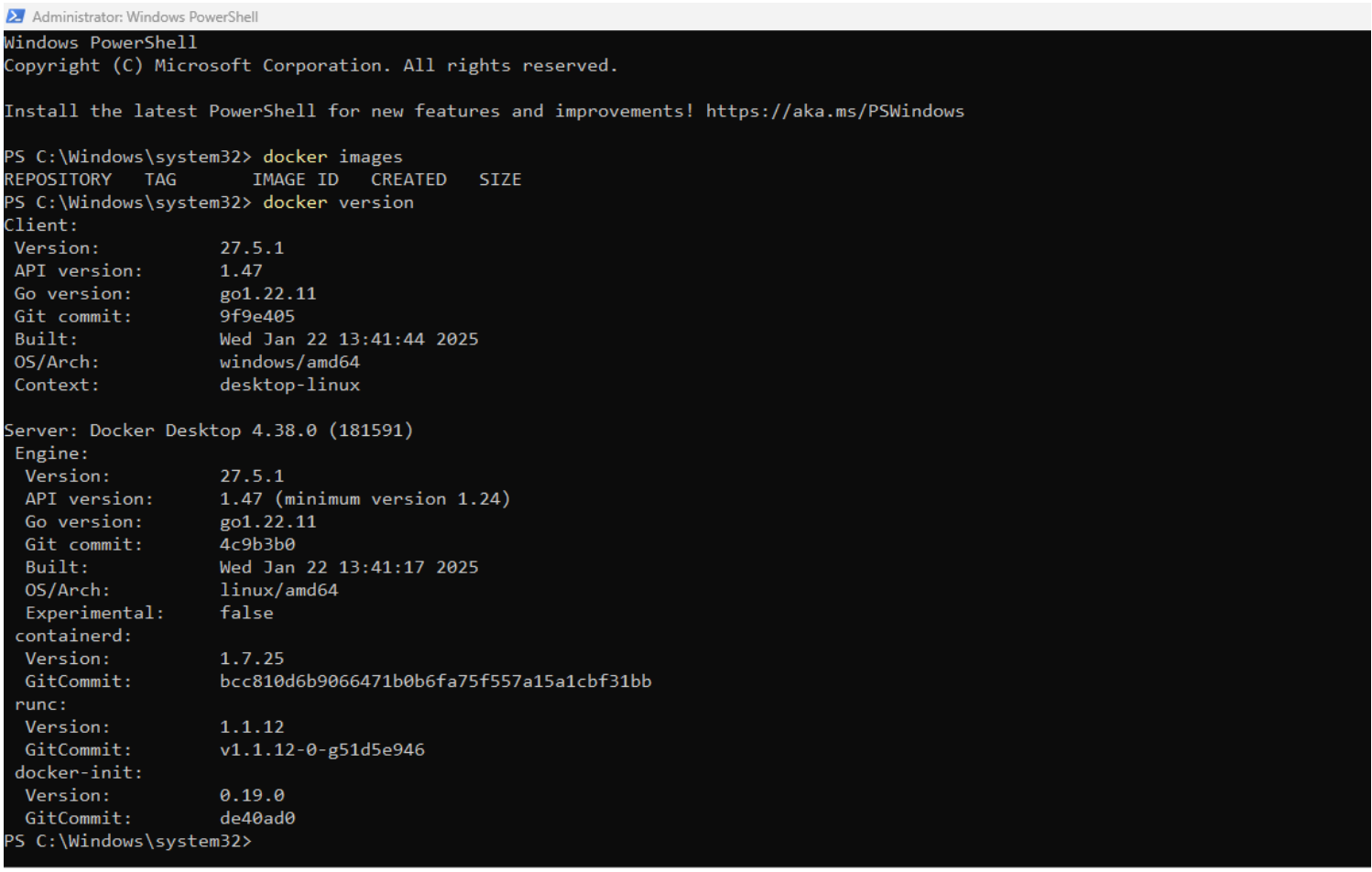


Step 2: Verify Docker Installation

- Open PowerShell and check if Docker is installed by running the version command.
- Verify that Docker is running properly by checking its system information.

