# WIPRO EXERCISE

## Steps to build my-website

### Step 1: Create Project Directory

Open your terminal and run:

```
mkdir my-website
cd my-website
```

---

### Step 2: Create Website Content

Create a file named`index.html` with the following content:

File: ***index.html***

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>My Docker Website</title>
</head>
<body>
  <h1>Welcome to My Docker Website!</h1>
  <p>This is a simple static website hosted using Nginx and Docker.</p>
</body>
</html>
```

---

### Step 3: Create Dockerfile

Create a file named`Dockerfile` in the same directory:

File: ***Dockerfile***

```
# Use the official Nginx image from Docker Hub
FROM nginx:alpine
# Copy the website files to the Nginx HTML directory
COPY index.html /usr/share/nginx/html/index.html
# Expose port 80 for the web server
EXPOSE 80
# Start Nginx when the container launches
CMD ["nginx", "-g", "daemon off;"]
```

---

## Step 4: Build and Run the Docker Container

1. Build the Docker image:
2. `docker build -t my-website .`
3. Run the Docker container (map container port 80 to host port 8080):
4. `docker run -d -p 8080:80 --name my-website-container my-website`

## Step 5: Access the Website

Open your browser and go to:
👉 http://localhost:8080
You should see the "Welcome to My Docker Website!" page.

## Step 6: Stop and Clean Up (Optional)

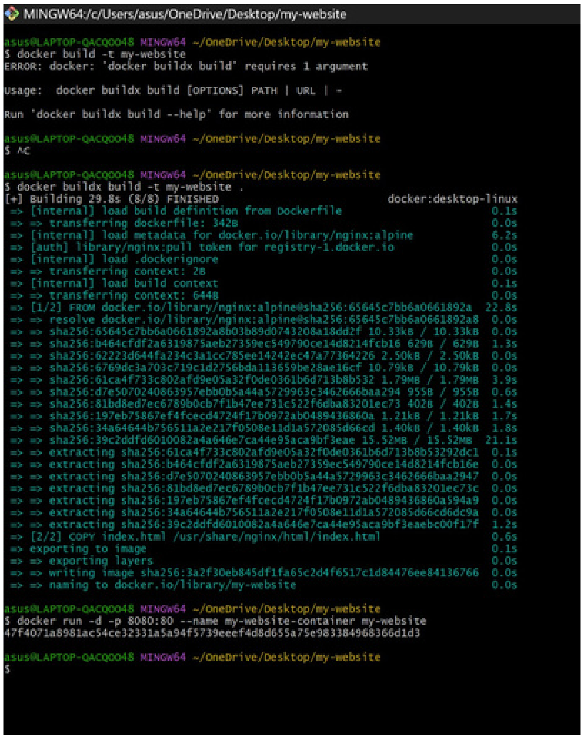
1. Stop and remove the container:
2. `docker stop my-website-container`
3. `docker rm my-website-container`

4.Remove the Docker image:

5. `docker rmi my-website`

# SCREENSHOT OF THE STEPS

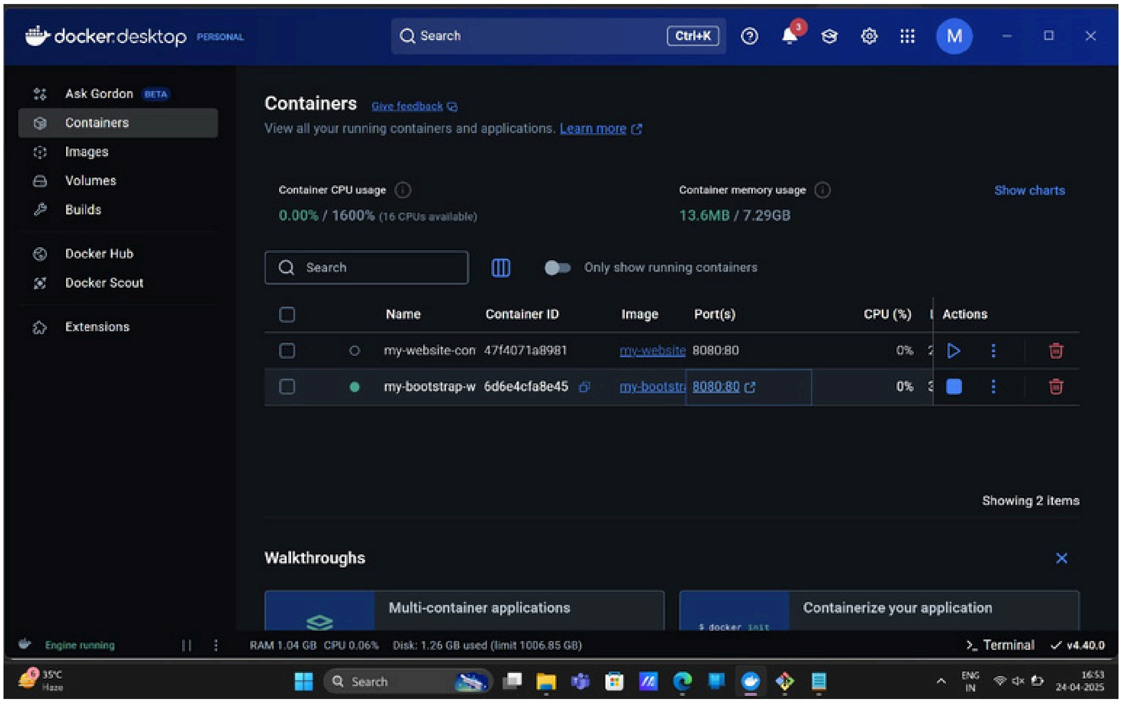
STEP-1



MINGW64:/c/Users/asus/OneDrive/Desktop/my-website

