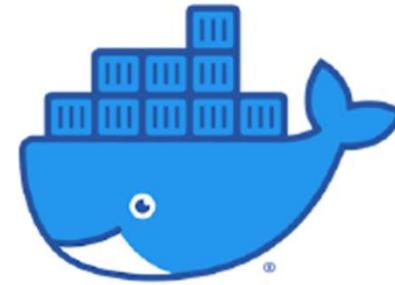


# DOCKER



---

## INTRODUCTION WITH CLI COMMANDS

**EXERCISE – 1** PULLING AN IMAGE, RUNNING IT AS CONTAINER, MODIFYING IT, COMMITTING THE CHANGES WITH NEW IMAGE NAME AND PUSHING INTO DOCKER HUB

**EXERCISE -2** CREATING A NEW IMAGE USING DOCKERFILE, PUSHING INTO DOCKERHUB AND RUNNING IT.

# Content

- a) DOCKER INSTALLATION
- b) DOCKER CLI COMMANDS
- c) PULL IMAGE FROM LOCAL REPOSITORY AND DOCKER HUB REPOSITORY AND RUN THE IMAGE AS A CONTAINER USING RUN COMMAND . PULL AND RUN/EXECUTE TOMCAT DOCKER IMAGE FROM DOCKER HUB REPOSITORY.
- d) PULLING THAT IMAGE, RUNNING THAT IMAGE,  
EXECUTING IN INTERACTIVE MODE
- e) CREATE DOCKER IMAGE USING DOCKER FILE.  
PUBLISHING IT INTO THE DOCKER REPOSITORY

# Docker

Docker is an open platform for developing, shipping, and running applications.

Docker enables you to separate your applications from your infrastructure so you can deliver software quickly.

# Need of Docker

Before docker has introduced developers and testers experienced difficulty due to the different computer environments. i.e. the code on one system doesn't work on another system.

The solution was **Virtual Machine**.

**Docker** was introduced as an alternate solution.

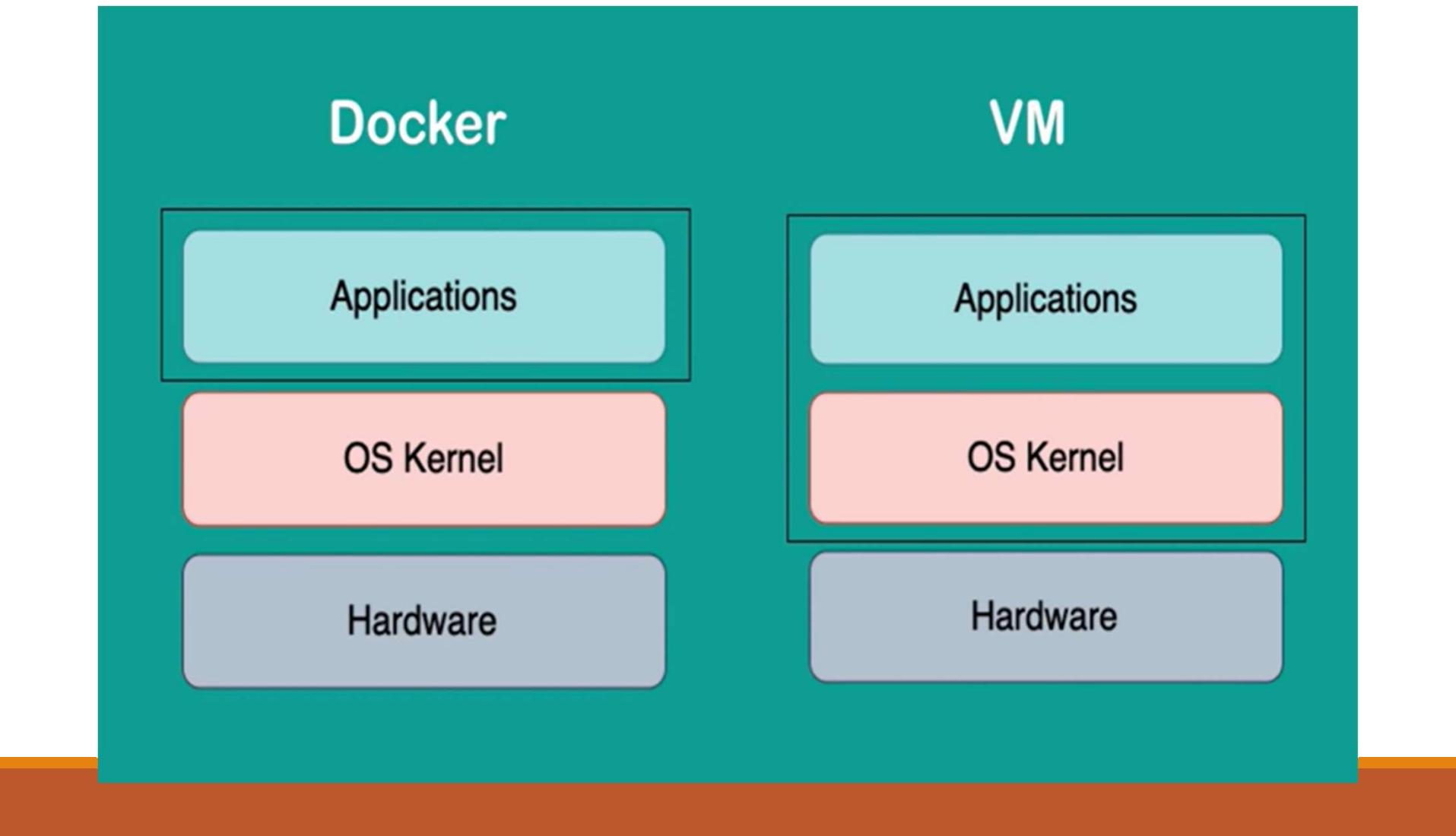
Docker is an open platform for developing, shipping, and running applications.

Docker provides the ability to package and run an application in a loosely isolated environment called a **Container**.

Many containers can run simultaneously on a given host.

Container contains everything needed to run the application, so you do not need to rely on what is currently installed on the host.

# Difference Between Docker and VM



# What is Docker

- DOCKER INCLUDES DOCKER CLIENT, DOCKER SERVER, DOCKER HUB, ETC.
- DOCKER CLIENT IS LIKE CLI. IT IS A TOOL WE ISSUE COMMANDS TO REACH OUT TO THE DOCKER DAEMON.
- DOCKER DAEMON IS A SERVICE THAT RUNS ON YOUR HOST OPERATING SYSTEM.
- DOCKER DAEMON CONVERT IMAGE TO THE CONTAINER, RUN, STOP THE CONTAINER
- IMAGE IS A FILE CONTAINING INFORMATION (RELATED TO REQUIRED RESOURCES)
- CONTAINER HOLDS ALL RESOURCES TO RUN THE IMAGE. IT OCCUPIES A SMALL PART OF THE CPU, RAM, HD.

# Container

- ▶ A way to **package** application with **all** the **necessary dependencies** and **configuration**



- ▶ **Portable artifact**, easily shared and moved around

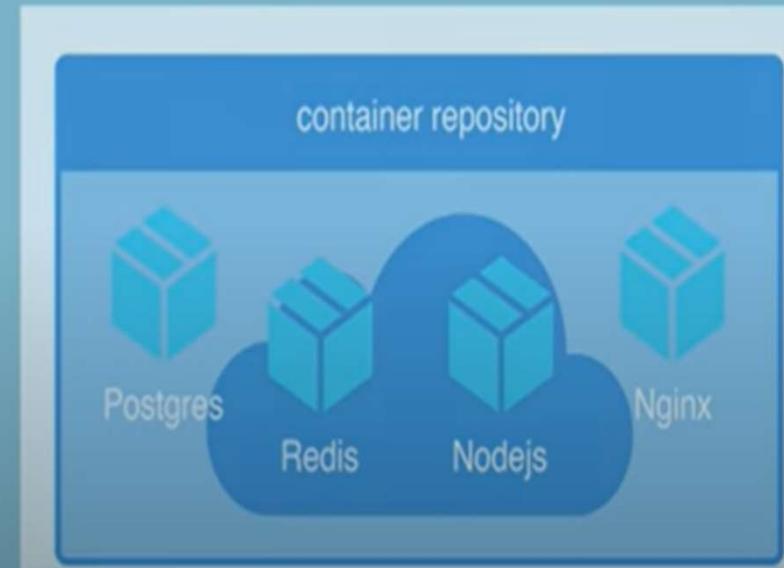


- ▶ Makes development and deployment **more efficient**

# Where do container live

- ▶ Container Repository
- ▶ Private repositories
- ▶ Public repository for Docker

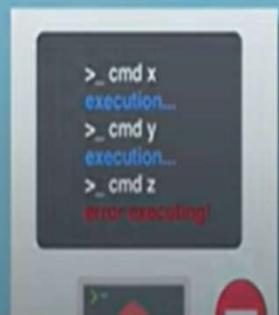
DockerHub



## Application Development

### Before containers

- ▶ Installation process different on each OS environment
- ▶ Many steps where something could go wrong



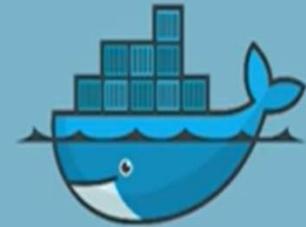
Developer

Developer

## Application Development

- ▶ own isolated environment
- ▶ packaged with all needed configuration
- ▶ one command to install the app
- ▶ run same app with 2 different versions

## After containers



### Container



configuration



PostgreSQL  
v9.3

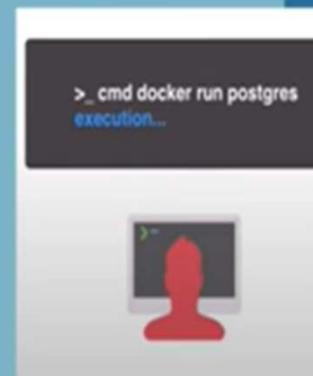


Start script

### PostgreSQL Container



### PostgreSQL Container



## Application Deployment

## Before containers

► configuration on the server needed

✗ dependency version conflicts

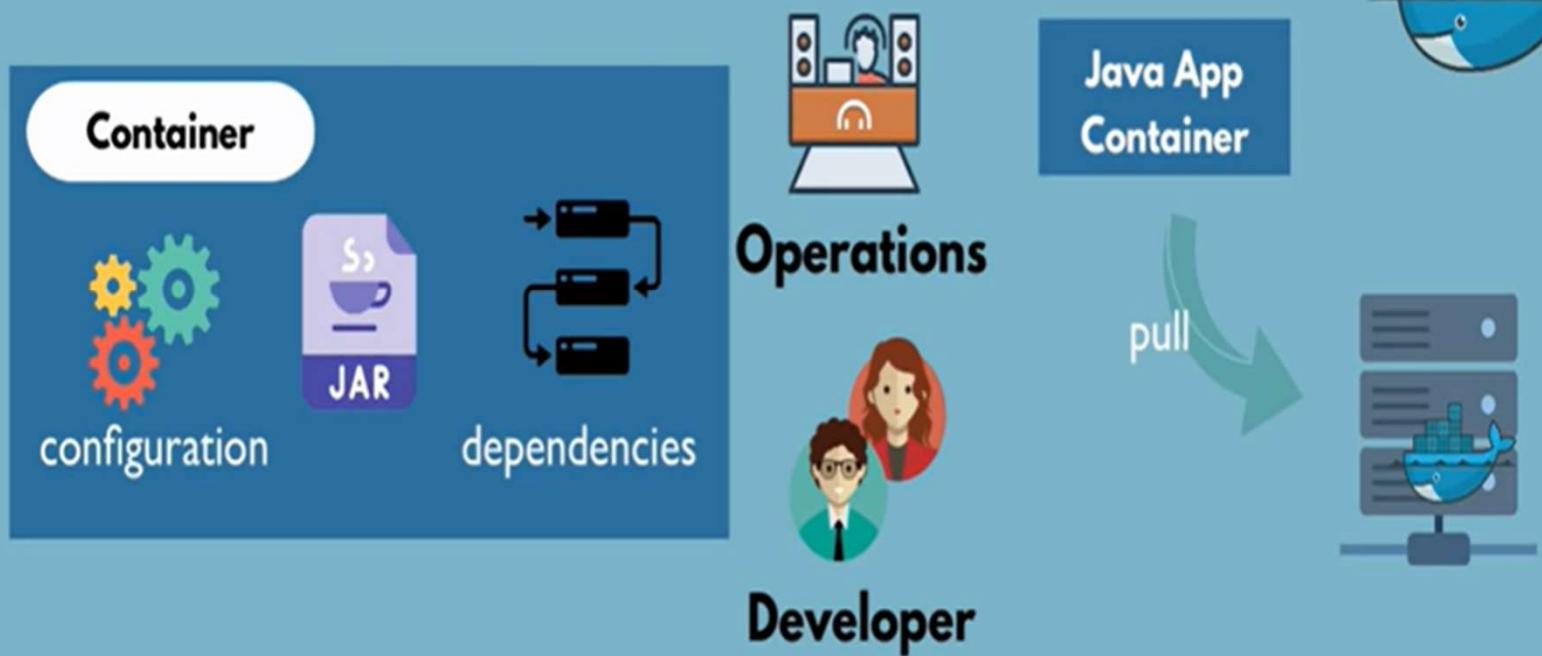
✗ misunderstandings



## Application Deployment

## After containers

- ▶ Developers and Operations work together to package the application in a container
- ▶ No environmental configuration needed on server  - except Docker Runtime



## What is a Container?

- ▶ Layers of images
- ▶ Mostly **Linux Base Image**, because small in size
- ▶ Application image on top



postgres:10.10

**Layer** - application image



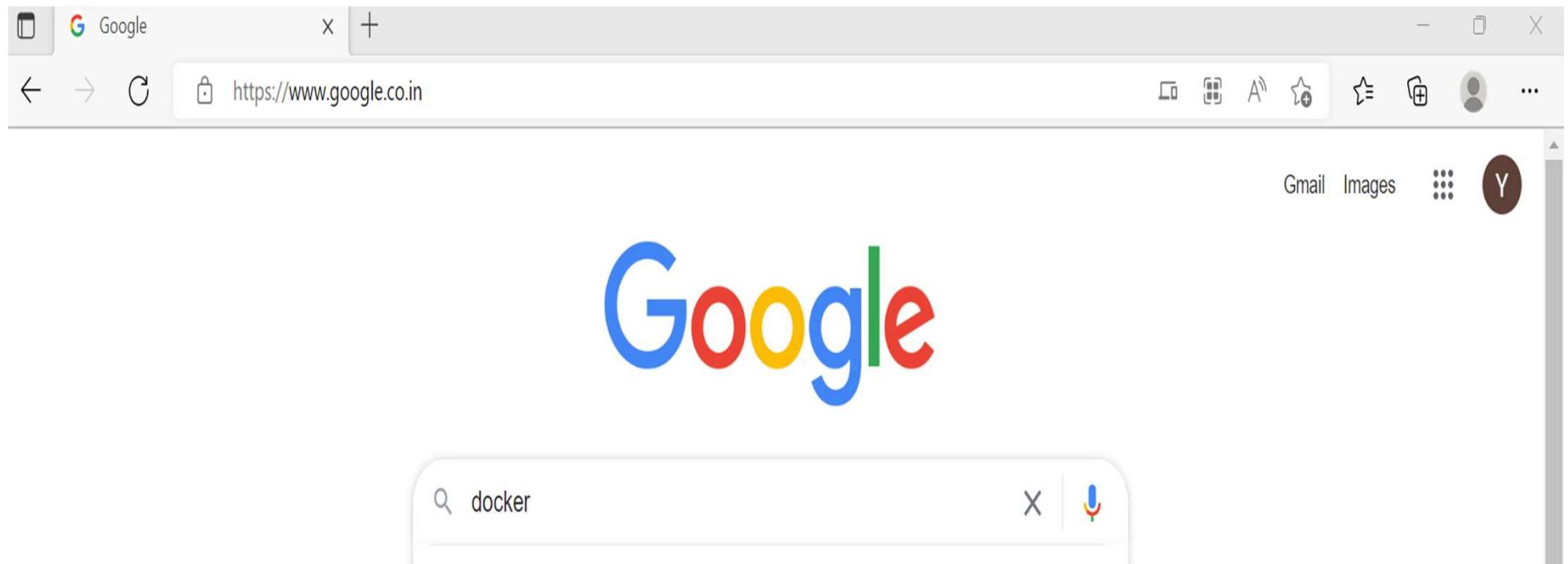
alpine:3.10

**Layer** - linux base image

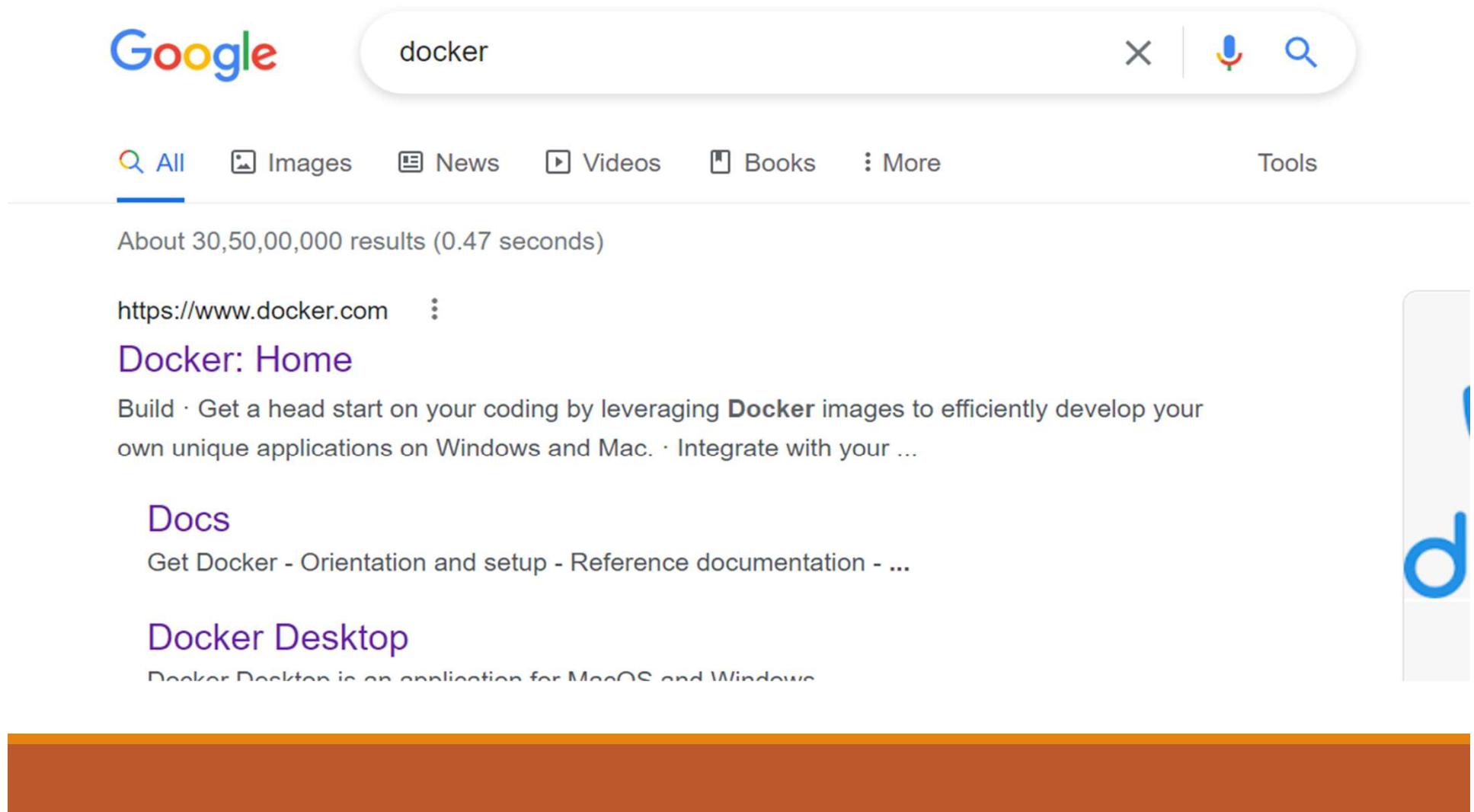


# Docker Installation

GO TO GOOGLE AND TYPE DOCKER



# Click on Docker Home



A screenshot of a Google search results page. The search bar at the top contains the query "docker". Below the search bar, there are several navigation links: "All" (which is underlined in blue), "Images", "News", "Videos", "Books", "More", and "Tools". The main search results section starts with a summary: "About 30,50,00,000 results (0.47 seconds)". The first result is a link to the Docker website: "https://www.docker.com · Docker: Home". The description for this result reads: "Build · Get a head start on your coding by leveraging Docker images to efficiently develop your own unique applications on Windows and Mac. · Integrate with your ...". Below this, there is another link titled "Docs" with the description: "Get Docker - Orientation and setup - Reference documentation - ...". At the bottom of the search results, there is a link titled "Docker Desktop" with the description: "Docker Desktop is an application for macOS and Windows". The entire search results area is set against a white background with black text, except for the orange footer bar at the very bottom.

Google docker

All Images News Videos Books More Tools

About 30,50,00,000 results (0.47 seconds)

<https://www.docker.com> · Docker: Home

Build · Get a head start on your coding by leveraging Docker images to efficiently develop your own unique applications on Windows and Mac. · Integrate with your ...

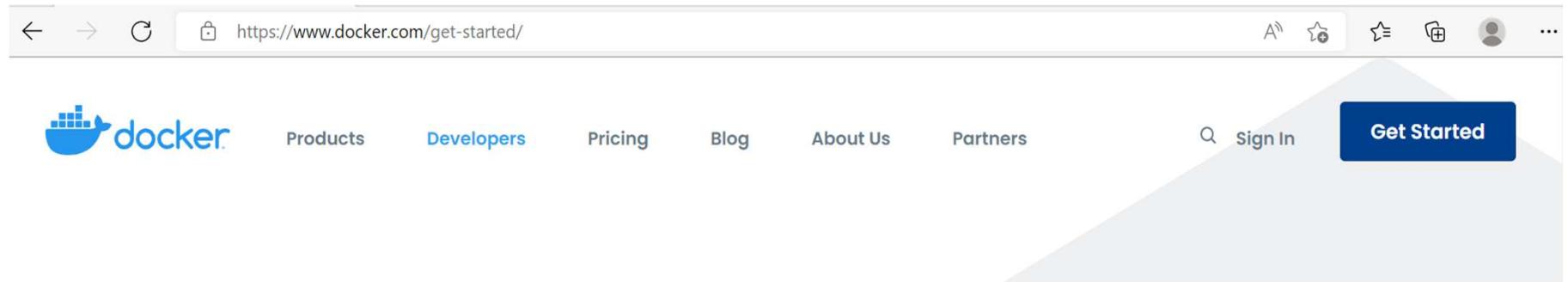
Docs

Get Docker - Orientation and setup - Reference documentation - ...

Docker Desktop

Docker Desktop is an application for macOS and Windows

# Press on GET STARTED



# Click on see Docker Desktop for Windows

The preferred choice for millions of developers that are building containerized apps.

Docker Desktop is an application for MacOS and Windows machines for the building and sharing of containerized applications. Access [Docker Desktop](#) and follow the [guided onboarding](#) to build your first containerized application in minutes.

