UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

Practical 9 : Docker

Docker is a container management system. One use a Docker Image to build a software/application with the same setup as the original developer, similar to Vagrant's Vagrantfile. Docker image unlike Vagrantfile contains all setups in different layers, e.g.;

1. Base layer: e.g. Ubuntu
2. Layer 2: e.g. Software files
3. Layer 3: e.g. Dependencies
4. Layer 4: e.g. Configurations

Subsequent to building a Docker image, a container is built. A container is a running instance of a Docker image. A container contains everything that is needed to run the project files.

Q: What is the difference between Docker and Virtual Machines? Answer at the end of this practical exploration of Docker.

Preliminaries.

Firstly, download Docker Desktoip from https://www.docker.com/products/docker-desktop as shown in Figure 1. Follow through default installation instructions to ensure complete Docker installation.

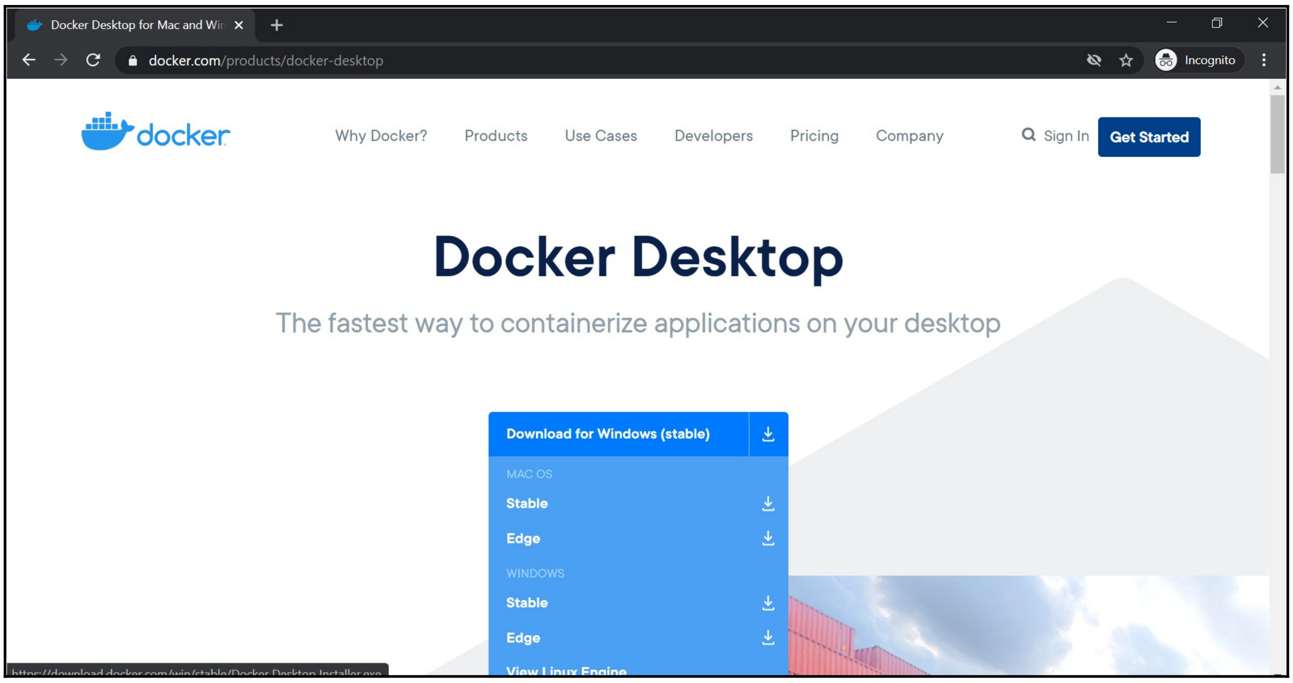
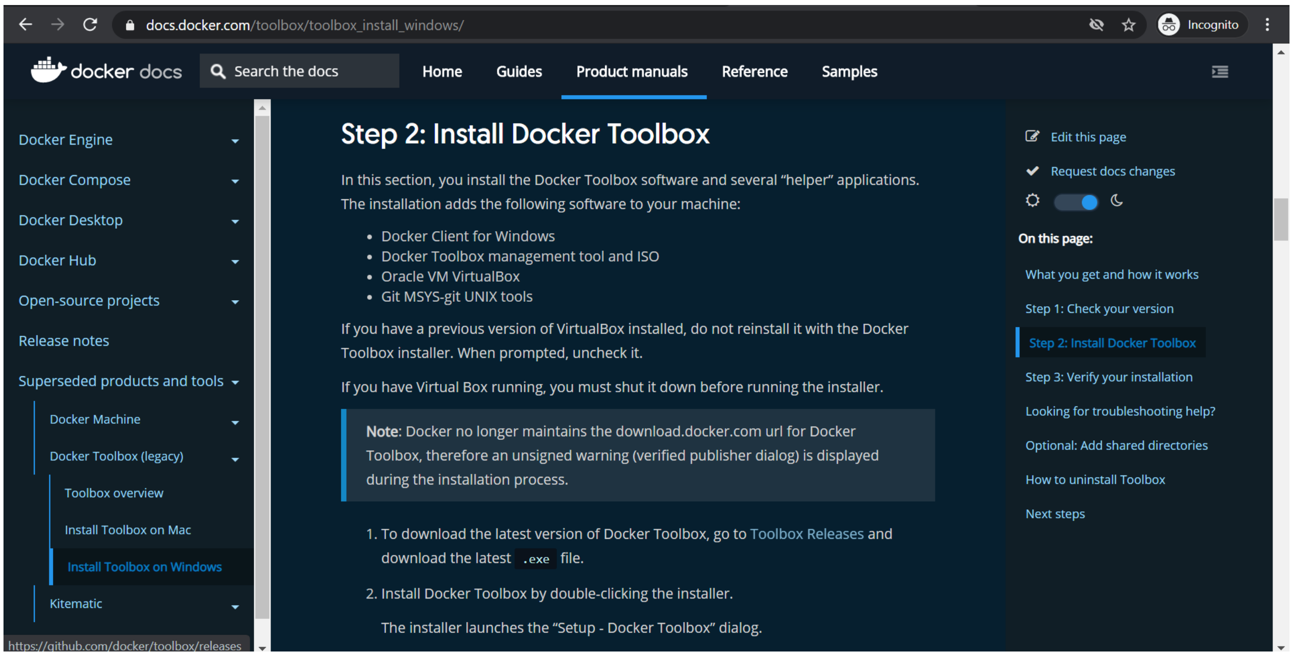

Docker Desktop for Mac and Wi
docker.com/products/docker-desktop
Incognito
docker
Why Docker?
Products
Use Cases
Developers
Pricing
Company
Sign In
Get Started
Docker Desktop
The fastest way to containerize applications on your desktop
Download for Windows (stable)
MAC OS
Stable
Edge
WINDOWS
Stable
Edge
View Linux Engine

Figure 1: Download and install Docker Desktop.

If installation is shows an error of “"Installation failed: one prerequisite is not fullfilled" Docker Desktop requires Windows 10 Pro/Enterprise (15063+) or Windows 10 Home (19018+).” install the Docker Toolbox from https://docs.docker.com/toolbox/toolbox_install_windows/ instead as shown in Figure 2.


docs.docker.com/toolbox/toolbox_install_windows/
Incognito
docker docs
Search the docs
Home
Guides
Product manuals
Reference
Samples
Docker Engine
Docker Compose
Docker Desktop
Docker Hub
Open-source projects
Release notes
Superseded products and tools
Docker Machine
Docker Toolbox (legacy)
Toolbox overview
Install Toolbox on Mac
Install Toolbox on Windows
Kitematic
https://github.com/docker/toolbox/releases
Edit this page
Request docs changes
On this page:
What you get and how it works
Step 1: Check your version
Step 2: Install Docker Toolbox
Step 3: Verify your installation
Looking for troubleshooting help?
Optional: Add shared directories
How to uninstall Toolbox
Next steps

Step 2: Install Docker Toolbox

In this section, you install the Docker Toolbox software and several "helper" applications. The installation adds the following software to your machine:

- Docker Client for Windows
- Docker Toolbox management tool and ISO
- Oracle VM VirtualBox
- Git MSYS-git UNIX tools

If you have a previous version of VirtualBox installed, do not reinstall it with the Docker Toolbox installer. When prompted, uncheck it.

If you have Virtual Box running, you must shut it down before running the installer.

