Aarna Bafna
Devops Lab
T11
07

LAB ASSIGNMENT 8

AIM: : Deploy static web application on docker.

LAB OUTCOME:
LO1, LO5 Mapped.

THEORY:
To deploy a static web application on Docker, you can follow these steps:

1. Install Docker Desktop:
If you haven't already, download and install Docker Desktop for Windows. You can get it from the official Docker website: https://www.docker.com/products/docker-desktop

2. Verify Docker Installation:
After installation, open Docker Desktop to ensure that it's running correctly. You should see the Docker icon in your system tray.

3. Create a Dockerfile:
Create a Dockerfile in the root directory of your web application. This file is used to define how your application should be built and run within a Docker container. Here's a simple example of a Dockerfile for a static web application:

```
Dockerfile
  # Use an official Nginx image as the base image
  FROM nginx:alpine

  # Copy your static web application files to the container
  COPY ./path/to/your/app /usr/share/nginx/html

  # Expose port 80 to the host
  EXPOSE 80
```

4. Build the Docker Image:
Open a terminal and navigate to the directory containing your Dockerfile. Run the following command to build a Docker image:

```
docker build -t my-web-app .
```

Replace `my-web-app` with your desired image name, and don't forget the period at the end, which indicates the current directory.

5. Run the Docker Container:
After building the image, you can start a Docker container based on that image using the following command:

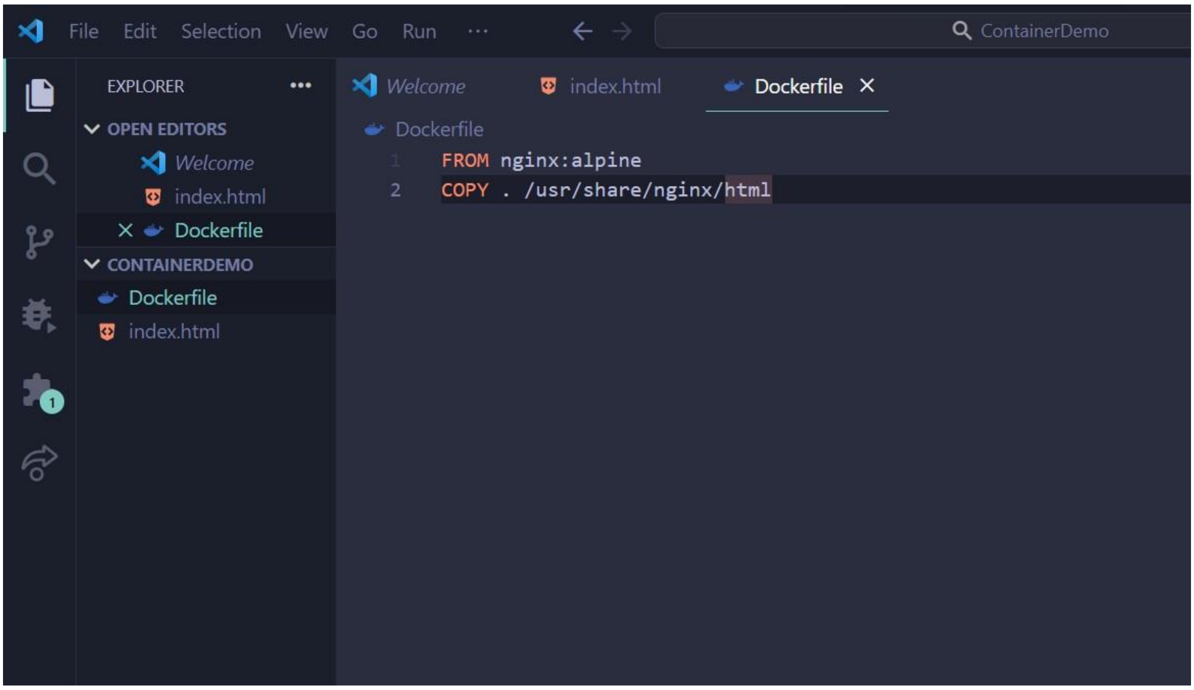
```
docker run -d -p 8080:80 my-web-app
```

This command runs the container in detached mode (`-d`) and maps port 8080 on your host to port 80 in the container. You can choose a different port if you like.