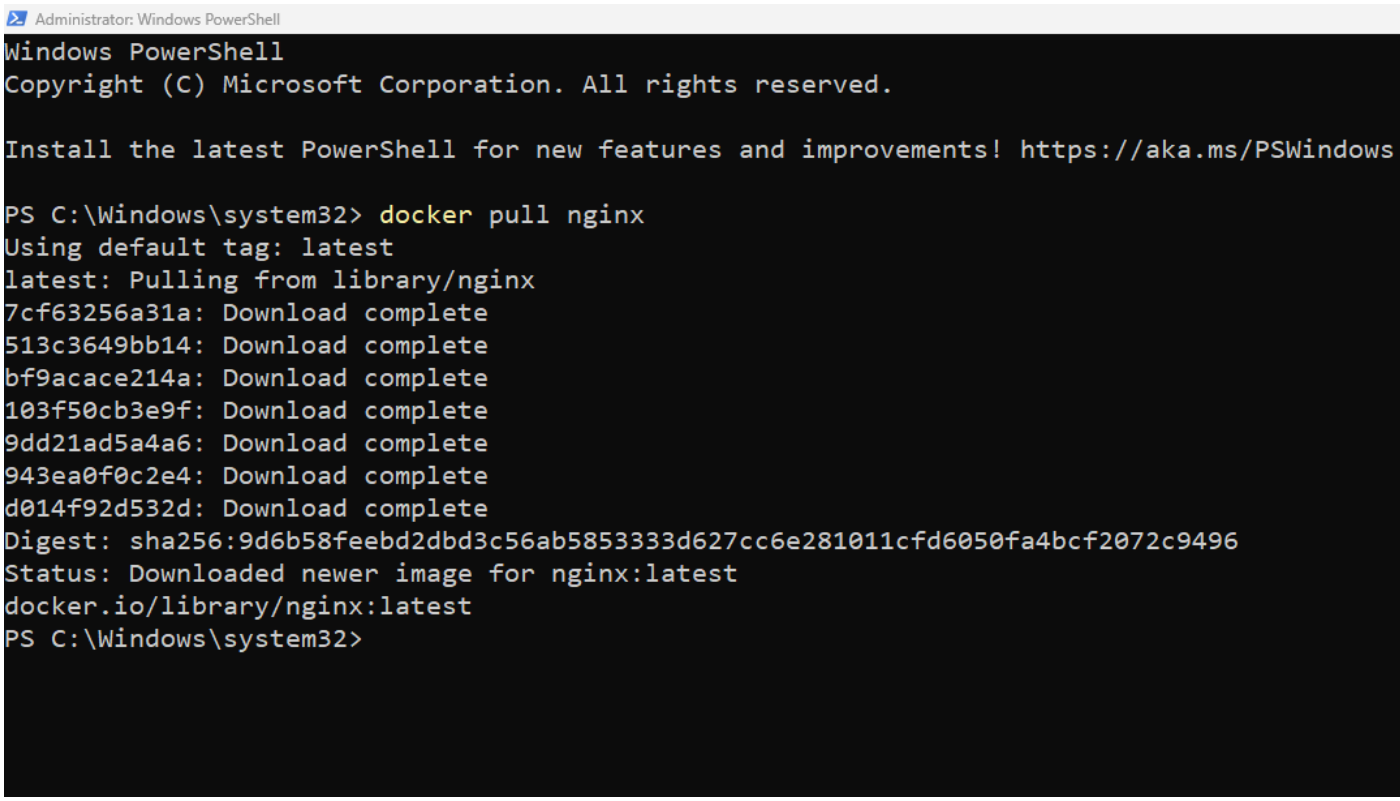

Docker Desktop: The #1 Contai
docker.com/products/docker-desktop/
Increase productivity and efficiency to reduce time to deployment
Docker Desktop enhances your development experience by offering a powerful, user-friendly platform for container management. Fully integrated with your development tools, it simplifies container deployment and accelerates your workflow efficiency.
Download Docker Desktop
Docker Engine
Powerful container runtime
The Docker Engine powers your containerized applications with
Docker CLI
Flexible command-line interface
The Docker CLI offers a robust command-line tool for precise
Docker Compose
Streamlined multi-container management
Docker Compose simplifies the
Docker Build
Simplified container building
Docker Build is a powerful tool within Docker Desktop that

- If there are any errors, ensure Docker Desktop is open and running in the background



Step 3: Pull the Nginx Docker Image

- Use the Docker pull command to download the latest Nginx image from Docker Hub.
- Once the image is downloaded, verify it by listing all available images in Docker.