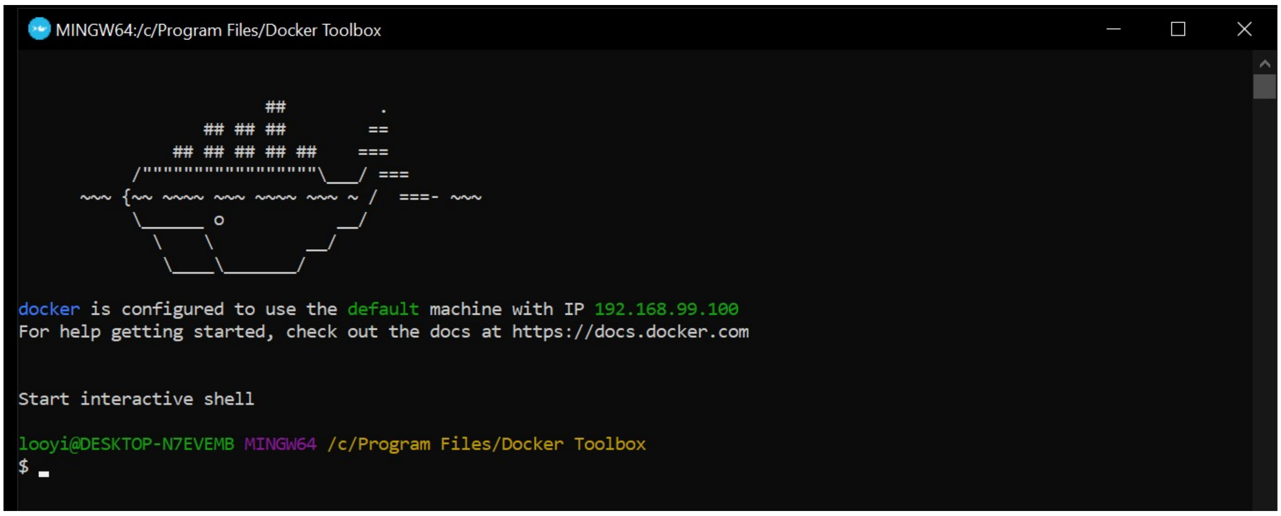
> **Note**: Docker no longer maintains the download.docker.com url for Docker Toolbox, therefore an unsigned warning (verified publisher dialog) is displayed during the installation process.

1. To download the latest version of Docker Toolbox, go to Toolbox Releases and download the latest `.exe` file.
2. Install Docker Toolbox by double-clicking the installer.

   The installer launches the "Setup - Docker Toolbox" dialog.

Figure 2: Download and install Docker Toolbox (if pre-requisite fails).

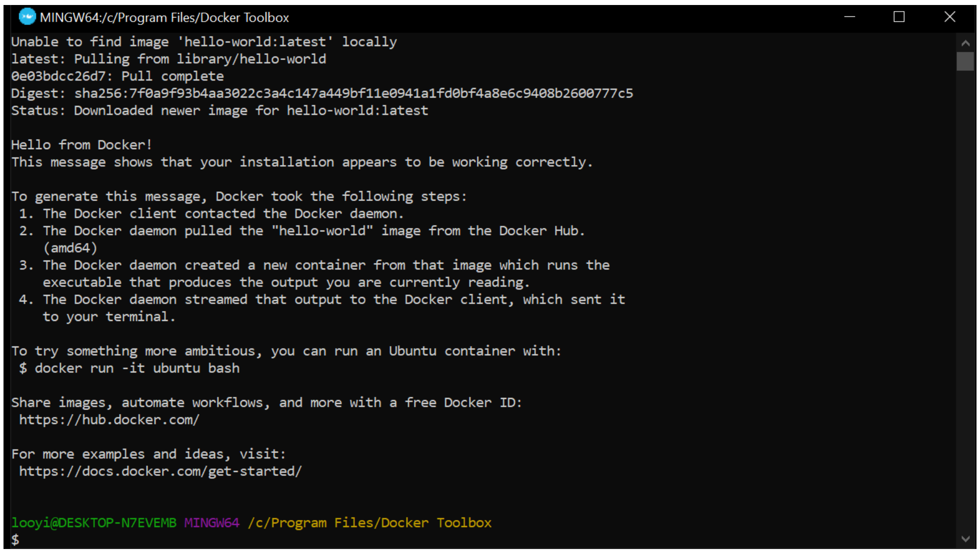
Verify the installation by running Docker Quickstart Terminal. Choose "Yes" for User Account Control prompt to allow VirtualBox to make changes to your computer. The installation is completed when the bash mode of Docker CLI is displayed as shown in Figure 3.



```
                        ##         .
                  ## ## ##        ==
               ## ## ## ## ##    ===
           /"""""""""""""""""\___/ ===
      ~~~ {~~ ~~~~ ~~~ ~~~~ ~~~ ~ /  ===- ~~~
           \______ o           __/
             \    \         __/
              \____\_______/

docker is configured to use the default machine with IP 192.168.99.100
For help getting started, check out the docs at https://docs.docker.com


Start interactive shell

looyi@DESKTOP-N7EVEMB MINGW64 /c/Program Files/Docker Toolbox
$ _
```

Figure 3: Docker Bash CLI.

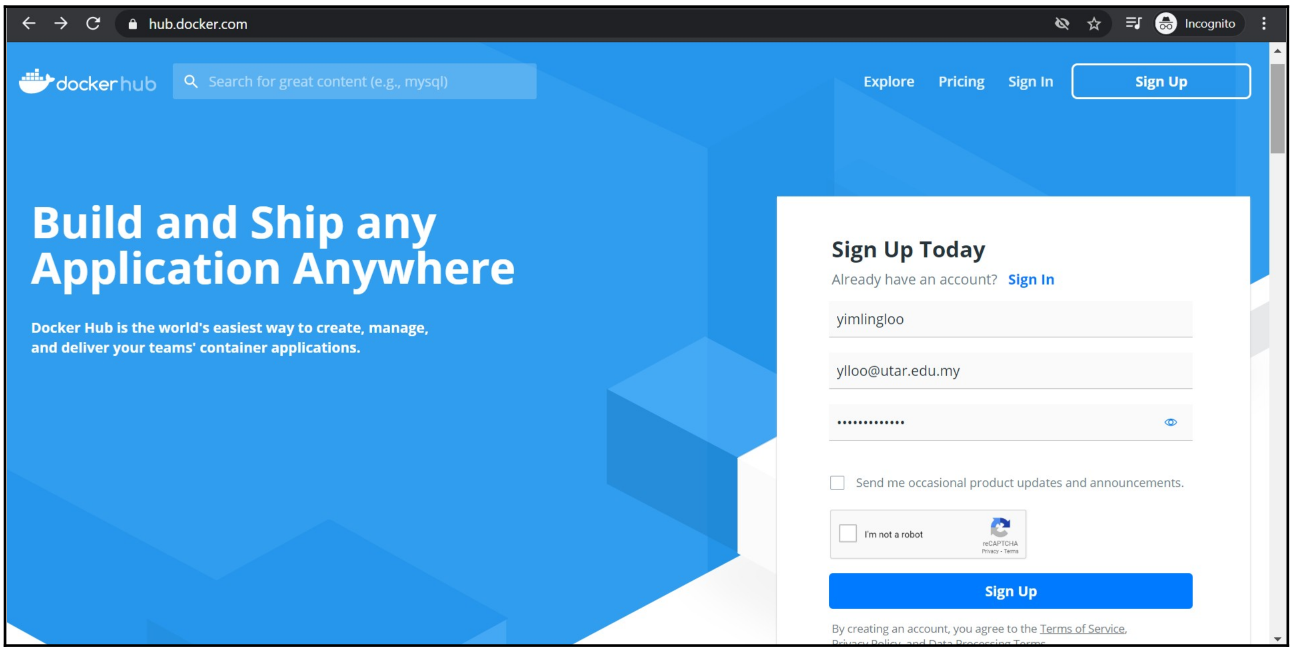
Try to execute `docker run hello-world` command and press "Enter". If the installation is successful, Docker Bash CLI will show as Figure 4.



```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest: sha256:7f0a9f93b4aa3022c3a4c147a449bf11e0941a1fd0bf4a8e6c9408b2600777c5
Status: Downloaded newer image for hello-world:latest

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/


looyi@DESKTOP-N7EVEMB MINGW64 /c/Program Files/Docker Toolbox
$
```

Figure 4: Successful Docker installation message.

Go to Docker Hub at https://hub.docker.com/ for all the official generic Docker images for different build ups. This is similar to generic vagrant boxes that one could find in Vagrant Cloud.
Do signup for a Docker account as we're going to push our application/software to Docker Hub for deployment as shown in Figure 5.



hub.docker.com

Build and Ship any Application Anywhere

Docker Hub is the world's easiest way to create, manage, and deliver your teams' container applications.

Sign Up Today

Already have an account? Sign In

yimlingloo

ylloo@utar.edu.my

••••••••••••••

Send me occasional product updates and announcements.

