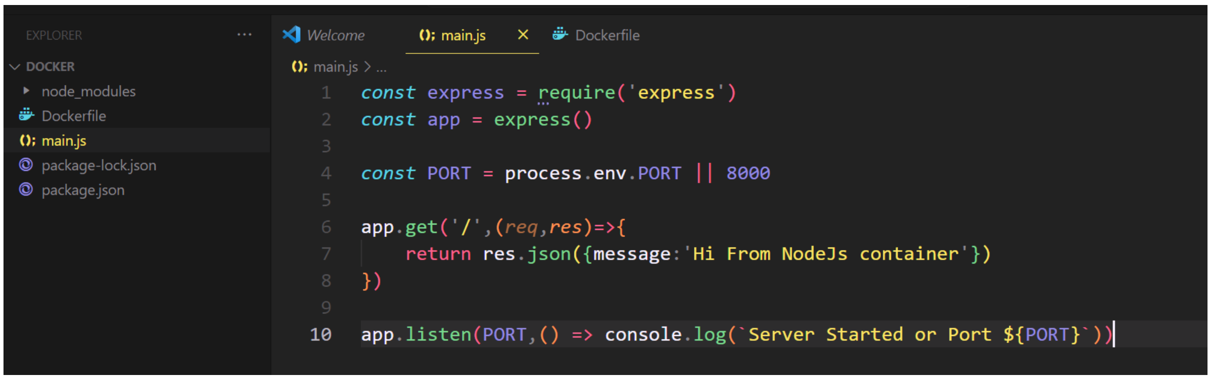
Assignment-1

Task: Create a simple Docker container for a Node.js web application.

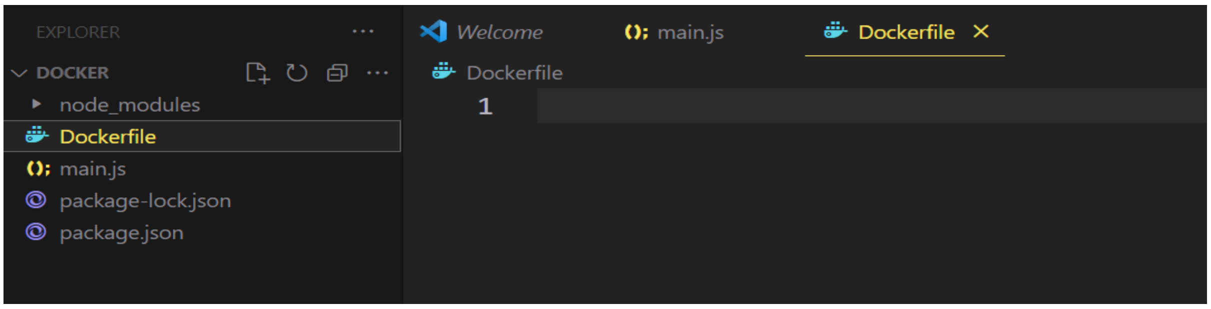
Steps:

- Open any code editor (Vs Code) and write Node-JS application.