```
asus@LAPTOP-QACQ0048 MINGW64 ~/OneDrive/Desktop/my-website
$ docker build -t my-website
ERROR: docker: 'docker buildx build' requires 1 argument

Usage:  docker buildx build [OPTIONS] PATH | URL | -

Run 'docker buildx build --help' for more information

asus@LAPTOP-QACQ0048 MINGW64 ~/OneDrive/Desktop/my-website
$ ^C

asus@LAPTOP-QACQ0048 MINGW64 ~/OneDrive/Desktop/my-website
$ docker buildx build -t my-website .
[+] Building 29.8s (8/8) FINISHED                                docker:desktop-linux
 => [internal] load build definition from Dockerfile                               0.1s
 => => transferring dockerfile: 342B                                               0.0s
 => [internal] load metadata for docker.io/library/nginx:alpine                    6.2s
 => [auth] library/nginx:pull token for registry-1.docker.io                       0.0s
 => [internal] load .dockerignore                                                  0.0s
 => => transferring context: 2B                                                    0.0s
 => [internal] load build context                                                  0.1s
 => => transferring context: 644B                                                  0.0s
 => [1/2] FROM docker.io/library/nginx:alpine@sha256:65645c7bb6a0661892a          22.8s
 => => resolve docker.io/library/nginx:alpine@sha256:65645c7bb6a0661892a8          0.0s
 => => sha256:65645c7bb6a0661892a8b03b89d0743208a18dd2f 10.33kB / 10.33kB          0.0s
 => => sha256:b464cfdf2a6319875aeb27359ec549790ce14d8214fcb16 629B / 629B          1.3s
 => => sha256:62223d644fa234c3a1cc785ee14242ec47a77364226 2.50kB / 2.50kB          0.0s
 => => sha256:6769dc3a703c719c1d2756bda113659be28ae16cf 10.79kB / 10.79kB          0.0s
 => => sha256:61ca4f733c802afd9e05a32f0de0361b6d713b8b532 1.79MB / 1.79MB          3.9s
 => => sha256:d7e5070240863957ebb0b5a44a5729963c3462666baa294 955B / 955B          0.6s
 => => sha256:81bd8ed7ec6789b0cb7f1b47ee731c522f6dba83201ec73 402B / 402B          1.4s
 => => sha256:197eb75867ef4fcecd4724f17b0972ab0489436860a 1.21kB / 1.21kB          1.7s
 => => sha256:34a64644b756511a2e217f0508e11d1a572085d66cd 1.40kB / 1.40kB          1.8s
 => => sha256:39c2ddfd6010082a4a646e7ca44e95aca9bf3eae 15.52MB / 15.52MB          21.1s
 => => extracting sha256:61ca4f733c802afd9e05a32f0de0361b6d713b8b53292dc1          0.1s
 => => extracting sha256:b464cfdf2a6319875aeb27359ec549790ce14d8214fcb16e          0.0s
 => => extracting sha256:d7e5070240863957ebb0b5a44a5729963c3462666baa2947          0.0s
 => => extracting sha256:81bd8ed7ec6789b0cb7f1b47ee731c522f6dba83201ec73c          0.0s
 => => extracting sha256:197eb75867ef4fcecd4724f17b0972ab0489436860a594a9          0.0s
 => => extracting sha256:34a64644b756511a2e217f0508e11d1a572085d66cd6dc9a          0.0s
 => => extracting sha256:39c2ddfd6010082a4a646e7ca44e95aca9bf3eaebc00f17f          1.2s
 => [2/2] COPY index.html /usr/share/nginx/html/index.html                         0.6s
 => exporting to image                                                             0.1s
 => => exporting layers                                                            0.0s
 => => writing image sha256:3a2f30eb845df1fa65c2d4f6517c1d84476ee84136766          0.0s
 => => naming to docker.io/library/my-website                                      0.0s