I'm not a robot

Sign Up

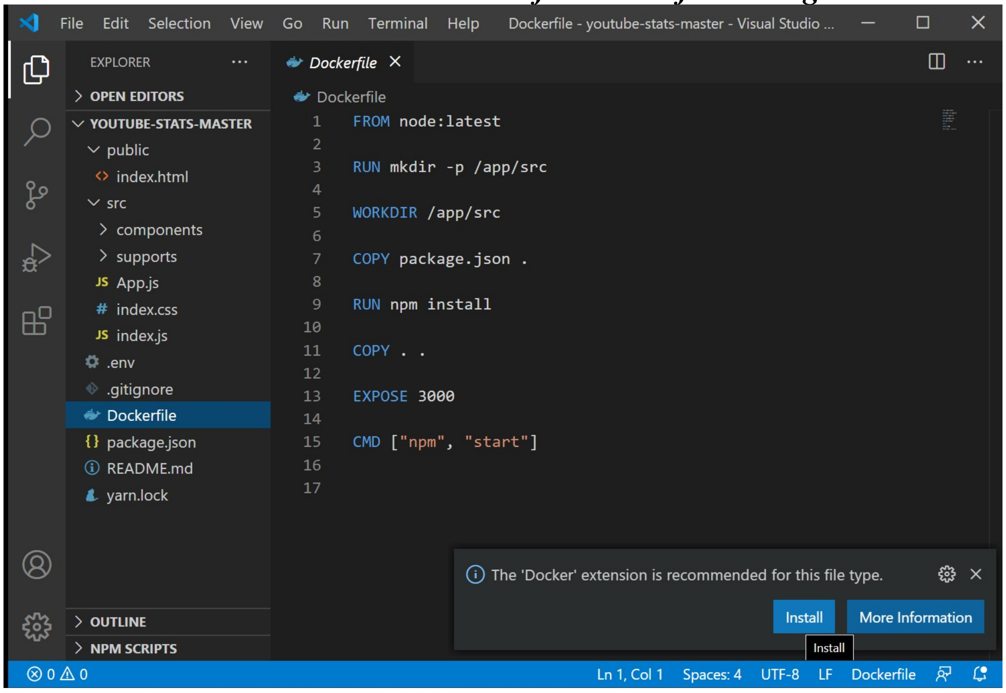
By creating an account, you agree to the Terms of Service, Privacy Policy, and Data Processing Terms.

Figure 5: Signup for a Docker Hub account.

Similar to Github, one may create Docker Repository to maintain the project deployment by storing Docker image in the repositories.

Getting Started.

Let's build a React App image and run the image in a Docker container. Clone the example React App from https://github.com/rodgtr1/youtube-stats and open the folder using VS Code as shown in Figure 6. ***Note: Install Docker extension in VS Code for Dockerfile editing.***



```
FROM node:latest

RUN mkdir -p /app/src

WORKDIR /app/src

COPY package.json .

RUN npm install

COPY . .

EXPOSE 3000

CMD ["npm", "start"]
```

The 'Docker' extension is recommended for this file type.

Figure 6: Open cloned React App folder.

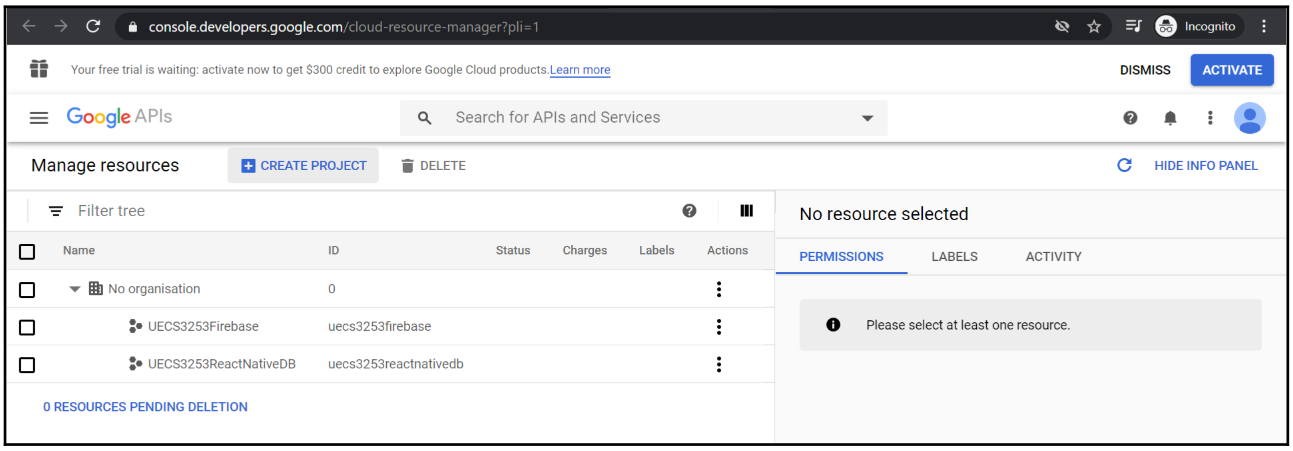
Dockerfile.

Dockerfile contains all the configurations needed to launch the example React App. The codes lists a set of instructions for Docker to launch the project files.

// The following line in Dockerfile initiate that the project files are to be launched on the latest node.js environment.
```
FROM node:latest
```
// The following line in Dockerfile initiate that the project files that are in "src" are to be placed inside the current folder in the new directory called "app".
```
RUN mkdir -p /app/src
```
// The following line in Dockerfile initiate that the working directory is the directory where the project files are
```
WORKDIR /app/src
```
// The following line in Dockerfile initiate the packages of json (dependencies) to the current working directory
```
COPY package.json .
```
// The following line in Dockerfile initiate installation of the packages installation
```
RUN npm install
```
// The following line in Dockerfile initiate copies all the sources into the working directory.
```
COPY . .
```
// The following line in Dockerfile initiate that the project files are to be launched on port 3000.
```
EXPOSE 3000
```
// The following line in Dockerfile initiate the launch of npm and start of CLI.
```
CMD ["npm", "start"]
```

Figure 7: Explanations on React App Dockerfile.

As the example React application uses fetches data from Youtube using API for the stats data, the instructions of getting API key need to be followed, which is provided in the Github repository's Readme.md "Instructions to Build and Run Docker Image." section. Series of how the trainer get the API key are shared in the Figures below (all followed the Instructions from Github repository).


console.developers.google.com/cloud-resource-manager?pli=1

Your free trial is waiting: activate now to get $300 credit to explore Google Cloud products. Learn more

Manage resources

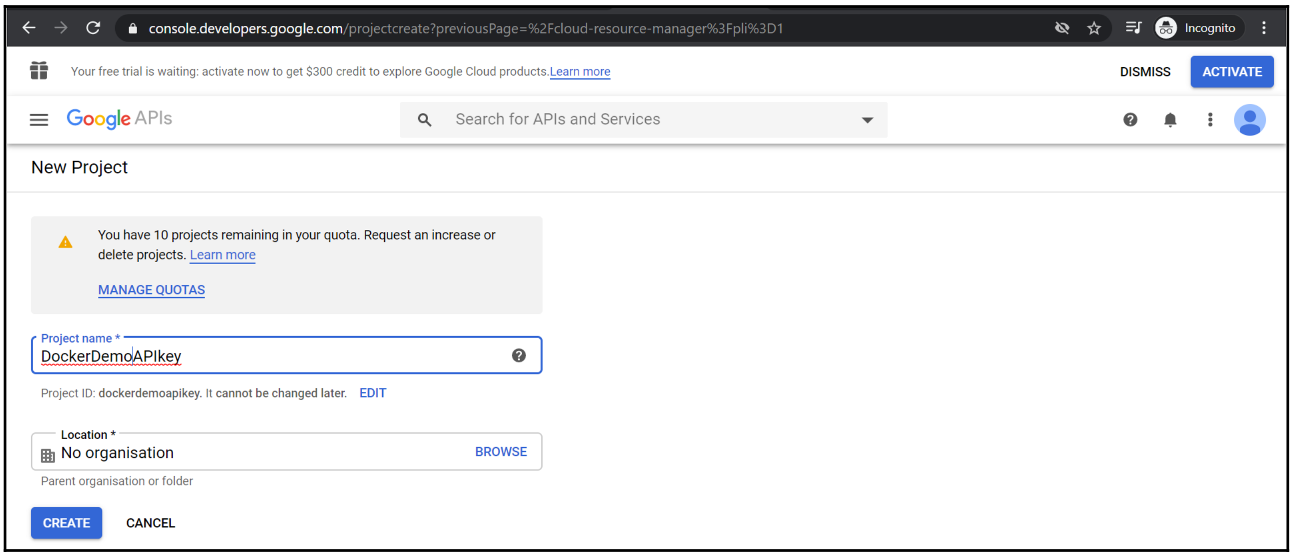
Name	ID	Status	Charges	Labels	Actions
No organisation	0				⋮
UECS3253Firebase	uecs3253firebase				⋮
UECS3253ReactNativeDB	uecs3253reactnativedb				⋮

0 RESOURCES PENDING DELETION

No resource selected

Please select at least one resource.