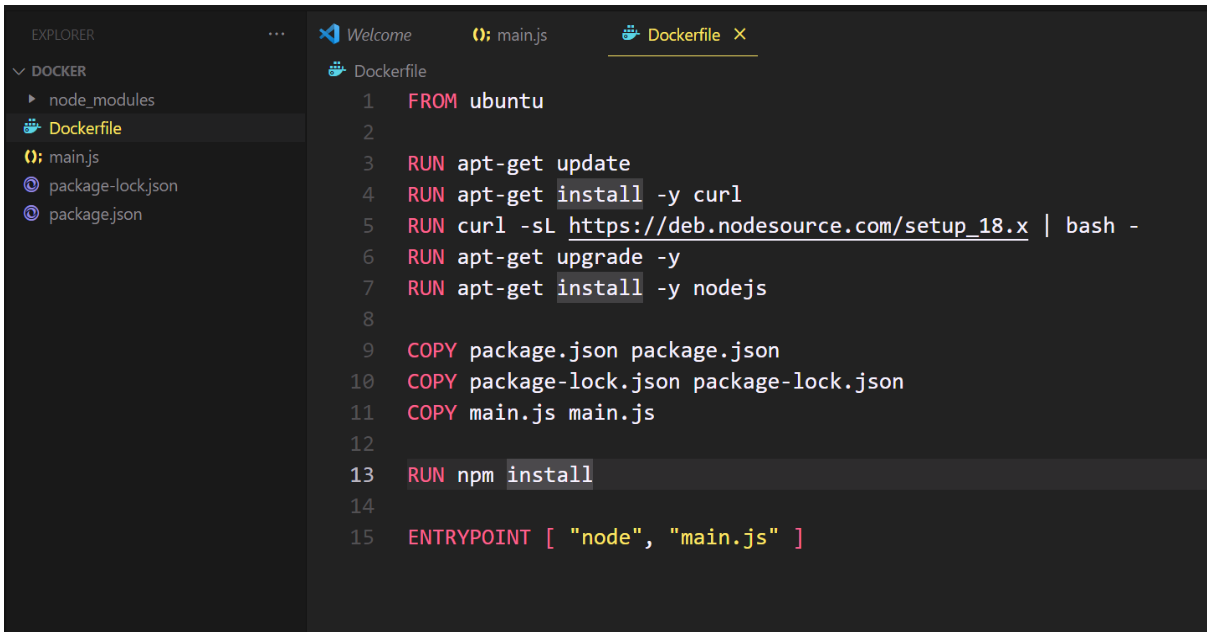


```
const express = require('express')
const app = express()

const PORT = process.env.PORT || 8000

app.get('/',(req,res)=>{
    return res.json({message:'Hi From NodeJs container'})
})

app.listen(PORT,() => console.log(`Server Started or Port ${PORT}`))
```

- Now in the same directory create another file called "Dockerfile", here we'll write our docker commands configuration.



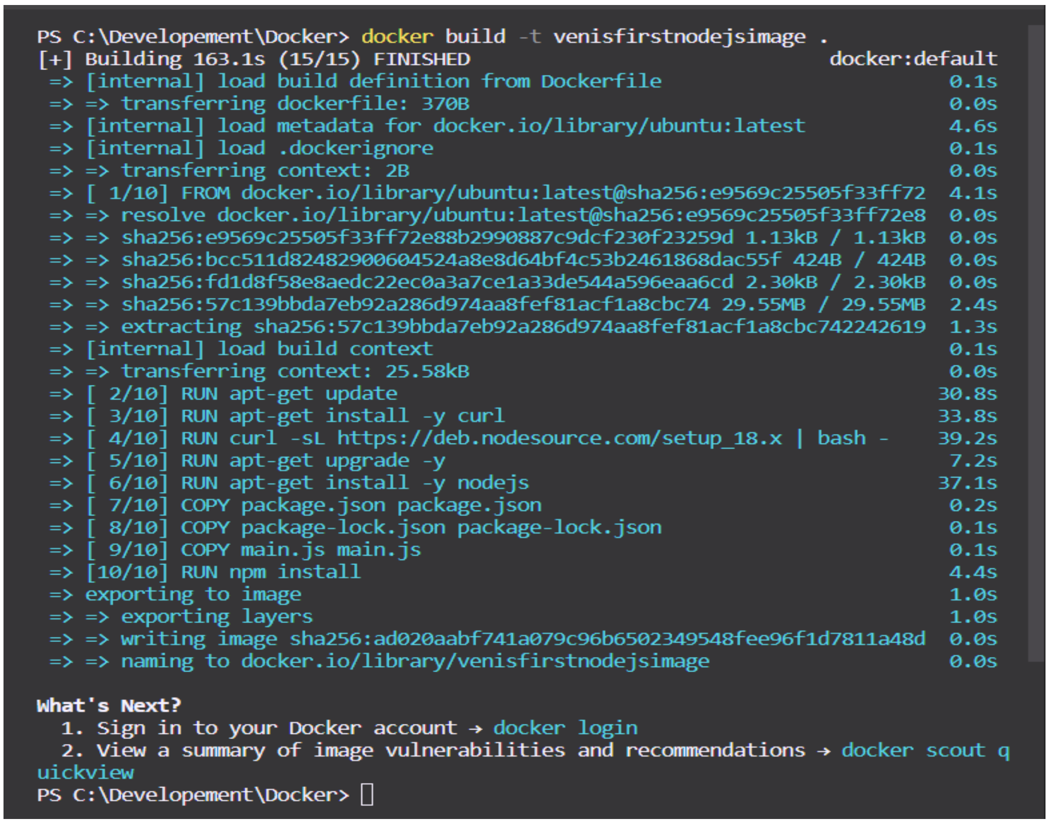
➢ Here we’ve have written down our dockerfile.



```
FROM ubuntu

RUN apt-get update
RUN apt-get install -y curl
RUN curl -sL https://deb.nodesource.com/setup_18.x | bash -
RUN apt-get upgrade -y
RUN apt-get install -y nodejs

COPY package.json package.json
COPY package-lock.json package-lock.json
COPY main.js main.js

RUN npm install

ENTRYPOINT [ "node", "main.js" ]
```

