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:

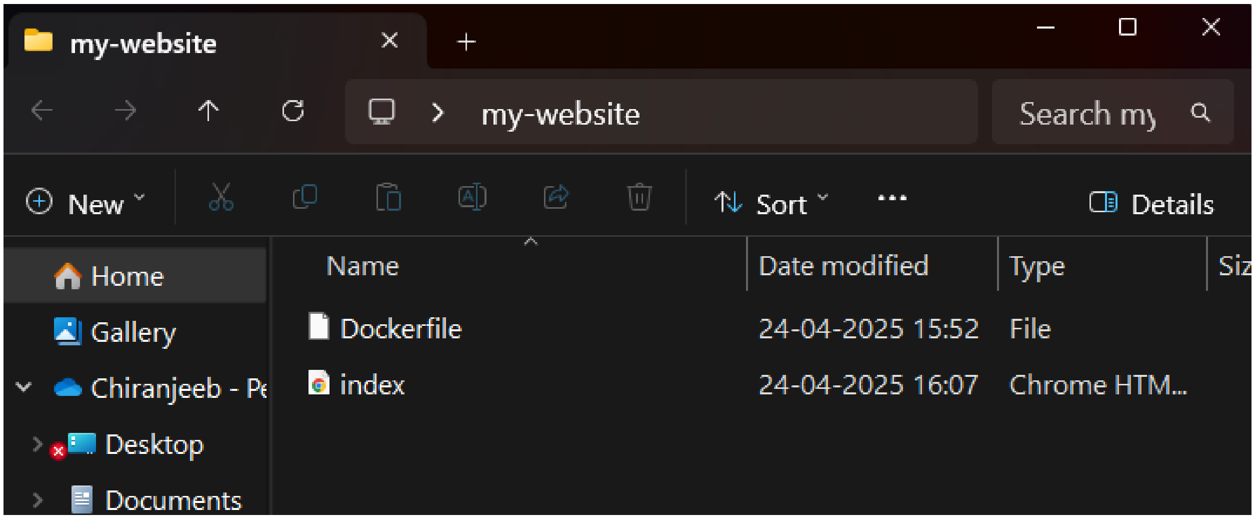
2. docker stop my-website-container

3. docker rm my-website-container

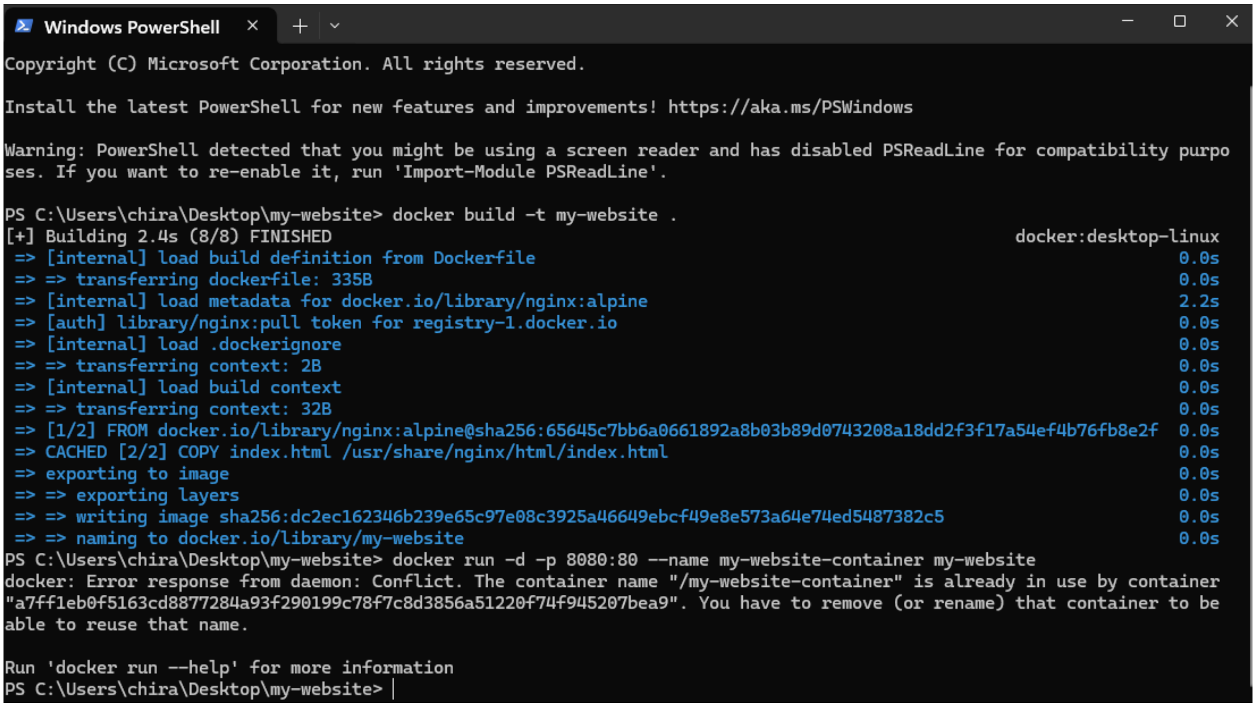
4. Remove the Docker image:

5. docker rmi my-website

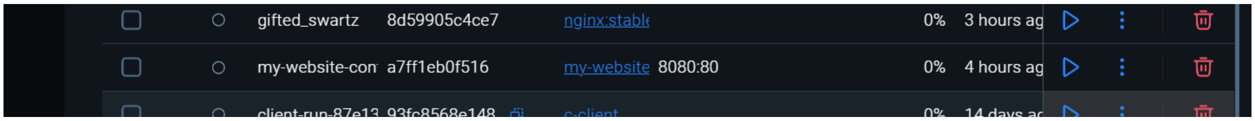
SCREENSHOT OF THE STEPS



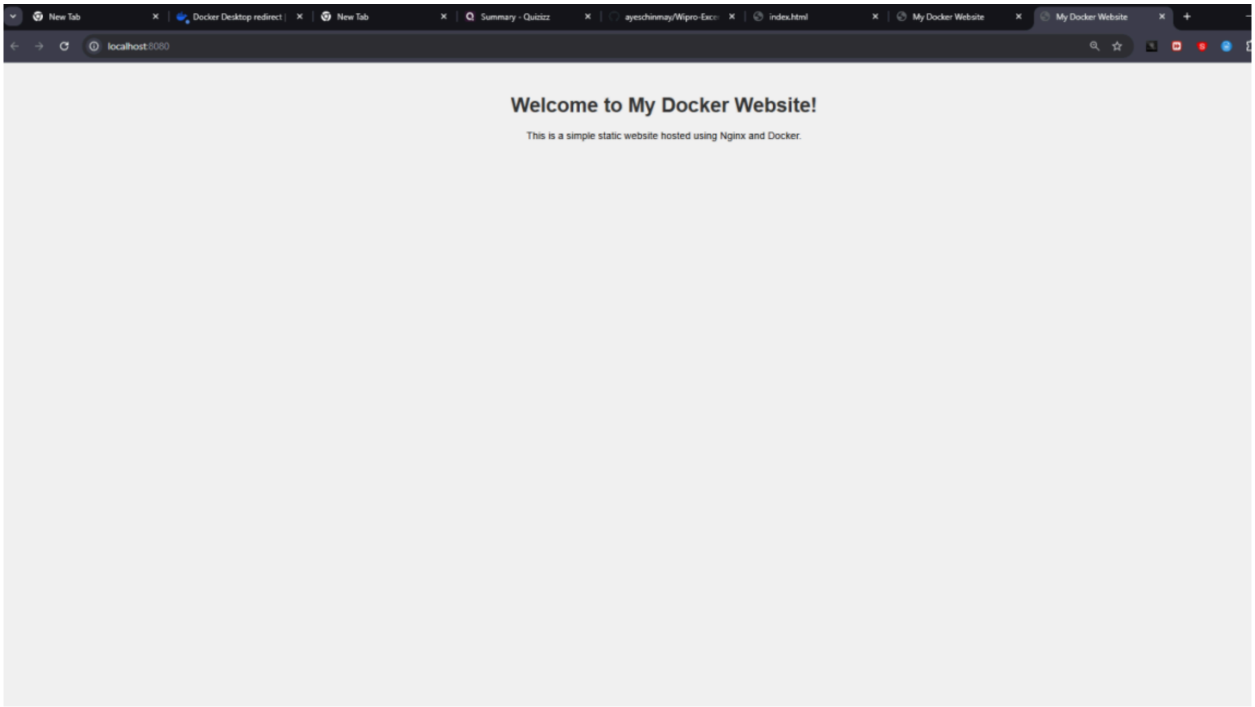
Name	Date modified	Type	Siz
Dockerfile	24-04-2025 15:52	File	
index	24-04-2025 16:07	Chrome HTM...	