Figure 8: Click "CREATE PROJECT".


console.developers.google.com/projectcreate?previousPage=%2Fcloud-resource-manager%3Fpli%3D1

Your free trial is waiting: activate now to get $300 credit to explore Google Cloud products. Learn more

New Project

You have 10 projects remaining in your quota. Request an increase or delete projects. Learn more

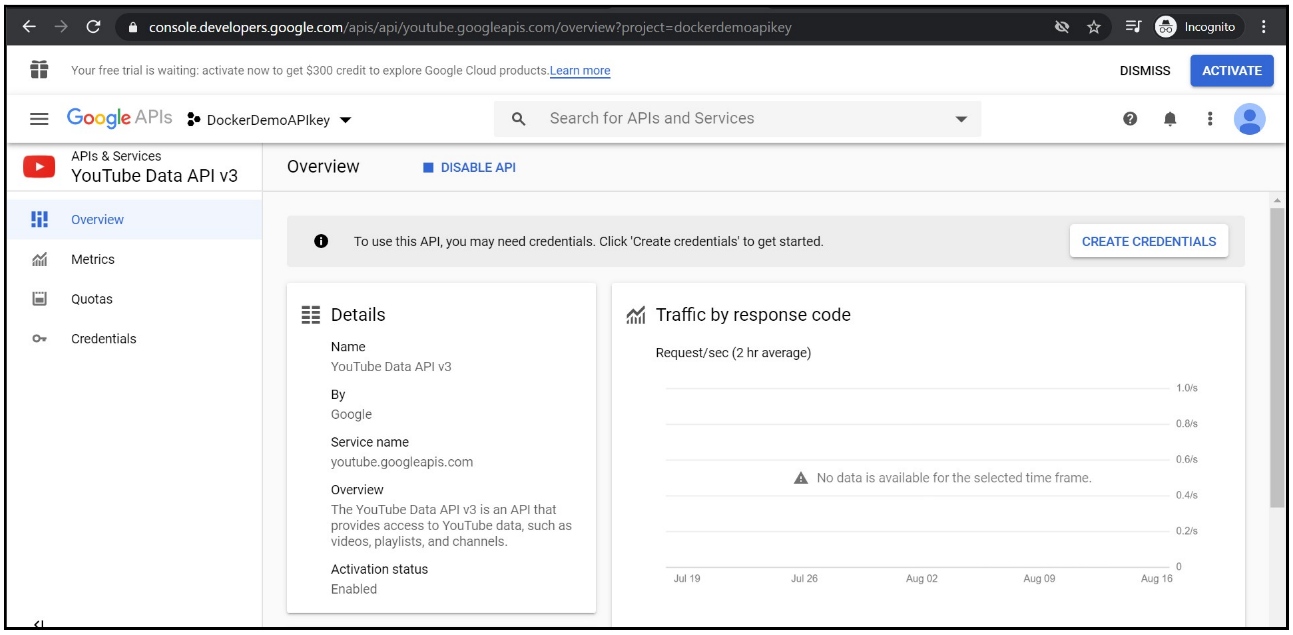
Project name *: DockerDemoAPIkey

Project ID: dockerdemoapikey. It cannot be changed later.

Location *: No organisation

Parent organisation or folder

Figure 9: Insert a project name.


console.developers.google.com/apis/api/youtube.googleapis.com/overview?project=dockerdemoapikey
Incognito
Your free trial is waiting: activate now to get $300 credit to explore Google Cloud products. Learn more
DISMISS
ACTIVATE
Google APIs
DockerDemoAPIkey
Search for APIs and Services
APIs & Services
YouTube Data API v3
Overview
DISABLE API
Overview
Metrics
Quotas
Credentials

To use this API, you may need credentials. Click 'Create credentials' to get started.

Details

Name
YouTube Data API v3

By
Google

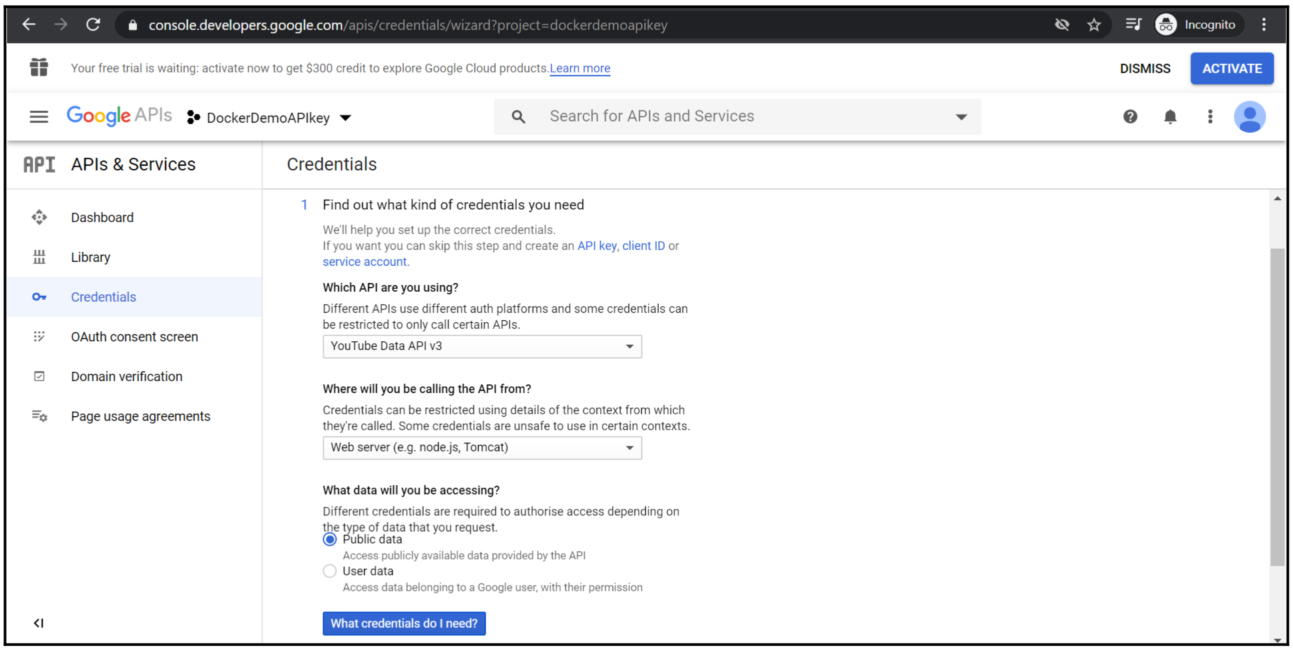
Service name
youtube.googleapis.com

Overview
The YouTube Data API v3 is an API that provides access to YouTube data, such as videos, playlists, and channels.

Activation status
Enabled

Traffic by response code
Request/sec (2 hr average)
1.0/s
0.8/s
0.6/s
0.4/s
0.2/s
0
No data is available for the selected time frame.
Jul 19
Jul 26
Aug 02
Aug 09
Aug 16

Figure 10: Enable YouTube Data API in Library of Google APIs.


console.developers.google.com/apis/credentials/wizard?project=dockerdemoapikey
Incognito
Your free trial is waiting: activate now to get $300 credit to explore Google Cloud products. Learn more
DISMISS
ACTIVATE
Google APIs
DockerDemoAPIkey
Search for APIs and Services
API APIs & Services
Credentials
Dashboard
Library
Credentials
OAuth consent screen
Domain verification
Page usage agreements

1 Find out what kind of credentials you need

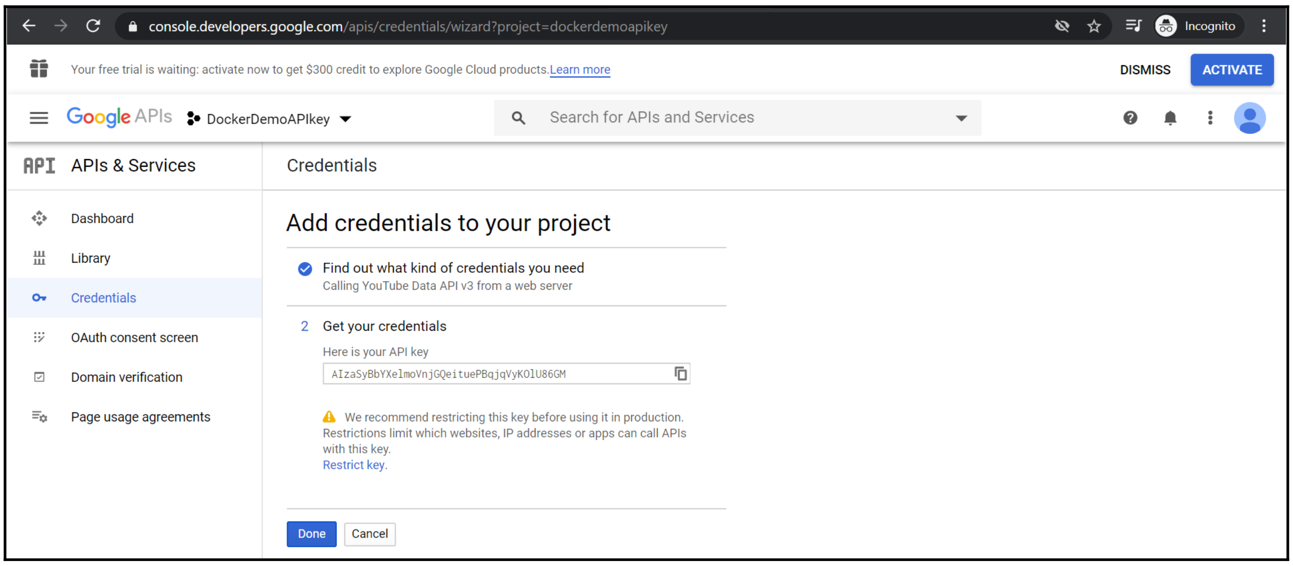
We'll help you set up the correct credentials.
If you want you can skip this step and create an API key, client ID or service account.

