

Step 1 : Install Docker <https://docs.docker.com/get-docker/>



Step 2 : Install Docker Desktop on Windows

- a) Read the System Requirements before installing Docker.

## System requirements

Your Windows machine must meet the following requirements to successfully install Docker Desktop.

WSL 2 backend

[Hyper-V backend and Windows containers](#)

### WSL 2 backend

- Windows 11 64-bit: Home or Pro version 21H2 or higher, or Enterprise or Education version 21H2 or higher.
- Windows 10 64-bit: Home or Pro 21H1 (build 19043) or higher, or Enterprise or Education 20H2 (build 19042) or higher.
- Enable the WSL 2 feature on Windows. For detailed instructions, refer to the [Microsoft documentation](#).
- The following hardware prerequisites are required to successfully run WSL 2 on Windows 10 or Windows 11:
  - 64-bit processor with [Second Level Address Translation \(SLAT\)](#)
  - 4GB system RAM
  - BIOS-level hardware virtualization support must be enabled in the BIOS settings. For more information, see [Virtualization](#).

## b) Download Docker

# Install Docker Desktop on Windows

Welcome to Docker Desktop for Windows. This page contains information about Docker Desktop for Windows system requirements, download URL, instructions to install and update Docker Desktop for Windows.

Docker Desktop for Windows

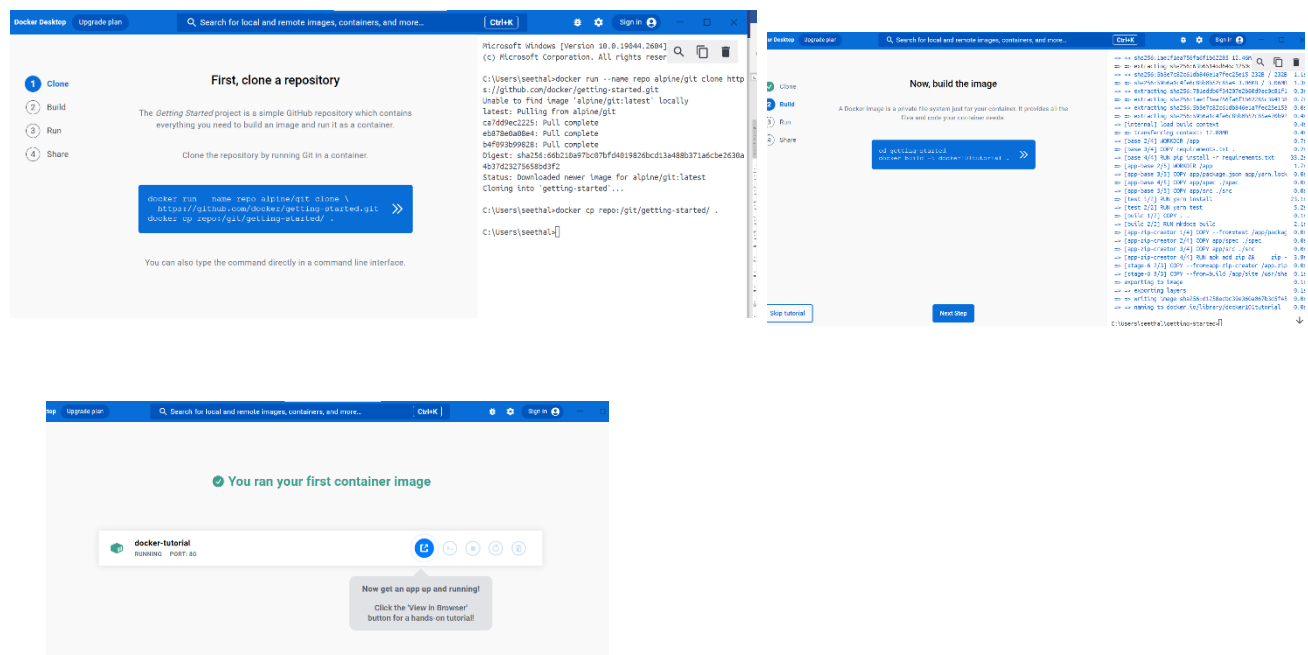
For checksums, see [Release notes](#)