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

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purpo
ses. If you want to re-enable it, run 'Import-Module PSReadLine'.

PS C:\Users\chira\Desktop\my-website> docker build -t my-website .
[+] Building 2.4s (8/8) FINISHED                                                                   docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                               0.0s
 => => transferring dockerfile: 335B                                                                               0.0s
 => [internal] load metadata for docker.io/library/nginx:alpine                                                    2.2s
 => [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                                                                                  0.0s
 => => transferring context: 32B                                                                                   0.0s
 => [1/2] FROM docker.io/library/nginx:alpine@sha256:65645c7bb6a0661892a8b03b89d0743208a18dd2f3f17a54ef4b76fb8e2f  0.0s
 => CACHED [2/2] COPY index.html /usr/share/nginx/html/index.html                                                  0.0s
 => exporting to image                                                                                             0.0s
 => => exporting layers                                                                                            0.0s
 => => writing image sha256:dc2ec162346b239e65c97e08c3925a46649ebcf49e8e573a64e74ed5487382c5                       0.0s
 => => naming to docker.io/library/my-website                                                                      0.0s
PS C:\Users\chira\Desktop\my-website> docker run -d -p 8080:80 --name my-website-container my-website
docker: Error response from daemon: Conflict. The container name "/my-website-container" is already in use by container
"a7ff1eb0f5163cd8877284a93f290199c78f7c8d3856a51220f74f945207bea9". You have to remove (or rename) that container to be
able to reuse that name.