Which API are you using?
Different APIs use different auth platforms and some credentials can be restricted to only call certain APIs.
YouTube Data API v3

Where will you be calling the API from?
Credentials can be restricted using details of the context from which they're called. Some credentials are unsafe to use in certain contexts.
Web server (e.g. node.js, Tomcat)

What data will you be accessing?
Different credentials are required to authorise access depending on the type of data that you request.
[✓ Public data]
Access publicly available data provided by the API
User data
Access data belonging to a Google user, with their permission

Figure 11: Setup the credentials for the API.


console.developers.google.com/apis/credentials/wizard?project=dockerdemoapikey
Incognito
Your free trial is waiting: activate now to get $300 credit to explore Google Cloud products. Learn more
DISMISS
ACTIVATE
Google APIs
DockerDemoAPIkey
Search for APIs and Services
API APIs & Services
Dashboard
Library
Credentials
OAuth consent screen
Domain verification
Page usage agreements

Credentials

Add credentials to your project

Find out what kind of credentials you need
Calling YouTube Data API v3 from a web server

2 Get your credentials

Here is your API key

`AIzaSyBbYXelmoVnjGQeituePBqjqVyKOlU86GM`

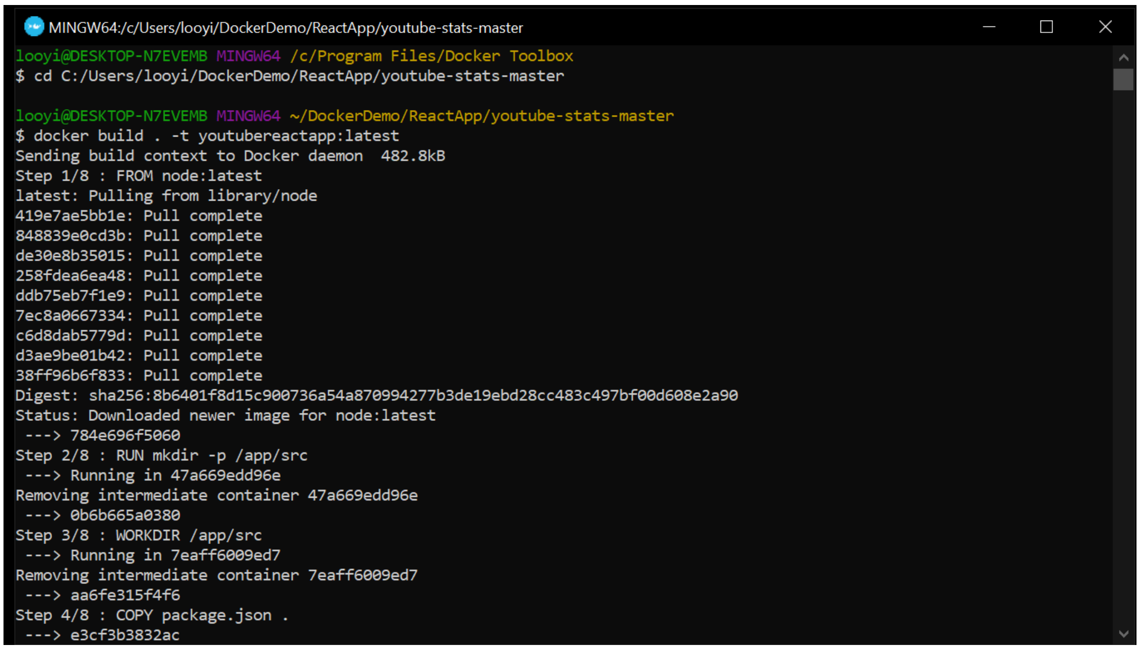
We recommend restricting this key before using it in production. Restrictions limit which websites, IP addresses or apps can call APIs with this key.
Restrict key.

Done Cancel

Figure 12: Retrieve API key to be filled in the .env file.

Docker Image.

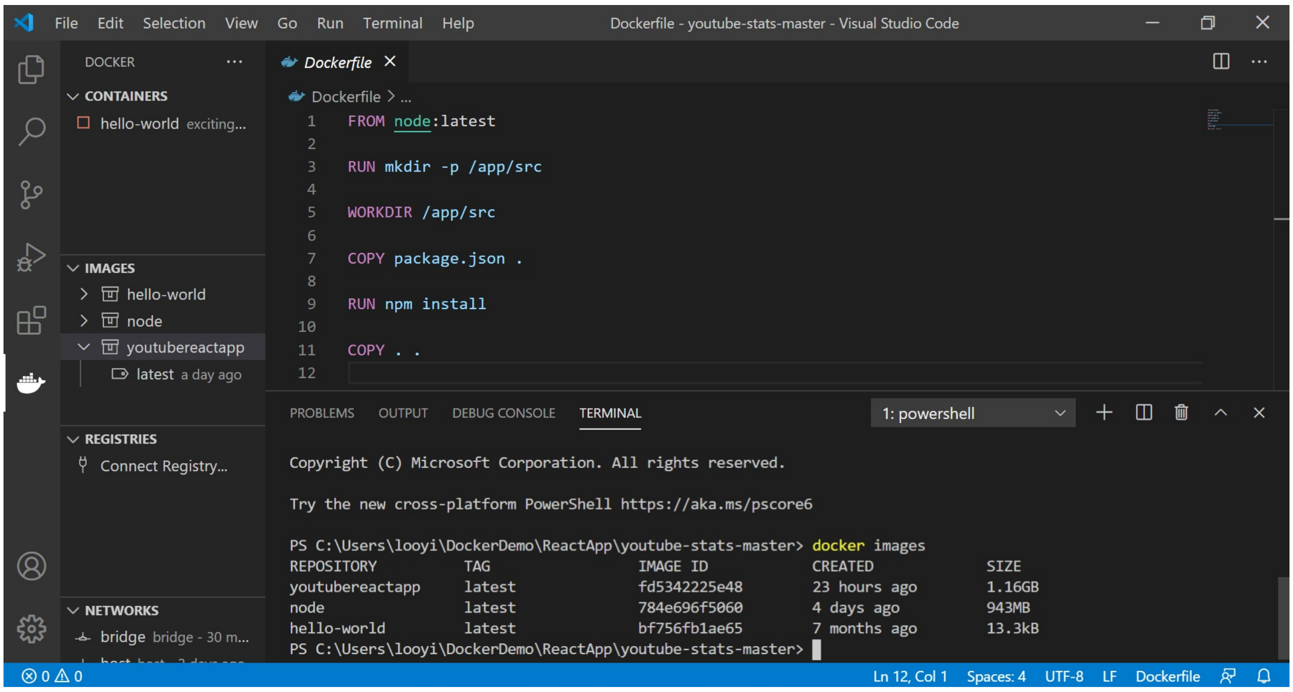
Docker images are built in Docker bash CLI. Execute `docker images` command to see the images in your workstation and execute `docker ps` command to show which container is running. Docker image can be built by executing `docker build` command. The command is accompanied with the directory that the Dockerfile is in and best to be named using a tag `-t` command. For this example, build the Docker image through Docker bash CLI by executing `docker build . -t youtubereactapp:latest` command in the directory of the project folder as shown in Figure 13.


MINGW64:/c/Users/looyi/DockerDemo/ReactApp/youtube-stats-master

```
looyi@DESKTOP-N7EVEMB MINGW64 /c/Program Files/Docker Toolbox
$ cd C:/Users/looyi/DockerDemo/ReactApp/youtube-stats-master

looyi@DESKTOP-N7EVEMB MINGW64 ~/DockerDemo/ReactApp/youtube-stats-master
$ docker build . -t youtubereactapp:latest
Sending build context to Docker daemon  482.8kB
Step 1/8 : FROM node:latest
latest: Pulling from library/node
419e7ae5bb1e: Pull complete
848839e0cd3b: Pull complete
de30e8b35015: Pull complete
258fdea6ea48: Pull complete
ddb75eb7f1e9: Pull complete
7ec8a0667334: Pull complete
c6d8dab5779d: Pull complete
d3ae9be01b42: Pull complete
38ff96b6f833: Pull complete
Digest: sha256:8b6401f8d15c900736a54a870994277b3de19ebd28cc483c497bf00d608e2a90
Status: Downloaded newer image for node:latest
 ---> 784e696f5060
Step 2/8 : RUN mkdir -p /app/src
 ---> Running in 47a669edd96e
Removing intermediate container 47a669edd96e
 ---> 0b6b665a0380
Step 3/8 : WORKDIR /app/src
 ---> Running in 7eaff6009ed7
Removing intermediate container 7eaff6009ed7
 ---> aa6fe315f4f6
Step 4/8 : COPY package.json .
 ---> e3cf3b3832ac
```

Figure 13: Build Docker Image.

The steps detail out the layers that this Docker image have. In order to run the image, `docker run` command have to be executed. Take note that this can actually be executed in VS Code as well as illustrated in Figure 14.



```
FROM node:latest

RUN mkdir -p /app/src

WORKDIR /app/src

COPY package.json .

RUN npm install

COPY . .