## c) Install Docker

Docker Manual: <https://docs.docker.com/desktop/>

Exercise:

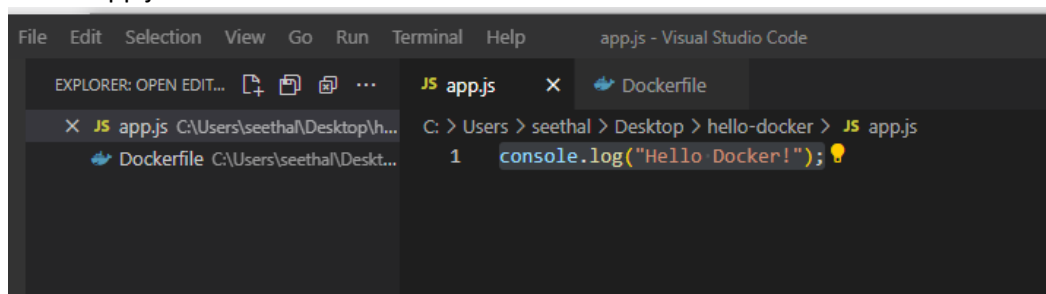
### 1. Run Docker Desktop & follow the first tutorial.



## 2. [Docker Tutorial for Beginners](#)

//Using command lines docker

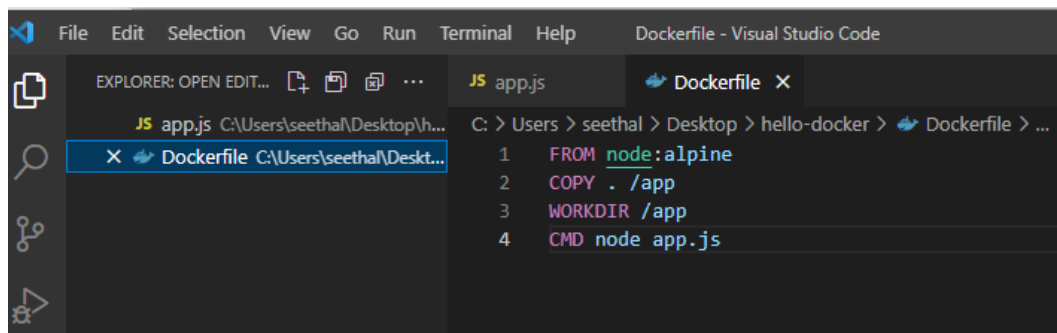
1. Create directory in Desktop  
mkdir hello-docker
2. Move to hello-docker  
cd hello-docker
3. Create app.js



The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying 'app.js' and 'Dockerfile'. The main editor area shows the content of 'app.js' with the following code:

```
C: > Users > seethal > Desktop > hello-docker > JS app.js
1 console.log("Hello Docker!");
```

## 4. Create Dockerfile



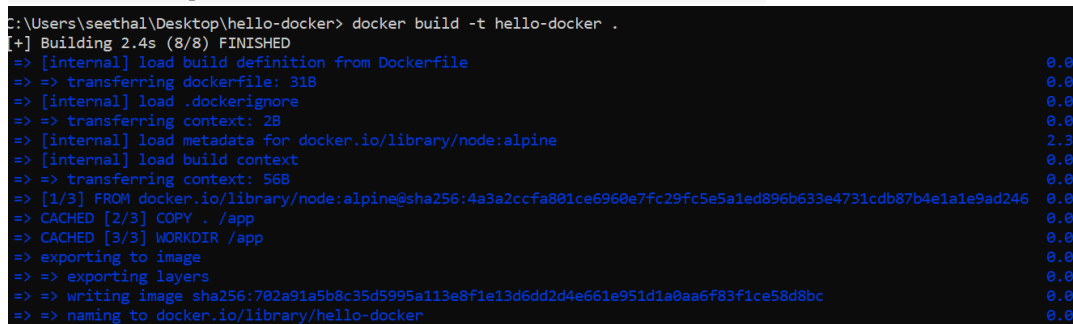
The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying 'app.js' and 'Dockerfile'. The main editor area shows the content of 'Dockerfile' with the following code:

```
1 FROM node:alpine
2 COPY . /app
3 WORKDIR /app
4 CMD node app.js
```