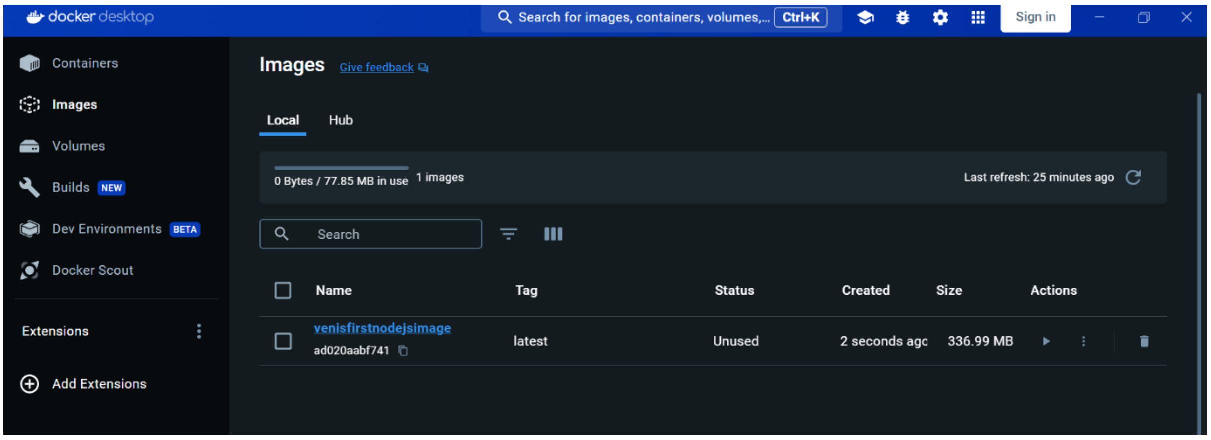
➢ Now go to terminal and select the directory in which our nodejs application and dockerfile are present and then run following command to build our image.

Docker build -t venisfirstnodejsimage .



```
PS C:\Developement\Docker> docker build -t venisfirstnodejsimage .
[+] Building 163.1s (15/15) FINISHED                            docker:default
 => [internal] load build definition from Dockerfile                       0.1s
 => => transferring dockerfile: 370B                                       0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest           4.6s
 => [internal] load .dockerignore                                          0.1s
 => => transferring context: 2B                                            0.0s
 => [ 1/10] FROM docker.io/library/ubuntu:latest@sha256:e9569c25505f33ff72  4.1s
 => => resolve docker.io/library/ubuntu:latest@sha256:e9569c25505f33ff72e8  0.0s
 => => sha256:e9569c25505f33ff72e88b2990887c9dcf230f23259d 1.13kB / 1.13kB  0.0s
 => => sha256:bcc511d82482900604524a8e8d64bf4c53b2461868dac55f 424B / 424B  0.0s
 => => sha256:fd1d8f58e8aedc22ec0a3a7ce1a33de544a596eaa6cd 2.30kB / 2.30kB  0.0s
 => => sha256:57c139bbda7eb92a286d974aa8fef81acf1a8cbc74 29.55MB / 29.55MB  2.4s
 => => extracting sha256:57c139bbda7eb92a286d974aa8fef81acf1a8cbc742242619  1.3s
 => [internal] load build context                                          0.1s
 => => transferring context: 25.58kB                                       0.0s
 => [ 2/10] RUN apt-get update                                            30.8s
 => [ 3/10] RUN apt-get install -y curl                                   33.8s
 => [ 4/10] RUN curl -sL https://deb.nodesource.com/setup_18.x | bash -   39.2s
 => [ 5/10] RUN apt-get upgrade -y                                         7.2s
 => [ 6/10] RUN apt-get install -y nodejs                                 37.1s
 => [ 7/10] COPY package.json package.json                                 0.2s
 => [ 8/10] COPY package-lock.json package-lock.json                       0.1s
 => [ 9/10] COPY main.js main.js                                           0.1s
 => [10/10] RUN npm install                                                4.4s
 => exporting to image                                                     1.0s
 => => exporting layers                                                    1.0s
 => => writing image sha256:ad020aabf741a079c96b6502349548fee96f1d7811a48d  0.0s
 => => naming to docker.io/library/venisfirstnodejsimage                   0.0s

What's Next?
  1. Sign in to your Docker account → docker login
  2. View a summary of image vulnerabilities and recommendations → docker scout q
uickview
PS C:\Developement\Docker> 
```