[See Docker Desktop for Mac](#)

[See Docker Desktop for Windows](#)

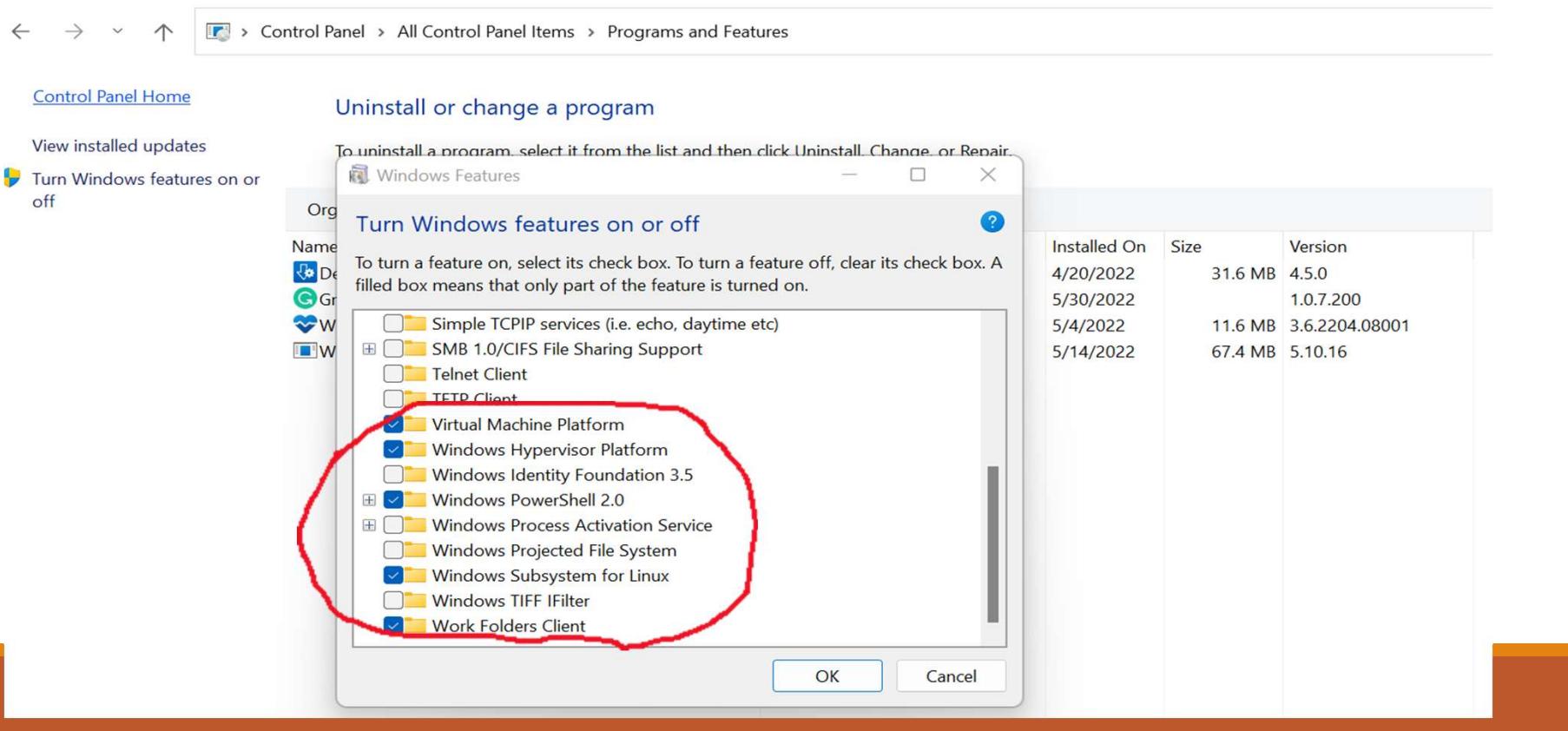
[See Docker Desktop for Linux](#)

# Click on Docker Desktop for Windows to download file

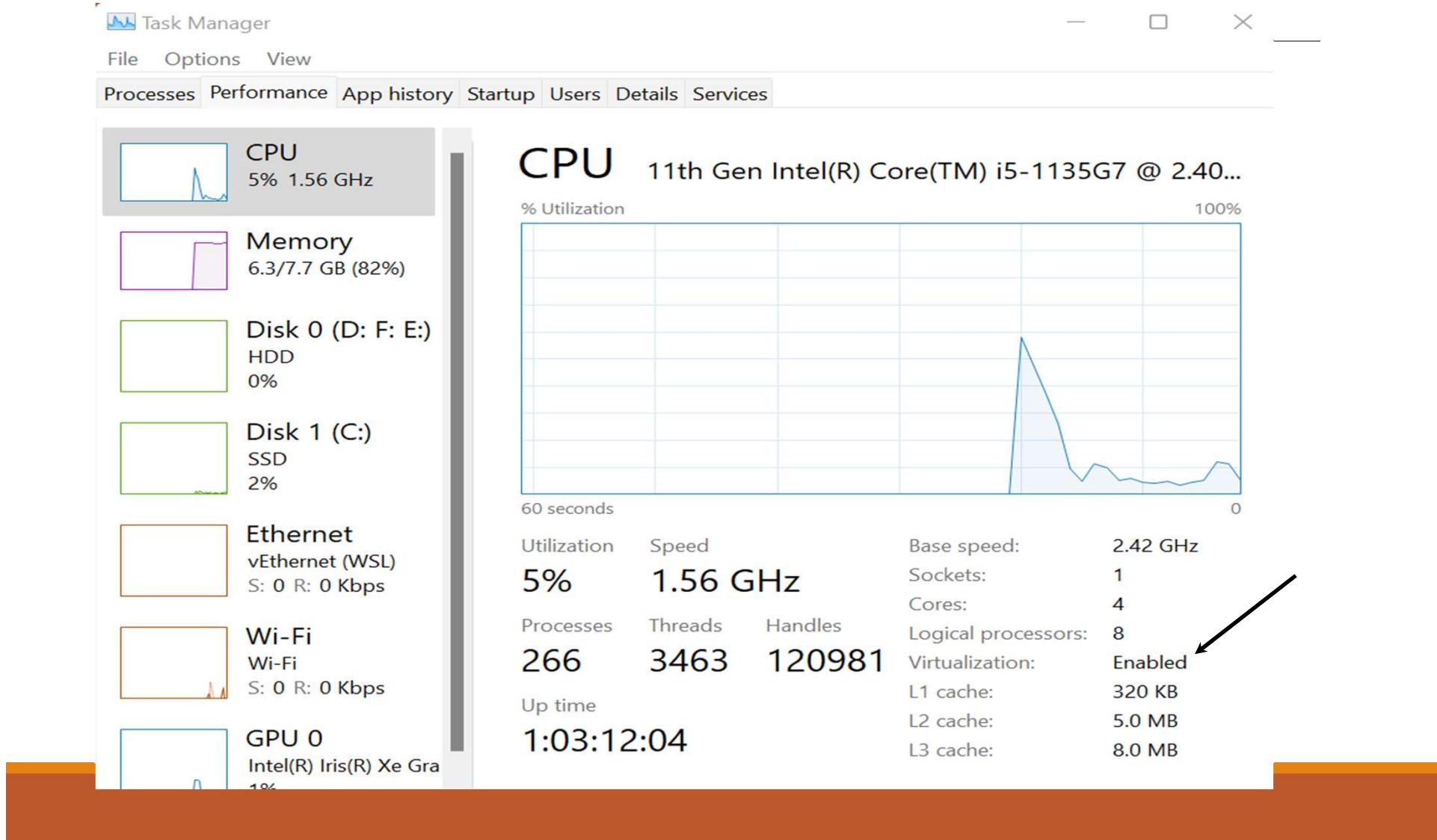
The screenshot shows a web browser window with the URL <https://docs.docker.com/desktop/windows/install/>. The page is titled "Install Docker Desktop on Windows". On the left, there's a sidebar with navigation links for Docker Desktop (Overview, Mac, Windows, Install Docker Desktop for Windows, User manual, Networking, Logs and troubleshooting, Docker Desktop WSL 2 backend), Linux, Dashboard, and Dev Environments (Preview). The main content area has a yellow callout box for "Update to the Docker Desktop terms" and a blue button for "Download Docker Desktop for Windows". A red arrow points to the "Docker Desktop for Windows" button. The right side of the page contains a sidebar with links for "On this page" (System requirements, WSL 2 backend, Hyper-V backend and Windows containers, About Windows containers), "Install Docker Desktop on Windows" (Install interactively, Install from the command line), "Start Docker Desktop" (Quick Start Guide), "Updates", "Uninstall Docker Desktop", and "Where to go next".

# Before installation do the following steps

1. Goto Control panel->Programs and Features->Turn Windows feature ON and OFF and click on it. Now check all given below are enabled



Step 2: Press Ctrl+ALT+delete to goto Task manager, click on Performance and then check Virtualization is Enabled

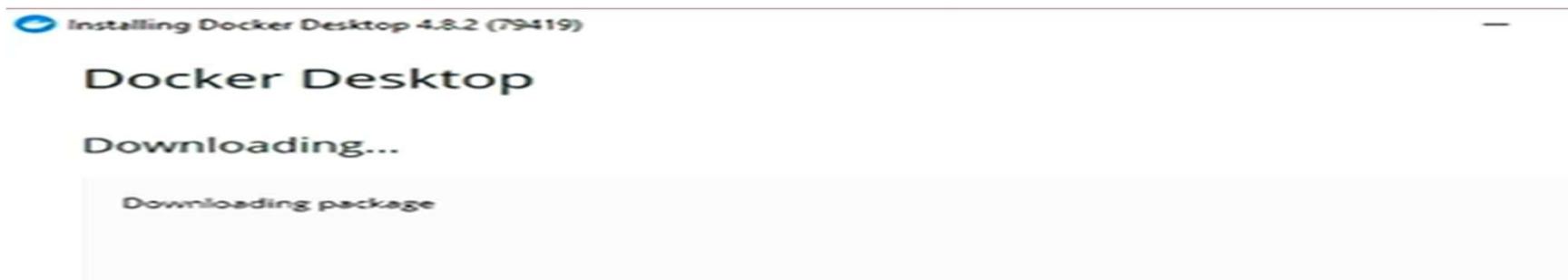


# Click on downloaded Docker Desktop Installer.exe



5 March 2022

# It shows docker packaging



The below window will not appear if WSL2 already enabled



- Use WSL 2 instead of Hyper-V (recommended)
- Add shortcut to desktop

# Docker Desktop 4.8.2

Installation succeeded

You must restart Windows to complete installation.

[Close and restart](#)

If you want to install , first type (install WSL)

then click on [Manual installation steps for older versions of WSL - Microsoft ...](#)

A screenshot of a web browser window showing search results for "install wsl" on Google. The browser has five tabs open: "Developers - Docker", "Docker Desktop for Windo...", "Favorites", "install wsl - Google Search", and "Manual installation steps". The "install wsl - Google Search" tab is active, displaying the search results page. The search bar contains "install wsl". Below the search bar are filters for "All", "Videos", "Images", "News", "Shopping", and "More", with "All" selected. The results section shows approximately 2,09,00,000 results found in 0.42 seconds. The top result is a link to Microsoft Docs titled "Install Linux on Windows with WSL - Microsoft Docs", with a snippet of text from the page: "27-Apr-2022 — Install Windows Subsystem for Linux with the command, `wsl --install`. Use a Bash terminal on your Windows machine run by your preferred Linux ...". Below this is another link to "Windows Command Line · Install on Windows Server". Further down the page, another link is visible: "https://docs.microsoft.com › ... › WSL › Install". A highlighted section of the page contains the text "Manual installation steps for older versions of WSL - Microsoft ...". At the bottom of this section, it says "5 days ago — Step 1 - Enable the Windows Subsystem for Linux · Step 2 - Check ...". The browser interface includes standard navigation buttons (back, forward, refresh) and a URL bar showing the full search query.

# Download WSL2 Linux kernel update package for x64 machines and install

---

## Step 4 - Download the Linux kernel update package

1. Download the latest package:

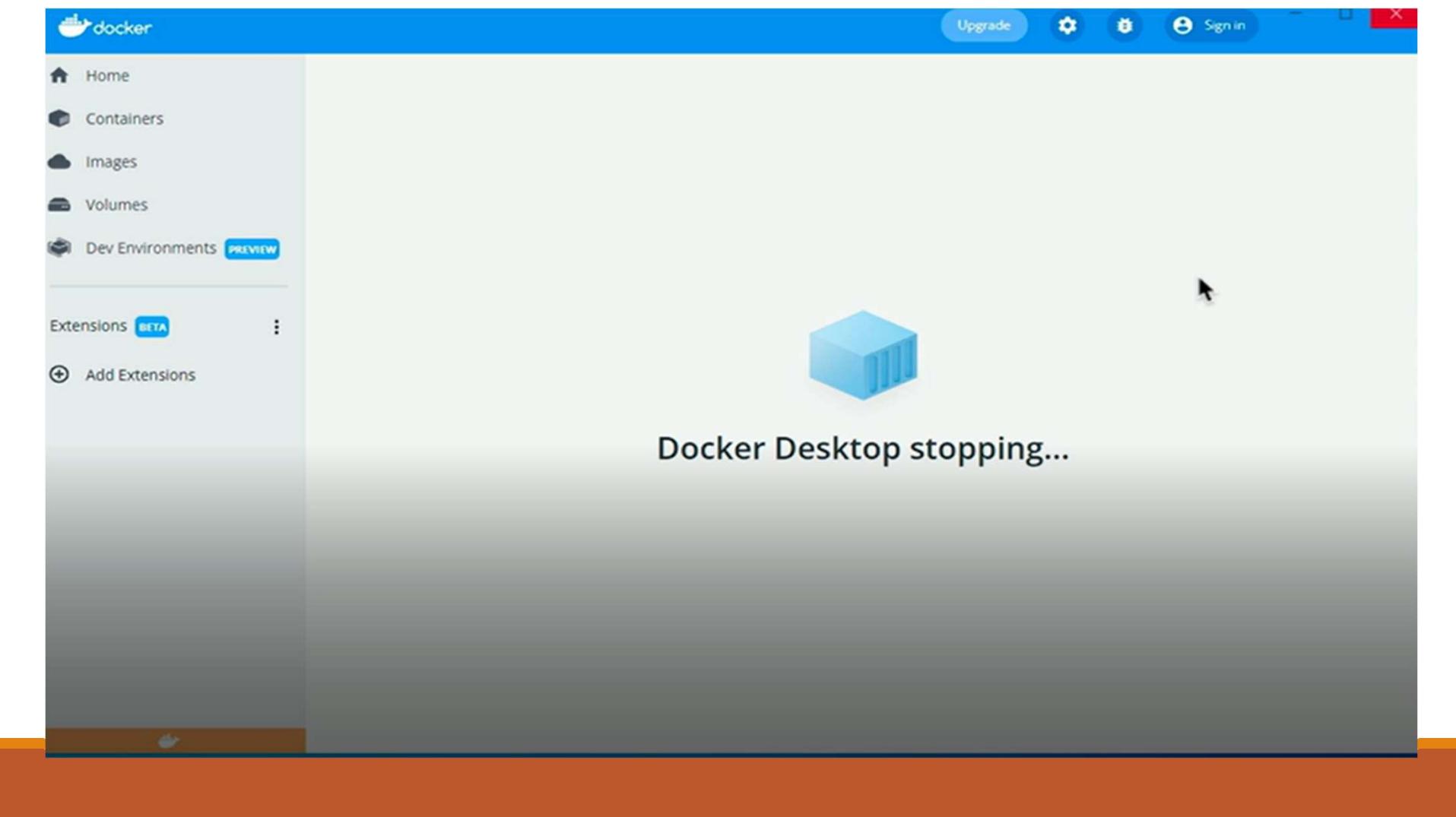
- [WSL2 Linux kernel update package for x64 machines ↗](#)

 **Note**

If you're using an ARM64 machine, please download the [ARM64 package ↗](#) instead. If you're not sure what kind of machine you have, open Command Prompt or PowerShell and enter: `systeminfo | find "System Type"`. **Caveat:** On non-English Windows versions, you might have to modify the search text, translating the "System Type" string. You may also need to escape the quotations for the find command. For example, in German `systeminfo | find '"Systemtyp"'`.

2. Run the update package downloaded in the previous step. (Double-click to run - you will be prompted for elevated permissions, select 'yes' to approve this installation.)

# Next Displays



# Now Clicking Docker on a desktop must display these images

The screenshot shows the Docker desktop application interface. The left sidebar includes links for Home, Containers, Images, Volumes, Dev Environments (with a PREVIEW badge), Extensions (BETA), and Add Extensions. The main content area is titled 'Home' and 'Give Feedback'. It features a 'Featured Images' section with four cards:

- NGINX** nginx:latest **DOCKER OFFICIAL IMAGE**  
An open-source web server, reverse proxy, load balancer and HTTP cache.  
[View on Hub](#) [Run](#)
- Redis** redis:latest **DOCKER OFFICIAL IMAGE**  
An open-source in-memory key-value store that functions as a data structure server.  
[View on Hub](#) [Run](#)
- PostgreSQL** postgres:latest **DOCKER OFFICIAL IMAGE**  
An open-source object-relational transactional ACID-compliant SQL database.  
[View on Hub](#) [Run](#)
- MongoDB** mongo:latest **DOCKER OFFICIAL IMAGE**  
An open source NoSQL document database which uses JSON-like documents with schemata.  
[View on Hub](#) [Run](#)

The right sidebar contains 'Get Started' links for Quick Start Guide and Install Extensions, and a 'Useful Links' section with Release Notes, Documentation, Docker Hub, Docker Roadmap, and DockerCon 2022.

# Download the docker from the link as shown

The screenshot shows a web browser window with two tabs open:

- Inbox (573) - budarajumadhurika**
- docker download - Google Search**

The Google search results page for "docker download" is displayed. The search bar shows "docker download". Below it, the "All" tab is selected, along with other options like Books, Videos, News, Images, and More. The "Tools" button is also visible.

About 17,10,00,000 results (0.48 seconds)

[https://docs.docker.com › desktop › windows › install](https://docs.docker.com/desktop/windows/install) ::

**Install Docker Desktop on Windows**

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

[User manual · WSL 2 backend · Release notes · Logs and troubleshooting](#)

**People also search for**

- [docker download for windows](#)
- [docker download mac](#)
- [docker download linux](#)
- [docker desktop](#)
- [docker hub](#)
- [docker install](#)

[https://www.docker.com › products › docker-desktop](https://www.docker.com/products/docker-desktop) ::

**Docker Desktop**

Docker Desktop is an application for MacOS and Windows machines for the building and sharing of containerized applications and microservices. Docker Desktop ...

[Docker volumes](#) · [Docker Desktop Alternatives](#) · [How Kubernetes works under...](#)

[https://docs.docker.com › get-docker](https://docs.docker.com/get-docker) ::

**Get Docker**

Activate Windows

Go to Settings to activate Windows.

Show all

MS-Office-Pro-Plus....iso

Tomcat-installation.pptx

docker installatio....pptx

# Click on the following link to download the executable file

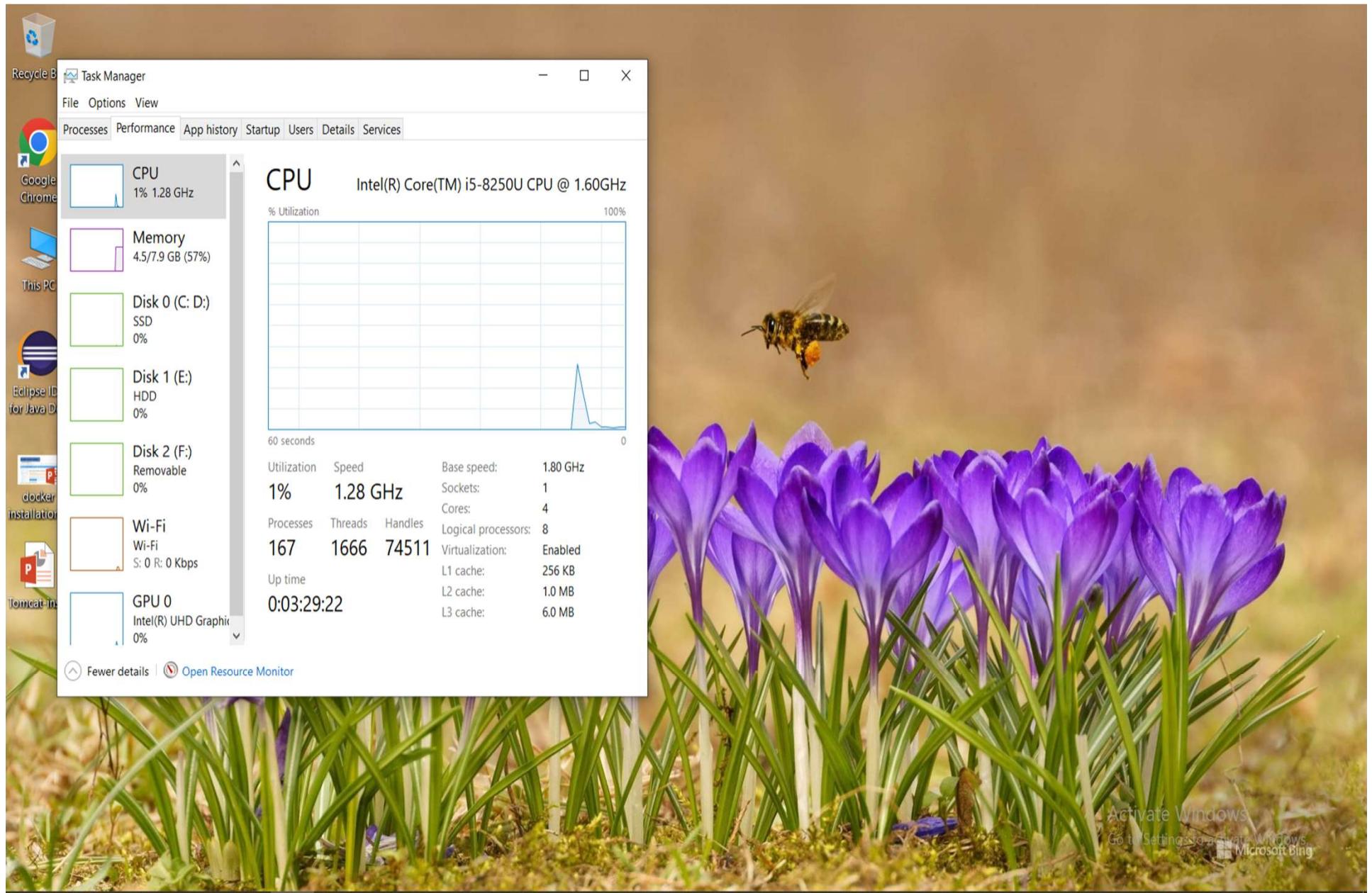
The screenshot shows a Microsoft Edge browser window with the following details:

- Title Bar:** Shows two tabs: "Inbox (573) - budarajumadhurika" and "Install Docker Desktop on Windo".
- Address Bar:** Displays the URL: "docs.docker.com/desktop/windows/install/".
- Toolbar:** Includes standard browser controls (Back, Forward, Stop, Refresh) and a search bar.
- Header:** Features the "docker docs" logo, a search bar ("Search the docs"), and navigation links: Home, Guides, Manuals, Reference, Samples.
- Breadcrumbs:** Shows the current path: "/ Manuals / Docker Desktop / Windows / Install Docker Desktop for Windows".
- Left Sidebar:** A navigation menu under "Docker Desktop" with sections: Overview, Mac, Windows, User manual, Networking, Logs and troubleshooting, Docker Desktop WSL 2 backend, Linux, Dashboard, Dev Environments (Preview), Extensions (Beta), and Extensions SDK (Beta). The "Windows" section is currently selected.
- Main Content:**
  - Section Header:** "Install Docker Desktop on Windows" with an estimated reading time of "10 minutes".
  - Warning:** "Update to the Docker Desktop terms" regarding commercial use.
  - Text:** 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.
  - Section Header:** "Download Docker Desktop for Windows".
  - Call-to-Action Button:** A large blue button labeled "Docker Desktop for Windows".
- Right Sidebar:** A list of links related to Docker Desktop for Windows, including: Edit this page, Request docs changes, System requirements, WSL 2 backend, Hyper-V backend and Windows containers, About Windows containers, Install Docker Desktop on Windows, Install interactively, Install from the command line, Start Docker Desktop, Quick Start Guide, Updates, Uninstall Docker Desktop, and Activate Windows.
- Bottom Taskbar:** Shows three open files: "MS-Office-Pro-Plus....iso", "Tomcat-installation.pptx", and "docker installatio....pptx".
- Bottom Right:** A "Where to go next" section with a "Show all" button and a "Go to Settings to activate Windows" link.