## 5. Build the image file

Command Line:

```
docker image build [OPTIONS] PATH | URL | -
```



The screenshot shows the terminal output for the command 'docker build -t hello-docker .' in the directory 'C:\Users\seethal\Desktop\hello-docker'. The output shows the build process, including downloading the 'node:alpine' image, copying files, and exporting the layers to the image.

```
C:\Users\seethal\Desktop\hello-docker> docker build -t hello-docker .
[+] Building 2.4s (8/8) FINISHED
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 31B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/library/node:alpine 2.3s
=> [internal] load build context 0.0s
=> => transferring context: 56B 0.0s
=> [1/3] FROM docker.io/library/node:alpine@sha256:4a3a2ccfa801ce6960e7fc29fc5e5a1ed896b633e4731c8b87b4e1a1e9ad246 0.0s
=> CACHED [2/3] COPY . /app 0.0s
=> CACHED [3/3] WORKDIR /app 0.0s
=> exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:702a91a5b8c35d5995a113e8f1e13d6dd2d4e661e951d1a0aa6f83f1ce58d8bc 0.0s
=> => naming to docker.io/library/hello-docker 0.0s
```

## 6. To view image files

```
C:\Users\seethal\Desktop\hello-docker>docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hello-docker	latest	702a91a5b8c3	About a minute ago	177MB
docker101tutorial	latest	d1258ecbc39e	2 days ago	47MB

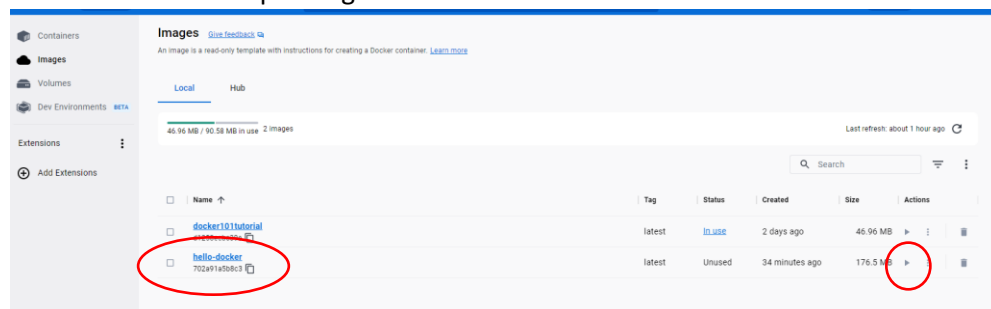
## 7. To containerise, run the image file

Command Line:

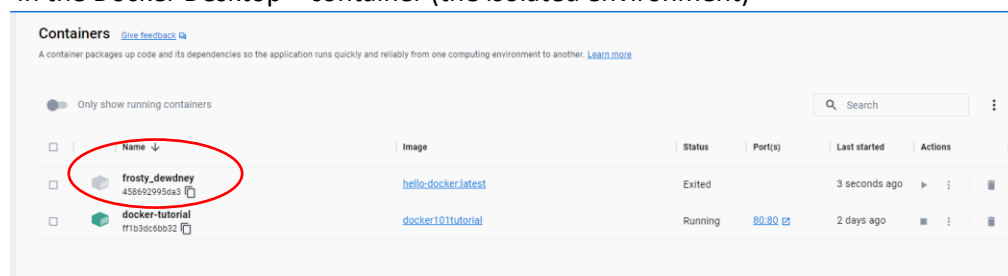
```
docker container run --name [container_name] [docker_image]
//if option of container name is not provided, default name
//will be given
```