- Now go to docker desktop and go to images section, here we can see that our image has been created.



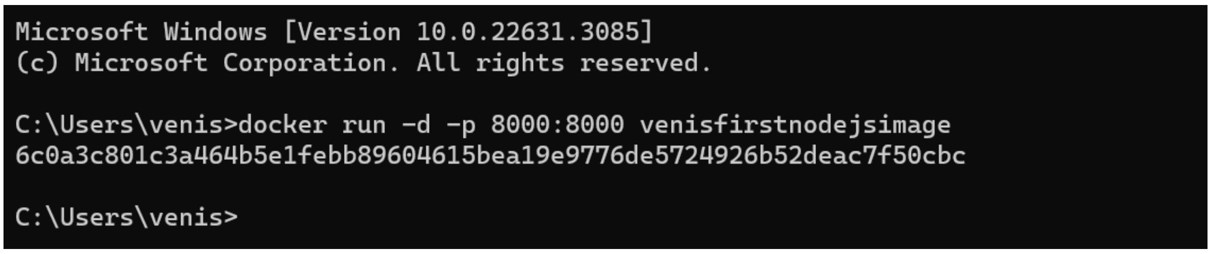
0 Bytes / 77.85 MB in use 1 images

Last refresh: 25 minutes ago

Name	Tag	Status	Created	Size	Actions
venisfirstnodejsimage ad020aabf741	latest	Unused	2 seconds ago	336.99 MB	

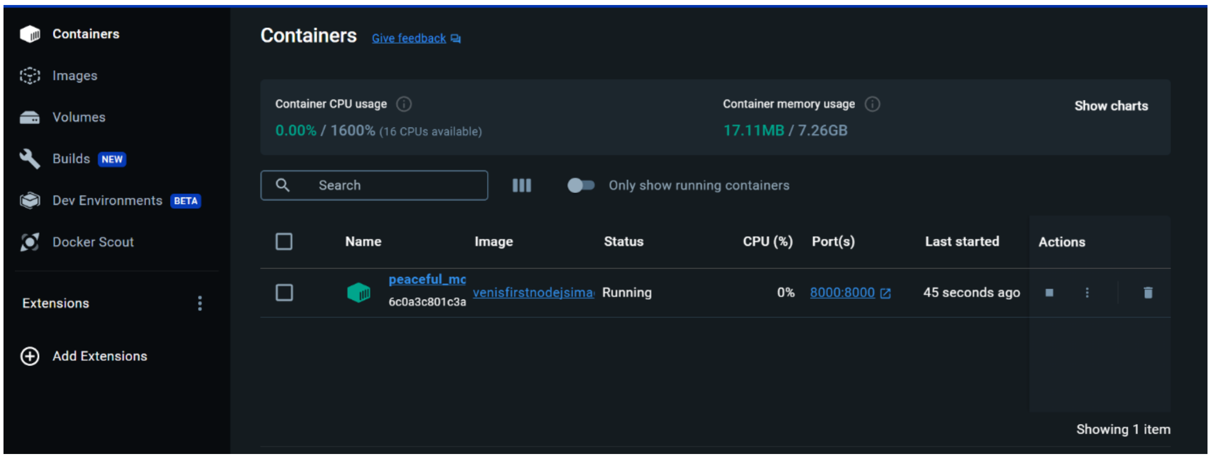
- Now to run this image and to do port mapping with our local machine's port so that we can access it through the same port, run this following command.
- This command will run the image and it maps the port : 8000 of docker container with local machine's port : 8000.