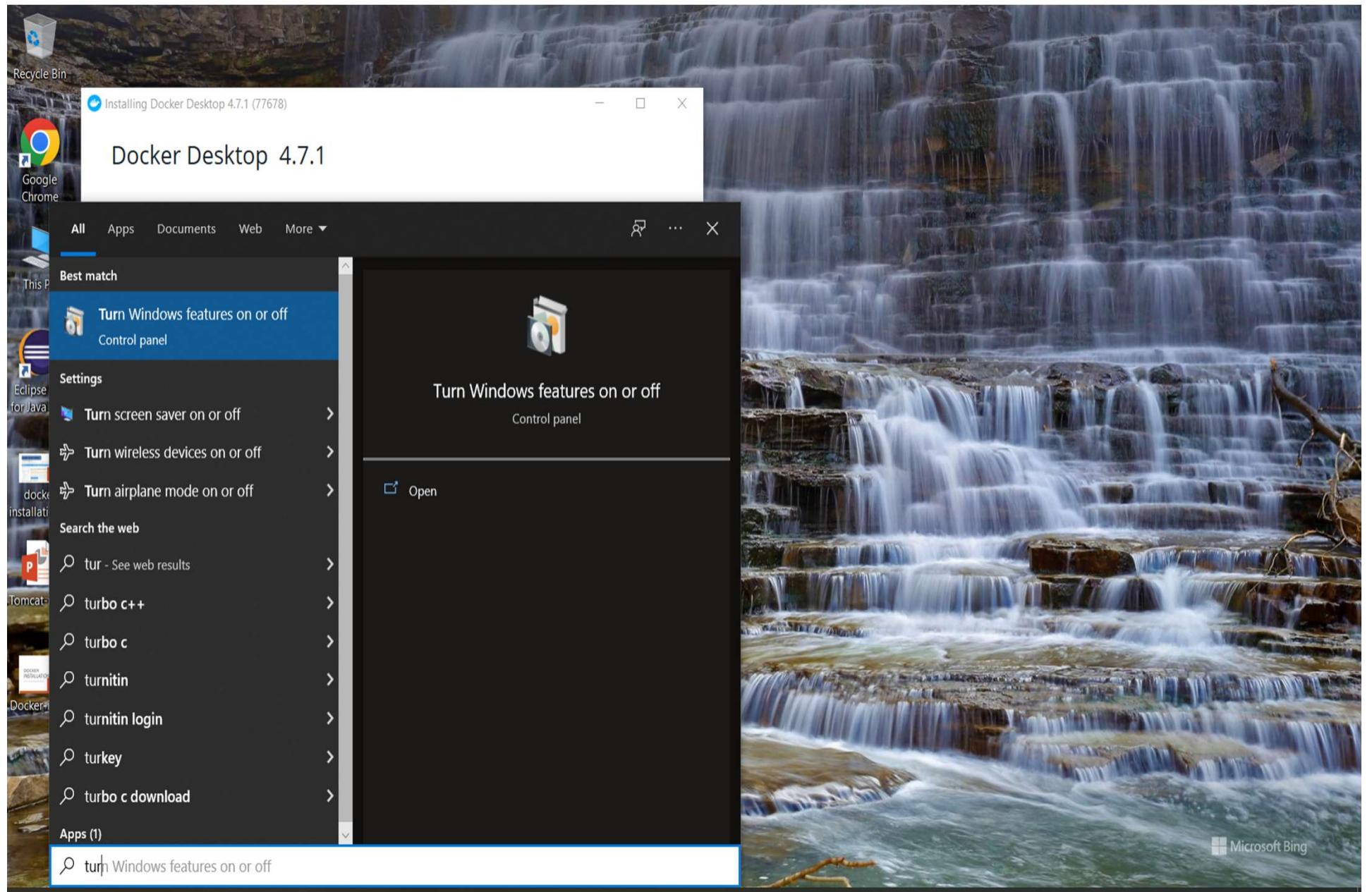
Type winver in Run to check the prerequisites - OS version- Windows 10 pro preferably



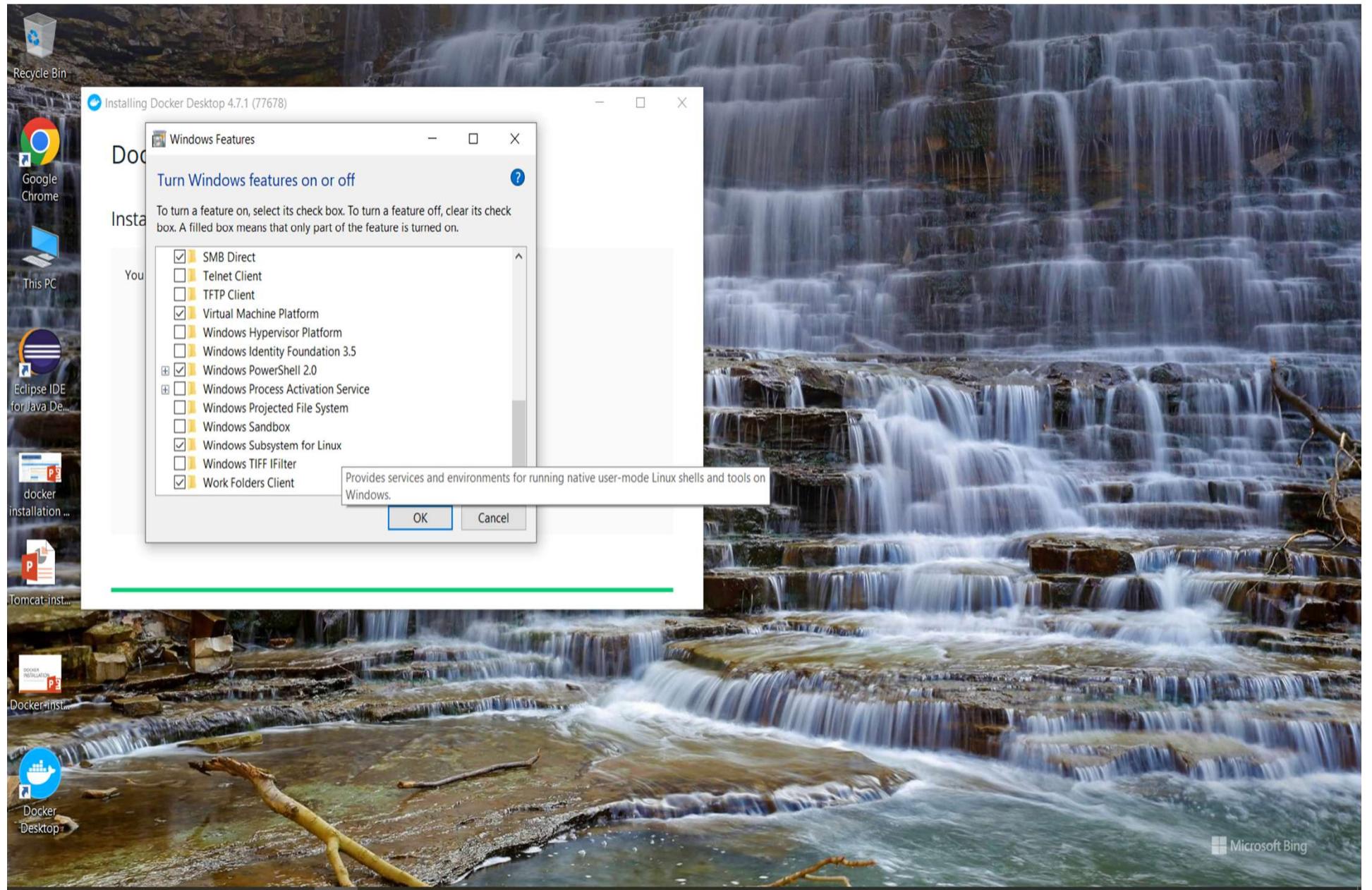
# Make sure Virtualization is enabled in the system, we can check it in task manager as shown



# Click on turn windows features on or off in start



# Make sure to check the Virtual Machine Platform and Windows subsystem for Linux checkboxes and click ok



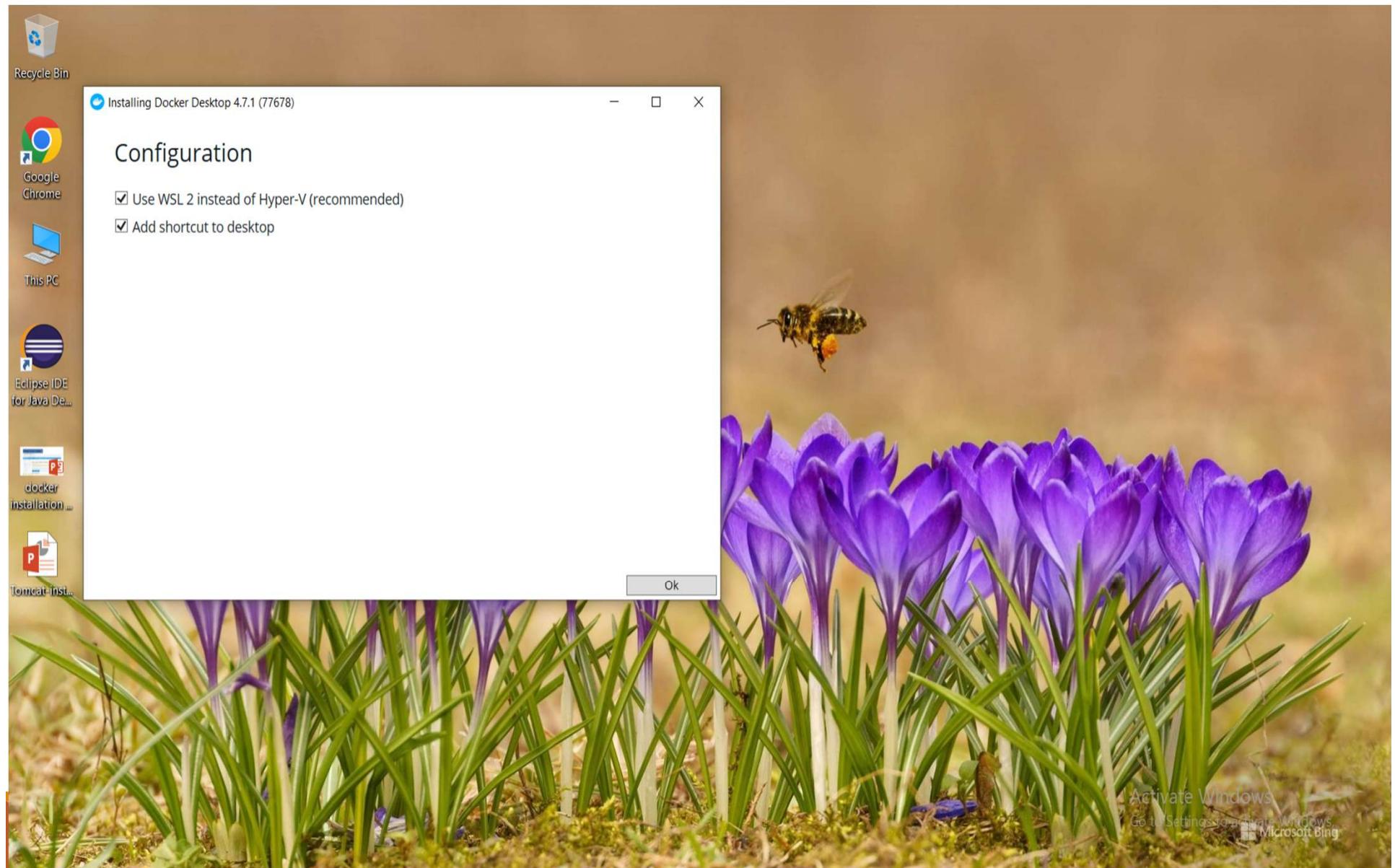
# Download Docker for Windows

The screenshot shows a Windows File Explorer interface. The left sidebar contains a 'Quick access' section with links to Desktop, Downloads, Documents, Pictures, Music, Videos, OneDrive, This PC, and Network. Below these are local disk drives: Local Disk (C:), Local Disk (D:), New Volume (E:), Sony\_16GR (F:), DVD Drive (G:), Sony\_16GR (F:), and Madhu-Backup. The 'Downloads' folder is currently selected. The main pane displays two files from today: 'MS-Office-Pro-Plus-2019-x64' (Disc Image File) and 'Git-2.36.1-64-bit' (Application). Under 'Earlier this month', the 'Docker Desktop Installer' is listed, which has been selected and is highlighted with a blue border. At the bottom left, it says '3 items'. On the right side, there is an 'Activate Windows' message with a link to 'Go to Settings to activate Windows.' and a small icon.

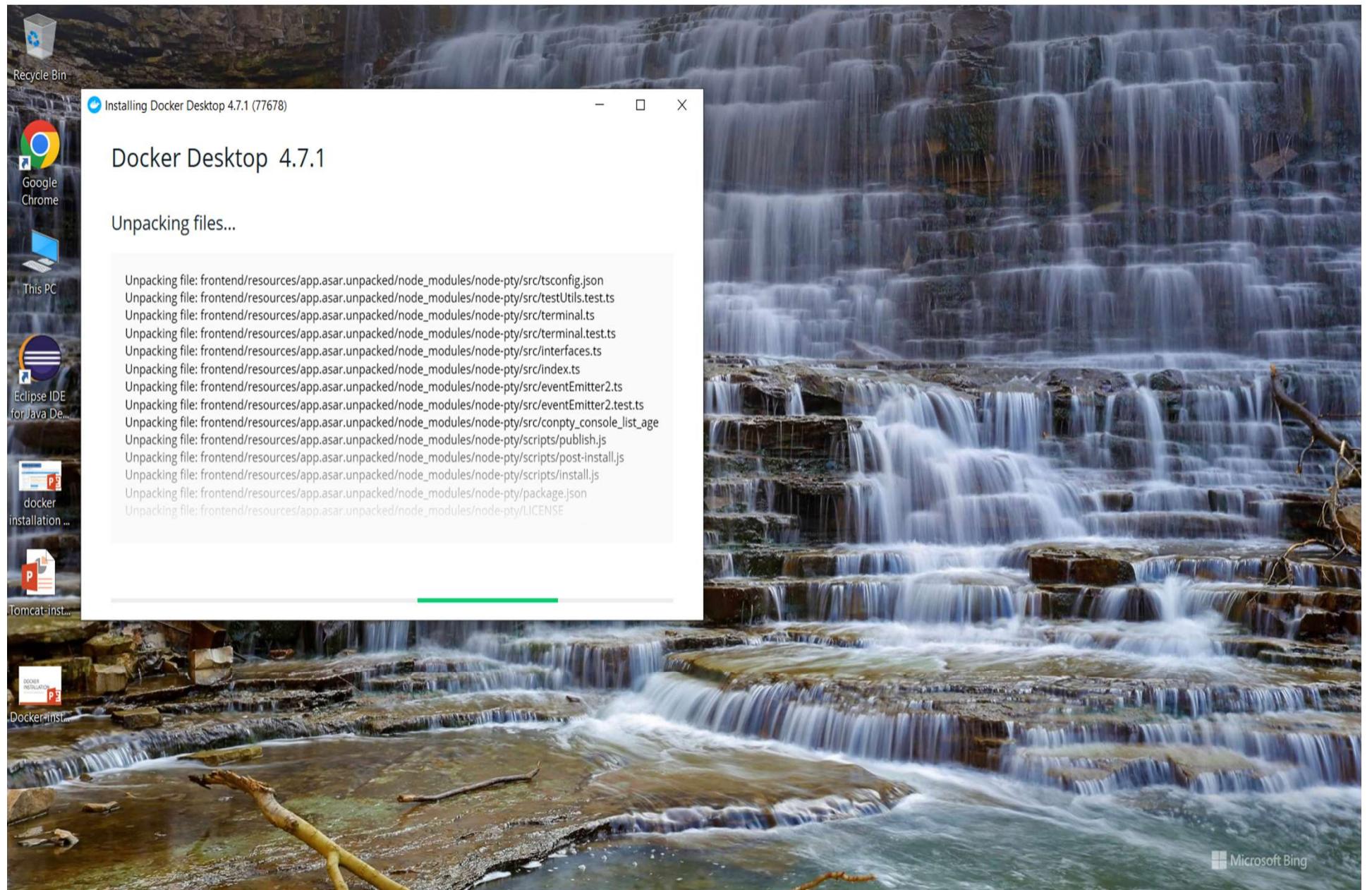
File	Modified	Type	Size
MS-Office-Pro-Plus-2019-x64	21-05-2022 13:23	Disc Image File	20,52,096 ...
Git-2.36.1-64-bit	21-05-2022 11:40	Application	48,475 KB
Docker Desktop Installer	06-05-2022 14:52	Application	5,11,323 KB

Activate Windows  
Go to Settings to activate Windows.

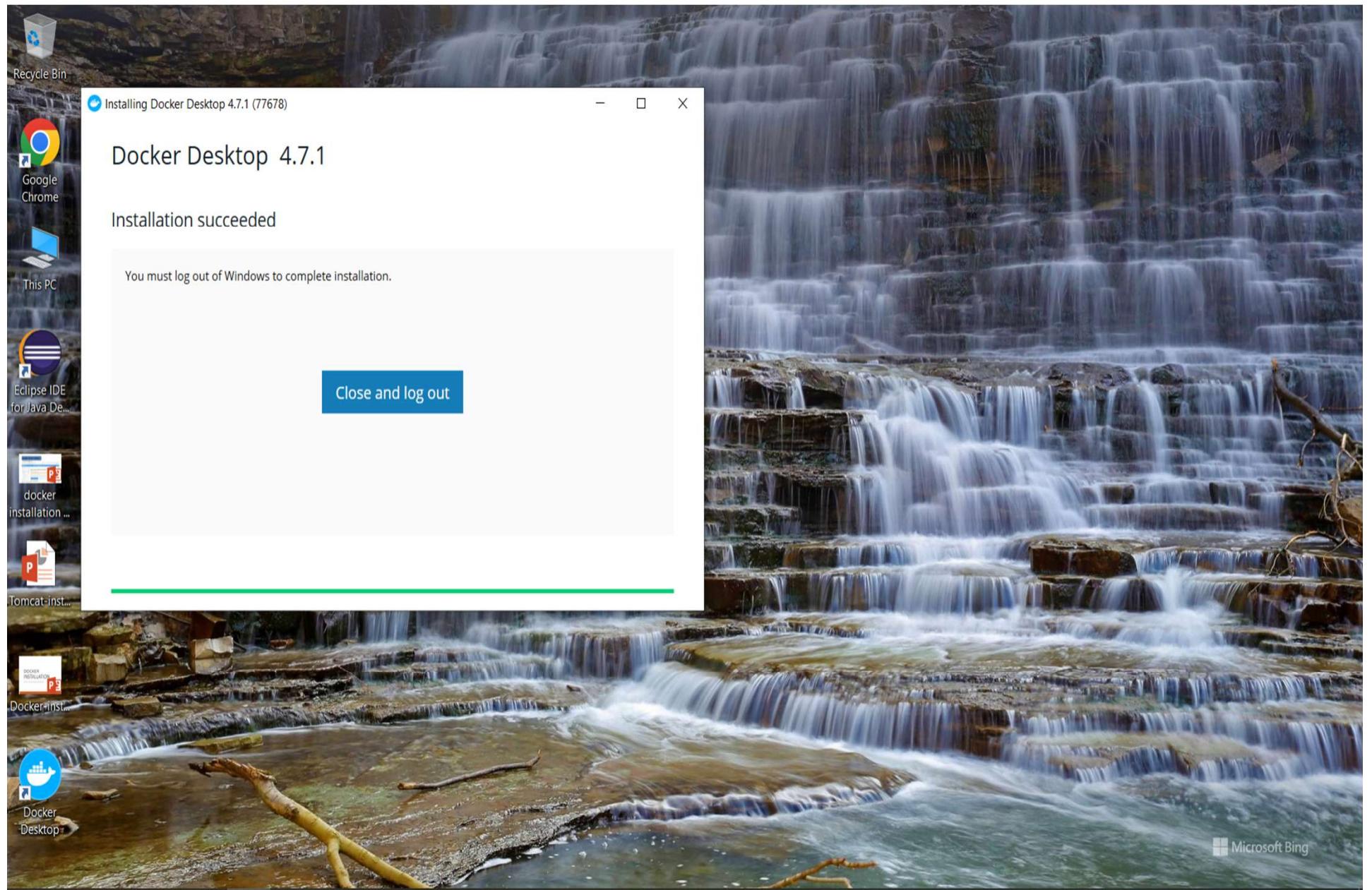
Double click the executable file and click ok as shown, make sure the following check boxes are selected



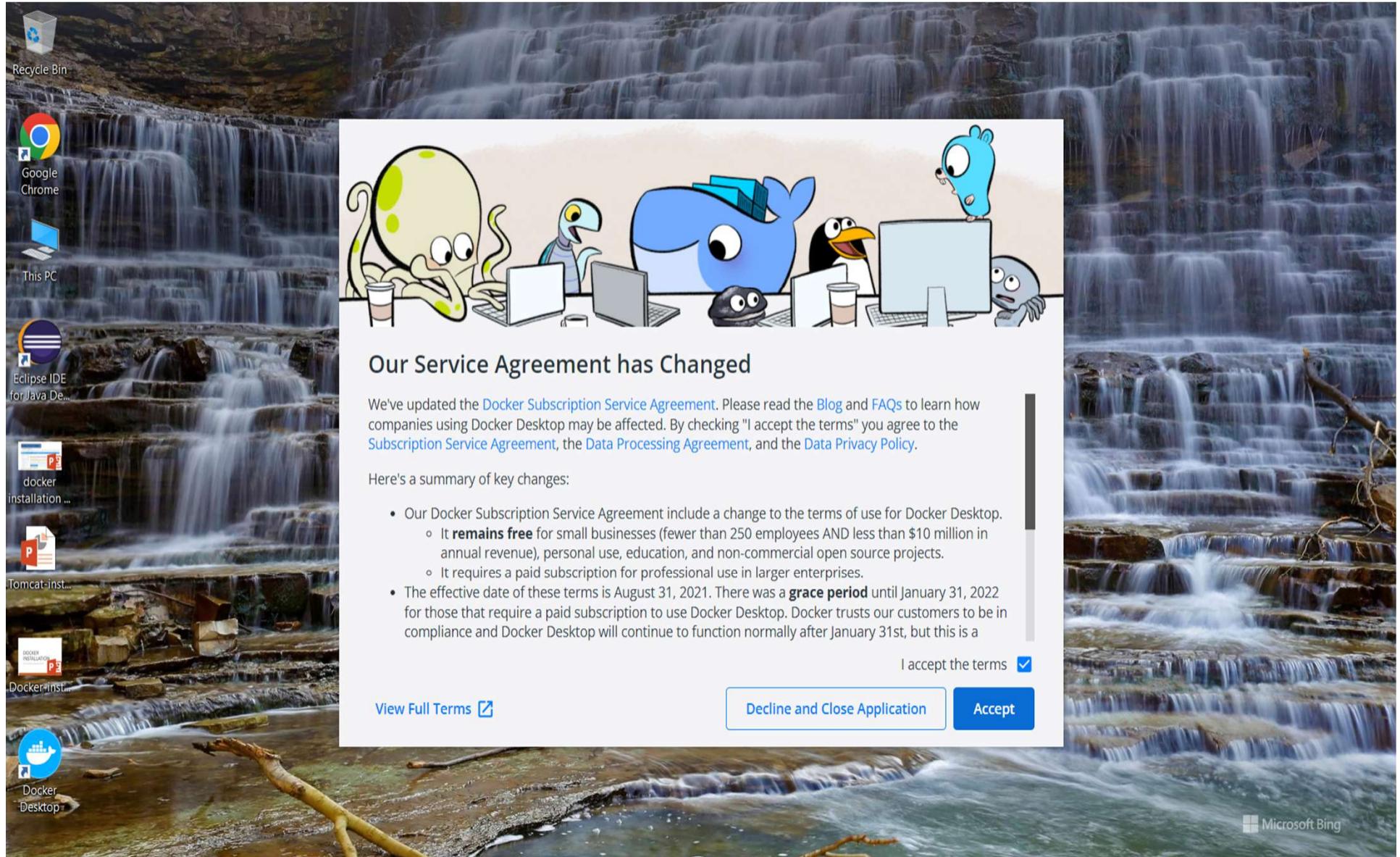
# The download has started



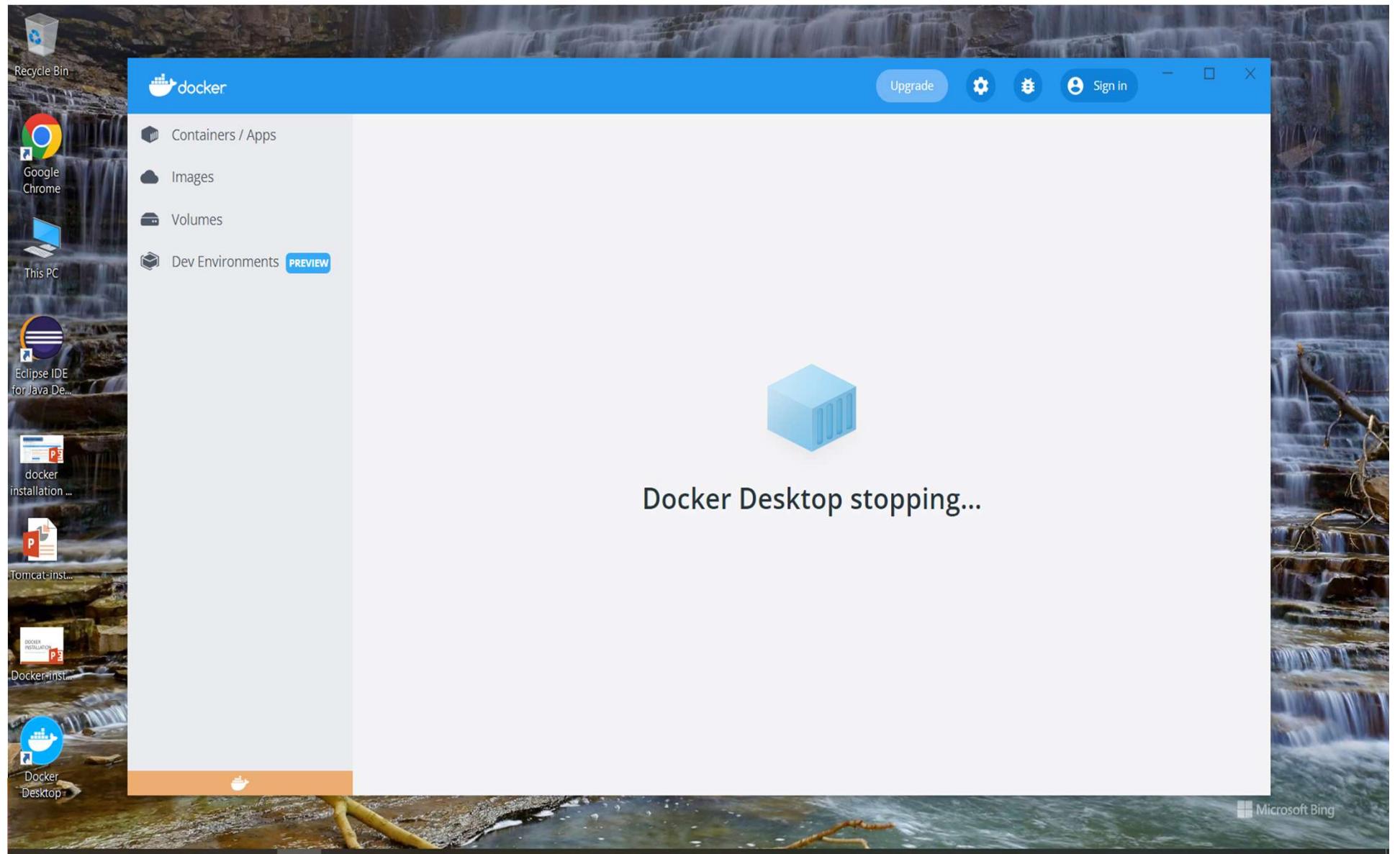
# Click on close and log out



# Accept the license -> click Accept



# The local Docker Hub is created



# Download WSL2 as shown

Downloads

Name Date modified Type Size

MS-Office-Pro-Plus-2019-x64 21-05-2022 13:23 Disc Image File 20,52,096 ...

