Docker

What is Docker?

Docker is an open-source platform that helps you build, ship, and run applications inside lightweight, portable, and isolated environments called containers.

What is a Container?

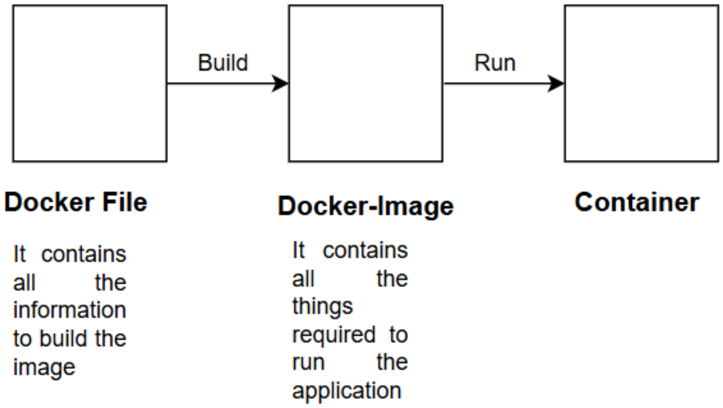
A container is like a mini virtual machine, but much more lightweight. It packages your application code together with all its dependencies (like libraries, frameworks, and config files), so it can run reliably in any environment—whether it's your laptop, a server, or the cloud.

Why Use Docker?

- Portability – "Works on my machine" no longer an issue.
- Speed – Containers start up super fast compared to VMs.
- Isolation – Keeps apps and services separate from each other.
- Scalability – Works well with microservices and orchestration tools (like Kubernetes).
- Consistency – Ensures environments are the same across development, testing, and production.

Key Components of Docker:

1. Docker Engine – The runtime that builds and runs containers.
2. Dockerfile – A text file that defines how to build a Docker image.
3. Docker Image – A snapshot/template for creating containers.
4. Docker Container – A running instance of a Docker image.
5. Docker Hub – A cloud-based registry where you can share and download Docker images.



It contains all the information to build the image

It contains all the things required to run the application

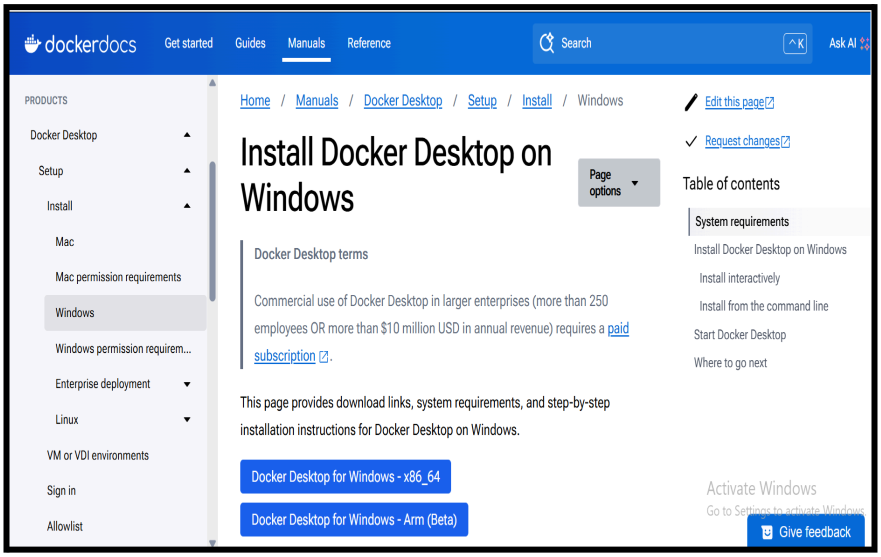
- ➢ Difference between Docker and containerization

Feature	Virtualization	Containerization
Definition	Running multiple virtual machines (VMs) on a single physical machine.	Running multiple isolated containers on the same OS.
Technology	Uses **hypervisors** (like VMware, VirtualBox, Hyper-V).	Uses **container engines** (like Docker, containerd).
SystemLevel	Virtualizes **entire hardware + OS**.	Virtualizes only the **application + dependencies**.
Guest OS	Each VM runs a full **separate OS** (Linux, Windows, etc.).	Containers share the **host OS kernel**.
Startup Time	Slower – boots the entire OS (minutes).	Faster – containers start in **seconds**.
Resource Usage	Heavy – uses more CPU, memory, and disk.	Lightweight – minimal overhead.