Docker run -d -p 8000:8000 venisfirstnodejsimage



```
Microsoft Windows [Version 10.0.22631.3085]
(c) Microsoft Corporation. All rights reserved.

C:\Users\venis>docker run -d -p 8000:8000 venisfirstnodejsimage
6c0a3c801c3a464b5e1febb89604615bea19e9776de5724926b52deac7f50cbc

C:\Users\venis>
```

- Now, go to docker desktop and go in docker containers section, here we can see our container is running.



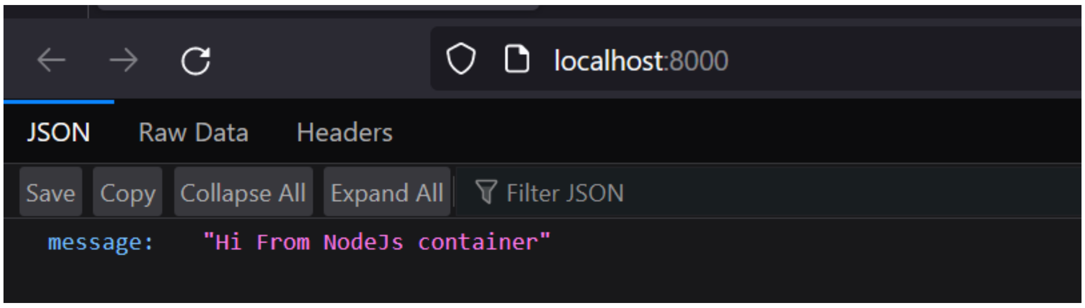
Container CPU usage 0.00% / 1600% (16 CPUs available)

Container memory usage 17.11MB / 7.26GB

Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
peaceful_mc 6c0a3c801c3a	venisfirstnodejsima	Running	0%	8000:8000	45 seconds ago	

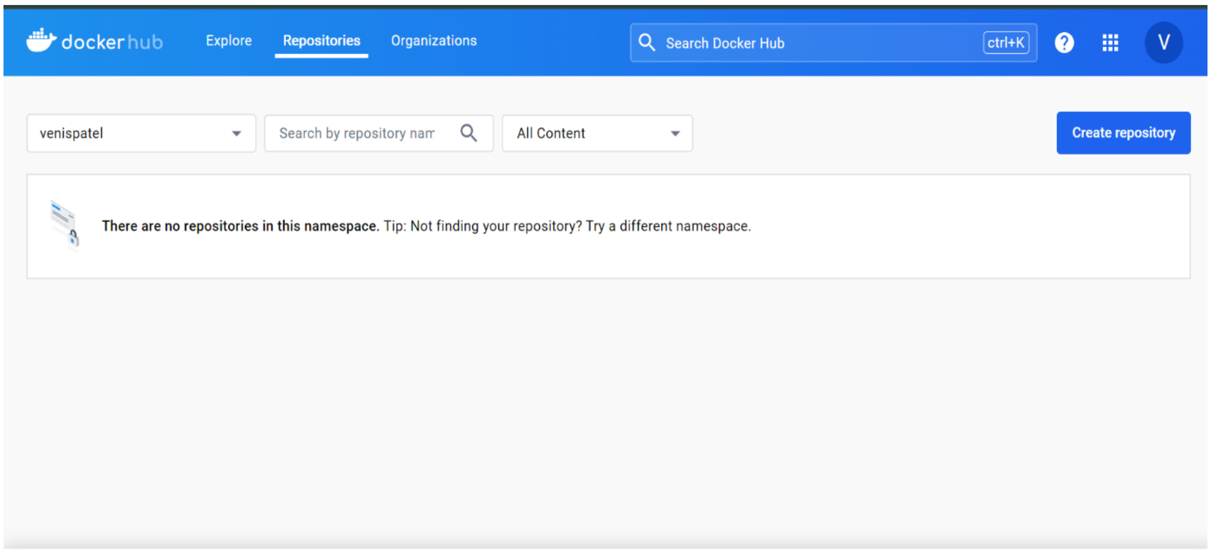
Showing 1 item

- Now go to any web-browser and type "*localhost:8000*" and here we can see that our node-js application is running.