Git-2.36.1-64-bit 21-05-2022 11:40 Application 48,475 KB

wsl\_update\_x64 21-05-2022 09:39 Windows Installer ... 16,704 KB

Earlier this month (1)

Docker Desktop Install

Item type: Windows Installer Package  
Authors: Microsoft Corporation  
Title: Installation Database  
Subject: Windows Subsystem for Linux Update  
This installer database contains the logic and data required to install Windows Subsystem for Linux Update.  
Date modified: 21-05-2022 09:39  
Size: 16.3 MB

This PC

Downloads

Music Pictures Videos Local Disk (C:) Local Disk (D:) New Volume (E:) Network

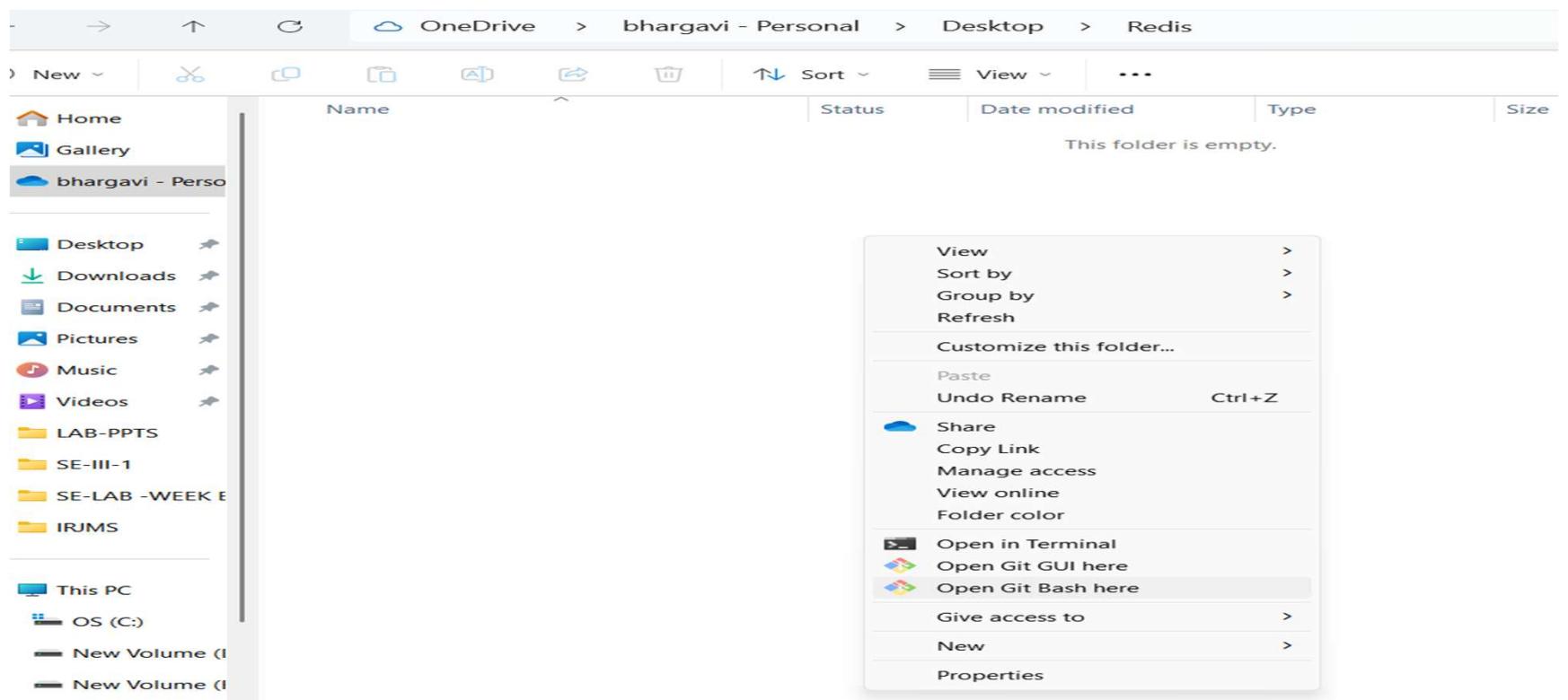
4 items

Docker Desktop - Install WSL 2 kernel update

**WSL 2 installation is incomplete.**

The WSL 2 Linux kernel is now installed using the latest version. Please click the link and follow the instructions at <https://aka.ms/wsl2kernel>.

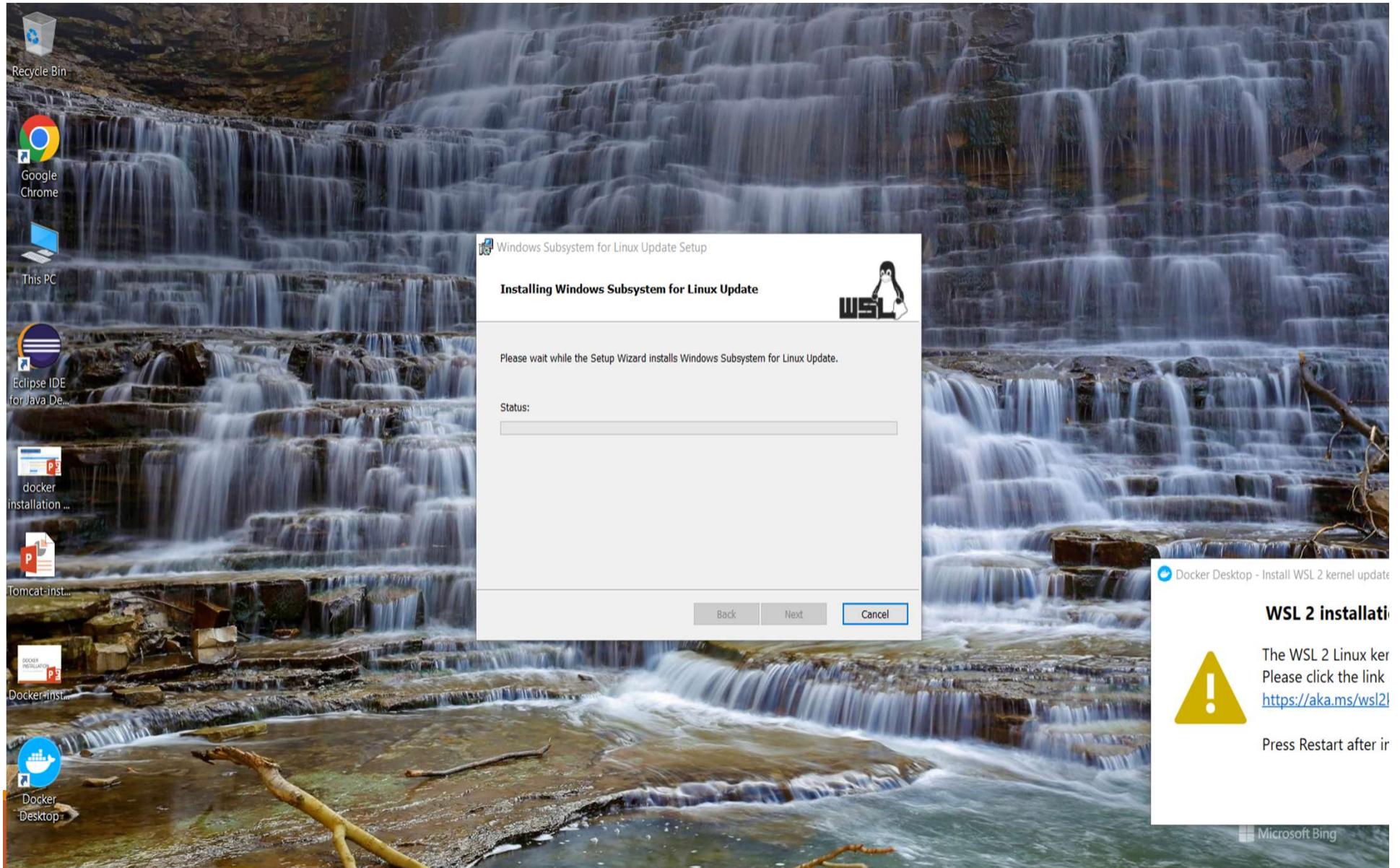
Press Restart after installing the Linux kernel.



# Double click the executable file and click next as shown



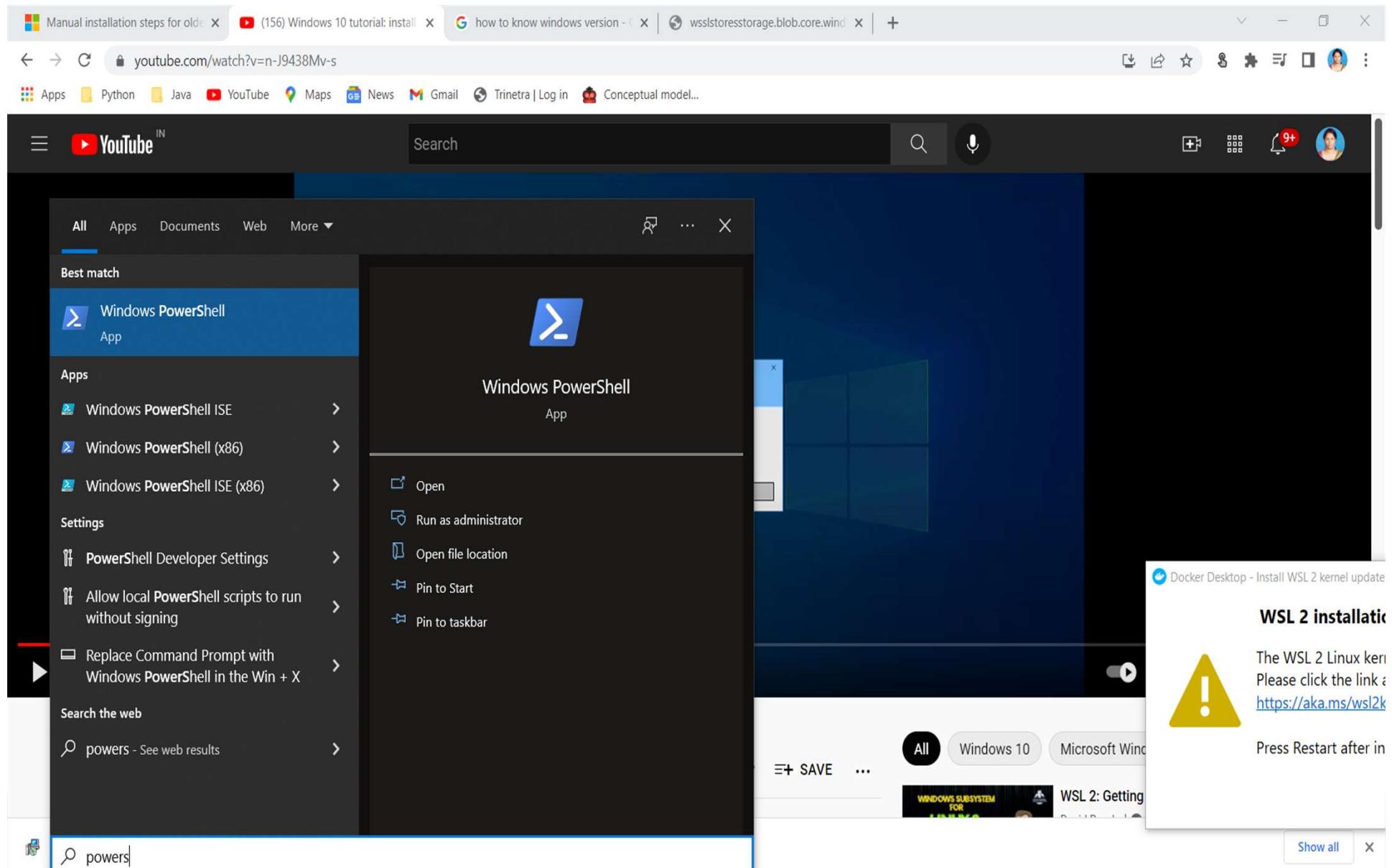
# Installation of WSL2 begins



# Once the installation is completed click on Finish



# Open Windows Powershell as shown from Start



# Run as administrator

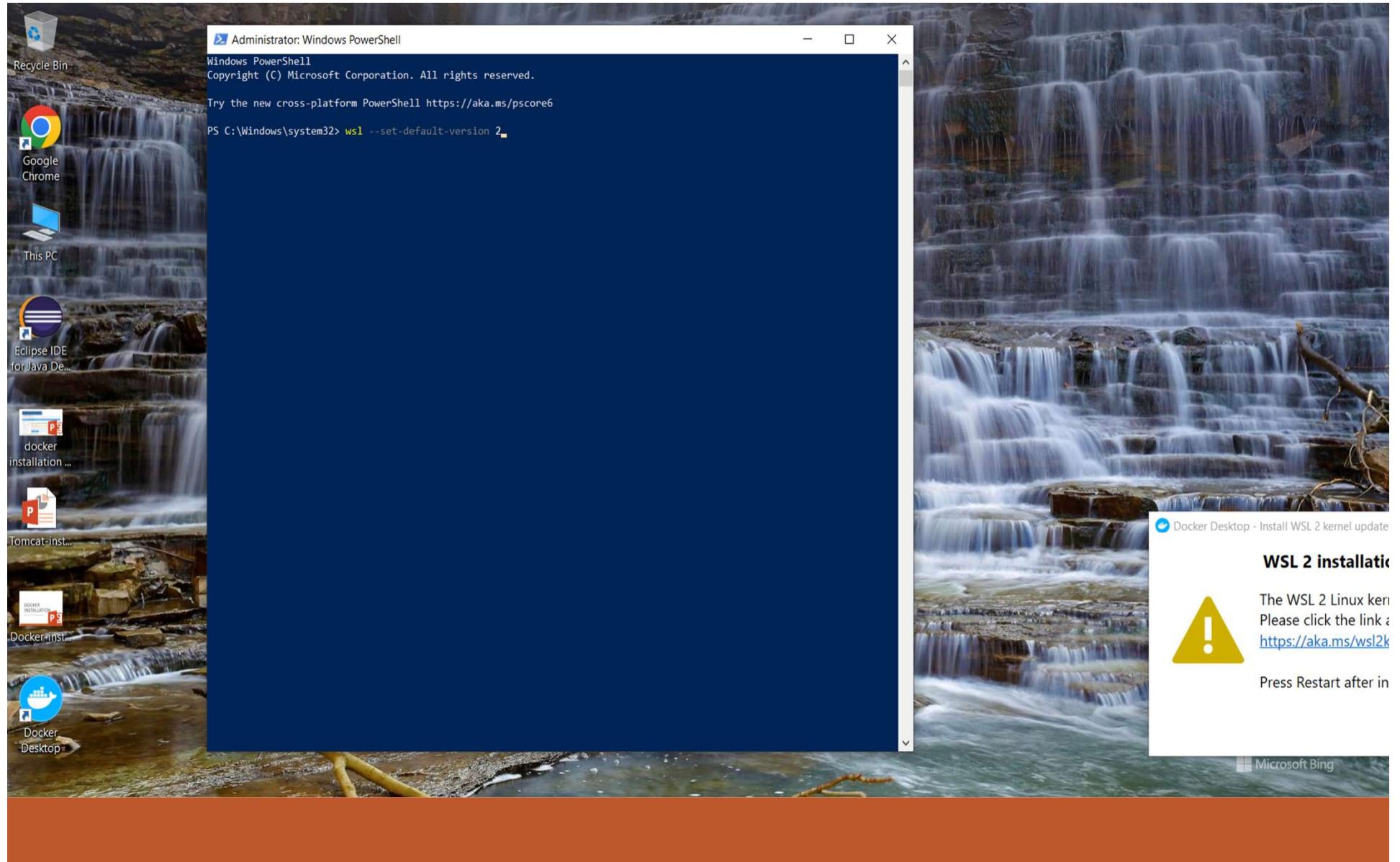
The screenshot shows a Windows desktop environment with a PowerPoint presentation titled "Docker-installation - PowerPoint" open in the foreground. The presentation has 19 slides, with slide 19 currently selected. The search bar at the bottom of the screen shows the query "powershe".

The desktop background features a dark theme with a yellow warning icon in the bottom right corner. The icon contains the text "WSL 2 installatio" and "The WSL 2 Linux keri Please click the link : <https://aka.ms/wsl2k>". Below the icon, it says "Press Restart after in".

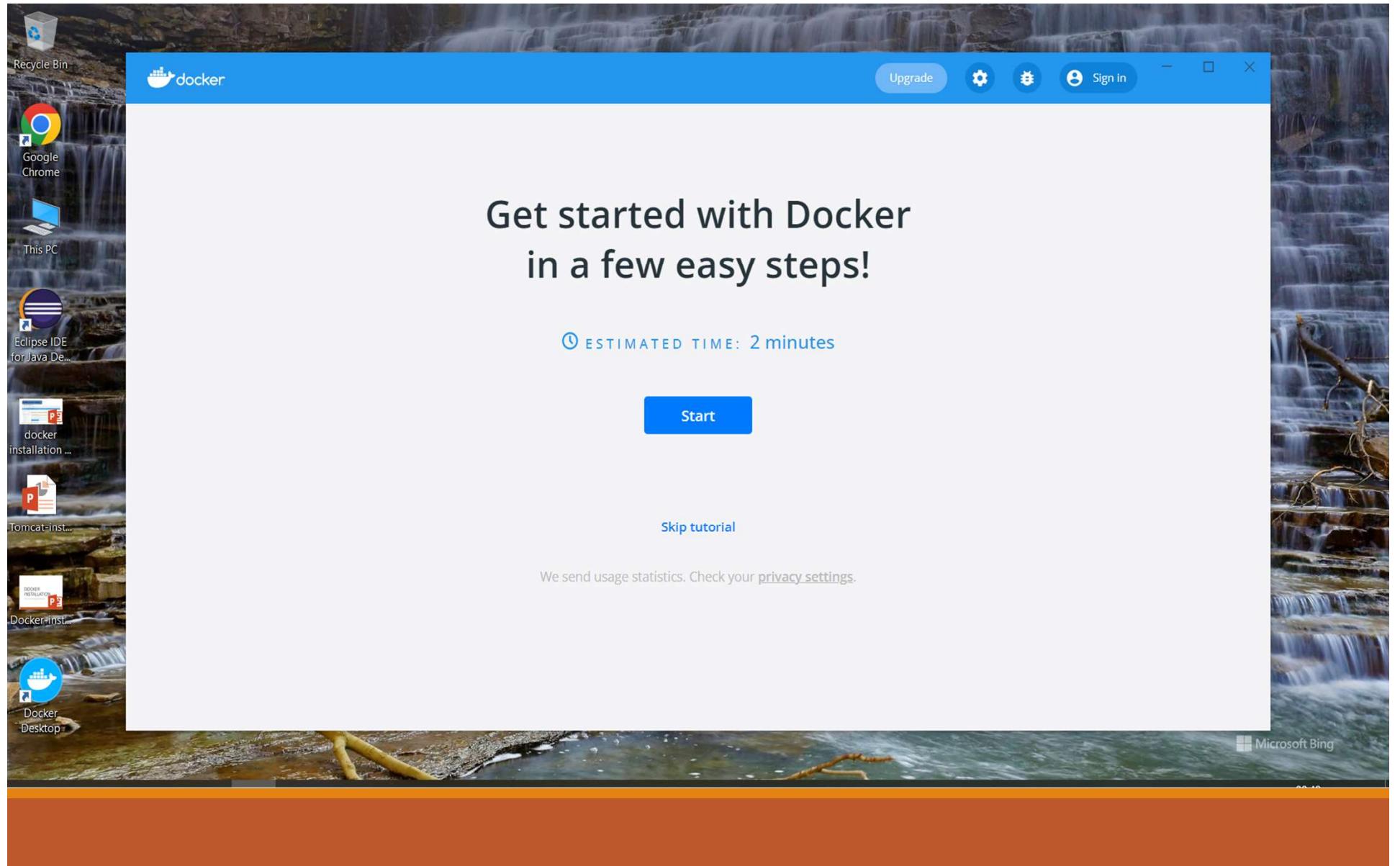
A context menu is open over a "Windows PowerShell" app entry in the Start Menu search results. The menu includes options like "Run as administrator", "Open file location", "Pin to Start", "Pin to taskbar", and "Open".

The top of the screen shows the Windows taskbar with icons for File Explorer, Edge, and other applications. The system tray shows a battery icon and a network connection icon.

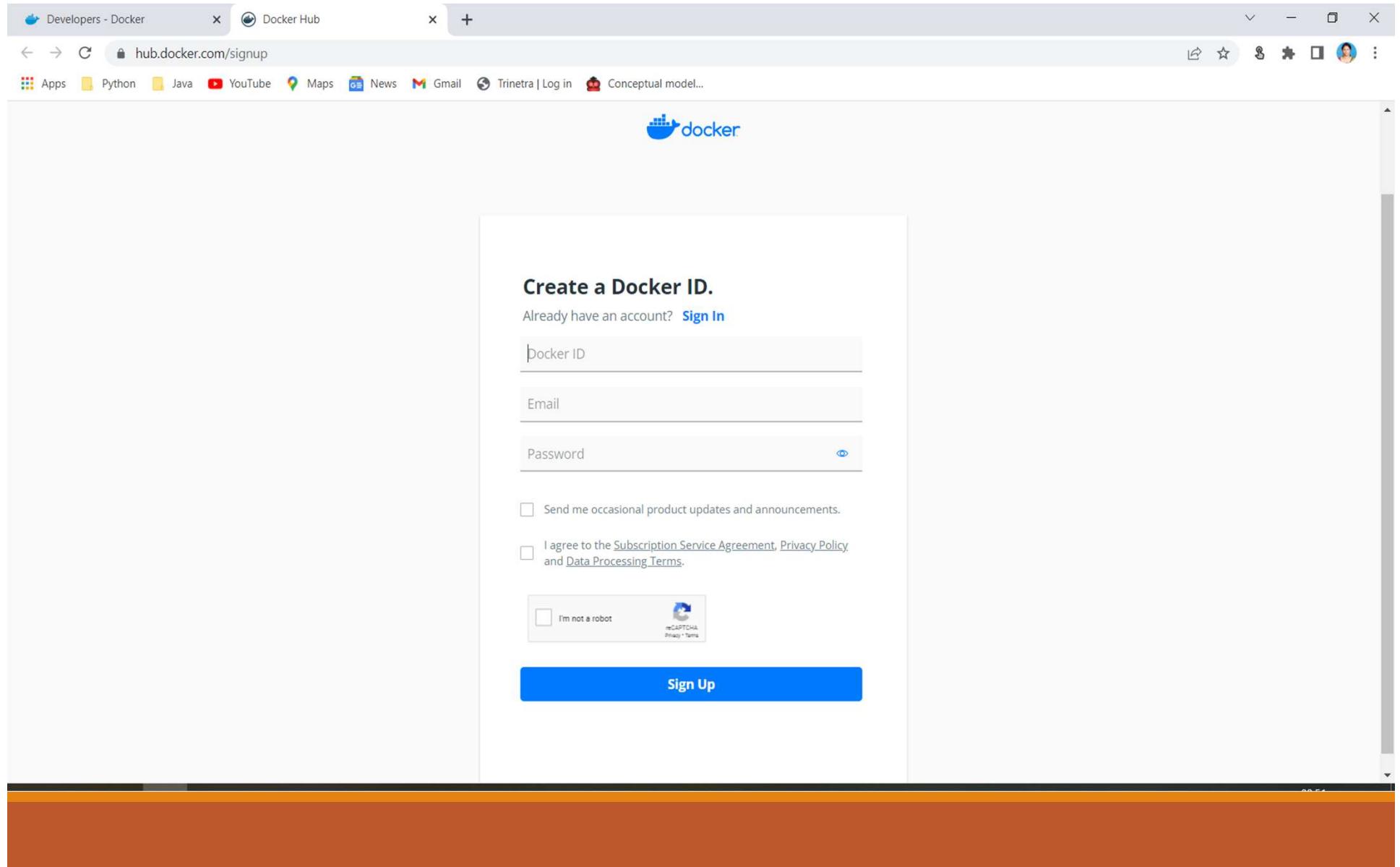
# Type the following command, to set WSL as default version



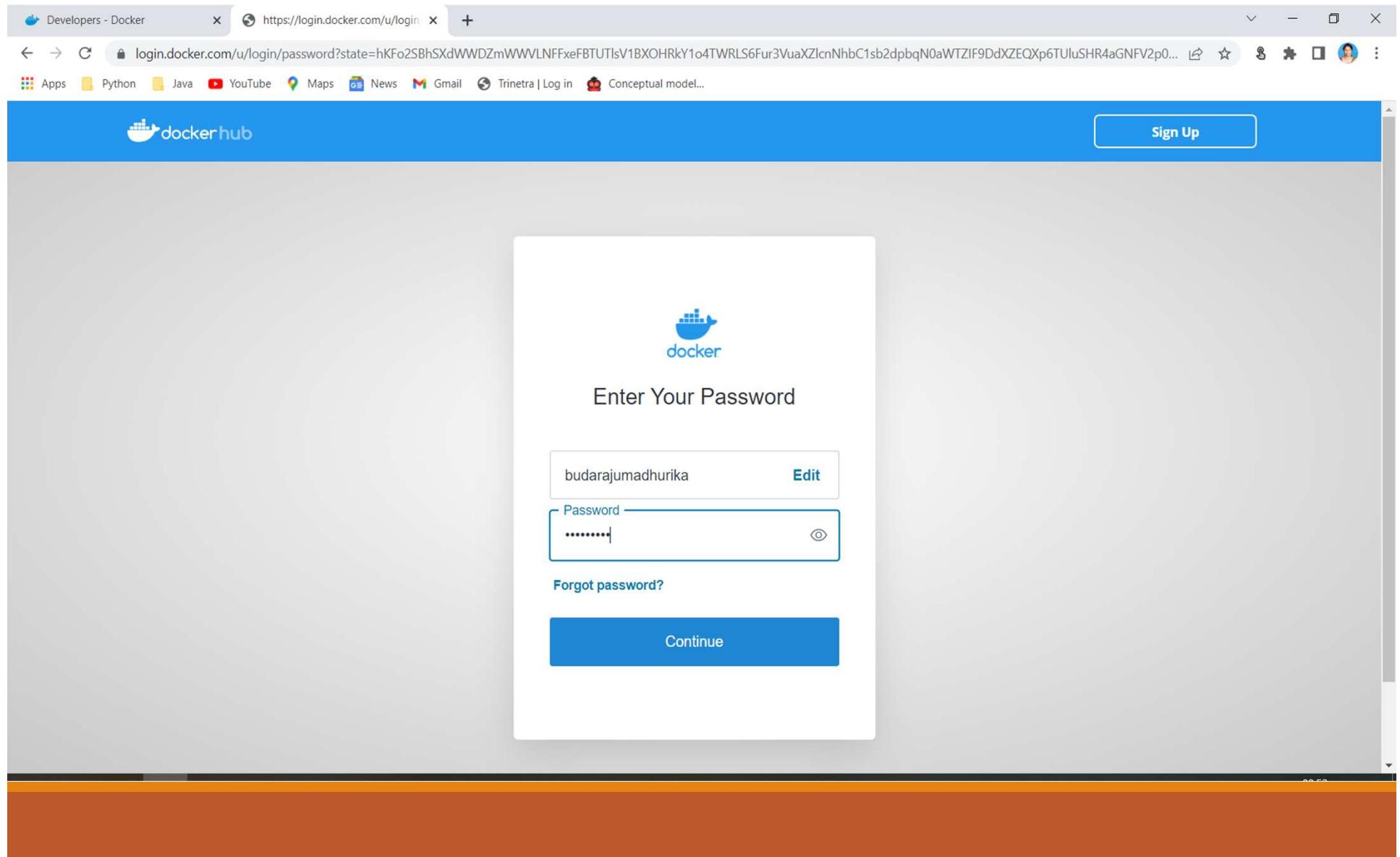
# Now we can start the Docker



# Click on hub.docker.com and signup as shown



# Sign In into Docker Hub



Click on personal-> continue with free

The screenshot shows the Docker Hub 'Choose a Plan' page. It features four subscription plans:

- Personal**: \$0 per month. Includes Docker essentials for individual developers. A 'Continue with Free' button is available.
- Pro**: \$5 per month. Extends Docker capabilities for individual developers. A 'Buy Now' button is available.
- Team** (highlighted with a blue border): \$7 per user/month (minimum 5 seats). Ideal for teams, including enhanced collaboration and security. A 'Buy Now' button is available.
- Business**: \$21 per user/month (only available with an annual subscription). Ideal for medium and large businesses. A 'Buy Now' button is available.