asus@LAPTOP-QACQ0048 MINGW64 ~/OneDrive/Desktop/my-website
$ docker run -d -p 8080:80 --name my-website-container my-website
47f4071a8981ac54ce32331a5a94f5739eeef4d8d655a75e9833849683 66d1d3

asus@LAPTOP-QACQ0048 MINGW64 ~/OneDrive/Desktop/my-website
$
```

<u>STEP-2</u>



Containers Give feedback

View all your running containers and applications. Learn more

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

Container memory usage 13.6MB / 7.29GB

Show charts

Search ⬤ Only show running containers

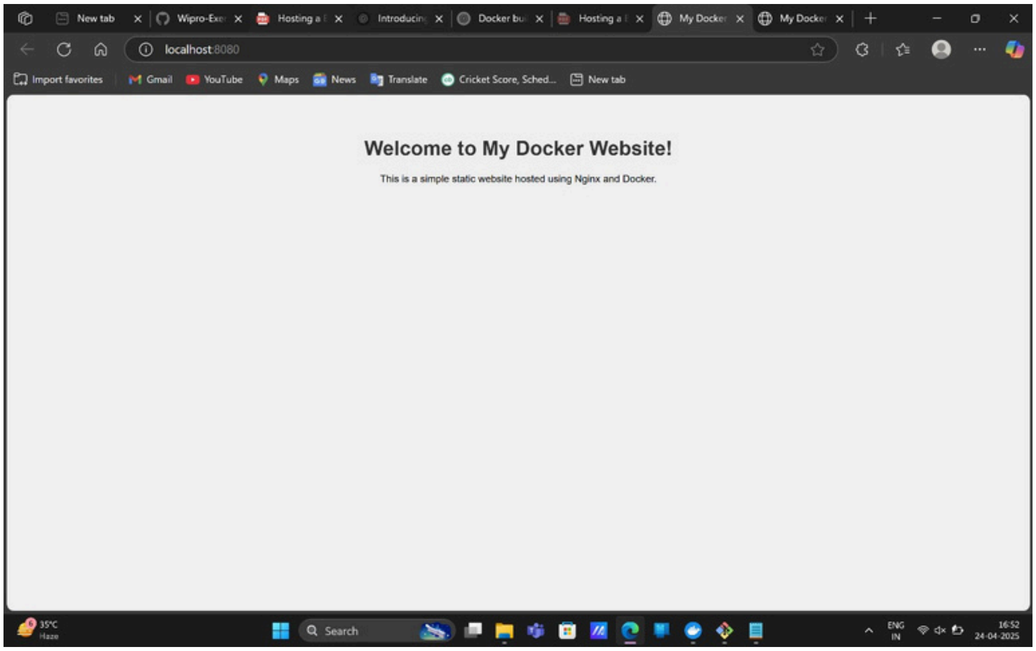
| □ | | Name | Container ID | Image | Port(s) | CPU (%) | L | Actions |
|---|---|---|---|---|---|---|---|---|
| □ | ○ | my-website-con | 47f4071a8981 | my-website | 8080:80 | 0% | 2 | ▷ ⋮ 🗑 |
| □ | ● | my-bootstrap-w | 6d6e4cfa8e45 | my-bootstr | 8080:80 | 0% | 3 | ■ ⋮ 🗑 |

Showing 2 items

RAM 1.04 GB CPU 0.06% Disk: 1.26 GB used (limit 1006.85 GB)

---

<u>STEP-3</u>



Welcome to My Docker Website!

This is a simple static website hosted using Nginx and Docker.

---

# Steps to build my-bootstrap-website

## Step 1: Create Project Directory

Open your terminal and run:

```
mkdir my-bootstrap-website
cd my-bootstrap-website
```

## Step 2: Create Website Files

### File: *index.html*

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>My Docker Website</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- Bootstrap CSS CDN -->
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css
" rel="stylesheet">

  <!-- Custom CSS -->
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container text-center mt-5">
  <h1>Welcome to My Docker Website!</h1>
  <p class="lead">
      This website uses Bootstrap, jQuery, and custom CSS, hosted with Nginx
and Docker.
    </p>
    <button id="toggleButton" class="btn btn-primary mt-3">Toggle
Message</button>
    <p id="message" class="mt-3 d-none">Hello! This message was toggled using
jQuery.</p>
  </div>

  <!-- jQuery CDN -->
  <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
  <!-- Custom JS -->
  <script src="script.js"></script>
</body>
</html>
```

### File: ***`styles.css`***

```
body {
  background-color: #f8f9fa;
  font-family: Arial, sans-serif;
}
h1 {
  color: #343a40;
}

.lead {
  color: #6c757d;
}
```

---

### File: ***`script.js`***

```
$(document).ready(function() {
  $('#toggleButton').click(function() {
    $('#message').toggleClass('d-none');
  });
});
```

---

## Step 3: Create Dockerfile

### File: ***`Dockerfile`***

```
# Use the official Nginx image from Docker Hub
FROM nginx:alpine
# Copy website files to the Nginx HTML directory
COPY index.html /usr/share/nginx/html/
COPY styles.css /usr/share/nginx/html/
COPY script.js /usr/share/nginx/html/
# Expose port 80 for the web server
EXPOSE 80
# Start Nginx when the container launches
CMD ["nginx", "-g", "daemon off;"]
```

---

## Step 4: Build and Run the Docker Container

1. Build the Docker image:

```
docker build -t my-bootstrap-website .
```

2. Run the Docker container:

```
docker run -d -p 8080:80 --name my-bootstrap-website-container my-bootstrap-
website
```

## Step 5: Access the Website

Open your browser and visit:

📍 http://localhost:8080

You should see a Bootstrap-styled page with a button and a message that toggles using jQuery.

## Step 6: Stop and Clean Up (Optional)

1. Stop and remove the container:

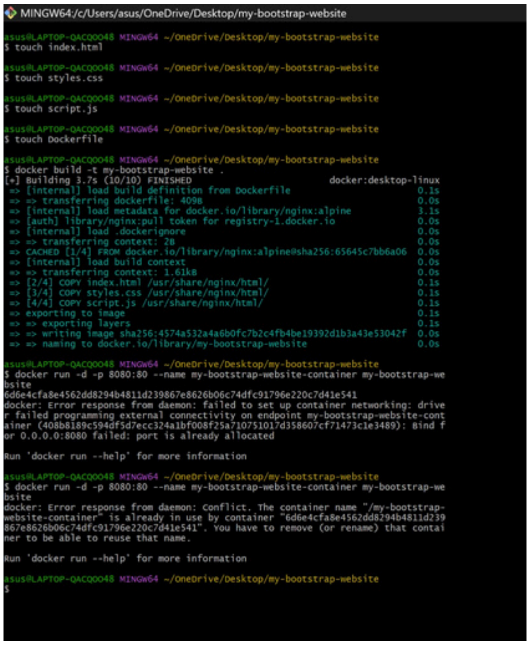
```
docker stop my-bootstrap-website-container
docker rm my-bootstrap-website-container
```