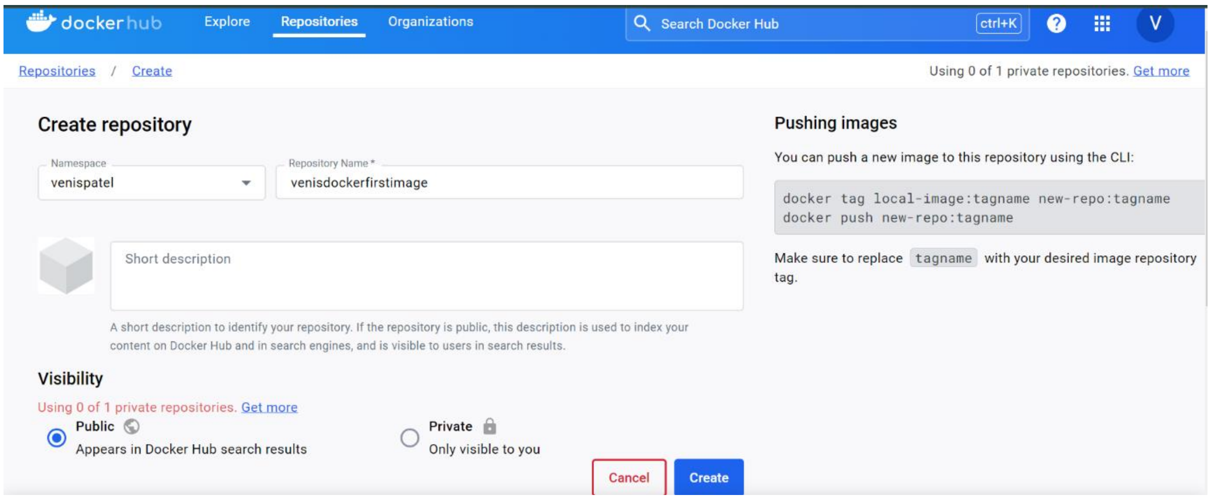
```
message:    "Hi From NodeJs container"
```

- Now go to "hub.docker.com" and make a free account. After making a account select on "Create Repository".



There are no repositories in this namespace. Tip: Not finding your repository? Try a different namespace.

➢ Now give name to Repository as "venisdockerfirstimage" and make visibility as Public for now. (Better to go for private visibility for security options).



Create repository

Namespace: venispatel

Repository Name *: venisdockerfirstimage

Short description

A short description to identify your repository. If the repository is public, this description is used to index your content on Docker Hub and in search engines, and is visible to users in search results.