Each plan card includes a brief description, a price, and a 'Buy Now' or 'Continue with Free' button. The 'Team' plan is labeled as a 'DEVELOPER FAVORITE'.

# Go to gmail and verify your account

The screenshot shows a web browser window with two tabs: "Developers - Docker" and "Docker Hub". The "Docker Hub" tab is active and displays the URL "hub.docker.com/verify-email". The page content is as follows:

Please check your inbox to verify the email associated with this account. You won't be able to create a repository or configure your Docker Hub without verifying your email address.

Missed DockerCon 2022? [Watch now on-demand.](#)

**docker hub**  Explore Repositories Organizations Help ▾ Upgrade  budarajumadhuri... ▾



**Please verify your email address**

Great! You're almost there. Before you can create a repository or configure Docker Hub, you'll need to verify your email address.

We've sent a verification email to  
[budarajumadhurika@gmail.com](mailto:budarajumadhurika@gmail.com).

If you didn't receive the email, try [resending it](#) or check your [email settings](#).

# We have successfully logged in into the DockerHub

The screenshot shows a web browser window with the Docker Hub homepage loaded. The address bar displays 'hub.docker.com'. The top navigation bar includes links for 'Explore', 'Repositories', 'Organizations', 'Help', and a user profile for 'budarajumadhuri...'. A search bar at the top says 'Search for great content (e.g., mysql)'. A prominent blue banner in the center says 'Welcome to Docker' and 'Download the desktop application', with a 'Download for Windows' button. Below the banner, it says 'Also available for Mac and Linux'. At the bottom of the page, there are three cards: 'Create a Repository', 'Docker Hub Basics', and 'Language-Specific Guides'. A large orange footer bar at the bottom says 'Access the world's largest library of container images'.

Missed DockerCon 2022? [Watch now on-demand.](#)

hub.docker.com

Apps Python Java YouTube Maps News Gmail Trineta | Log in Conceptual model...

dockerhub

Search for great content (e.g., mysql)

Explore Repositories Organizations Help

Upgrade budarajumadhuri...

Welcome to Docker

Download the desktop application

Download for Windows

Also available for [Mac](#) and [Linux](#)

Create a Repository

Push container images to a repository on Docker Hub.

Docker Hub Basics

Watch the guide on how to create and push your first image into a Docker Hub repository.

Language-Specific Guides

Learn how to containerize language-specific applications using Docker.

Access the world's largest library of container images

# As of now, there are no containers and Images in Docker Hub

The screenshot shows a Microsoft PowerPoint slide with the title "docker installation & working - PowerPoint". The slide content is as follows:

- Containers / Apps** (highlighted)
- Images
- Volumes
- Dev Environments PREVIEW

**No containers running**

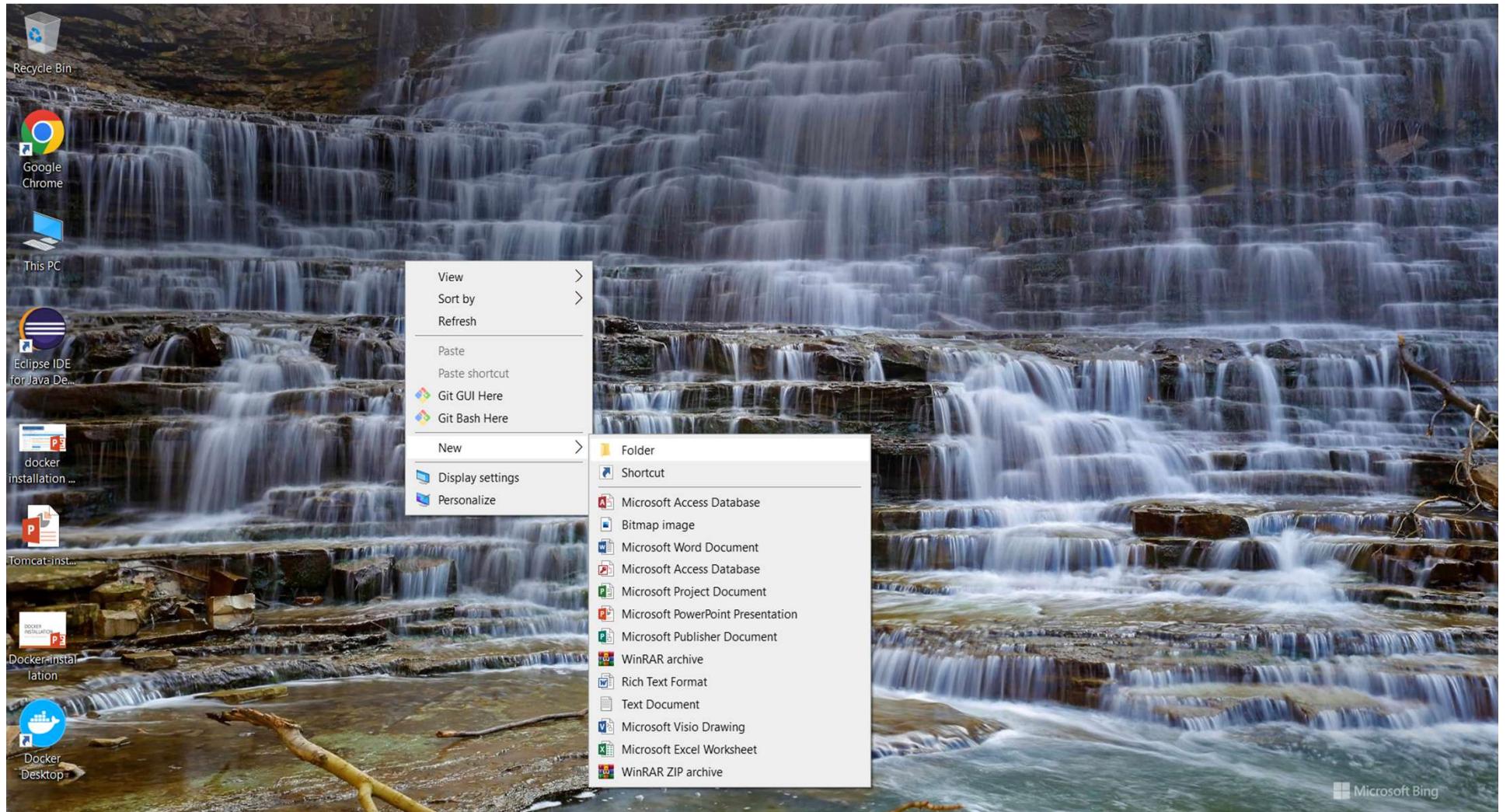
Try running a container: Copy and paste this command into your terminal and then come back

```
docker run -d -p 80:80 docker/getting-started
```

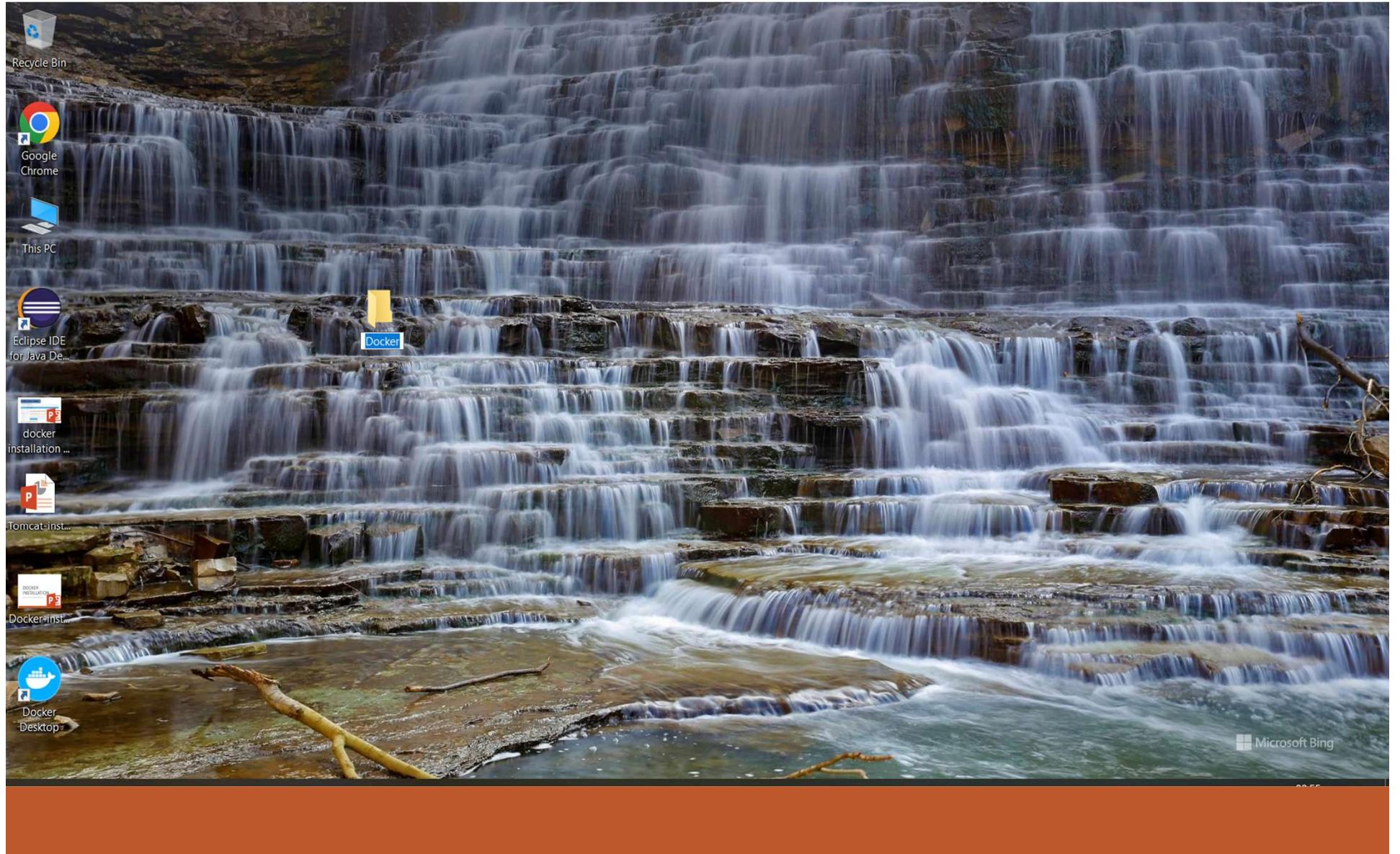
[Explore more in the Docker Docs](#)

On the left side of the slide, there is a vertical list of slide thumbnails numbered 15 to 20. The 19th slide is highlighted with a red border and shows a screenshot of a Docker interface with the heading "Installing Docker on". The 20th slide shows a screenshot of a Docker interface with the heading "Installing Docker on".

# Create a folder on desktop



# Docker



Working with Docker -> First check whether docker installed by command

> **docker --version**

---

You can distribute your software or application in either Linux, Windows or MAC and can run it on dockers

\$ **docker --version**

displays the version of the docker

```
Microsoft Windows [Version 10.0.26100.4946]
(c) Microsoft Corporation. All rights reserved.

C:\Users\BHARGAVI KUMBHAM>docker --version
Docker version 28.3.2, build 578ccf6
```

**docker version** displays the version information of docker client & server running

```
:\Users\BHARGAVI KUMBHAM>docker version
client:
Version:          28.3.2
API version:      1.51
Go version:       go1.24.5
Git commit:       578ccf6
Built:            Wed Jul  9 16:12:31 2025
OS/Arch:          windows/amd64
Context:          desktop-linux

server: Docker Desktop 4.44.1 (201842)
Engine:
  Version:          28.3.2
  API version:      1.51 (minimum version 1.24)
  Go version:       go1.24.5
  Git commit:       e77ff99
  Built:            Wed Jul  9 16:13:55 2025
  OS/Arch:          linux/amd64
  Experimental:    false
containerd:
  Version:          1.7.27
  GitCommit:        05044ec0a9a75232cad458027ca83437aae3f4da
runc:
  Version:          1.2.5
  GitCommit:        v1.2.5-0-g59923ef
docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0

:\Users\BHARGAVI KUMBHAM>
```

# \$ docker run hello-world

---

1. pull docker image hello-world locally
2. First, check in the cache, if not present, get from docker hub.
3. Then create a container for it and run

# **docker pull [image name]**

pull a docker image from local repository

---

```
C:\Users\BHARGAVI KUMBHAM>docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
Digest: sha256:a0dfb02aac212703bfcb339d77d47ec32c8706ff250850ecc0e19c8737b18567
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
```

```
C:\Users\BHARGAVI KUMBHAM>
```

**docker image ls** displays the list of docker images in the local server  
Tag latest means it has downloaded latest version. It shows size of image

---

```
C:\Users\BHARGAVI KUMBHAM>docker image ls
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
hello-world     latest       a0dfb02aac21  9 days ago   20.3kB
```

```
C:\Users\BHARGAVI KUMBHAM>|
```

# How to run Hello-world Image

## **docker run [image name]**

### executing a docker image in the local server

```
C:\Users\BHARGAVI KUMBHAM>docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

C:\Users\BHARGAVI KUMBHAM>
```

# docker ps -a

Displays the list of images running currently or previously or stopped  
docker pull <imagename>  
docker ps -a docker run ubuntu

```
C:\Users\BHARGAVI KUMBHAM>docker ps -a
CONTAINER ID        IMAGE             COMMAND            CREATED           STATUS            PORTS      NAMES
2f4043aeda5d      hello-world      "/hello"          49 seconds ago   Exited (0) 49 seconds ago
gracious_dewdney

C:\Users\BHARGAVI KUMBHAM>docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
b71466b94f26: Pull complete
Digest: sha256:7c06e91f61fa88c08cc74f7e1b7c69ae24910d745357e0dfe1d2c0322aaf20f9
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

C:\Users\BHARGAVI KUMBHAM>docker ps -a
CONTAINER ID        IMAGE             COMMAND            CREATED           STATUS            PORTS      NAMES
2f4043aeda5d      hello-world      "/hello"          About a minute ago   Exited (0) About a minute ago
gracious_dewdney

C:\Users\BHARGAVI KUMBHAM>docker run ubuntu

C:\Users\BHARGAVI KUMBHAM>docker ps -a
CONTAINER ID        IMAGE             COMMAND            CREATED           STATUS            PORTS      NAMES
0b6d963e8f1d      ubuntu           "/bin/bash"        16 seconds ago   Exited (0) 15 seconds ago
2f4043aeda5d      hello-world      "/hello"          2 minutes ago    Exited (0) 2 minutes ago
gracious_dewdney

C:\Users\BHARGAVI KUMBHAM>
```

## **docker run -it <container id / name> bash**

running a ubuntu container in interactive mode.... Using bash commands to ammend the existing image

---

Now run the Ubuntu with echo command

Docker echo command will run the command shell on the ubuntu container

```
C:\Users\BHARGAVI KUMBHAM>docker run -it ubuntu bash
root@779228eb144b:/# echo Dr.K.Bhargavi
Dr.K.Bhargavi
root@779228eb144b:/# pwd
/
root@779228eb144b:/# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@779228eb144b:/# exit
exit
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
		NAMES			
779228eb144b	ubuntu	"bash"	2 minutes ago	Exited (0) About a minute ago	
3f20a952436e	redis:6.0	"docker-entrypoint.s..."	49 minutes ago	Created	
d021eb67ea90	redis	"docker-entrypoint.s..."	56 minutes ago	Created	
89ea5c5c92f0	redis:7.0	"docker-entrypoint.s..."	About an hour ago	Up About an hour	0.0.0.0:7000->6379/tcp, [::]:7000->6379/tcp
20f910a2d204	redis:7.0	"docker-entrypoint.s..."	About an hour ago	Exited (0) About an hour ago	
a69e2a3a4c30	redis:6.0	"docker-entrypoint.s..."	About an hour ago	Up About an hour	0.0.0.0:6000->6379/tcp, [::]:6000->6379/tcp
631861951d6e	redis	"docker-entrypoint.s..."	About an hour ago	Exited (0) About an hour ago	
5f2fcbb3c753f	redis	"docker-entrypoint.s..."	2 hours ago	Up 2 hours	6379/tcp
c7cb40ae4e8b	redis:6.0	"docker-entrypoint.s..."	2 hours ago	Exited (0) About an hour ago	

```
docker exec -it 5f2fcb3c753f sh
```

docker exec → Run a command inside an already running container.

-it → Interactive terminal (so you can type inside).

5f2fcb3c753f → Container ID.

sh → Start a shell session inside the container.

---

It listed files in the current directory (/data), showing dump.rdb

dump.rdb is Redis's default snapshot database file (used to persist data)

/data (the working directory of Redis in the container)..

Exited from the container shell and returned to your Windows host (C:\Users\...).

```
C:\Users\BHARGAVI KUMBHAM>docker exec -it 5f2fcb3c753f sh
# ls
dump.rdb
# pwd
/data
# exit

C:\Users\BHARGAVI KUMBHAM>
```

**docker ps -a** display the list of

containers to learn the container id

**docker stop <cont id>** stop the running  
container

---

```
C:\Users\BHARGAVI KUMBHAM>docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS          NAMES
f5136af770d9        ubuntu              "bash"              About a minute ago   Exited (0) About a minute ago
1f83381f48e6        ubuntu              "bash"              3 minutes ago      Exited (127) 2 minutes ago
0b6d963e8f1d        ubuntu              "/bin/bash"         5 minutes ago      Exited (0) 5 minutes ago
2f4043aeda5d        hello-world        "/hello"            7 minutes ago      Exited (0) 7 minutes ago

C:\Users\BHARGAVI KUMBHAM>docker stop f5136af770d9
f5136af770d9
```

**docker commit <cont id> new name**      commit the modified container

---

```
C:\Users\BHARGAVI KUMBHAM>docker commit f5136af770d9 kumbham/ubuntu
sha256:86bdeeb0a751d3d4e117d2a59b7f124dc252954d8e2a6269e1a3363aec72c36c

C:\Users\BHARGAVI KUMBHAM>
```

- **docker image ls**      display the list of images
- 

```
C:\Users\BHARGAVI KUMBHAM>docker image ls
REPOSITORY      TAG      IMAGE ID      CREATED       SIZE
kumbham/ubuntu  latest   86bdeeb0a751  2 minutes ago  117MB
hello-world     latest   a0dfb02aac21  9 days ago    20.3kB
ubuntu          latest   7c06e91f61fa  2 weeks ago   117MB
```

```
C:\Users\BHARGAVI KUMBHAM>|
```

# Pushing the changed image from local repository to docker hub

## Step-1-Docker Login

---

**docker login**      login to docker hub with required credentials

```
C:\Users\BHARGAVI KUMBHAM>docker login
Authenticating with existing credentials... [Username: bhargavikumbham1]

[Info → To login with a different account, run 'docker logout' followed by 'docker login'

Login Succeeded
```

# Pushing the images from local repository to docker hub

## Step-2-Push Command

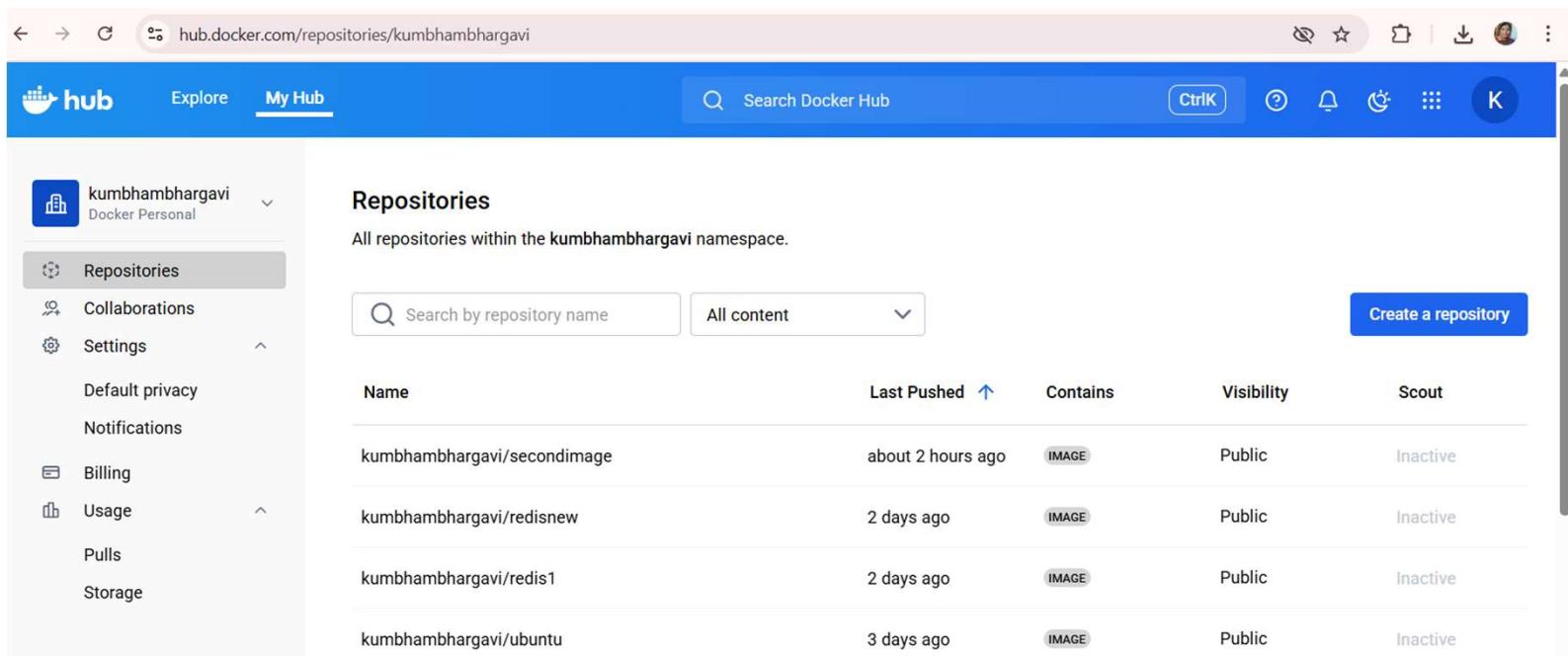
---

- **docker push <image name>** push the image to the docker hub repository
- docker push