2. Remove the Docker image:

```
docker rmi my-bootstrap-website
```

## SCREENSHOT OF THE STEPS

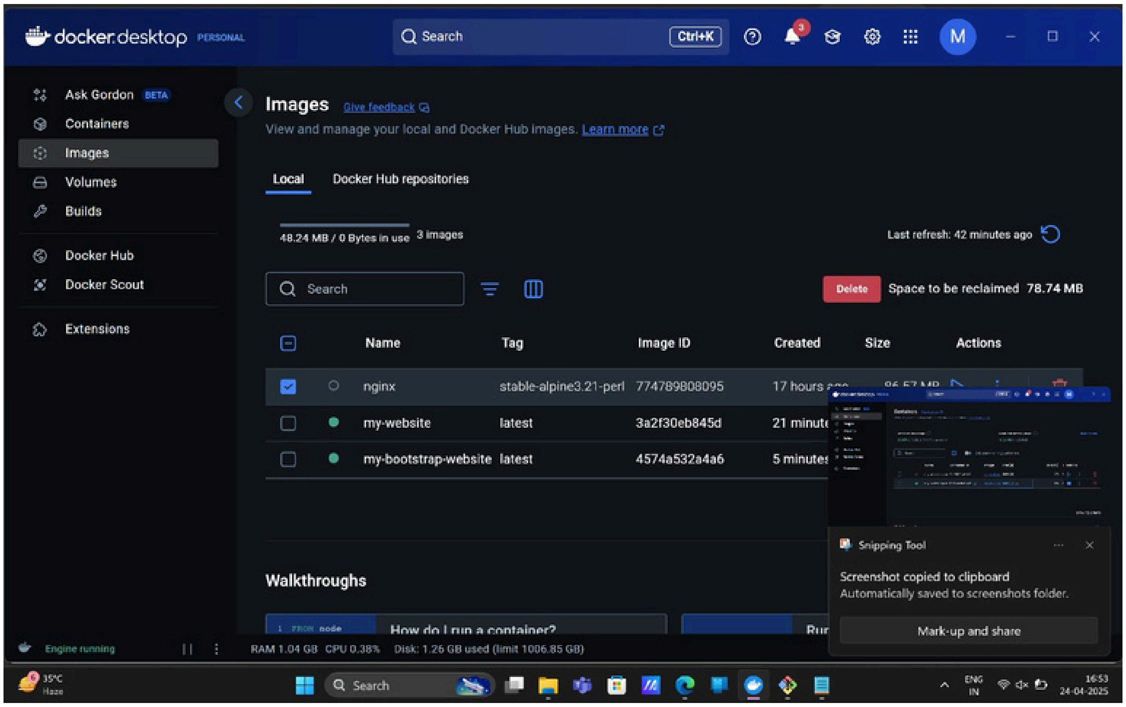
### STEP-1



MINGW64:/c/Users/asus/OneDrive/Desktop/my-bootstrap-website

```
asus@LAPTOP-QACQ0048 MINGW64 ~/OneDrive/Desktop/my-bootstrap-website
$ touch index.html

asus@LAPTOP-QACQ0048 MINGW64 ~/OneDrive/Desktop/my-bootstrap-website
$ touch styles.css

asus@LAPTOP-QACQ0048 MINGW64 ~/OneDrive/Desktop/my-bootstrap-website
$ touch script.js

asus@LAPTOP-QACQ0048 MINGW64 ~/OneDrive/Desktop/my-bootstrap-website
$ touch Dockerfile

asus@LAPTOP-QACQ0048 MINGW64 ~/OneDrive/Desktop/my-bootstrap-website
$ docker build -t my-bootstrap-website .
[+] Building 3.7s (10/10) FINISHED                          docker:desktop-linux
 => [internal] load build definition from Dockerfile                        0.1s
 => => transferring dockerfile: 409B                                        0.0s
 => [internal] load metadata for docker.io/library/nginx:alpine             3.1s
 => [auth] library/nginx:pull token for registry-1.docker.io                0.0s
 => [internal] load .dockerignore                                           0.0s
 => => transferring context: 2B                                             0.0s
 => CACHED [1/4] FROM docker.io/library/nginx:alpine@sha256:65645c7bb6a06   0.0s
 => [internal] load build context                                           0.0s
 => => transferring context: 1.61kB                                         0.0s
 => [2/4] COPY index.html /usr/share/nginx/html/                            0.1s
 => [3/4] COPY styles.css /usr/share/nginx/html/                            0.1s
 => [4/4] COPY script.js /usr/share/nginx/html/                             0.1s
 => exporting to image                                                      0.1s
 => => exporting layers                                                     0.1s
 => => writing image sha256:4574a532a4a6b0fc7b2c4fb4be19392d1b3a43e53042f   0.0s
 => => naming to docker.io/library/my-bootstrap-website                     0.0s

asus@LAPTOP-QACQ0048 MINGW64 ~/OneDrive/Desktop/my-bootstrap-website
$ docker run -d -p 8080:80 --name my-bootstrap-website-container my-bootstrap-we
bsite
6d6e4cfa8e4562dd8294b4811d239867e8626b06c74dfc91796e220c7d41e541
docker: Error response from daemon: failed to set up container networking: drive
r failed programming external connectivity on endpoint my-bootstrap-website-cont
ainer (408b8189c594df5d7ecc324a1bf008f25a710751017d358607cf71473c1e3489): Bind f
or 0.0.0.0:8080 failed: port is already allocated

Run 'docker run --help' for more information

asus@LAPTOP-QACQ0048 MINGW64 ~/OneDrive/Desktop/my-bootstrap-website
$ docker run -d -p 8080:80 --name my-bootstrap-website-container my-bootstrap-we
bsite
docker: Error response from daemon: Conflict. The container name "/my-bootstrap-
website-container" is already in use by container "6d6e4cfa8e4562dd8294b4811d239
867e8626b06c74dfc91796e220c7d41e541". You have to remove (or rename) that contai
ner to be able to reuse that name.

Run 'docker run --help' for more information

asus@LAPTOP-QACQ0048 MINGW64 ~/OneDrive/Desktop/my-bootstrap-website
$
```

