

## What is Docker?

Docker is a tool for quickly building, testing and deploying software. It works by isolating the program in a container environment to assemble the code without risk to the larger application. After testing and fixing any bugs, developers can deploy the code safely and efficiently. Docker has revolutionized the world of software development by simplifying containerization.

Docker is an open-source technology that can run on Windows, Linux, and macOS. It works by isolating newly built software in its environment. This environment begins as a Docker image—an executable package that determines how to create a container, what software it will use, and how it will run.

After executing the image, Docker creates a container that holds everything needed to run an application. This includes system tools, code libraries, any dependencies, and runtime.

Docker image as the container's source code or the Docker container as the instance of the Docker image.

## **What is Docker used for?**

1. **Microservices architecture**
2. **Continuous integration and deployment**
3. **Containers**
4. **Software testing**

## **Benefits of Docker**

**Scalability**-Docker containers start up quickly, enabling seamless deployment of applications on demand. This responsiveness allows applications to scale up or down based on fluctuating traffic or increased workloads.

**Consistency**-Docker ensures consistency in development and deployment. It allows developers to create and clone packages that work in any environment. Users can verify the exact version of necessary libraries and packages in a container. This minimizes the risk of bugs from having slightly different dependency revisions.

Bug elimination and code testing would be tedious and time-consuming without this consistency. Software delivery would become unreliable due to the number of inconsistencies.

**Containers**-Docker containers are standalone units of code and all their dependencies that can run reliably in any environment. Docker allows users to build and deploy applications with packaged content and infrastructure that your IT teams manage and secure.

Before containerization tools, developers had to build their applications on local machines or virtual machines (VMs). They would then manually deploy them to a server environment to test and run.

With Docker, developers can now use simple commands to build, start, or stop containers and their code. This simplifies and streamlines application deployment.

**Portability**-Docker is a lightweight, portable software tool because it packages everything the application needs to run consistently across different environments.

Since everything is self-contained, the containers don't rely on pre-installed software or configurations on the host machine. This makes it easy to set up and deploy containers wherever necessary.

**Isolation**-A Docker container isolates code in a self-contained environment, independent of other containers or the host machine's operating system. This isolation makes it safer to test without possibly derailing the entire application. It also eliminates compatibility issues and dependency conflicts when running applications directly on different platforms.

### **Resource efficiency-**

1. Docker containers typically use less memory and disk space than VMs.
2. Docker containers share the host machine's kernel, eliminating the overhead of running multiple VMs.
3. Users can allocate specific CPU, memory, and disk space to each Docker container.
4. You can set Docker orchestration tools to automatically manage the lifecycle of containers and optimize resource utilization across a cluster of machines.

## Docker Desktop

Docker Desktop is an easy-to-install application for your Mac, Windows or Linux environment that enables you to build and share containerized applications and microservices. Docker Desktop includes the Docker daemon (dockerd), the Docker client (docker), Docker Compose, Docker Content Trust, Kubernetes, and Credential Helper.

## How to install Linux on Windows with WSL

Open PowerShell or Windows Command Prompt in **administrator** mode by right-clicking and selecting "Run as administrator", enter the wsl --install command, then restart your machine.

PowerShell  
wsl --install

To set the default version to WSL 1 or WSL 2 when a new Linux distribution is installed, use the command: wsl --set-default-version <Version#>, replacing <Version#> with either 1 or 2.

## Install Docker Desktop on Windows

- Double-click Docker Desktop Installer.exe to run the installer.
- When prompted, ensure the Use WSL 2 instead of Hyper-V option on the Configuration page is selected or not depending on your choice of backend.

- If your system only supports one of the two options, you will not be able to select which backend to use.
- Follow the instructions on the installation wizard to authorize the installer and proceed with the install.
- When the installation is successful, select Close to complete the installation process.

### Install Docker Desktop on Windows from the Command Line

After downloading Docker Desktop Installer.exe, run the following command in a terminal to install Docker Desktop:

```
$ "Docker Desktop Installer.exe" install
```

If you're using PowerShell you should run it as:

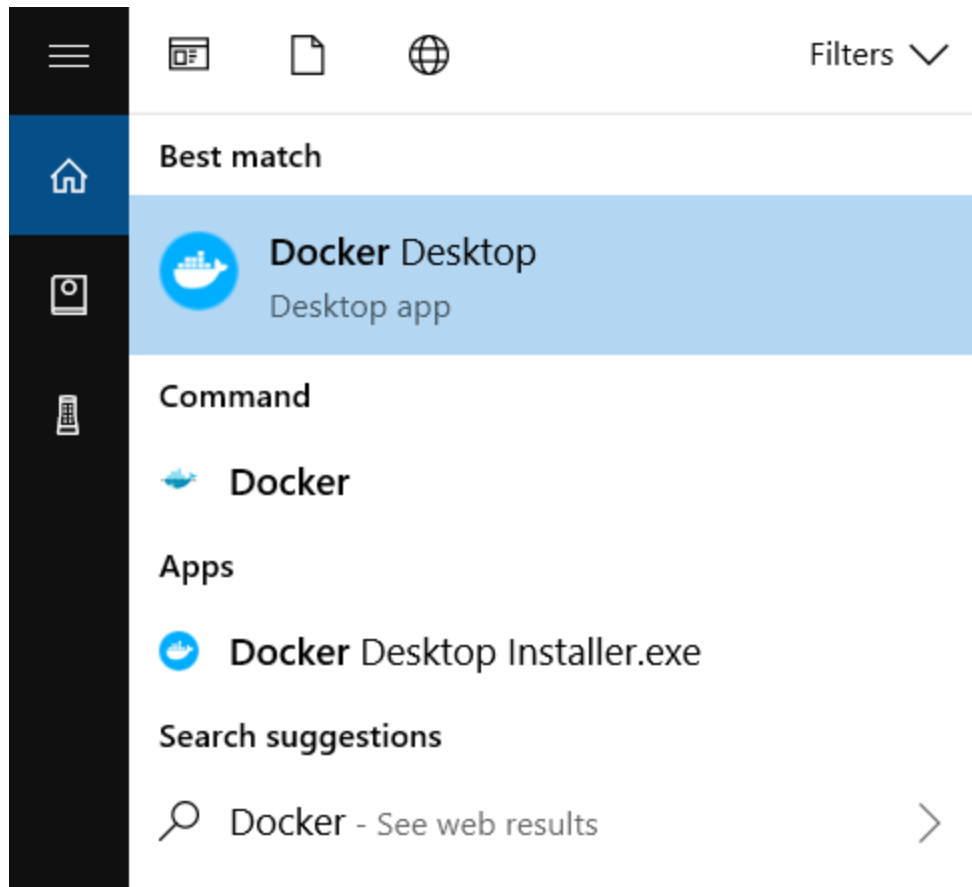
```
Start-Process 'Docker Desktop Installer.exe' -Wait install
```

If using the Windows Command Prompt:

```
start /w "" "Docker Desktop Installer.exe" install
```

### Start Docker Desktop

1 Search for Docker, and select Docker Desktop in the search results.



2 The Docker menu ( whale menu ) displays the Docker Subscription Service Agreement.

- Docker Desktop is free for small businesses (fewer than 250 employees AND less than \$10 million in annual revenue), personal use, education, and non-commercial open source projects.
- Otherwise, it requires a paid subscription for professional use.

- Paid subscriptions are also required for government entities.
- The Docker Pro, Team, and Business subscriptions include commercial use of Docker Desktop.

3 Select Accept to continue. Docker Desktop starts after you accept the terms.

Note that Docker Desktop won't run if you do not agree to the terms. You can choose to accept the terms at a later date by opening Docker Desktop.

---

What is a Docker image?

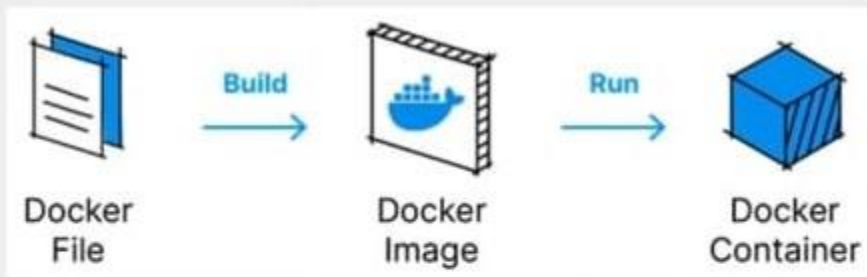
A Docker image is a file used to execute code in a Docker container. Docker images act as a set of instructions to build a Docker container, like a template. Docker images also act as the starting point when using Docker. An image is comparable to a snapshot in virtual machine (VM) environments. Docker is used to create, run and deploy applications in containers. A Docker image contains application code, libraries, tools, dependencies and other files needed to make an application run. When a user runs an image, it can become one or many instances of a container.

---

What is a Docker File?

A `Dockerfile` is a text file that Docker reads in from top to bottom. It contains a bunch of instructions which informs Docker HOW the Docker image should get built.





**Create a Dockerfile**



**Add instructions in Dockerfile to create Docker image**



**Run Dockerfile to create Docker image**



**Run Docker image to create Docker container**



**Access the application running in Docker container**

## Dockerfile > Docker Image > Docker Container > Access the App

Step 1 - Create a new directory

```
mkdir myapp  
cd myapp
```

Step 2 - Create a file called "index.html"

```
echo "Hello, world!" > index.html
```

Step 3 - Create a file named Dockerfile

```
touch Dockerfile
```

Step 4 - Open the "Dockerfile" file in a text editor and add the following lines:

```
FROM nginx  
COPY index.html /usr/share/nginx/html
```

This Dockerfile defines a new Docker image that

- uses the official **nginx** image as a base
- then copy the index.html file to the appropriate location in the image

Step 5 - Start docker & Build docker image from dockerfile

```
docker build -t myapp .
```

This command builds a new Docker image with the tag "myapp" using the Dockerfile in the current directory.

Step 6 - Run docker container from the image

```
docker run -p 8080:80 myapp
```

This tells Docker to run the myapp container and map port 8080 on your local machine to port 80 inside the container.

Step 7 - Access the app

Open a web browser and navigate to <http://localhost:8080> to see the "Hello, world!" message displayed in your web browser.

A Dockerfile is a text file with instructions to build a Docker Image

When we run a Dockerfile, Docker image is created

When we run the docker image, containers are created

## **Creating a Custom Image**

Create folder docker(any drive)

Open folder with visual studio

Tools->Command Line->Developer powershell

Pwd

C:\Users\DELL\Desktop\Docker

Add a new file i.e. Dockerfile

To run docker in undetach mode

docker run -d -p 8080:80 myapp

check container

docker ps