```
C:\Users\BHARGAVI KUMBHAM>docker tag kumbham/ubuntu:latest kumbhambhargavi/ubuntu:latest  
C:\Users\BHARGAVI KUMBHAM>docker push kumbhambhargavi/ubuntu:latest  
The push refers to repository [docker.io/kumbhambhargavi/ubuntu]  
b71466b94f26: Mounted from library/ubuntu  
f47bf9485dc3: Pushed  
latest: digest: sha256:86bdeeb0a751d3d4e117d2a59b7f124dc252954d8e2a6269e1a3363aec72c36c size: 746  
C:\Users\BHARGAVI KUMBHAM>
```

# Goto Github and check the repositories

---



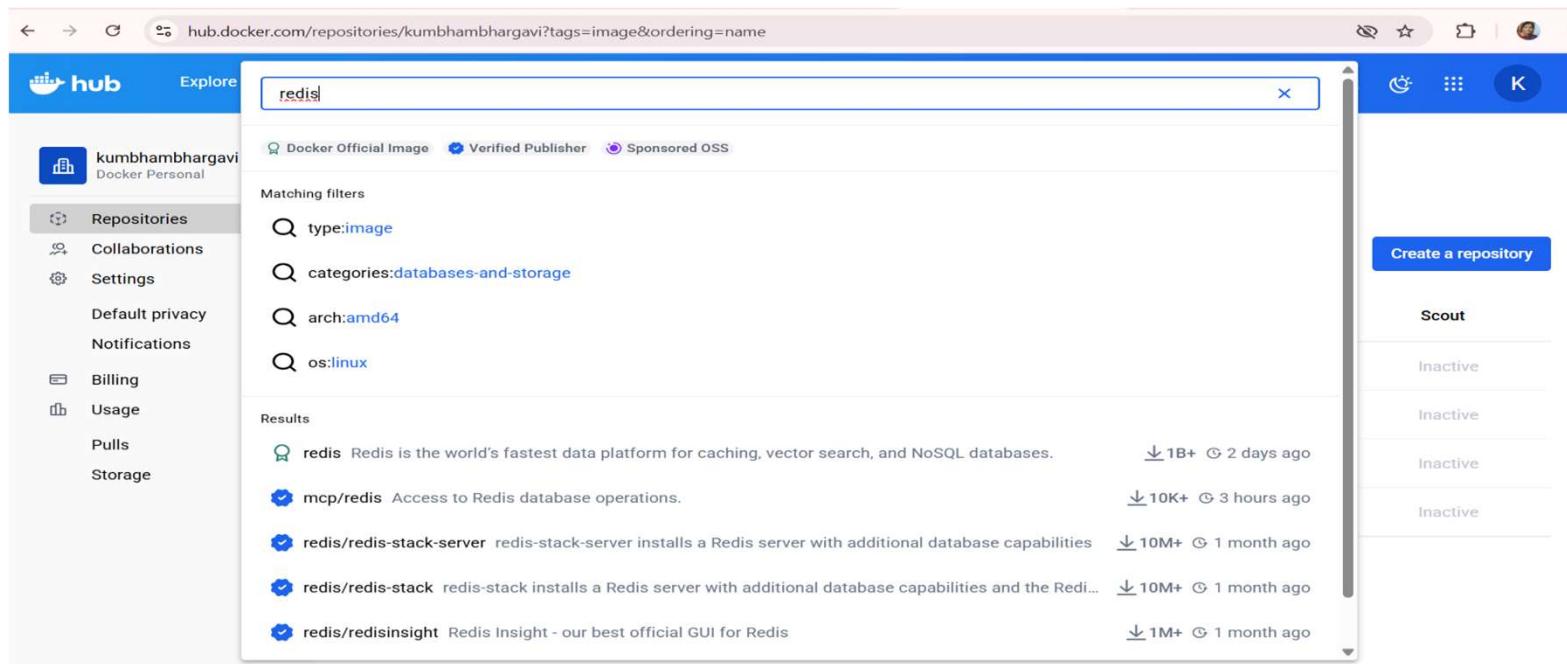
A screenshot of a web browser displaying the Docker Hub interface. The URL in the address bar is `hub.docker.com/repositories/kumbhambhargavi`. The page title is "Repositories". The left sidebar shows the user's profile "kumbhambhargavi" (Docker Personal) and navigation links for "Repositories", "Collaborations", "Settings", "Default privacy", "Notifications", "Billing", "Usage", "Pulls", and "Storage". The main content area is titled "Repositories" and displays a table of four Docker images. The table columns are "Name", "Last Pushed", "Contains", "Visibility", and "Scout". The images listed are:

Name	Last Pushed	Contains	Visibility	Scout
kumbhambhargavi/secondimage	about 2 hours ago	IMAGE	Public	Inactive
kumbhambhargavi/redisnew	2 days ago	IMAGE	Public	Inactive
kumbhambhargavi/redis1	2 days ago	IMAGE	Public	Inactive
kumbhambhargavi/ubuntu	3 days ago	IMAGE	Public	Inactive

At the top right of the main content area, there is a "Create a repository" button.

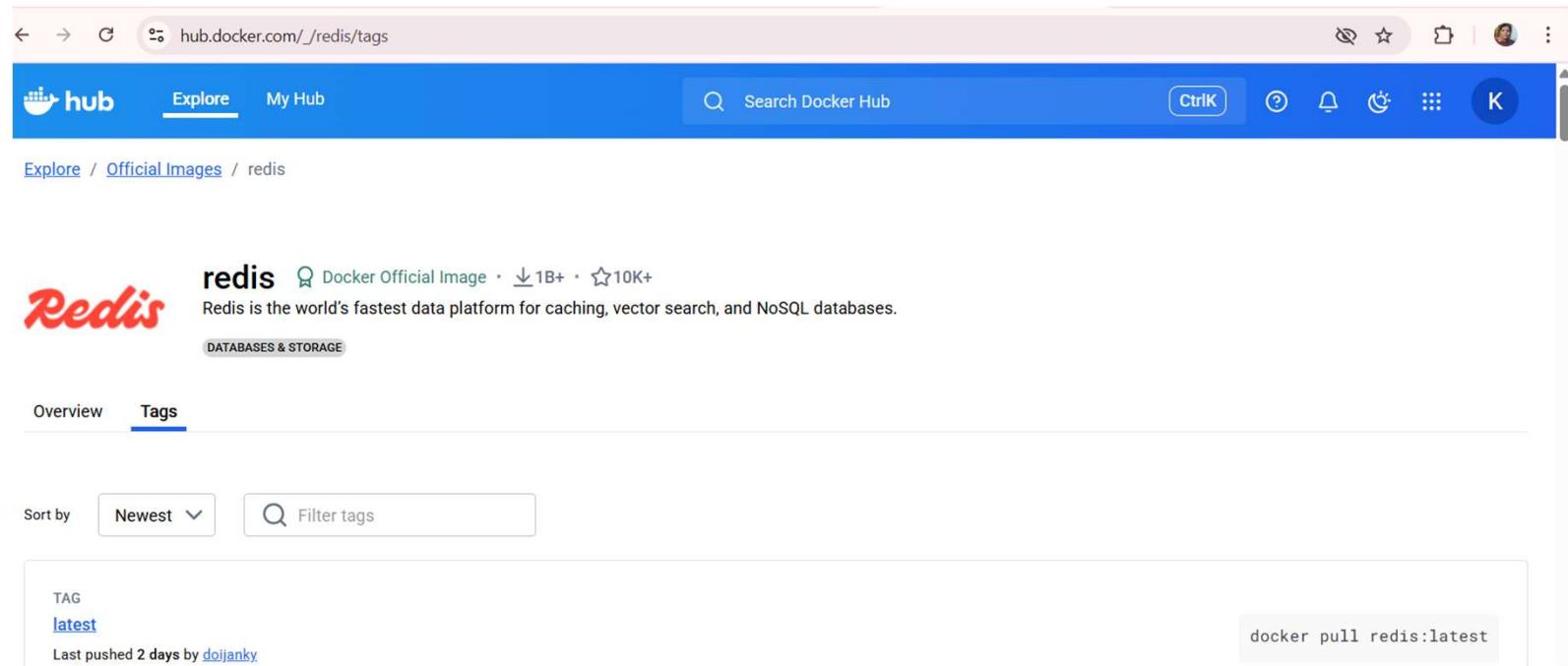
# Get images from docker hub

Goto dockerhub.com and you get many images. Now click on redis image



# Copy docker pull redis:latest and paste in command prompt

---



# If you go below in docker hub page, you can see redis versions

---

## Quick reference

- Maintained by:  
the Docker Community
- Where to get help:  
the Docker Community Forums, the Docker Community Slack, or Stack Overflow

## Supported tags and respective Dockerfile links

- `7.0.4`, `7.0`, `7`, `latest`, `7.0.4-bullseye`, `7.0-bullseye`, `7-bullseye`, `bullseye`
- `7.0.4-alpine`, `7.0-alpine`, `7-alpine`, `alpine`, `7.0.4-alpine3.16`, `7.0-alpine3.16`, `7-alpine3.16`, `alpine3.16`
- `6.2.7`, `6.2`, `6`, `6.2.7-bullseye`, `6.2-bullseye`, `6-bullseye`
- `6.2.7-alpine`, `6.2-alpine`, `6-alpine`, `6.2.7-alpine3.16`, `6.2-alpine3.16`, `6-alpine3.16`
- `6.0.16`, `6.0`, `6.0.16-bullseye`, `6.0-bullseye`
- `6.0.16-alpine`, `6.0-alpine`, `6.0.16-alpine3.16`, `6.0-alpine3.16`
- `5.0.14`, `5.0`, `5`, `5.0.14-bullseye`, `5.0-bullseye`, `5-bullseye`
- `5.0.14-32bit`, `5.0-32bit`, `5-32bit`, `5.0.14-32bit-bullseye`, `5.0-32bit-bullseye`, `5-32bit-bullseye`
- `5.0.14-alpine`, `5.0-alpine`, `5-alpine`, `5.0.14-alpine3.16`, `5.0-alpine3.16`, `5-alpine3.16`



## docker pull redis

pulling dockerhub to ocal repository

---

```
PS C:\Users\BHARGAVI KUMBHAM\OneDrive\Desktop\Dockerproject> docker pull redis:latest
latest: Pulling from library/redis
Digest: sha256:cc2dfb8f5151da2684b4a09bd04b567f92d07591d91980eb3eca21df07e12760
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
PS C:\Users\BHARGAVI KUMBHAM\OneDrive\Desktop\Dockerproject> |
```

## docker images

Shows all images in local repository. Tag latest means it has downloaded latest version. It shows size of image

---

```
C:\Users\BHARGAVI KUMBHAM>docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
kimage              01      f5a4b63b7121  3 hours ago   27.8MB
csmimage            01      3ab594f86bac  3 hours ago   27.8MB
kumbhambhargavi/secondimage  latest   3ab594f86bac  3 hours ago   27.8MB
kumbhambhargavi/redisnew    latest   003532995b6a  44 hours ago  200MB
kumbhambhargavi/redis1      latest   5cb5f49974cf  2 days ago    200MB
redisnew             latest   d2315726eed8  2 days ago    200MB
redis                latest   cc2dfb8f5151  2 days ago    200MB
dockerhub-username/ubuntu  latest   86bdeeb0a751  2 days ago    117MB
kumbham/ubuntu        latest   86bdeeb0a751  2 days ago    117MB
kumbhambhargavi/ubuntu  latest   86bdeeb0a751  2 days ago    117MB
wordpress            latest   c5f075fe71c9  8 days ago    1.04GB
hello-world           latest   a0dfb02aac21  12 days ago   20.3kB
ubuntu               latest   7c06e91f61fa  3 weeks ago   117MB
mysql                5.7     4bc6bc963e6d  20 months ago  689MB

C:\Users\BHARGAVI KUMBHAM>
```

# docker ps

Shows container ID of image and port on which it is running

```
C:\Users\BHARGAVI KUMBHAM>docker ps
CONTAINER ID        IMAGE       COMMAND                  CREATED             STATUS              PORTS
 NAMES
394d520f2131      redis:4.0    "docker-entrypoint.s..."   About a minute ago   Up About a minute   0.0.0.0:6379->6379/tcp, [::]:6379->6379/tcp
      newredis
91901cd726bc      redis       "docker-entrypoint.s..."   10 minutes ago     Up 10 minutes      6379/tcp
      my-redis
006192b4d055      redis       "docker-entrypoint.s..."   32 minutes ago     Up 32 minutes      6379/tcp
      myredis
981dc37fde46      mysql:5.7   "docker-entrypoint.s..."   42 hours ago       Up 42 hours       3306/tcp, 33060/tcp
      mysql_container

C:\Users\BHARGAVI KUMBHAM>
```

# docker run redis

---

```
C:\Users\BHARGAVI KUMBHAM>docker run redis
Starting Redis Server
1:C 21 Aug 2025 09:53:05.202 * o000o000o000o Redis is starting o000o000o000o
1:C 21 Aug 2025 09:53:05.203 * Redis version=8.2.1, bits=64, commit=00000000, modified=1, pid=1, just started
1:C 21 Aug 2025 09:53:05.203 * Configuration loaded
1:M 21 Aug 2025 09:53:05.203 * monotonic clock: POSIX clock_gettime
1:M 21 Aug 2025 09:53:05.207 * Running mode=standalone, port=6379.
1:M 21 Aug 2025 09:53:05.211 * <bf> RedisBloom version 8.2.0 (Git=unknown)
1:M 21 Aug 2025 09:53:05.211 * <bf> Registering configuration options: [
1:M 21 Aug 2025 09:53:05.211 * <bf>   { bf-error-rate : 0.01 }
1:M 21 Aug 2025 09:53:05.211 * <bf>   { bf-initial-size : 100 }
1:M 21 Aug 2025 09:53:05.211 * <bf>   { bf-expansion-factor : 2 }
1:M 21 Aug 2025 09:53:05.211 * <bf>   { cf-bucket-size : 2 }
1:M 21 Aug 2025 09:53:05.211 * <bf>   { cf-initial-size : 1024 }
1:M 21 Aug 2025 09:53:05.211 * <bf>   { cf-max-iterations : 20 }
1:M 21 Aug 2025 09:53:05.211 * <bf>   { cf-expansion-factor : 1 }
1:M 21 Aug 2025 09:53:05.211 * <bf>   { cf-max-expansions : 32 }
1:M 21 Aug 2025 09:53:05.211 * <bf> ]
1:M 21 Aug 2025 09:53:05.212 * Module 'bf' loaded from /usr/local/lib/redis/modules//redisbloom.so
1:M 21 Aug 2025 09:53:05.249 * <search> Redis version found by RedisSearch : 8.2.1 - oss
1:M 21 Aug 2025 09:53:05.249 * <search> RediSearch version 8.2.1 (Git=dba8dd0)
1:M 21 Aug 2025 09:53:05.250 * <search> Low level api version 1 initialized successfully
```

# Goto previous command prompt and press Control C to stop redis container running

---

```
1:M 21 Aug 2025 09:53:05.283 * <ReJSON> Exported RedisJSON_V1 API
1:M 21 Aug 2025 09:53:05.283 * <ReJSON> Exported RedisJSON_V2 API
1:M 21 Aug 2025 09:53:05.283 * <ReJSON> Exported RedisJSON_V3 API
1:M 21 Aug 2025 09:53:05.283 * <ReJSON> Exported RedisJSON_V4 API
1:M 21 Aug 2025 09:53:05.283 * <ReJSON> Exported RedisJSON_V5 API
1:M 21 Aug 2025 09:53:05.283 * <ReJSON> Enabled diskless replication
1:M 21 Aug 2025 09:53:05.283 * <ReJSON> Initialized shared string cache, thread safe: false
1:M 21 Aug 2025 09:53:05.283 * Module 'ReJSON' loaded from /usr/local/lib/redis/modules//r
1:M 21 Aug 2025 09:53:05.283 * <search> Acquired RedisJSON_V5 API
1:M 21 Aug 2025 09:53:05.284 * Server initialized
1:M 21 Aug 2025 09:53:05.286 * Ready to accept connections tcp
1:signal-handler (1755770179) Received SIGINT scheduling shutdown...
1:M 21 Aug 2025 09:56:19.920 * User requested shutdown...
1:M 21 Aug 2025 09:56:19.920 * Saving the final RDB snapshot before exiting.
1:M 21 Aug 2025 09:56:19.930 * DB saved on disk
1:M 21 Aug 2025 09:56:19.930 # Redis is now ready to exit, bye bye...

C:\Users\BHARGAVI KUMBHAM>
C:\Users\BHARGAVI KUMBHAM>
C:\Users\BHARGAVI KUMBHAM>
C:\Users\BHARGAVI KUMBHAM>
```

# stops running container

---

docker stop 96724fc65bd8

```
C:\Users\BHARGAVI KUMBHAM>docker stop 96724fc65bd8  
96724fc65bd8
```

```
C:\Users\BHARGAVI KUMBHAM>
```

# stoped 96724fc65bd8

---

C:\Users\BHARGAVI KUMBHAM>docker ps -a	COMMAND	CREATED	STATUS	PORTS
CONTAINER ID IMAGE NAMES				
96724fc65bd8 redis distracted_hellman	"docker-entrypoint.s..."	11 minutes ago	Exited (0) About a minute ago	
cfaa881b09d7 redis blissful_cerf	"docker-entrypoint.s..."	16 minutes ago	Exited (0) 13 minutes ago	
48f92a0dd3e2 kumbhambargavi/secondimage:latest loving_ptolemy	"redis-server"	3 hours ago	Exited (0) 8 minutes ago	
62e4b7cda543 3ab594f86bac nifty_lamport	"redis-server"	3 hours ago	Exited (0) 3 hours ago	
4b2055d7f932 wordpress:latest wordpress_container	"docker-entrypoint.s..."	41 hours ago	Created	
981dc37fde46 mysql:5.7	"docker-entrypoint.s..."	42 hours ago	Up 42 hours	3306/tcp
33060/tcp mysql_container				
2254ff0942da kumbhambargavi/redis1 redis1	"docker-entrypoint.s..."	42 hours ago	Exited (0) 42 hours ago	
f5136af770d9 ubuntu boring_swirles	"bash"	2 days ago	Exited (0) 2 days ago	
1f83381f48e6 ubuntu xenodochial_proskuriakova	"bash"	2 days ago	Exited (127) 2 days ago	
0b6d963e8f1d ubuntu determined_leakey	"/bin/bash"	2 days ago	Exited (0) 2 days ago	
2f4043aeda5d hello-world gracious_dewdney	"/hello"	3 days ago	Exited (0) 3 days ago	

# Creating same image with different version in one application

**docker run redis:6.0**

---

```
C:\Users\BHARGAVI KUMBHAM>docker run redis:6.0
Unable to find image 'redis:6.0' locally
6.0: Pulling from library/redis
a4de354087d8: Pull complete
e351272bb751: Pull complete
d4d7970efdb4: Pull complete
4f4fb700ef54: Pull complete
99f376ebb190: Pull complete
8d5a9db41753: Pull complete
d2acfa1fd76e: Pull complete
18c00d30c02d: Pull complete
Digest: sha256:b99ffd0554dc8d300230b9d1b9f2a129a6abf595bf8589883beb980ed1feae3d
Status: Downloaded newer image for redis:6.0
1:C 21 Aug 2025 10:57:22.505 # o000o000o000o Redis is starting o000o000o000o
1:C 21 Aug 2025 10:57:22.505 # Redis version=6.0.20, bits=64, commit=00000000, modified=0, pid=1, just started
1:C 21 Aug 2025 10:57:22.505 # Warning: no config file specified, using the default config. In order to specify a config file use redis-server /path/to/redis.conf
1:M 21 Aug 2025 10:57:22.507 * Running mode=standalone, port=6379.
1:M 21 Aug 2025 10:57:22.507 # Server initialized
1:M 21 Aug 2025 10:57:22.508 * Ready to accept connections
1:signal-handler (1755773883) Received SIGINT scheduling shutdown...
1:M 21 Aug 2025 10:58:03.095 # User requested shutdown...
1:M 21 Aug 2025 10:58:03.095 * Saving the final RDB snapshot before exiting.
1:M 21 Aug 2025 10:58:03.098 * DB saved on disk
1:M 21 Aug 2025 10:58:03.098 # Redis is now ready to exit, bye bye...

C:\Users\BHARGAVI KUMBHAM>
```

---

If Redis is still running and showing logs → press

Ctrl + C

That will stop the container and give you back the prompt.

If Redis got stuck and doesn't release the prompt → just close that Command Prompt window and open a new one.

Run in detached mode:-a → detached, so it runs in background.

**docker run -d --name uredis -p 6379:6379  
redis:6.0**

---

```
C:\Users\BHARGAVI KUMBHAM>docker run -d --name uredis -p 6379:6379 redis:6.0
8260ef5a2d378d5c13e2326e70ec4c2b46a237ad0ebf8e841f40ec8adda9d9cf
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint u
redis (3e59420f2349de31229c2d3df22112c3f576d969a8bb94d511544c6a5211a87a): Bind for 0.0.0.0:6379 failed: port is already allocated
Run 'docker run --help' for more information
```

getting an error

# getting an error then check

---

Check which container is using port 6379

```
docker ps
```

If you see another container using port 6379:6379, stop and remove it:

```
docker stop <container_id>
```

```
docker rm <container_id>
```

If you want to start fresh and avoid this name conflict issue in future, you can remove all stopped containers with (Better check which you want to stop and do)

---

```
C:\Users\BHARGAVI KUMBHAM>docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
b206415fd05db73aad43572a9cff1b6954294c41525cdf18df132015f32ddfc
83bcd4eb551ed05604afdb1c2ffd0ec979e83fbfcc95cac130ffb0998c28df02
79c8d306ee20f2714fa27fc960671c583bf4691de2179de2d9765dbd63ee298b
ca66a5400315525061c4476914811320544d4423a4cd03915fa2f2a87aea030c
a7c68ce096541c0e30893cf5d107fdf1e00b5db1de43b67fd510de95b0c333
8ee3af4396bc306e1991aec468fdedc04bcc19d01ec12693c1b2e8ae5647051
622e7107cc77ba70552bde55ff8b495d42b261ed733253c0eb92f2e1d7faf3e
088993106ee1c57ad5ded83c8e666f3749834929a0a3e581c5a8410ee1cb8f9b
96724fc65bd89262e0a22bd391b4954a57f202fa394e001d0a7ef4ec52e07ada
cfaa881b09d7c83ceda2b15793cba5aeed69983b4dcae4ce9b710e375ff2d1e1
48f92a0dd3e2108aa95a20684d760a4974b7910face61b21bd89f60b911de502
62e4b7cda543843aff20463b67f28ef699c983c5dc12924832ef3dded85931ea
4b2055d7f932ad32508cf2822125be89e8a08ec035805436b627ceddfc55a06b
2254ff0942da54481aa64e512f6d9794893886135594faf5e1025230c2cf66bc
f5136af770d9960f31d213b112278948d40d2246d9811b2018b7f2c472333d8c
1f83381f48e623626ac18f282d6d515a4bd733d1f6764ad4efe69db9de9ee835
0b6d963e8f1d9b730f109465a873dd2f4328c2ff206a29661549ef32cb7dd87a
2f4043aeda5d218574cac442a8cbf2c2fad8aa05d167f56d5ed21c32b50ed9a7
```

# docker ps

Here are two redis images with different versions run using a container having the same port 6379 number. The containers having the same port can be bind to a different port on a host machine/ laptop.

```
C:\Users\BHARGAVI KUMBHAM>docker ps
CONTAINER ID   IMAGE      COMMAND           CREATED        STATUS          PORTS          NAMES
5f2fcb3c753f   redis      "docker-entrypoint.s..."  34 minutes ago  Up 27 minutes  6379/tcp       epic_
moser
c7cb40ae4e8b   redis:6.0  "docker-entrypoint.s..."  49 minutes ago  Up 49 minutes  0.0.0.0:6379->6379/tcp, [::]:6379->6379/tcp  uredi
s
981dc37fde46   mysql:5.7  "docker-entrypoint.s..."  46 hours ago   Up 46 hours    3306/tcp, 33060/tcp      mysql
_container
```

Now run

docker run -d --name nredis -p 6379:6379  
redis:6.0

---

```
C:\Users\BHARGAVI KUMBHAM>docker run -d --name uredis -p 6379:6379 redis:6.0
c7cb40ae4e8b62e76a5dab294a3b5b743efe8a0d251286eb6a5b9691d99dc499
```

```
C:\Users\BHARGAVI KUMBHAM>docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
b206415fd05db73aad43572a9cffff1b6954294c41525cdf18df132015f32ddfc
```

```
C:\Users\BHARGAVI KUMBHAM>docker run -d --name nredis -p 6379:6379 redis:6.0
9f627872925e13e7108c254776ff11fee8f1da81d06ef1b07371da8a081e1e63

C:\Users\BHARGAVI KUMBHAM>docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
9f627872925e redis:6.0 "docker-entrypoint.s..." About a minute ago Up About a minute 0.0.0.0:6379->6379/tcp, [::]:6379->6379/tcp
nredis
91901cd726bc redis "docker-entrypoint.s..." 28 minutes ago Up 28 minutes 6379/tcp
my-redis
006192b4d055 redis "docker-entrypoint.s..." 50 minutes ago Up 50 minutes 6379/tcp
myredis
981dc37fde46 mysql:5.7 "docker-entrypoint.s..." 43 hours ago Up 43 hours 3306/tcp, 33060/tcp
mysql_container
```

# Run in detached mode:-d →

detached, so it runs in background.

**docker run -d --name nredis -p 6379:6379 redis:4.0**

---

```
C:\Users\BHARGAVI KUMBHAM>docker run -d --name newredis -p 6379:6379 redis:4.0  
394d520f2131ee7ed06a227b05c3a59ef5410f5df00294df51efdb02a0a8fdf6
```

```
C:\Users\BHARGAVI KUMBHAM>
```

Use -p port number to bind the redis to port 6000  
**docker run -p6000:6379 redis**

---

```
C:\Users\BHARGAVI KUMBHAM>docker run -p6000:6379 redis
Starting Redis Server
1:C 21 Aug 2025 13:35:07.010 * o000o000o000o Redis is starting o000o000o000o
1:C 21 Aug 2025 13:35:07.010 * Redis version=8.2.1, bits=64, commit=00000000, modified=1, p
1:C 21 Aug 2025 13:35:07.010 * Configuration loaded
1:M 21 Aug 2025 13:35:07.012 * monotonic clock: POSIX clock_gettime
1:M 21 Aug 2025 13:35:07.014 * Running mode=standalone, port=6379.
1:M 21 Aug 2025 13:35:07.016 * <bf> RedisBloom version 8.2.0 (Git=unknown)
1:M 21 Aug 2025 13:35:07.016 * <bf> Registering configuration options: [
1:M 21 Aug 2025 13:35:07.016 * <bf>     { bf-error-rate : 0.01 }
1:M 21 Aug 2025 13:35:07.016 * <bf>     { bf-initial-size : 100 }
1:M 21 Aug 2025 13:35:07.016 * <bf>     { bf-expansion-factor : 2 }
1:M 21 Aug 2025 13:35:07.016 * <bf>     { cf-bucket-size : 2 }
1:M 21 Aug 2025 13:35:07.016 * <bf>     { cf-initial-size : 1024 }
1:M 21 Aug 2025 13:35:07.016 * <bf>     { cf-max-iterations : 20 }
1:M 21 Aug 2025 13:35:07.016 * <bf>     { cf-expansion-factor : 1 }
1:M 21 Aug 2025 13:35:07.016 * <bf>     { cf-max-expansions : 32 }
1:M 21 Aug 2025 13:35:07.016 * <bf> ]
1:M 21 Aug 2025 13:35:07.017 * Module 'bf' loaded from /usr/local/lib/redis/modules//redisb
1:M 21 Aug 2025 13:35:07.029 * <search> Redis version found by RedisSearch : 8.2.1 - oss
1:M 21 Aug 2025 13:35:07.029 * <search> RedisSearch version 8.2.1 (Git=dba8dd0)
1:M 21 Aug 2025 13:35:07.029 * <search> Low level api version 1 initialized successfully
1:M 21 Aug 2025 13:35:07.030 * <search> ac::ON prefix min length: 2 min word length to st
```

# docker ps

## Shows redis not binded to port 6000

---