Visibility

Using 0 of 1 private repositories. Get more

[✓ Public] Appears in Docker Hub search results

Private Only visible to you

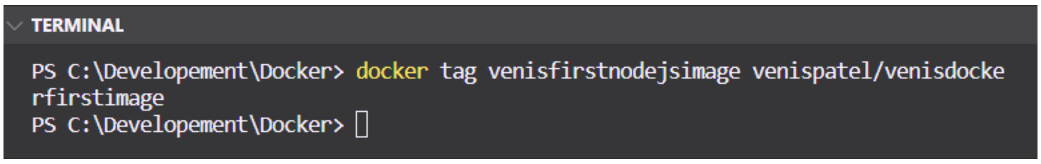
Pushing images

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
```

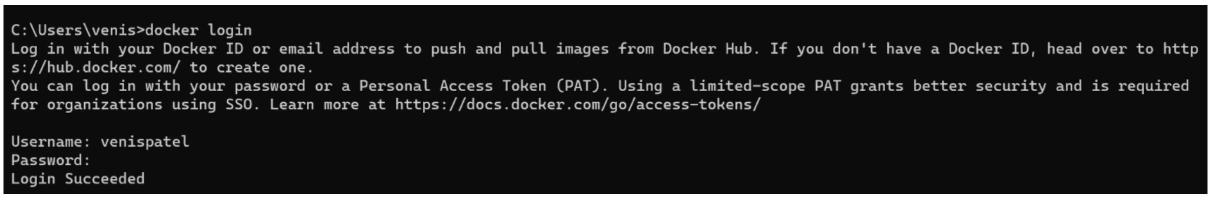
Make sure to replace `tagname` with your desired image repository tag.

➢ Now go to terminal and run this following command to tag our existential image with our created repository on DockerHub.



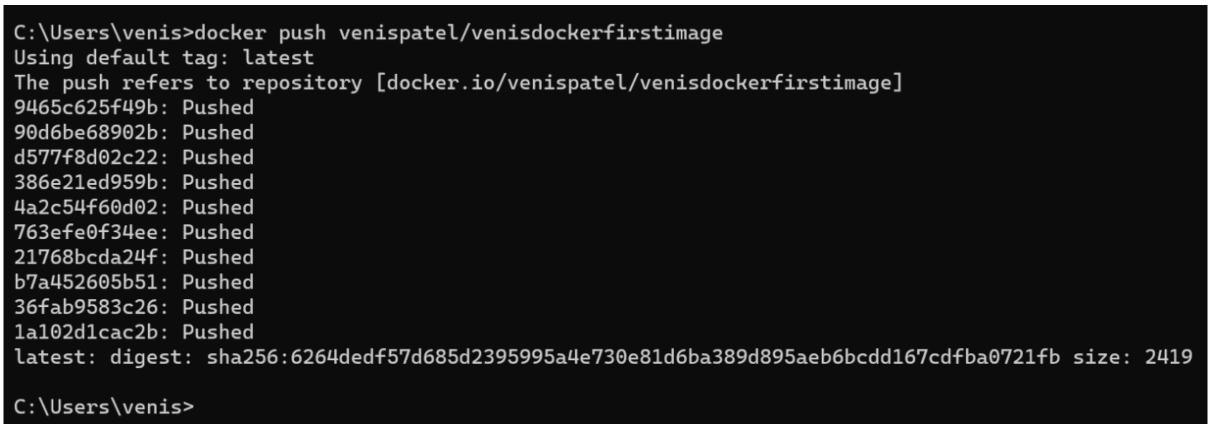
```
PS C:\Developement\Docker> docker tag venisfirstnodejsimage venispatel/venisdocke
rfirstimage
PS C:\Developement\Docker>
```

➢ Now it will ask for docker login, so run command: docker login
➢ Now it will ask for username and password so enter that now you are successfully logged in.



```
C:\Users\venis>docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to http
s://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required
for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/

Username: venispatel
Password:
Login Succeeded
```

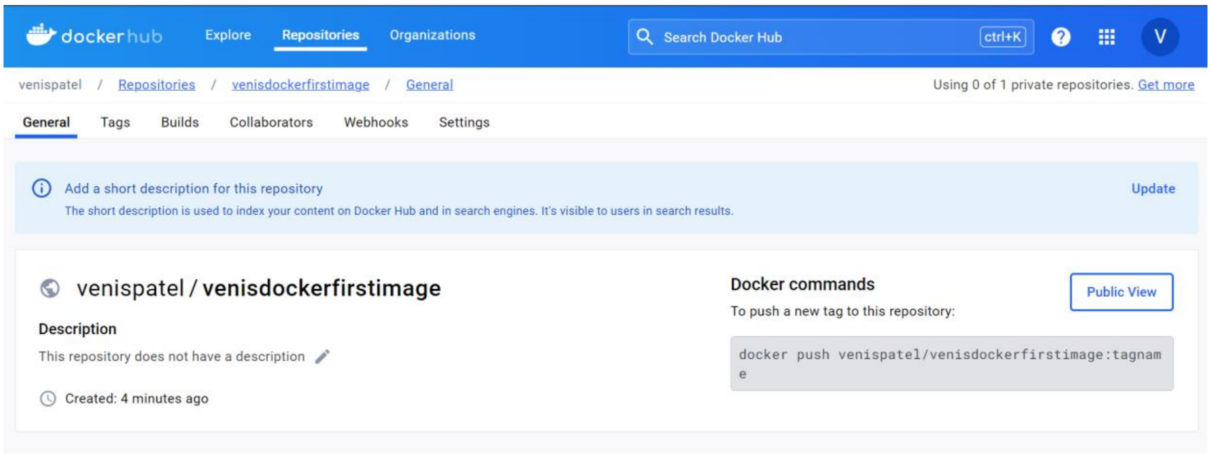
➢ Now run the command docker push venispatel/venisdockerfirstimage to push image to dockerhub.