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> docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
7cf63256a31a: Download complete
513c3649bb14: Download complete
bf9acace214a: Download complete
103f50cb3e9f: Download complete
9dd21ad5a4a6: Download complete
943ea0f0c2e4: Download complete
d014f92d532d: Download complete
Digest: sha256:9d6b58feebd2dbd3c56ab5853333d627cc6e281011cfd6050fa4bcf2072c9496
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
PS C:\Windows\system32>

Step 4: Run the Nginx Container

- Start an Nginx container by running it in detached mode and mapping it to port 8080.
- Verify that the container is running by listing all active containers.

```
PS C:\Windows\system32> docker run -d -p 8080:80 --name my-nginx nginx
```

```
PS C:\Windows\system32> docker run -d -p 8080:80 --name my-nginx nginx
b0d7457863296eace678cf1c4f63f88f1f1d85147239e64a3e97d4f6bc02a85a
PS C:\Windows\system32>
```

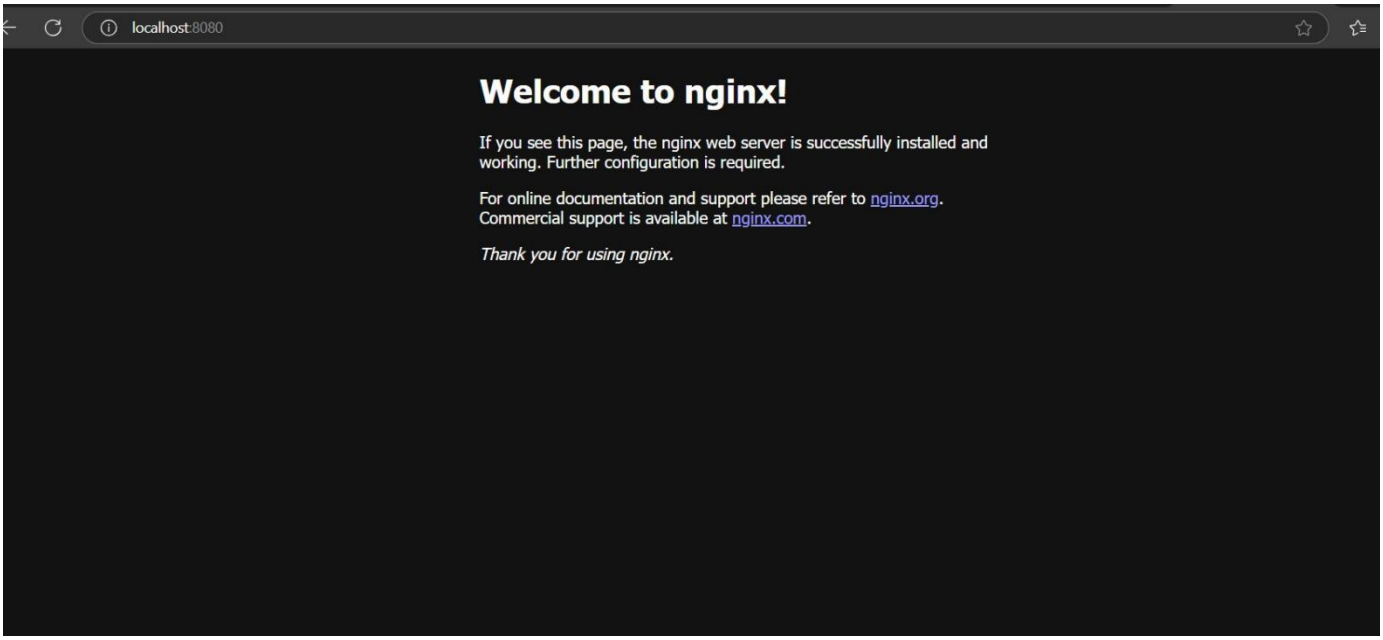
Step 5: Access the Nginx Web Page

- Open a web browser and go to http://localhost:8080.
- If everything is set up correctly, the default Nginx welcome page should appear.

```
PS C:\Windows\system32> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                    NAMES
b0d745786329   nginx     "/docker-entrypoint.…"   5 minutes ago   Up 5 minutes   0.0.0.0:8080->80/tcp   my-nginx
```

```
PS C:\Windows\system32> docker inspect my-nginx | Select-String '"HostPort": "8080"'

                    "HostPort": "8080"
                    "HostPort": "8080"
```



Step 6: Stop and Remove the Container

- If you no longer need the container, stop it using the stop command.
- Remove the stopped container from Docker.
- Optionally, remove the Nginx image if you want to free up space.

```
PS C:\Windows\system32> docker stop my-nginx
my-nginx
PS C:\Windows\system32> docker start my-nginx
my-nginx
PS C:\Windows\system32> docker rmi nginx
```

Expected Outcome

- Successful installation of **Docker Desktop** on Windows.
- Verification that Docker is running correctly through **PowerShell commands**.
- Pulling and running an **Nginx container** without errors.
- Accessing the **Nginx default welcome page** in a web browser at http://localhost:8080.
- Understanding basic **Docker commands** like pull, run, stop, and remove.