Run 'docker run --help' for more information
PS C:\Users\chira\Desktop\my-website> |
```



□	○	gifted_swartz	8d59905c4ce7	nginx:stable		0%	3 hours ag
□	○	my-website-con	a7ff1eb0f516	my-website	8080:80	0%	4 hours ag
□	○	client-run-87e13	93fc8568e148	c-client		0%	14 days ac



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=1.0">
<title>My Docker Bootstrap 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.
<button id="toggleButton" class="btn btn-primary mt-3">Toggle Message
<p id="message" class="mt-3 d-none">Hello! This message was toggled using jQuery. </p>
</div>
 <!-- jQuery y 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:

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-bootstrapwebsite

Step 5: Access the Website Open your browser and visit:

http://localhost:8080

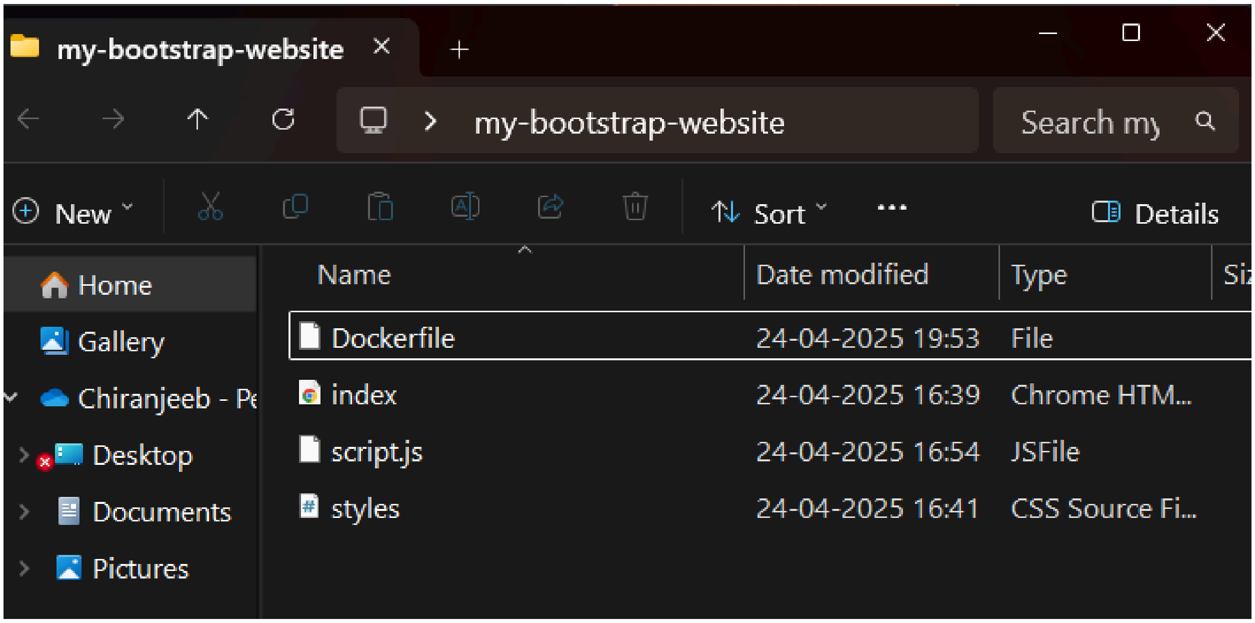
You should see a Bootstrap-styled page with a button and a message that toggles using jQuery

Step 6: Stop and Clean Up (Optional)

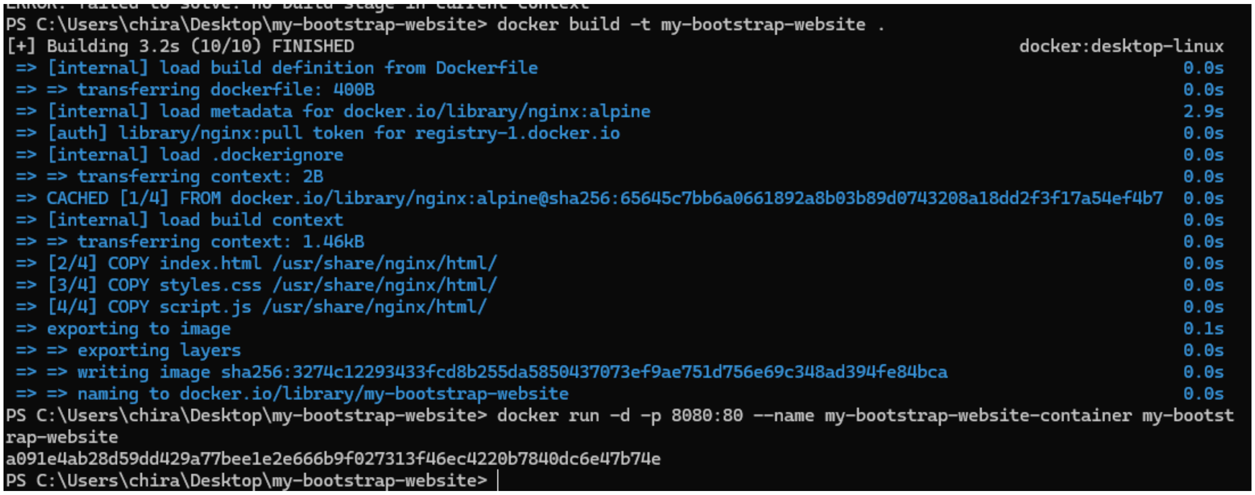
1. Stop and remove the container: docker stop my-bootstrap-website-container docker rm my-bootstrap-website-container

2. Remove the Docker image: docker rmi my-bootstrap-website

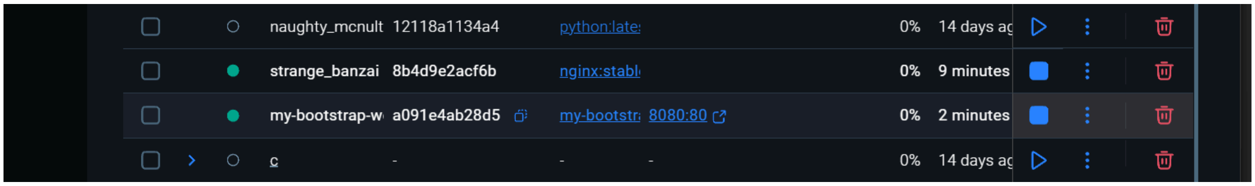
SCREENSHOT OF THE STEPS:



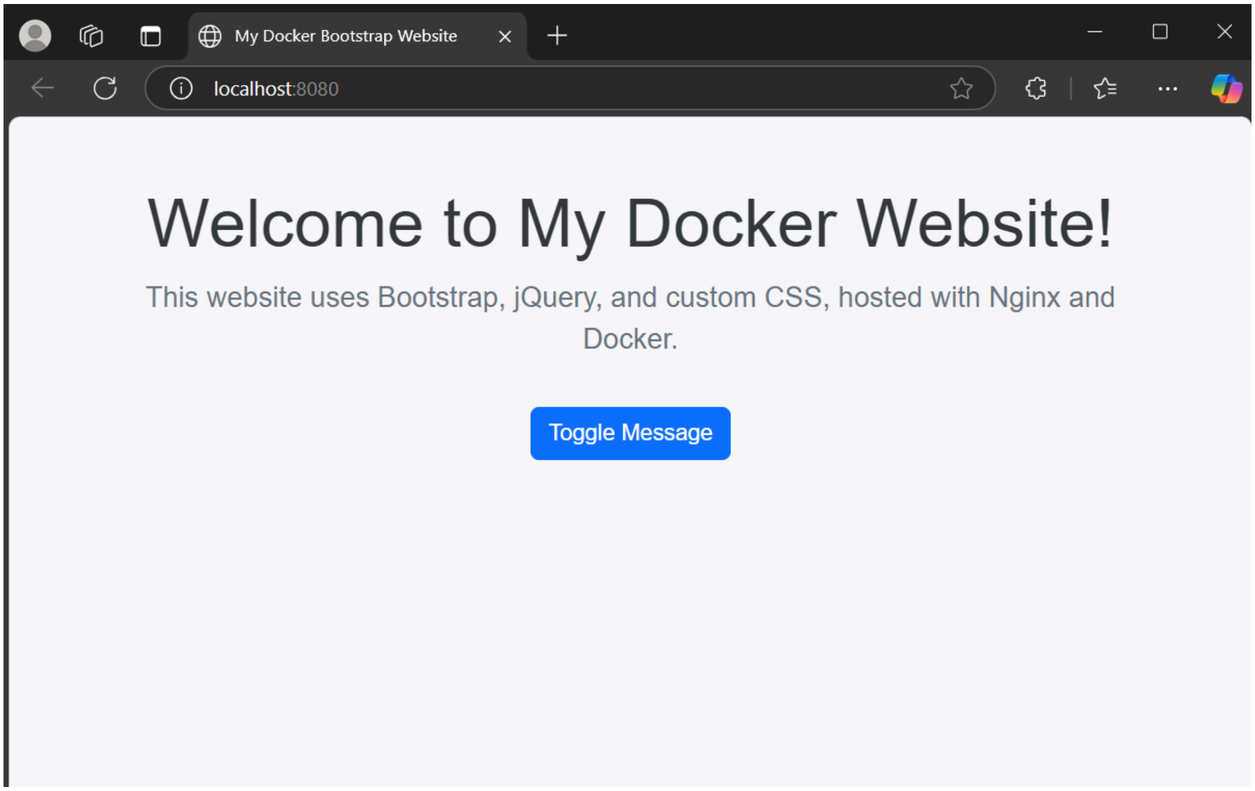
Name	Date modified	Type	Si
Dockerfile	24-04-2025 19:53	File	
index	24-04-2025 16:39	Chrome HTM...	
script.js	24-04-2025 16:54	JSFile	
styles	24-04-2025 16:41	CSS Source Fi...	