```
C:\Users\seethal\Desktop\hello-docker>docker run hello-docker
Hello Docker!
```

## 8. In the Docker Desktop - image



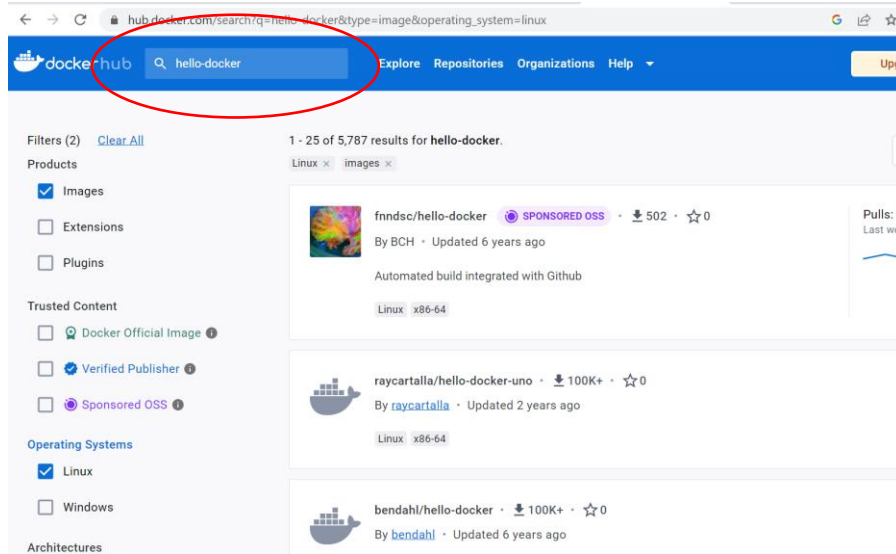
## 9. In the Docker Desktop – container (the isolated environment)



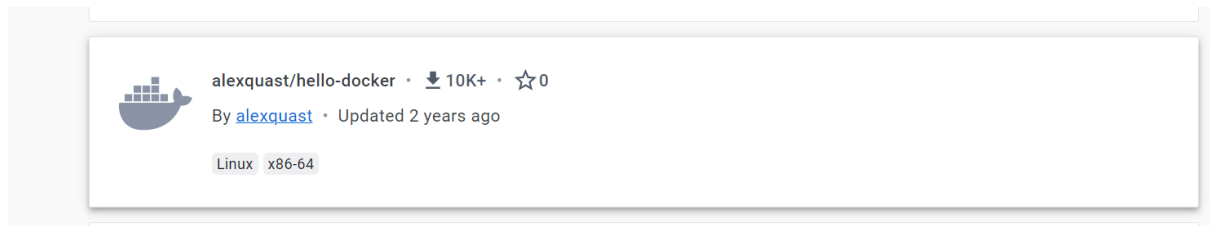
## 10. Docker hub – Collaborate via push/pull

Step 1 : <https://hub.docker.com/>

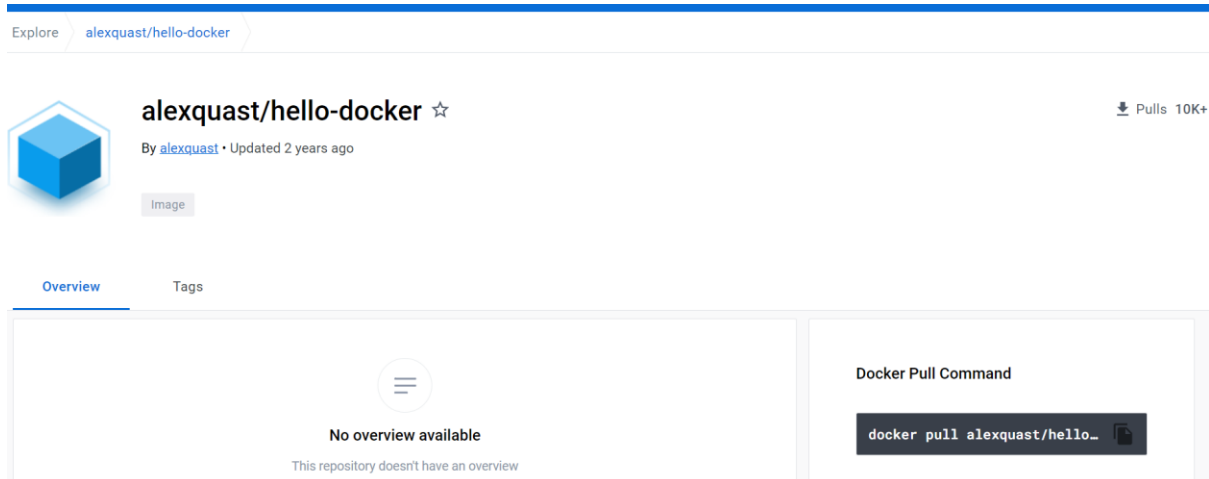
## Step 2: Search for image file



## Step 3: Click on the image which you need to pull



## Step 4: Copy and paste to your command prompt



## Step 4: docker pull alexquast/hello-docker


