**Docker**

Getting Docker

Installing docker into the windows machine.

1. Double-click **Docker Desktop Installer.exe** to run the installer.

If you haven’t already downloaded the installer (Docker Desktop Installer.exe), you can get it from [**Docker Hub**](https://hub.docker.com/?overlay=onboarding)(<https://hub.docker.com/?overlay=onboarding>). It typically downloads to your Downloads folder, or you can run it from the recent downloads bar at the bottom of your web browser.

1. Follow the instructions on the installation wizard to accept the license, authorize the installer, and proceed with the install.

When prompted, authorize the Docker Desktop Installer with your system password during the install process. Privileged access is needed to install networking components, links to the Docker apps, and manage the Hyper-V VMs.

1. Click **Finish** on the setup complete dialog and launch the Docker Desktop application.

System Requirements

* Windows 10 64-bit: Pro, Enterprise, or Education (Build 15063 or later).
* Hyper-V and Containers Windows features must be enabled.
* The following hardware prerequisites are required to successfully run Client Hyper-V on Windows 10:
  + 64 bit processor with [Second Level Address Translation (SLAT)](http://en.wikipedia.org/wiki/Second_Level_Address_Translation)
  + 4GB system RAM
  + BIOS-level hardware virtualization support must be enabled in the BIOS settings. For more information, see [Virtualization](https://docs.docker.com/docker-for-windows/troubleshoot/#virtualization-must-be-enabled).

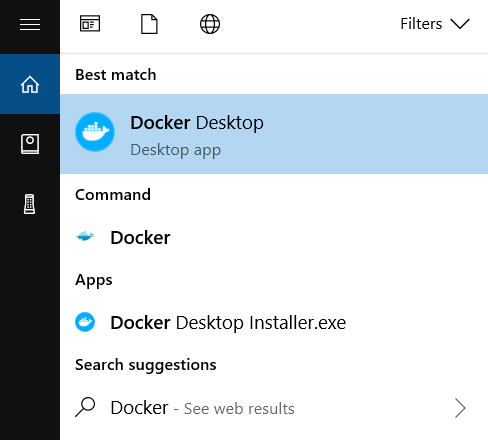
### What’s included in the installer

The Docker Desktop installation includes [Docker Engine](https://docs.docker.com/engine/userguide/), Docker CLI client, [Docker Compose](https://docs.docker.com/compose/overview/), [Docker Machine](https://docs.docker.com/machine/overview/), and [Kitematic](https://docs.docker.com/kitematic/userguide/).

Containers and images created with Docker Desktop are shared between all user accounts on machines where it is installed. This is because all Windows accounts use the same VM to build and run containers.

## **Start Docker Desktop**

Docker Desktop does not start automatically after installation. To start Docker Desktop, search for Docker, and select **Docker Desktop** in the search results.



When the whale icon in the status bar stays steady, Docker Desktop is up-and-running, and is accessible from any terminal window.

whale on taskbar

If the whale icon is hidden in the Notifications area, click the up arrow on the taskbar to show it. To learn more, see [Docker Settings](https://docs.docker.com/docker-for-windows/#docker-settings-dialog).

After installing the Docker Desktop app, you also get a pop-up success message with suggested next steps, and a link to this documentation.

**WHAT IS DOCKER**

* Docker is a container management service.
* Docker is a tool designed to make it easier to create, deploy, and run applications by using containers.
* Docker has many advantages but the most important one is the container analogy. That you don't only ship your code but ship the whole OS and all related layers with each deployment in a standard way.

**WHAT IS CONTAINER**

* Containers can be termed as the standard boxes where the goods can be packaged in a standard way, so you just have to focus on developing the item you develop do not worry about how it will run.
* A Docker container is a Docker image **(An image contains the Docker file, libraries, and code your application needs to run, all bundled together)** brought to life.
* **DOCKER IMAGES--**An image includes everything needed to run an application, the code or binary, runtimes, dependencies, and any other file system objects required.

**Sharing image**

* Multiple team members may wish to share images
* Images can be in production, under development or under test
* Docker Hub is a repository where images can be stored and shared.
* Each image is tagged to allow versioning
* Any image can be “pulled” to any host (with appropriate credentials)
* Tagging as “latest” allows updates to be propagated.
* Pull: Latest gets the last image checked into repository with that name.

**DOCKER ADVANTAGES**

* Runs on my machine = runs anywhere
* Reproducibility
* Isolation
* Docker Hub
* Continuous Integration

**Adding Docker Support to .NET API**

WHY TO USE DOCKER WITH .NET CORE APPLICATION??

Docker is a way of encapsulating everything you need to run an application such as .net core app including the code ,frameworks ,packages it depends on and underlying O.S ,so Docker container is a way of putting them altogether in one neat package and to use or it or put in wherever you need to.

Open Visual Studio and follow the steps below:-

* File=>New=>Project=>ASP.NET Core Web Application (Name your application) =>API =>Click OK.
* In the solution explorer

Right click on your project name=>click Add=>Docker Support=> Specify the OS you want to target=>Click Ok (This will create a Docker file in the project.)

* Now you have successfully added Docker support to your project.

**Making your first Docker Image**

Follow the Steps:-

* Open the command prompt (win+r=>cmd).
* Docker build -t **dockerdemo** (Location of your project ).
* Check the image is successfully build using **Docker images** command.
* You successfully created an image.

**Pushing Docker image to Docker HUB**

* Docker Hub is a registry service on the cloud that allows you to download Docker images that are built by other communities. You can also upload your own Docker built images to Docker hub.

Follow the steps below:-

Step 1:- Login Docker using your Docker credentials.

**Docker login**

Step 2:- **Docker images**

REPOSITORY TAG IMAGE ID CREATED SIZE

**dockerdemo** dev 1689e0cfaf32 2 days ago 514MB

Step 3:- docker tag <IMAGE ID> <docker\_username/docker repositoryname:tag>

Step 4:- docker push <docker\_username/docker repositoryname>

Docker image successfully pushed to docker Hub.

**Pulling Docker Images**

Commands :-- docker pull <image name>..

Example - docker pull hello-world