```
ERROR: failed to solve: no build stage in current context
PS C:\Users\chira\Desktop\my-bootstrap-website> docker build -t my-bootstrap-website .
[+] Building 3.2s (10/10) FINISHED                                                         docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                       0.0s
 => => transferring dockerfile: 400B                                                                       0.0s
 => [internal] load metadata for docker.io/library/nginx:alpine                                            2.9s
 => [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@sha256:65645c7bb6a0661892a8b03b89d0743208a18dd2f3f17a54ef4b7  0.0s
 => [internal] load build context                                                                          0.0s
 => => transferring context: 1.46kB                                                                        0.0s
 => [2/4] COPY index.html /usr/share/nginx/html/                                                           0.0s
 => [3/4] COPY styles.css /usr/share/nginx/html/                                                           0.0s
 => [4/4] COPY script.js /usr/share/nginx/html/                                                            0.0s
 => exporting to image                                                                                     0.1s
 => => exporting layers                                                                                    0.0s
 => => writing image sha256:3274c12293433fcd8b255da5850437073ef9ae751d756e69c348ad394fe84bca              0.0s
 => => naming to docker.io/library/my-bootstrap-website                                                    0.0s
PS C:\Users\chira\Desktop\my-bootstrap-website> docker run -d -p 8080:80 --name my-bootstrap-website-container my-bootst
rap-website
a091e4ab28d59dd429a77bee1e2e666b9f027313f46ec4220b7840dc6e47b74e
PS C:\Users\chira\Desktop\my-bootstrap-website> |
```



□	○	naughty_mcnult	12118a1134a4	python:lates		0%	14 days ag
□	●	strange_banzai	8b4d9e2acf6b	nginx:stabl		0%	9 minutes
□	●	my-bootstrap-w	a091e4ab28d5	my-bootstr	8080:80	0%	2 minutes
□ ›	○	c	-	-	-	0%	14 days ag



Welcome to My Docker Website!

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

Toggle Message