```
C:\Users\venis>docker push venispatel/venisdockerfirstimage
Using default tag: latest
The push refers to repository [docker.io/venispatel/venisdockerfirstimage]
9465c625f49b: Pushed
90d6be68902b: Pushed
d577f8d02c22: Pushed
386e21ed959b: Pushed
4a2c54f60d02: Pushed
763efe0f34ee: Pushed
21768bcda24f: Pushed
b7a452605b51: Pushed
36fab9583c26: Pushed
1a102d1cac2b: Pushed
latest: digest: sha256:6264dedf57d685d2395995a4e730e81d6ba389d895aeb6bcdd167cdfba0721fb size: 2419

C:\Users\venis>
```

➢ Now go to hub.docker.com and here we can see that our image has been successfully pushed to dockerhub.

Before:



venispatel / Repositories / venisdockerfirstimage / General

Using 0 of 1 private repositories. Get more

General Tags Builds Collaborators Webhooks Settings

Add a short description for this repository

The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.

Update

venispatel / venisdockerfirstimage

Description

This repository does not have a description

Created: 4 minutes ago

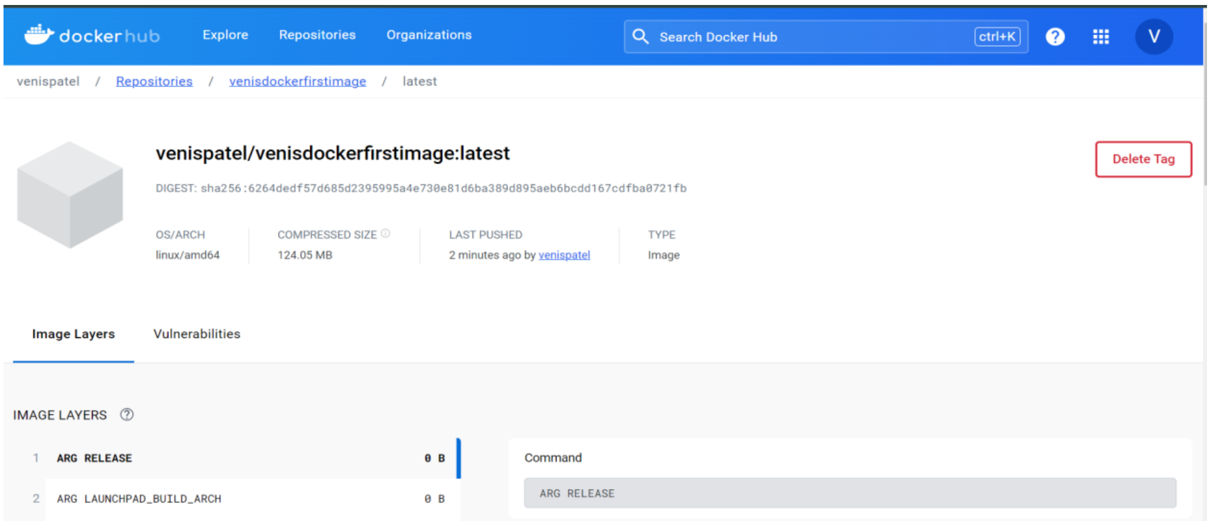
Docker commands

To push a new tag to this repository:

```
docker push venispatel/venisdockerfirstimage:tagname
```

Public View

After:



venispatel / Repositories / venisdockerfirstimage / latest

venispatel/venisdockerfirstimage:latest

DIGEST: sha256:6264dedf57d685d2395995a4e730e81d6ba389d895aeb6bcdd167cdfba0721fb

Delete Tag

OS/ARCH	COMPRESSED SIZE	LAST PUSHED	TYPE
linux/amd64	124.05 MB	2 minutes ago by venispatel	Image

Image Layers Vulnerabilities

IMAGE LAYERS

1	ARG RELEASE	0 B
2	ARG LAUNCHPAD_BUILD_ARCH	0 B

Command

```
ARG RELEASE
```