```
C:\Users\seethal>docker pull alexquast/hello-docker
Using default tag: latest
latest: Pulling from alexquast/hello-docker
9a6724ff3fcd: Pull complete
8b3c21ce1f8c: Pull complete
560ed84bbbcc: Pull complete
281a19a6c438: Pull complete
83be522e5500: Pull complete
1dd14de5680c: Pull complete
9738682cd647: Pull complete
Digest: sha256:30e3ec5a18ab913b58057ecf062854abb49108ca134d06769a6a43493ac03b4c
Status: Downloaded newer image for alexquast/hello-docker:latest
docker.io/alexquast/hello-docker:latest
```

```
C:\Users\seethal>docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hello-docker	latest	702a91a5b8c3	22 hours ago	177MB
docker101tutorial	latest	d1258ecbc39e	3 days ago	47MB
alexquast/hello-docker	latest	433da19f7fd4	2 years ago	122MB

//Use the Docker desktop to run the image, use the optional to rename the container

//Use the terminal to input Linux commands


**Run a new container**  
alexquast/hello-docker:latest

---

Optional settings ^

Container name

A random name is generated if you do not provide one.

**Ports**

No ports exposed in this image

**Volumes**

Host path ...

Container path

+

**Environment variables**

Variable

Value

+

Cancel

Run

//Click on the image to view the files

	Name ↑	Tag	Status	Created	Size	Actions
<input type="checkbox"/>	<a href="#">alexquast/hello-docker</a> 433da19f7fd4	latest	<a href="#">In use</a>	about 2 years ago	122.22 MB	

Containers

Images

Volumes

Dev Environments BETA

Extensions 1

Add Extensions

alexquast/hello-docker:latest IN USE

433da19f7fd4

CREATED about 2 years ago

SIZE 122.22 MB

Recommended fixes

Run

Image vulnerability analysis provided by Docker Scout. Free while in early access. [Read about Docker Scout](#)

Image hierarchy

FROM alpine:3.11, 3.11.7

FROM node:14.15-alpine, 14.15-alpine3.11, ...

ALL alexquast/hello-docker:latest

Layers (13)

0

ADD file:8ed80010e443da19d...

5.61 MB

1

CMD ["/bin/sh"]

0 B

2

ENV NODE\_VERSION=14.15.5

0 B

3

addgroup -g 1000 node && ad...

102.9 MB

4

ENV YARN\_VERSION=1.22.5

0 B

Images (3)

Vulnerabilities (47)

Packages (466)

Give feedback

Filters

Only fixable packages

Package or CVE name

Package

Vulnerabilities

> openssl 1.1.1-r0

1 C 3 H

> apk-tools 2.10.5-r0

1 C 1 H

1-10 of 22