```

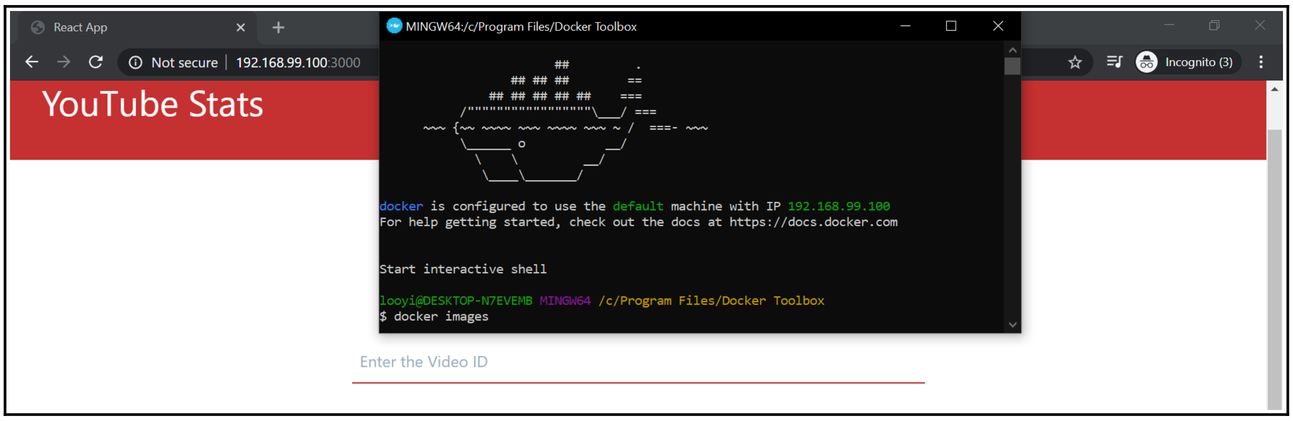
```
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\looyi\DockerDemo\ReactApp\youtube-stats-master> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
youtubereactapp     latest              fd5342225e48        23 hours ago        1.16GB
node                latest              784e696f5060        4 days ago          943MB
hello-world         latest              bf756fb1ae65        7 months ago        13.3kB
PS C:\Users\looyi\DockerDemo\ReactApp\youtube-stats-master>
```

Figure 14: Build or Run Docker Image in VS Code.

In VS Code, subsequent to installing the Docker extension, there will be a Docker logo on the left pane where developer may choose to view current containers and images that are available. Adding the terminal in VS Code will enable the execution of commands without the need to execute everything in Docker bash CLI.

<u>Docker Container.</u>

One may run the Docker Image in a container by executing `docker run` command. Similar to `docker build` command, the `docker run` command need to be accompanied with options such as removing intermediate containers after a successful build with `--rm` and adding interactive process with `-it` to show details of the run. Adding port flag is to map the indicated tcp port to the container and adding the image name is essential in order to run the intended Docker image. For this example, `docker run --rm -it -p 3000:3000/tcp youtubereactapp:latest` command in Docker bash CLI as well as right-clicking the image file on VS Code and select "Run" / "Run Interactively". After doing so, you will be able to view the application on a browser following URL given in the CLI as shown in Figure 15.

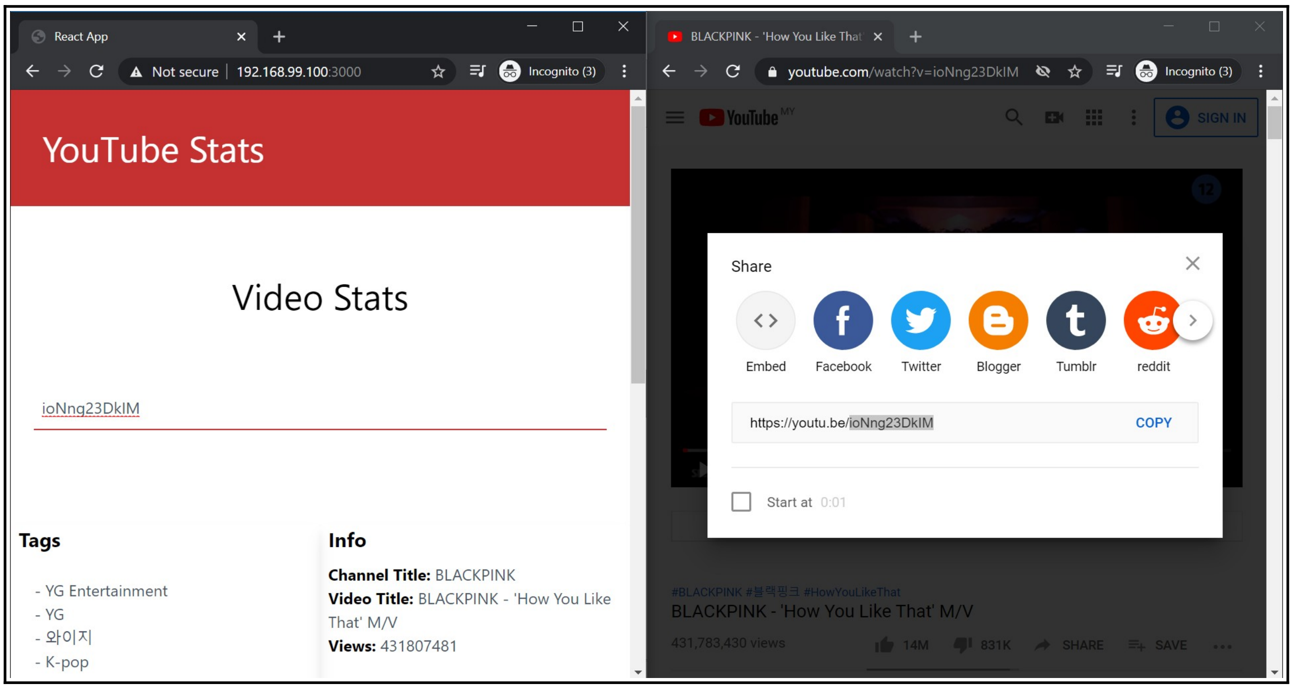


React App
Not secure | 192.168.99.100:3000
YouTube Stats
Enter the Video ID
MINGW64:/c/Program Files/Docker Toolbox

```
                        ##         .
                  ## ## ##        ==
               ## ## ## ## ##    ===
           /"""""""""""""""""\___/ ===
      ~~~ {~~ ~~~~ ~~~ ~~~~ ~~~ ~ /  ===- ~~~
           \______ o           __/
             \    \         __/
              \____\_______/

docker is configured to use the default machine with IP 192.168.99.100
For help getting started, check out the docs at https://docs.docker.com


Start interactive shell

looyi@DESKTOP-N7EVEMB MINGW64 /c/Program Files/Docker Toolbox
$ docker images
```

Figure 15: Example React App's Docker Image running in a Docker Container.

Take note that if there are any error of connecting to localhost or by entering localhost:3000 returns a refused connection error, that means Docker is running on a different IP from localhost. Check the IP in Docker bash CLI and replace localhost with the default IP that Docker is running on. Once the exampler react app is up running, you may check the YouTube Statistics about a video that you would like to check on as illustrated in Figure 16.



Tags

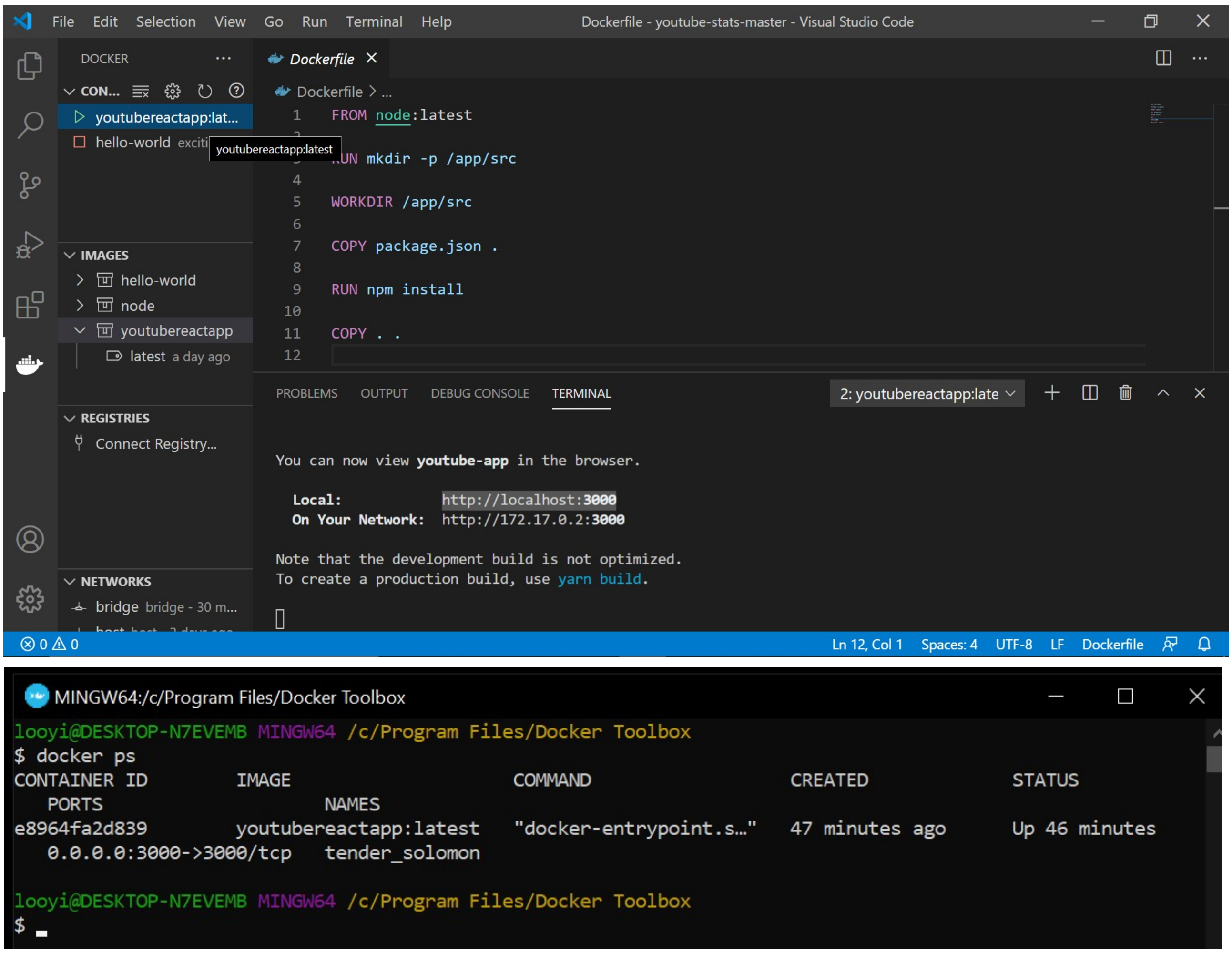
- YG Entertainment
- YG
- 와이지
- K-pop

Info

Channel Title: BLACKPINK
Video Title: BLACKPINK - 'How You Like That' M/V
Views: 431807481