Isolation	Stronger isolation (each VM is fully separate).	Lighter isolation (process-level).
Use Case	Great for running different OSs or legacy apps.	Ideal for microservices, cloud-native apps, CI/CD.

➢ Steps to install docker on windows

1. Type Docker Desktop on windows



Install Docker Desktop on Windows

Docker Desktop terms

Commercial use of Docker Desktop in larger enterprises (more than 250 employees OR more than $10 million USD in annual revenue) requires a paid subscription.

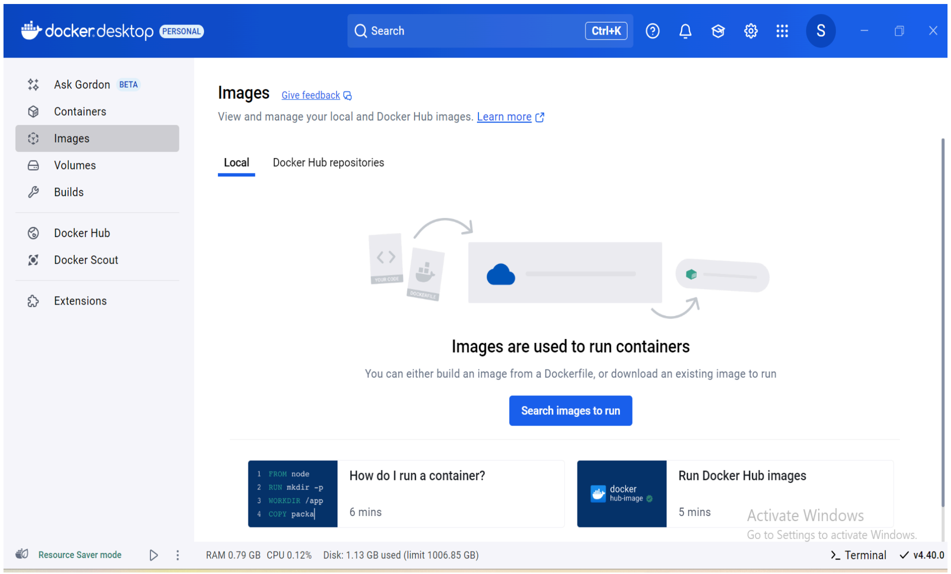
This page provides download links, system requirements, and step-by-step installation instructions for Docker Desktop on Windows.

Table of contents

- System requirements
- Install Docker Desktop on Windows
  - Install interactively
  - Install from the command line
- Start Docker Desktop
- Where to go next

Activate Windows
Go to Settings to activate Windows.

2. Click for windows -x86_64.
3. After Downloading sign and sign up and create your account and then you can see this type of window.



Images

View and manage your local and Docker Hub images. Learn more

Images are used to run containers

You can either build an image from a Dockerfile, or download an existing image to run

```
FROM node
RUN mkdir -p
WORKDIR /app
COPY packa
```

How do I run a container?
6 mins

Run Docker Hub images
5 mins

Activate Windows
Go to Settings to activate Windows.

RAM 0.79 GB CPU 0.12% Disk: 1.13 GB used (limit 1006.85 GB)

- Steps to install docker on AWS VM
    1. Launch an Ec2 instance on aws(ubuntu) .

    2. Connect it and update your system.

        ```
        #sudo apt update
        ```

    3. Install docker

        ```
        # sudo apt install Docker.io
        ```

    4. For accessing docker content first you have to add user to docker group

        ```
        #sudo usermod -aG docker $USER
        ```

        Where $USER is a Current user

    5. Then refresh it

        ```
        # newgrp Docker
        ```