## STEP-2



Images Give feedback

View and manage your local and Docker Hub images. Learn more

48.24 MB / 0 Bytes in use 3 images

Last refresh: 42 minutes ago

Space to be reclaimed 78.74 MB

| ☐ | | Name | Tag | Image ID | Created | Size | Actions |
|---|---|---|---|---|---|---|---|
| ☑ | ○ | nginx | stable-alpine3.21-perl | 774789808095 | 17 hours ago | 96.57 MB | |
| ☐ | ● | my-website | latest | 3a2f30eb845d | 21 minute | | |
| ☐ | ● | my-bootstrap-website | latest | 4574a532a4a6 | 5 minutes | | |

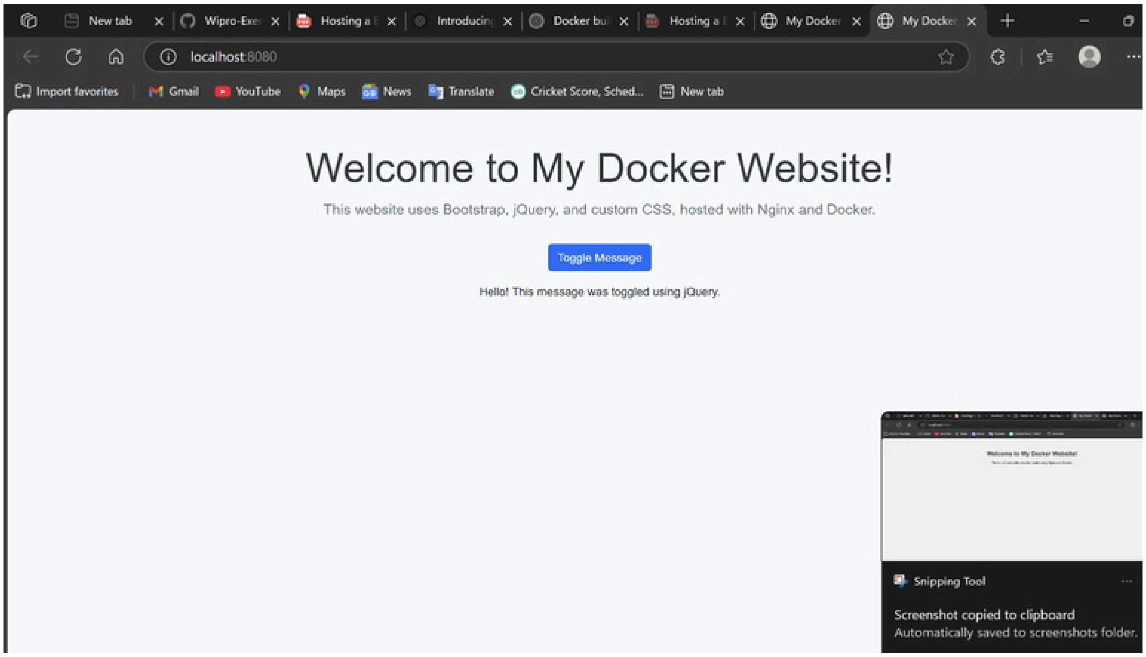
Walkthroughs

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

RAM 1.04 GB CPU 0.38% Disk: 1.26 GB used (limit 1006.85 GB)

## STEP-3



# Welcome to My Docker Website!

This website uses Bootstrap, jQuery, and custom CSS, hosted with Nginx and Docker.

Hello! This message was toggled using jQuery.

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.