Figure 16: React App running in a Docker Container without react native/expo CLI.

In VS Code, the container panel will have the container indicated as “running” mode. Similarly, in Docker bash CLI, exectuing the docker ps will return the information of currently running container as shown in Figure 17.



```
FROM node:latest

RUN mkdir -p /app/src

WORKDIR /app/src

COPY package.json .

RUN npm install

COPY . .
```

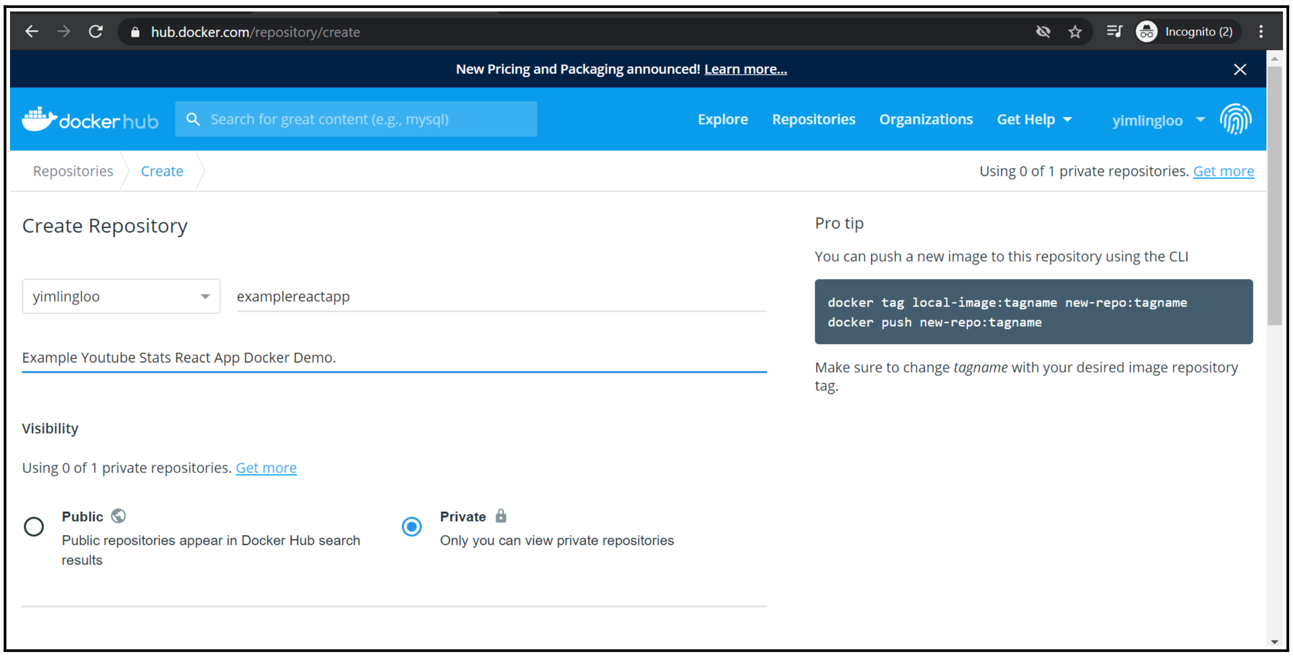
```
You can now view youtube-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://172.17.0.2:3000

Note that the development build is not optimized.
To create a production build, use yarn build.
```

```
looyi@DESKTOP-N7EVEMB MINGW64 /c/Program Files/Docker Toolbox
$ docker ps
CONTAINER ID        IMAGE                     COMMAND                  CREATED             STATUS
   PORTS                    NAMES
e8964fa2d839        youtubereactapp:latest    "docker-entrypoint.s…"   47 minutes ago      Up 46 minutes
   0.0.0.0:3000->3000/tcp   tender_solomon

looyi@DESKTOP-N7EVEMB MINGW64 /c/Program Files/Docker Toolbox
$
```

Figure 17: Docker Container indicated as “running” in VS Code and CLI.

Dockerhub.

One may choose to push the Docker image up to a repository; similar as the concept of Git. Git repository may be stored in Github, Docker images may be stored in Dockerhub. Subsequent to signing up a Dockerhub account, one will be given a free repository. Create a new repository in Dockerhub by clicking on the "Create Repository" button. Create a repository as shown in Figure 18.


hub.docker.com/repository/create

New Pricing and Packaging announced! Learn more...

Repositories > Create — Using 0 of 1 private repositories. Get more

Create Repository

yimlingloo / examplereactapp

Example Youtube Stats React App Docker Demo.

Visibility

Using 0 of 1 private repositories. Get more

Public — Public repositories appear in Docker Hub search results

Private (selected) — Only you can view private repositories

Pro tip

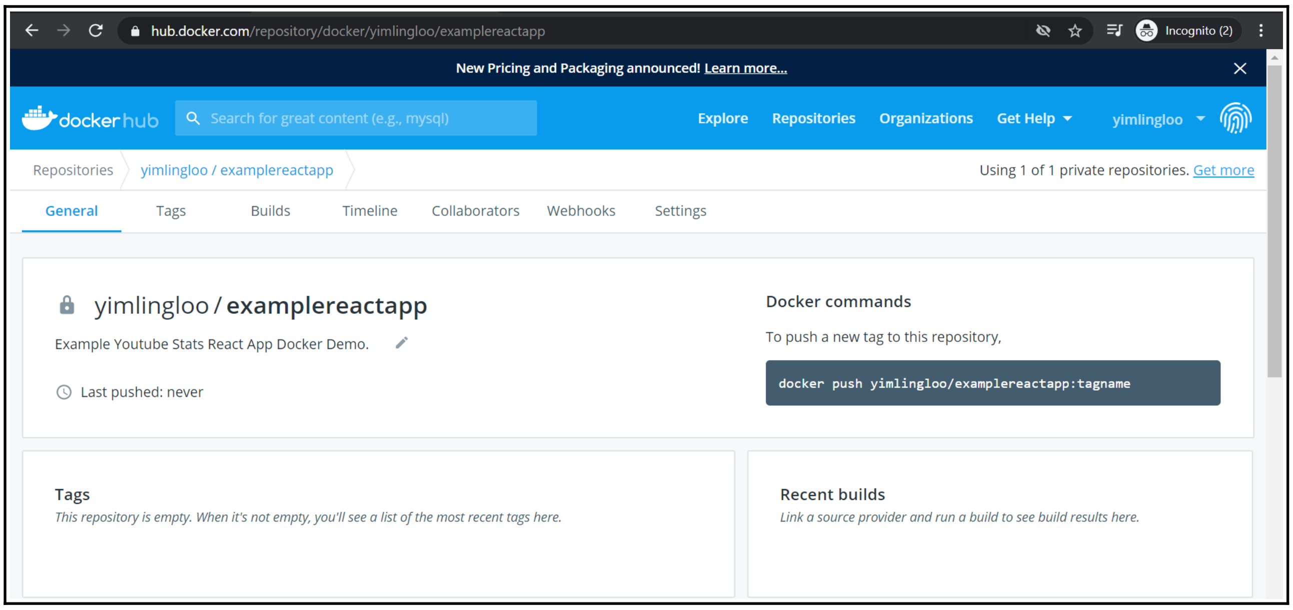
You can push a new image to this repository using the CLI

```
docker tag local-image:tagname new-repo:tagname
docker push new-repo:tagname
```

Make sure to change *tagname* with your desired image repository tag.

Figure 18: Create a New Docker repository.

Click on "Create" button to create the new repository. A new repository is created with the return of empty repository page as shown in Figure 19.


hub.docker.com/repository/docker/yimlingloo/examplereactapp

New Pricing and Packaging announced! Learn more...

Repositories > yimlingloo / examplereactapp — Using 1 of 1 private repositories. Get more

General | Tags | Builds | Timeline | Collaborators | Webhooks | Settings

yimlingloo / examplereactapp

Example Youtube Stats React App Docker Demo.