```
C:\Users\BHARGAVI KUMBHAM>docker ps
CONTAINER ID   IMAGE      COMMAND       CREATED      STATUS      PORTS          NAMES
5f2fcb3c753f   redis      "docker-entrypoint.s..."  37 minutes ago  Up 29 minutes  6379/tcp        epic_
moser
c7cb40ae4e8b   redis:6.0  "docker-entrypoint.s..."  51 minutes ago  Up 51 minutes  0.0.0.0:6379->6379/tcp, [::]:6379->6379/tcp  uredi
s
981dc37fde46   mysql:5.7  "docker-entrypoint.s..."  46 hours ago   Up 46 hours   3306/tcp, 33060/tcp        mysql
                                     _container
C:\Users\BHARGAVI KUMBHAM>
```

# Now run Redis with port binding:

---

```
C:\Users\BHARGAVI KUMBHAM>docker run -d --name redis6000 -p 6000:6379 redis:6.0  
a69e2a3a4c307ebef770d4a470e0b3909c43a4ae1132d167d6c7d962e2502763
```

```
C:\Users\BHARGAVI KUMBHAM>
```

successfully started a Redis container named redis6000 with port mapping 6000:6379.

# Now check docker ps

---

```
C:\Users\BHARGAVI KUMBHAM>docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                         NAMES
a69e2a3a4c30      redis:6.0          "docker-entrypoint.s..."   37 seconds ago    Up 37 seconds       0.0.0.0:6000->6379/tcp, [::]:6000->6379/tcp   redis
6000
5f2fc3c753f       redis              "docker-entrypoint.s..."   44 minutes ago   Up 36 minutes      6379/tcp                      epic_
moser
981dc37fde46      mysql:5.7         "docker-entrypoint.s..."   46 hours ago     Up 46 hours        3306/tcp, 33060/tcp           mysql
_container
```

Shows redis binded to port 6000

Now redis:7.0 connected to port 7000, if connected same port get an error

```
docker run -d --name redis7000 -p 7000:6379  
redis:7.0
```

---

```
C:\Users\BHARGAVI KUMBHAM>docker run -d --name redis7000 -p 7000:6379 redis:7.0  
89ea5c5c92f03627a2d1bb2a8da43647806f2bfe5ab10aeed5a2714a0cc94e6e  
  
C:\Users\BHARGAVI KUMBHAM>docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
89ea5c5c92f0 redis:7.0 "docker-entrypoint.s..." 5 seconds ago Up 4 seconds 0.0.0.0:7000->6379/tcp, [::]:7000->6379/tcp redis  
7000  
a69e2a3a4c30 redis:6.0 "docker-entrypoint.s..." 9 minutes ago Up 9 minutes 0.0.0.0:6000->6379/tcp, [::]:6000->6379/tcp redis  
6000  
5f2fc3c753f redis "docker-entrypoint.s..." 53 minutes ago Up 45 minutes 6379/tcp epic_moser  
981dc37fde46 mysql:5.7 "docker-entrypoint.s..." 47 hours ago Up 47 hours 3306/tcp, 33060/tcp mysql  
_container  
  
C:\Users\BHARGAVI KUMBHAM>
```

redis:7.0 is connected to port number 7000

`docker logs <container_id>` is the main way to see what's happening inside a container (especially if it exited unexpectedly).

---

```
C:\Users\BHARGAVI KUMBHAM>docker logs a69e2a3a4c30
1:C 21 Aug 2025 14:54:22.742 # <0000000000000000> Redis is starting <0000000000000000>
1:C 21 Aug 2025 14:54:22.742 # Redis version=6.0.20, bits=64, commit=00000000, modified=0, pid=1, just started
1:C 21 Aug 2025 14:54:22.742 # Warning: no config file specified, using the default config. In order to specify a config file use
server /path/to/redis.conf
1:M 21 Aug 2025 14:54:22.745 * Running mode=standalone, port=6379.
1:M 21 Aug 2025 14:54:22.746 # Server initialized
1:M 21 Aug 2025 14:54:22.747 * Ready to accept connections
```

# Naming the container redis 6.0 named as older

```
docker run -d -p 6000:6379 --name redis-older  
redis:6.0
```

---

```
C:\Users\BHARGAVI KUMBHAM>docker run -d -p 6000:6379 --name redis-older redis:6.0  
3f20a952436e0e91d2ce93d923bfa5794b208c1bee0bd9b1bc23ed6968f278ae  
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint  
redis-older (5dfbe8c4083b1cf2fabc36298725d72a5fe960359879dc92d7a1d4b03a48e3a8): Bind for 0.0.0.0:6000 failed: port is already allocated  
Run 'docker run --help' for more information  
C:\Users\BHARGAVI KUMBHAM>
```

Run in detached mode (-d)

Map host port 6000 → 6379 (Redis default)

Name the container redis-older

Use the image redis:6.0

## **Ex: 3**

### **Pull image from docker repository and run it as a container**

---

- Docker repository contains images like ubuntu machine, node, etc. If anyone wants ubuntu they need not have to download it. We can pull the ubuntu image from the docker repository and run it as a container that will provide all the features of ubuntu.
- We bring it in our local repository and run

# docker search ubuntu

## displays public images in ubuntu

```
C:\Users\BHARGAVI KUMBHAM>docker search ubuntu
NAME                           DESCRIPTION                                     STARS      OFFICIAL
ubuntu                         Ubuntu is a Debian-based Linux operating sys... 17661      [OK]
ubuntu/squid                   Squid is a caching proxy for the Web. Long-t... 117
ubuntu/nginx                   Nginx, a high-performance reverse proxy & we... 133
ubuntu/cortex                  Cortex provides storage for Prometheus. Long...
ubuntu/kafka                    Apache Kafka, a distributed event streaming ... 55
ubuntu/bind9                   BIND 9 is a very flexible, full-featured DNS... 111
ubuntu/apache2                 Apache, a secure & extensible open-source HT... 97
ubuntu/prometheus              Prometheus is a systems and service monitori...
ubuntu/zookeeper               ZooKeeper maintains configuration informatio... 13
ubuntu/mysql                   MySQL open source fast, stable, multi-thread...
ubuntu/postgres                PostgreSQL is an open source object-relation...
ubuntu/jre                      Distroless Java runtime based on Ubuntu. Lon...
ubuntu/dotnet-aspnet            Chiselled Ubuntu runtime image for ASP.NET a... 25
ubuntu/redis                   Redis, an open source key-value store. Long-...
ubuntu/python                  A chiselled Ubuntu rock with the Python runt...
ubuntu/dotnet-deps             Chiselled Ubuntu for self-contained .NET & A...
ubuntu/grafana                 Grafana, a feature rich metrics dashboard & ...
ubuntu/dotnet-runtime           Chiselled Ubuntu runtime image for .NET apps...
ubuntu/memcached               Memcached, in-memory keyvalue store for smal...
ubuntu/prometheus-alertmanager Alertmanager handles client alerts from Prom...
ubuntu/cassandra               Cassandra, an open source NoSQL distributed ...
ubuntu/mlflow                   MLFlow: for managing the machine learning li...
ubuntu/telegraf                 Telegraf collects, processes, aggregates & w...
ubuntu/chiselled-jre           [MOVED TO ubuntu/jre] Chiselled JRE: distrol...
ubuntu/loki                     Grafana Loki, a log aggregation system like ...

C:\Users\BHARGAVI KUMBHAM>
```

---

# Working with Docker file



## **Exercise 4. Create Docker Image using Dockerfile**

1. What is Dockerfile
  2. How to create Dockerfile
  3. How to build image from Dockerfile
  4. Basic Commands
-

# Dockerfile

---

It is plain text file that contains instruction to build image.

In docker hub existing images are present.

Dockerfile helps to create our new image



# Steps to follow

---

Step 1 : Create a file named Dockerfile

Step 2 : Add instructions in Dockerfile

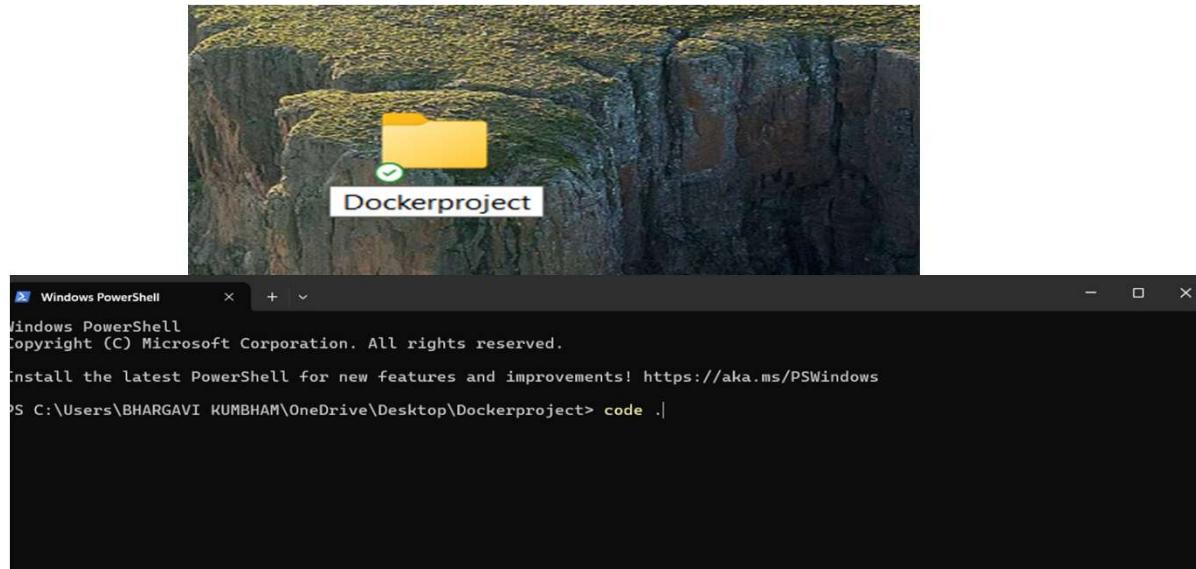
Step 3 : Build dockerfile to create image

Step 4 : Run image to create container

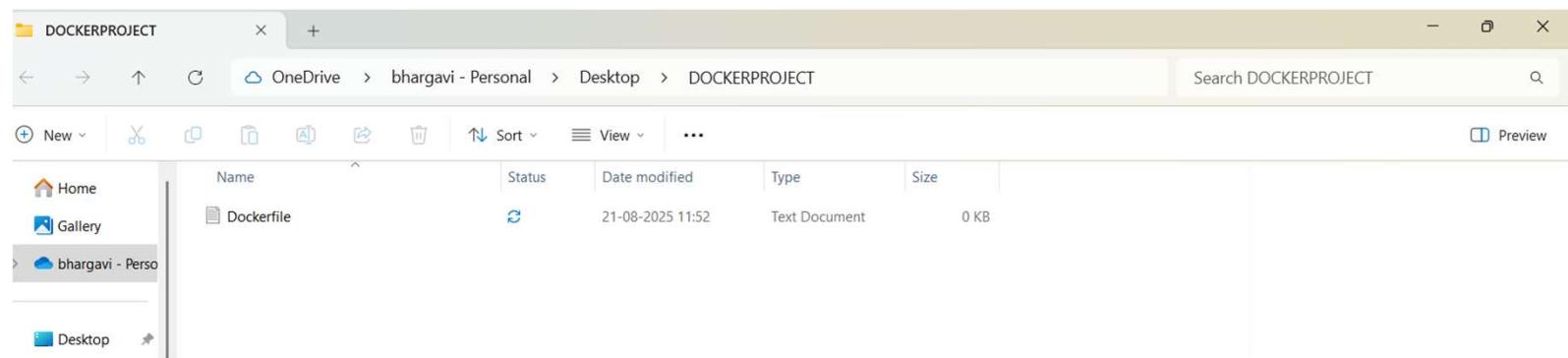
# Step 1 create Dockerfile

---

- a. Go desktop, right-click and create a folder named Dockerproject

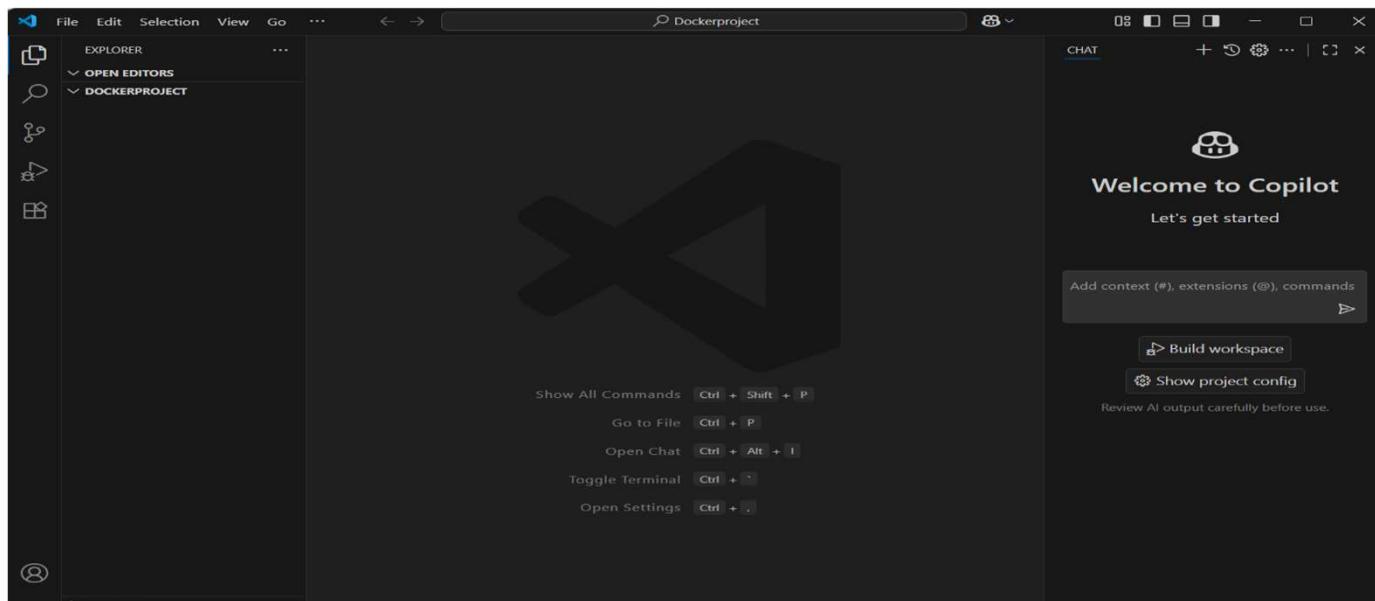


- a. Open the folder in Visual studio code
- b. Create a file named Dockerfile



open the terminal and type code .

```
PS C:\Users\BHARGAVI KUMBHAM\OneDrive\Desktop\Dockerproject> code .
PS C:\Users\BHARGAVI KUMBHAM\OneDrive\Desktop\Dockerproject> |
```



## Step 2: Create a file named Dockerfile and Add instructions to Dockerfile

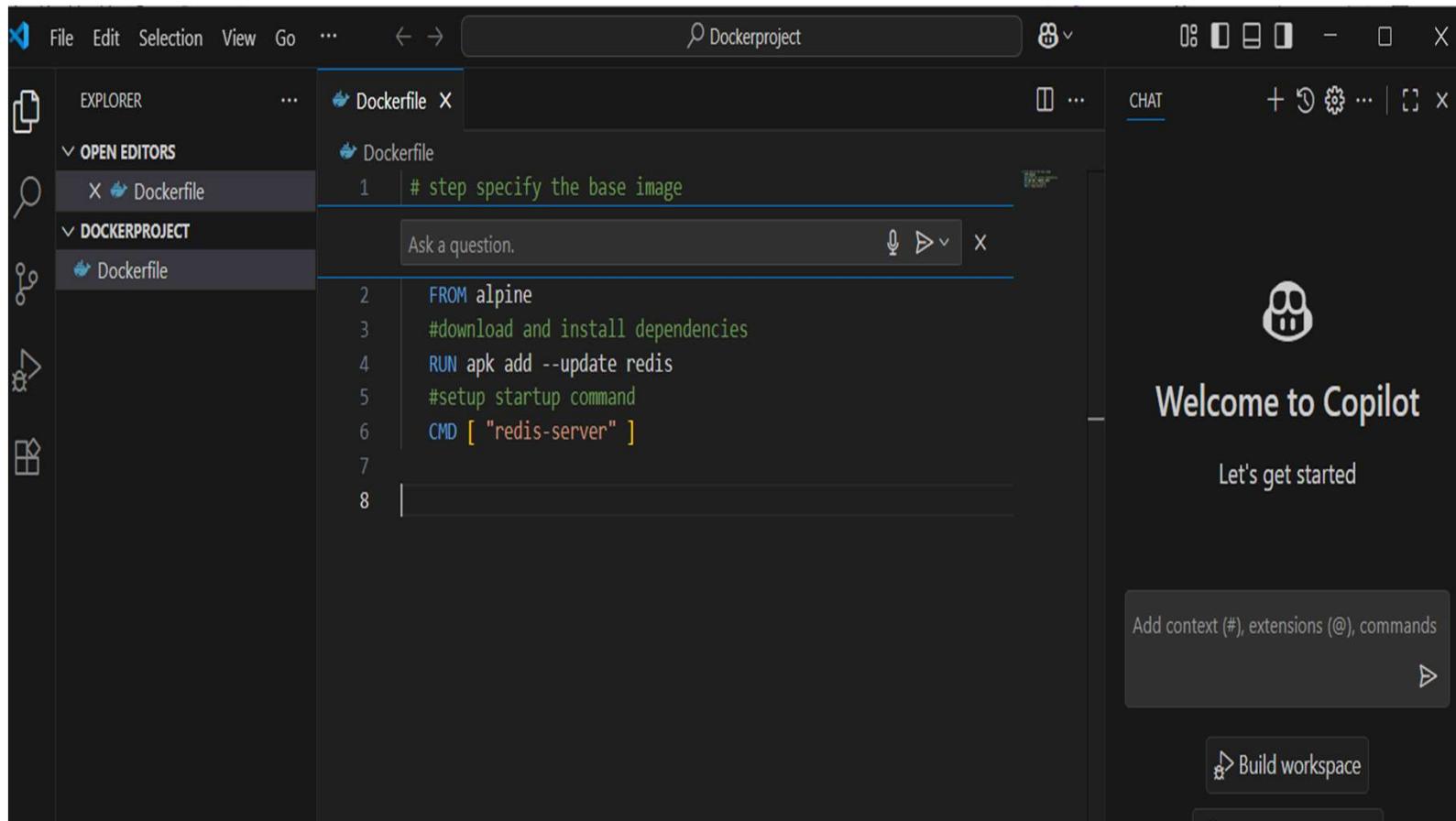
---

- a. In file Dockerfile, type commands as below

```
# step specify the base image  
FROM alpine  
#download and install dependencies  
RUN apk add --update redis  
#setup startup command  
CMD [ "redis-server" ]
```

- b. Control+S to save

# Code in Docker file



The screenshot shows the Microsoft Copilot extension integrated into the Visual Studio Code interface. The workspace is titled "Dockerproject". The left sidebar displays the "EXPLORER" view with "OPEN EDITORS" showing a "Dockerfile" and "DOCKERPROJECT" showing another "Dockerfile". The main editor area contains a Dockerfile with the following code:

```
1 # step specify the base image
2 FROM alpine
3 #download and install dependencies
4 RUN apk add --update redis
5 #setup startup command
6 CMD [ "redis-server" ]
```

A "CHAT" panel on the right side of the editor shows a "Welcome to Copilot" message and a "Let's get started" button. Below the message is a text input field with placeholder text: "Add context (#), extensions (@), commands".

# Step 3: Build Dockerfile to create image

---

Open terminal in Visual Studio Code and type

`docker build -t csmimage:01 .`

-t helps to tag name `csmimage:01` .

`csmimage` means give as username

`01` means give as version of your image

The screenshot shows the Visual Studio Code interface with a dark theme. The left sidebar has icons for file operations, search, and navigation. The Explorer sidebar shows an open editor for a 'Dockerfile' and a 'DockerProject' folder containing another 'Dockerfile'. The main editor area displays a Dockerfile with the following content:

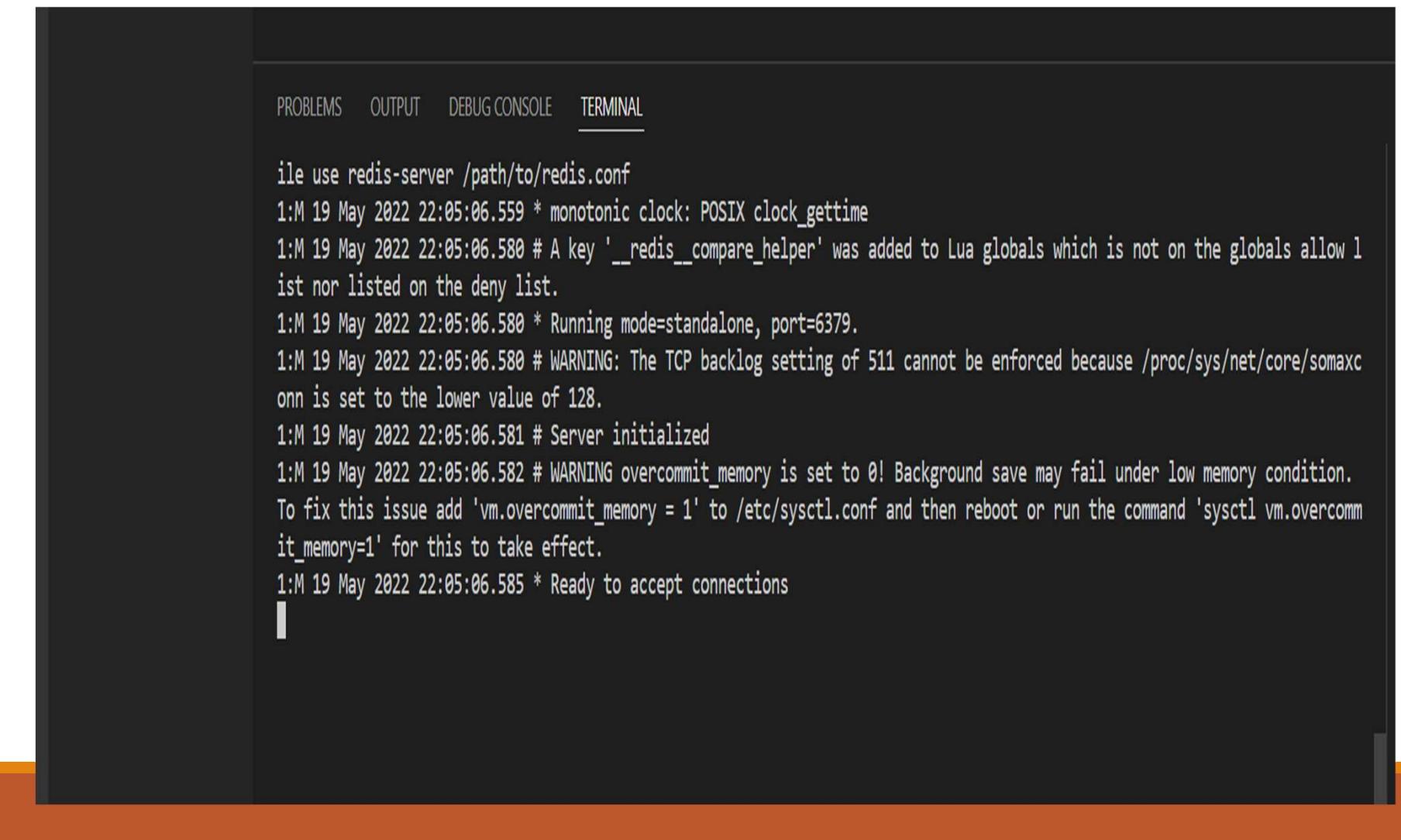
```
1 # step specify the base image
2 FROM alpine
3 #download and install dependencies
4 RUN apk add --update redis
5 #setup startup command
6 CMD [ "redis-server" ]
```

The terminal tab is active, showing the command `docker build -t csmimage:01 .` being run in a powershell window. The output shows the build process:

```
PS C:\Users\BHARGAVI KUMBHAM\OneDrive\Desktop\Dockerproject> docker build -t csmimage:01 .
● [+] Building 2.3s (6/6) FINISHED                                            docker:desktop-linux
=> [internal] load build definition from Dockerfile                         0.1s
=> => transferring dockerfile: 213B                                         0.0s
=> [internal] load metadata for docker.io/library/alpine:latest            1.9s
=> [internal] load .dockerignore                                           0.0s
=> => transferring context: 2B                                             0.0s
=> [1/2] FROM docker.io/library/alpine:latest@sha256:4bcff63911fcb4448bd4fda 0.0s
=> => resolve docker.io/library/alpine:latest@sha256:4bcff63911fcb4448bd4fda 0.0s
=> => CACHE [2/2] RUN apk add --update redis                                0.0s
=> => exporting to image                                                 0.1s
=> => exporting layers                                                 0.0s
=> => exporting manifest sha256:d5e828ba3cfe1bbd6dea45f994e535bd127db8f06acd 0.0s
=> => exporting config sha256:b77690587e622a863c0293ef5b3402ba5736bcd42aadaf 0.0s
=> => exporting attestation manifest sha256:d15888d5d846f8f3ce252dd1b1313704 0.0s
=> => exporting manifest list sha256:3ab594f86bacbfba26a18949ba9d3ec1a8cbce1 0.0s
=> => naming to docker.io/library/csmimage:01                             0.0s
=> => unpacking to docker.io/library/csmimage:01                           0.0s
```

The terminal also shows the prompt `PS C:\Users\BHARGAVI KUMBHAM\OneDrive\Desktop\Dockerproject>`.

# Continued... if prompt not appear close and open again



A screenshot of a terminal window titled "TERMINAL". The window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL, with the TERMINAL tab underlined. The terminal displays the following log output from a Redis server:

```
file use redis-server /path/to/redis.conf
1:M 19 May 2022 22:05:06.559 * monotonic clock: POSIX clock_gettime
1:M 19 May 2022 22:05:06.580 # A key '__redis__compare_helper' was added to Lua globals which is not on the globals allow list nor listed on the deny list.
1:M 19 May 2022 22:05:06.580 * Running mode=standalone, port=6379.
1:M 19 May 2022 22:05:06.580 # WARNING: The TCP backlog setting of 511 cannot be enforced because /proc/sys/net/core/somaxconn is set to the lower value of 128.
1:M 19 May 2022 22:05:06.581 # Server initialized
1:M 19 May 2022 22:05:06.582 # WARNING overcommit_memory is set to 0! Background save may fail under low memory condition.
To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
1:M 19 May 2022 22:05:06.585 * Ready to accept connections
```

# Continued...

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL
1:M 19 May 2022 22:05:06.582 # WARNING overcommit_memory is set to 0! Background save may fail under low memory condition.  
To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcom  
it_memory=1' for this to take effect.  
1:M 19 May 2022 22:05:06.585 * Ready to accept connections  
  
Session contents restored from 5/20/2022 at 4:05:27 AM  
  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Try the new cross-platform PowerShell https://aka.ms/pscore6  
PS C:\Users\SHRADHA\OneDrive\Desktop\docker-custom> |
```

# \$docker images

## Here image with csmimage created



```
File Edit Selection View Go ... Dockerfile X Dockerfile X Dockerfile # step specify the base image FROM alpine #download and install dependencies RUN apk add --update redis #setup startup command CMD [ "redis-server" ]
```

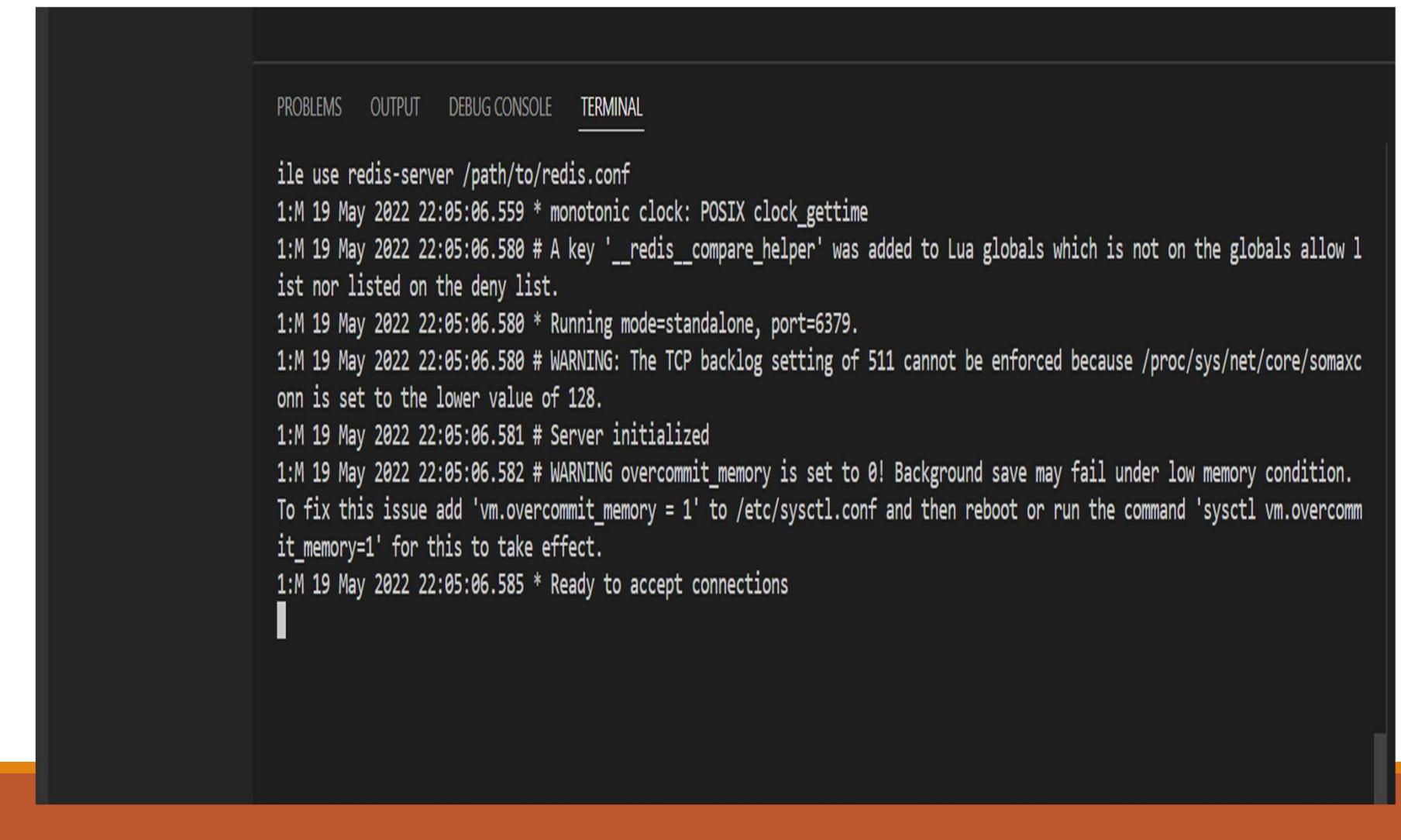
Ask a question.

```
=> => resolve docker.io/library/alpine:latest@sha256:4bcff63911fcb4448bd4fda 0.0s  
=> CACHED [2/2] RUN apk add --update redis 0.0s  
=> exporting to image 0.1s  
=> exporting layers 0.0s  
=> => exporting manifest sha256:d5e828ba3cfe1bbd6dea45f994e535bd127db8f06acd 0.0s  
=> => exporting config sha256:b77690587e622a863c0293ef5b3402ba5736bcd42aadaf 0.0s  
=> => exporting attestation manifest sha256:d15888d5d846f8f3ce252dd1b1313704 0.0s  
=> => exporting manifest list sha256:3ab594f86bacbfba26a18949ba9d3ec1a8cbce1 0.0s  
=> => naming to docker.io/library/csmimage:01 0.0s  
=> => unpacking to docker.io/library/csmimage:01 0.0s
```

```
PS C:\Users\BHARGAVI KUMBHAM\OneDrive\Desktop\Dockerproject> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
Kimage	01	f5a4b63b7121	11 minutes ago	27.8MB
csmimage	01	3ab594f86bac	11 minutes ago	27.8MB
kumbhambhargavi/redisnew	latest	003532995b6a	40 hours ago	200MB
kumbhambhargavi/redis1	latest	5cb5f49974cf	45 hours ago	200MB
redisnew	latest	d2315726eed8	2 days ago	200MB
kumbhambhargavi/ubuntu	latest	86bdeeb0a751	2 days ago	117MB
dockerhub-username/ubuntu	latest	86bdeeb0a751	2 days ago	117MB
kumbham/ubuntu	latest	86bdeeb0a751	2 days ago	117MB
wordpress	latest	c5f075fe71c9	8 days ago	1.04GB
hello-world	latest	a0dfb02aac21	12 days ago	20.3kB

# Continued... if prompt not appear close and open again



A screenshot of a terminal window titled "TERMINAL". The window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL, with the TERMINAL tab underlined. The terminal displays the following log output from a Redis server:

```
file use redis-server /path/to/redis.conf
1:M 19 May 2022 22:05:06.559 * monotonic clock: POSIX clock_gettime
1:M 19 May 2022 22:05:06.580 # A key '__redis__compare_helper' was added to Lua globals which is not on the globals allow list nor listed on the deny list.
1:M 19 May 2022 22:05:06.580 * Running mode=standalone, port=6379.
1:M 19 May 2022 22:05:06.580 # WARNING: The TCP backlog setting of 511 cannot be enforced because /proc/sys/net/core/somaxconn is set to the lower value of 128.
1:M 19 May 2022 22:05:06.581 # Server initialized
1:M 19 May 2022 22:05:06.582 # WARNING overcommit_memory is set to 0! Background save may fail under low memory condition.
To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
1:M 19 May 2022 22:05:06.585 * Ready to accept connections
```

# Exercise 5.

## Publishing docker image into docker repository

First create docker hub account at dockerhub.com

---

Login to docker at shell by \$ **docker login**

```
● PS C:\Users\BHARGAVI KUMBHAM\OneDrive\Desktop\Dockerproject> docker login

    USING WEB-BASED LOGIN

    ⓘ Info → To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: GQJW-JKSH
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate

Waiting for authentication in the browser...

Login Succeeded
○ PS C:\Users\BHARGAVI KUMBHAM\OneDrive\Desktop\Dockerproject>
```

\$ docker tag csmimage:01 kumbhambhargavi/secondimage

image name    version/tag    docker id    tagname  
latest

---

```
● PS C:\Users\BHARGAVI KUMBHAM\OneDrive\Desktop\Dockerproject> docker tag csmimage:01 kumbhambhargavi/secondimage
● PS C:\Users\BHARGAVI KUMBHAM\OneDrive\Desktop\Dockerproject> docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
kimage              01       f5a4b63b7121   29 minutes ago  27.8MB
kumbhambhargavi/secondimage  latest    3ab594f86bac   29 minutes ago  27.8MB
csmimage            01       3ab594f86bac   29 minutes ago  27.8MB
kumbhambhargavi/redisnew  latest    003532995b6a   41 hours ago   200MB
kumbhambhargavi/redis1   latest    5cb5f49974cf   45 hours ago   200MB
redisnew             latest    d2315726eed8   2 days ago    200MB
dockerhub-username/ubuntu  latest    86bdeeb0a751   2 days ago    117MB
kumbham/ubuntu        latest    86bdeeb0a751   2 days ago    117MB
kumbhambhargavi/ubuntu  latest    86bdeeb0a751   2 days ago    117MB
wordpress             latest    c5f075fe71c9   8 days ago    1.04GB
hello-world           latest    a0dfb02aac21   12 days ago   20.3kB
ubuntu                latest    7c06e91f61fa   3 weeks ago   117MB
mysql                 5.7      4bc6bc963e6d   20 months ago  689MB
○ PS C:\Users\BHARGAVI KUMBHAM\OneDrive\Desktop\Dockerproject>
```

# \$ docker images

## shows tagged image kumbhambhargavi/secondimage

```
PS C:\Users\BHARGAVI KUMBHAM\OneDrive\Desktop\Dockerproject> docker tag csmimage:01 kumbhambhargavi/secondimage
PS C:\Users\BHARGAVI KUMBHAM\OneDrive\Desktop\Dockerproject> docker images
REPOSITORY          TAG      IMAGE ID   CREATED        SIZE
kimage              01       f5a4b63b7121  29 minutes ago  27.8MB
kumbhambhargavi/secondimage  latest    3ab594f86bac  29 minutes ago  27.8MB
csmimage            01       3ab594f86bac  29 minutes ago  27.8MB
kumbhambhargavi/redisnew  latest    003532995b6a  41 hours ago   200MB
kumbhambhargavi/redis1   latest    5cb5f49974cf  45 hours ago   200MB
redisnew             latest    d2315726eed8  2 days ago    200MB
dockerhub-username/ubuntu  latest    86bdeeb0a751  2 days ago    117MB
kumbham/ubuntu       latest    86bdeeb0a751  2 days ago    117MB
kumbhambhargavi/ubuntu  latest    86bdeeb0a751  2 days ago    117MB
wordpress            latest    c5f075fe71c9  8 days ago    1.04GB
hello-world          latest    a0dfb02aac21  12 days ago   20.3kB
ubuntu               latest    7c06e91f61fa  3 weeks ago   117MB
mysql                5.7      4bc6bc963e6d  20 months ago  689MB
PS C:\Users\BHARGAVI KUMBHAM\OneDrive\Desktop\Dockerproject>
```

**\$ docker push kumbhambhargavi/secondimage**

Pushes image into docker repo with id  
**kumbhambhargavi**

---

```
PS C:\Users\BHARGAVI KUMBHAM\OneDrive\Desktop\Dockerproject> docker push kumbhambhargavi/secondimage
Using default tag: latest
The push refers to repository [docker.io/kumbhambhargavi/secondimage]
0c701b0fffdf: Pushed
1a271db51698: Pushed
9824c27679d3: Pushed
latest: digest: sha256:3ab594f86bacbfba26a18949ba9d3ec1a8cbce1221076c9e7d7006a7975d2042 size: 855
PS C:\Users\BHARGAVI KUMBHAM\OneDrive\Desktop\Dockerproject>
```

Goto docker hub account/repository. Now here no image with name **kumbhambhargavi/secondimage**. To get it press on refresh

---

hub.docker.com/repositories/kumbhambhargavi

hub Explore My Hub

Search Docker Hub CtrlK

kumbhambhargavi Docker Personal

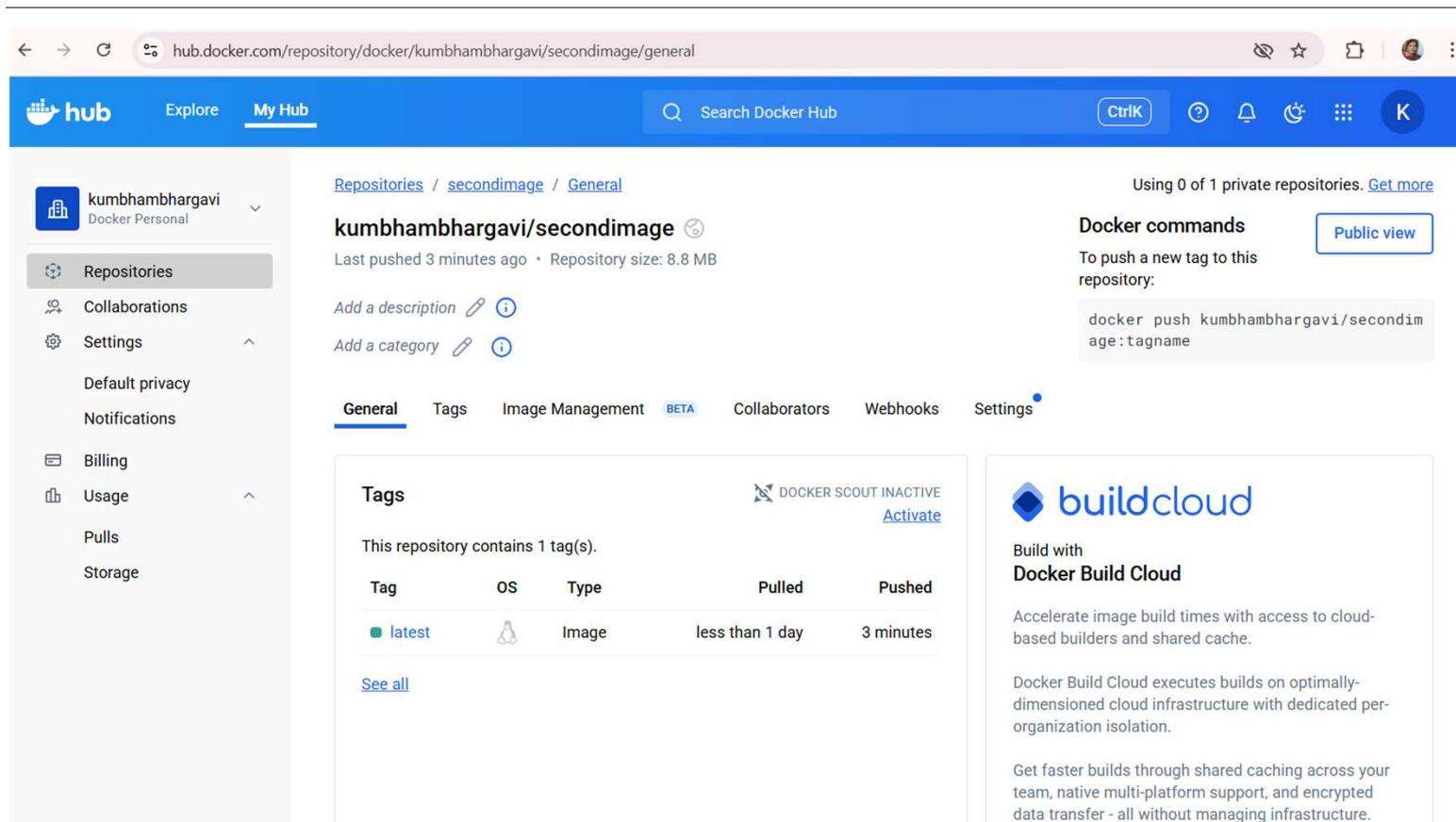
Repositories

All repositories within the kumbhambhargavi namespace.

Search by repository name All content Create a repository

Name	Last Pushed	Contains	Visibility	Scout
kumbhambhargavi/secondimage	2 minutes ago	IMAGE	Public	Inactive
kumbhambhargavi/redisnew	1 day ago	IMAGE	Public	Inactive
kumbhambhargavi/redis1	2 days ago	IMAGE	Public	Inactive
kumbhambhargavi/ubuntu	3 days ago	IMAGE	Public	Inactive

After refreshing, press on  
**kumbhambhargavi/secondimage**  
and see next slide ,Here shows image pushed



The screenshot shows a browser window displaying the Docker Hub interface. The URL in the address bar is `hub.docker.com/repository/docker/kumbhambhargavi/secondimage/general`. The top navigation bar includes links for 'hub', 'Explore', and 'My Hub' (which is currently selected). A search bar says 'Search Docker Hub'. On the right side of the header are icons for notifications, messages, and profile.

The main content area shows a repository for user `kumbhambhargavi` named `secondimage`. It was last pushed 3 minutes ago and has a size of 8.8 MB. There are options to 'Add a description' and 'Add a category'. Below these are tabs for 'General', 'Tags', 'Image Management (BETA)', 'Collaborators', 'Webhooks', and 'Settings' (selected).

The 'Tags' section shows one tag: `latest`, which is an `Image` type. It was pulled less than 1 day ago and pushed 3 minutes ago.

To the right of the repository details, there's a sidebar for Docker commands, a 'Public view' button, and a note about using 0 of 1 private repositories. It also shows a command line snippet: `docker push kumbhambhargavi/secondimage:tagname`.

On the right side of the page, there's an advertisement for 'buildcloud' with the text: 'Build with Docker Build Cloud. Accelerate image build times with access to cloud-based builders and shared cache. Docker Build Cloud executes builds on optimally-dimensioned cloud infrastructure with dedicated per-organization isolation. Get faster builds through shared caching across your team, native multi-platform support, and encrypted data transfer - all without managing infrastructure.'

# Here one can press on tag, or share this image using command pull

The screenshot shows the Docker Hub interface for the repository `kumbhambhargavi/secondimage`. The user is on the 'Tags' tab. A tooltip for the 'latest' tag displays the command `docker pull kumbhambhargavi/secondimage:latest`.

Digest	OS/ARCH	Last pull	Compressed size
<a href="#">d5e828ba3cfe</a>	linux/amd64	less than 1 day	8.79 MB

# Exercise 6. Pulling pushed image, running that image, executing in interactive mode

The screenshot shows the Docker Hub interface for the repository `kumbhambhargavi/secondimage`. The repository was last pushed 20 minutes ago and has a size of 8.8 MB. The `Tags` tab is selected, showing the `latest` tag which was pushed 20 minutes ago by `kumbhambhargavi`. Below the tags, the `Digest` is listed as `d5e828ba3cfe`, the `OS/ARCH` is `linux/amd64`, the `Last pull` is less than 1 day, and the `Compressed size` is 8.79 MB. A blue button labeled `Copy` is visible next to the `docker pull kumbhambhargavi/secondimage` command.

hub Explore My Hub

Search Docker Hub CtrlK

Repositories / secondimage / Tags

Using 0 of 1 private repositories. [Get more](#)

**kumbhambhargavi/secondimage**

Last pushed 20 minutes ago • Repository size: 8.8 MB

Add a description

Add a category

**Docker commands**

To push a new tag to this repository:

```
docker push kumbhambhargavi/secondimage:tagname
```

Public view

Repositories

Collaborations

Settings

Default privacy

Notifications

Billing

Usage

Pulls

Storage

General Tags Image Management BETA Collaborators Webhooks Settings

Sort by Newest Filter tags

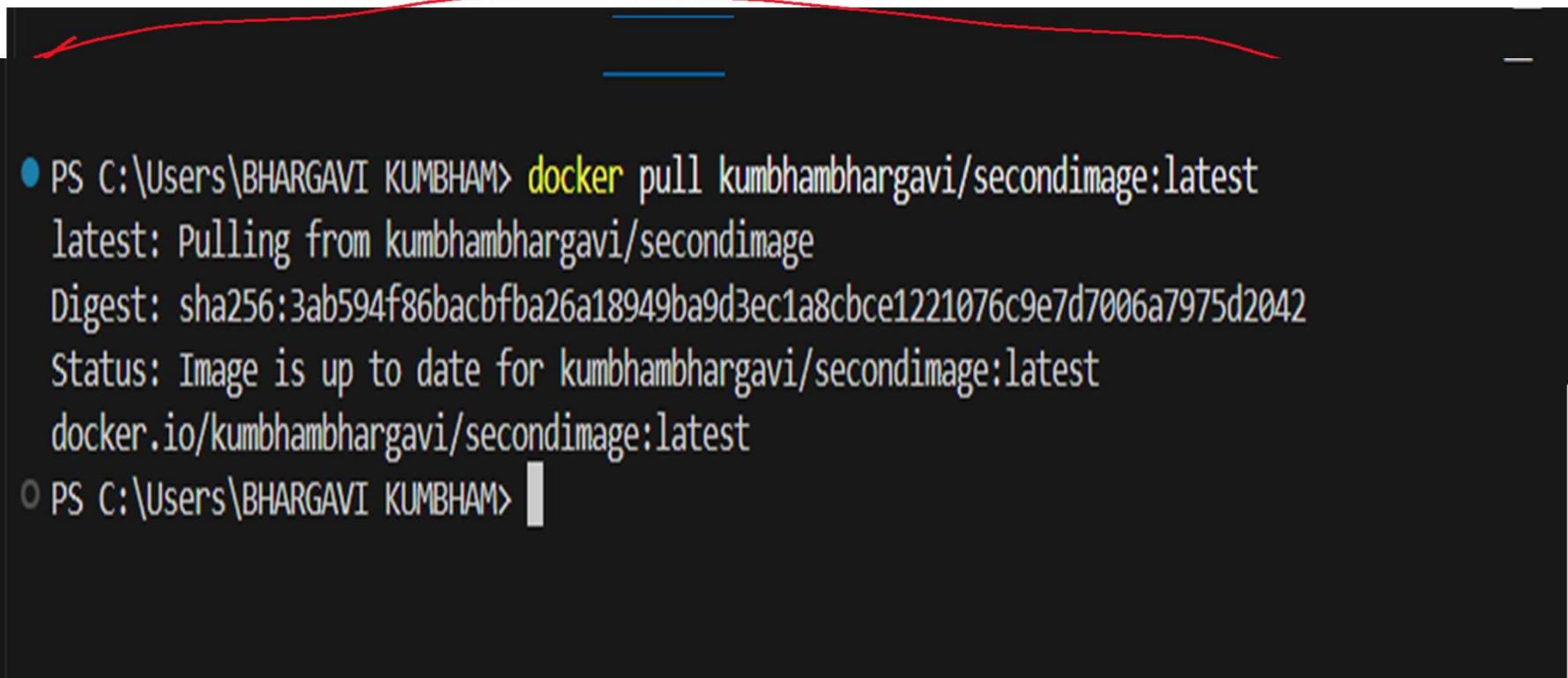
TAG	Digest	OS/ARCH	Last pull	Compressed size
<code>latest</code>	<code>d5e828ba3cfe</code>	<code>linux/amd64</code>	less than 1 day	8.79 MB

`docker pull kumbhambhargavi/secondimage`

Open another terminal in Visual Studio Code

Paste Pull command

**docker pull kumbhambhargavi/secondimage:latest**



```
● PS C:\Users\BHARGAVI KUMBHAM> docker pull kumbhambhargavi/secondimage:latest
latest: Pulling from kumbhambhargavi/secondimage
Digest: sha256:3ab594f86bacbfba26a18949ba9d3ec1a8cbce1221076c9e7d7006a7975d2042
Status: Image is up to date for kumbhambhargavi/secondimage:latest
        docker.io/kumbhambhargavi/secondimage:latest
○ PS C:\Users\BHARGAVI KUMBHAM>
```

# Run pulled image

```
docker run -it
```

```
kumbhambhargavi/secondimage:latest
```

```
Digest: sha256:3ab594f86bacbfba26a18949ba9d3ec1a8cbce1221076c9e7d7006a7975d2042
```

```
Status: Image is up to date for kumbhambhargavi/secondimage:latest
```

```
docker.io/kumbhambhargavi/secondimage:latest
```

```
PS C:\Users\BHARGAVI KUMBHAM> docker run -it kumbhambhargavi/secondimage:latest
```

```
1:C 21 Aug 2025 07:30:34.886 * o000o000o000 Redis is starting o000o000o000
```

```
1:C 21 Aug 2025 07:30:34.886 * Redis version=8.0.3, bits=64, commit=00000000, modified=1, pid=1, just started
```

```
1:C 21 Aug 2025 07:30:34.886 # Warning: no config file specified, using the default config. In order to specify a  
use redis-server /path/to/redis.conf
```

```
1:M 21 Aug 2025 07:30:34.886 * monotonic clock: POSIX clock_gettime
```

```
Redis Open Source  
8.0.3 (00000000/1) 64 bit  
Running in standalone mode  
Port: 6379  
PID: 1
```

```
https://redis.io
```

```
1:M 21 Aug 2025 07:30:34.889 * Server initialized
```

```
1:M 21 Aug 2025 07:30:34.890 * Ready to accept connections tcp
```