Last pushed: never

Docker commands

To push a new tag to this repository,

```
docker push yimlingloo/examplereactapp:tagname
```

Tags

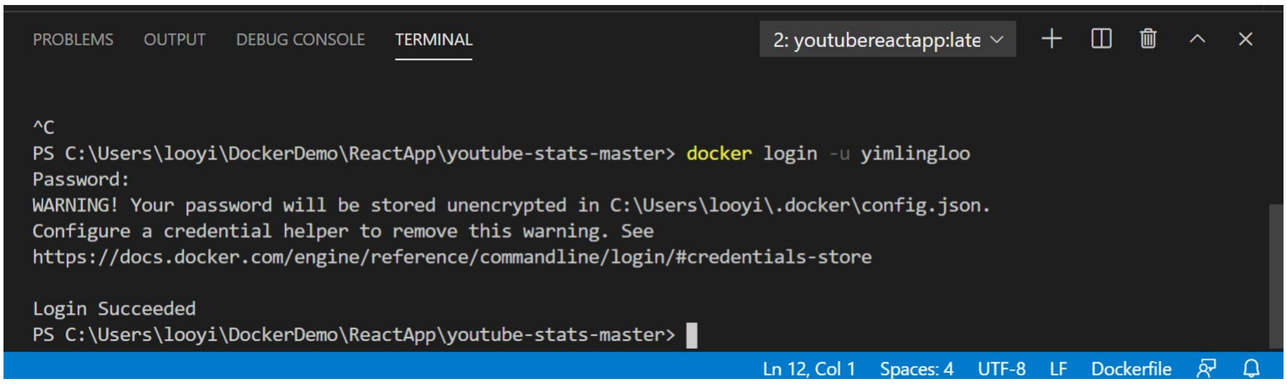
This repository is empty. When it's not empty, you'll see a list of the most recent tags here.

Recent builds

Link a source provider and run a build to see build results here.

Figure 19: New Docker repository.

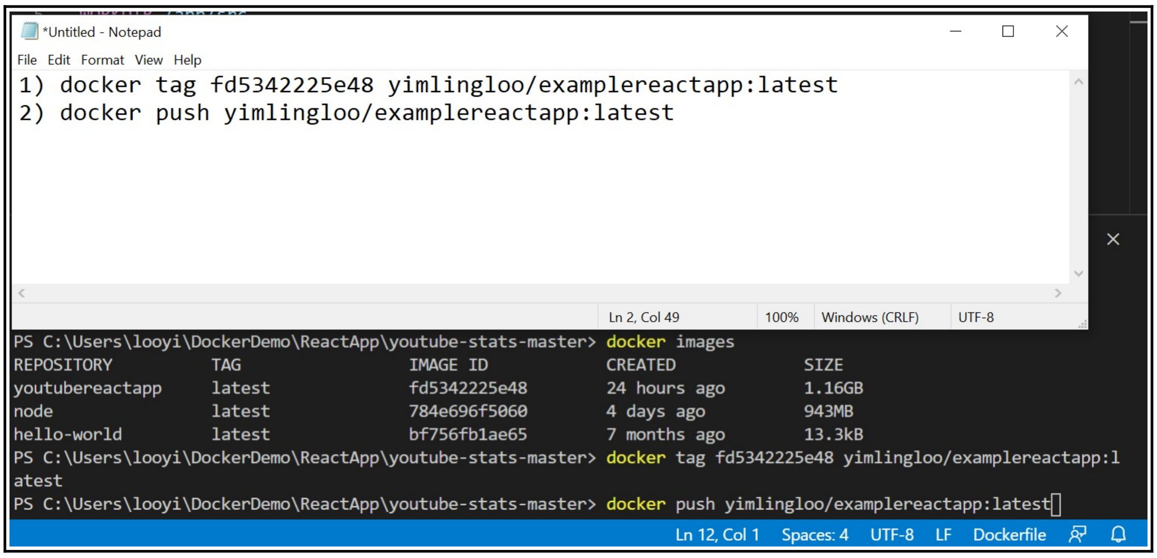
Note that similar to Github, Dockerhub also provide a `docker push` command on how to push a Docker Image to Dockerhub. In order to push the Dockerfile to Dockerhub, the local workstation need to be logged in to Dockerhub. In order to login to Dockerhub, one may do this in VS Code terminal by executing `docker login` command accompanied by the detail of username `-u` as shown in Figure 20.



```
^C
PS C:\Users\looyi\DockerDemo\ReactApp\youtube-stats-master> docker login -u yimlingloo
Password:
WARNING! Your password will be stored unencrypted in C:\Users\looyi\.docker\config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
PS C:\Users\looyi\DockerDemo\ReactApp\youtube-stats-master>
```

Figure 20: Docker hub login.

Upon successful login, execute the `docker images` command again in order to retrieve the TAG ID of the example react app. Upon retrieving the correct TAG ID, put the TAG ID inside the `docker tag` command in order to tag the correct Docker image to the push. After tagging to the intended Docker image, then only execture the `docker push` command as illustrated in Figure 21.



```
1) docker tag fd5342225e48 yimlingloo/examplereactapp:latest
2) docker push yimlingloo/examplereactapp:latest
```

```
PS C:\Users\looyi\DockerDemo\ReactApp\youtube-stats-master> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
youtubereactapp     latest              fd5342225e48        24 hours ago        1.16GB
node                latest              784e696f5060        4 days ago          943MB
hello-world         latest              bf756fb1ae65        7 months ago        13.3kB
PS C:\Users\looyi\DockerDemo\ReactApp\youtube-stats-master> docker tag fd5342225e48 yimlingloo/examplereactapp:l
atest
PS C:\Users\looyi\DockerDemo\ReactApp\youtube-stats-master> docker push yimlingloo/examplereactapp:latest
```

Figure 21: Tag and push Docker image to Dockerhub.

The Docker image will be pushed to Dockerhub with all the layers indicated in the terminal, similar to when the Docker image was being built. Upon successful push, the repository will have a tag available and one may click in the tag to see that now the Dockerhub suggests a docker pull command, whoever would like to have the shared Docker image as shown in Figure 22.

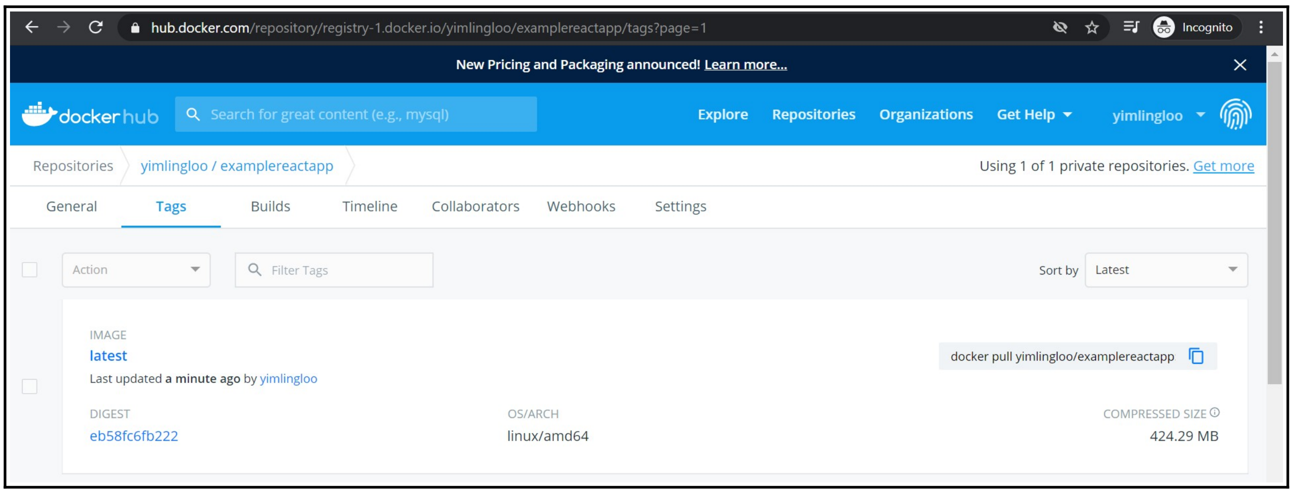

hub.docker.com/repository/registry-1.docker.io/yimlingloo/examplereactapp/tags?page=1
Incognito
New Pricing and Packaging announced! Learn more...
docker hub
Search for great content (e.g., mysql)
Explore
Repositories
Organizations
Get Help
yimlingloo
Repositories
yimlingloo / examplereactapp
Using 1 of 1 private repositories. Get more
General
Tags
Builds
Timeline
Collaborators
Webhooks
Settings
Action
Filter Tags
Sort by
Latest

IMAGE
latest
Last updated a minute ago by yimlingloo

docker pull yimlingloo/examplereactapp

DIGEST eb58fc6fb222
OS/ARCH linux/amd64
COMPRESSED SIZE 424.29 MB

Figure 22: Docker image pushed to Dockerhub and ready for pull.

Up until this step, it shows that a collaborator does not need to preconfigure anything when project development files are passed to him/her. All that was needed was to have the project folders and Dockerfile, in order for the collaborator to run the project files and test. Back to the question asked at the beginning of this practical;

Q: What do you think is the difference between Docker and Virtual Machines?