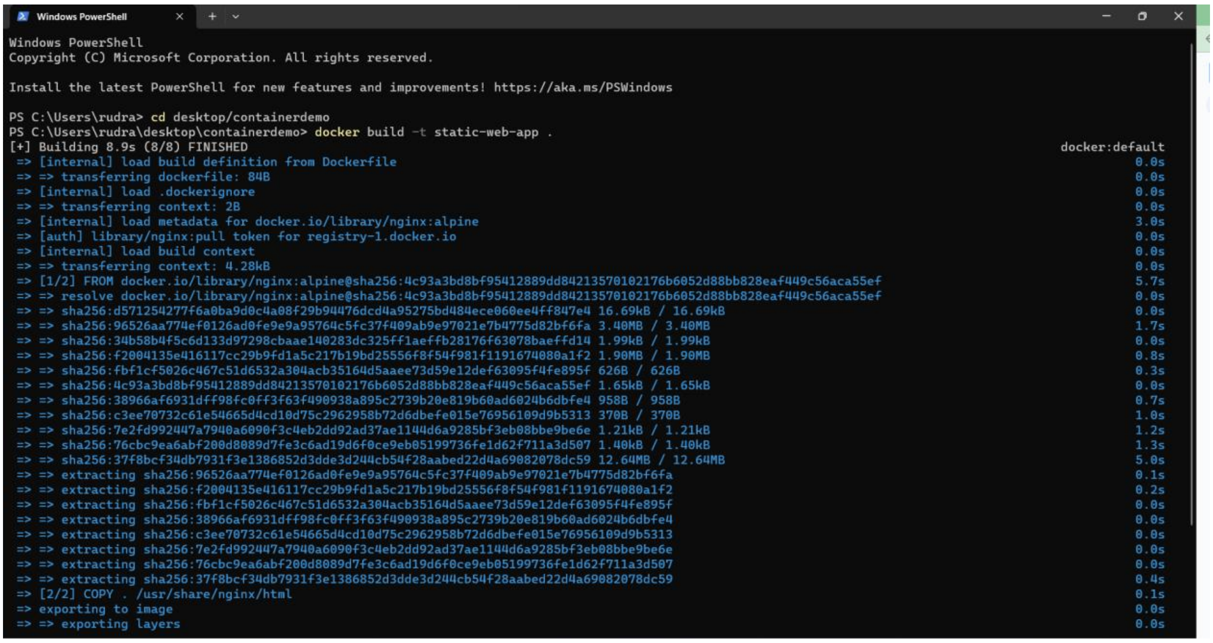
6. Access Your Web Application:
Open a web browser and navigate to `http://localhost:8080` (or the port you specified in step 5). You should be able to access your static web application running inside the Docker container.

7. Manage Docker Containers:
You can manage your Docker containers using Docker commands like `docker ps` to list running containers, `docker stop <container_id>` to stop a container, and `docker rm <container_id>` to remove a container.

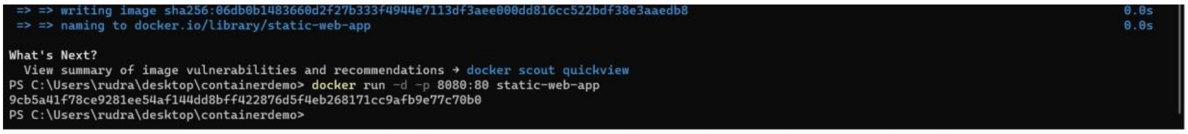
SCREENSHOTS:



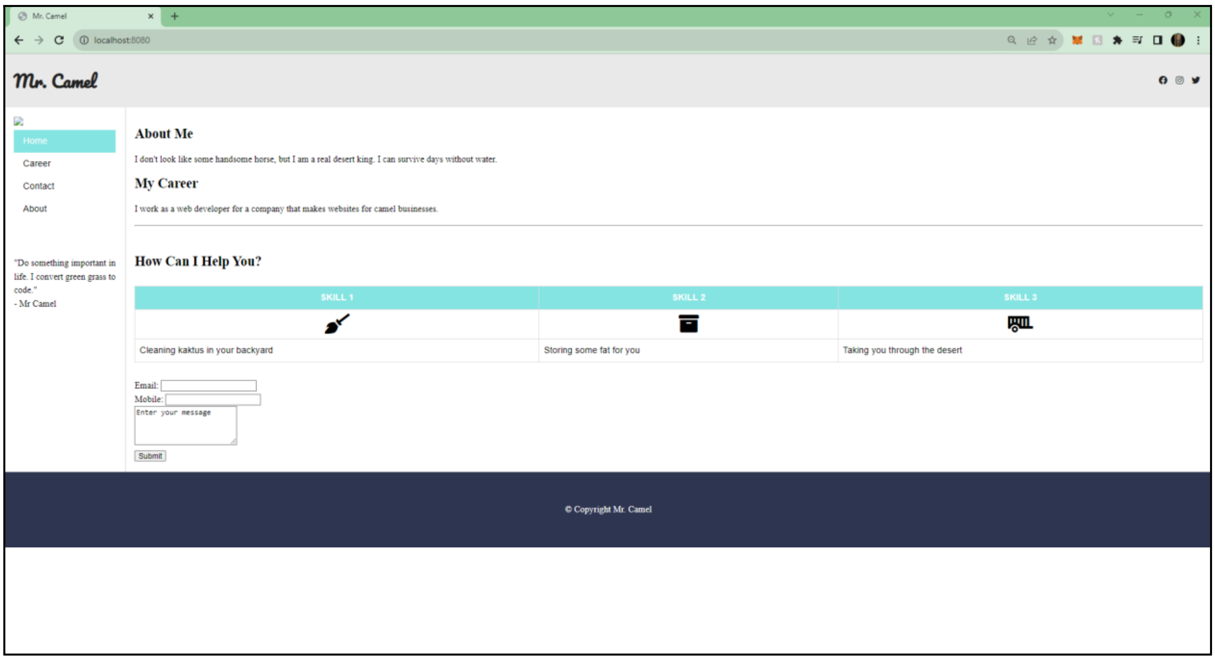
```
FROM nginx:alpine
COPY . /usr/share/nginx/html
```



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

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

PS C:\Users\rudra> cd desktop/containerdemo
PS C:\Users\rudra\desktop\containerdemo> docker build -t static-web-app .
[+] Building 8.9s (8/8) FINISHED                                                                                                            docker:default
 => [internal] load build definition from Dockerfile                                                                                                  0.0s
 => => transferring dockerfile: 84B                                                                                                                   0.0s
 => [internal] load .dockerignore                                                                                                                     0.0s
 => => transferring context: 2B                                                                                                                       0.0s
 => [internal] load metadata for docker.io/library/nginx:alpine                                                                                       3.0s
 => [auth] library/nginx:pull token for registry-1.docker.io                                                                                          0.0s
 => [internal] load build context                                                                                                                     0.0s
 => => transferring context: 4.28kB                                                                                                                   0.0s
 => [1/2] FROM docker.io/library/nginx:alpine@sha256:4c93a3bd8bf95412889dd84213570102176b6052d88bb828eaf449c56aca55ef                                 5.7s
 => => resolve docker.io/library/nginx:alpine@sha256:4c93a3bd8bf95412889dd84213570102176b6052d88bb828eaf449c56aca55ef                                 0.0s
 => => sha256:d571254277f6a0ba9d0c4a08f29b94476dcd4a95275bd484ece060ee4ff847e4 16.69kB / 16.69kB                                                      0.0s
 => => sha256:96526aa774ef0126ad0fe9e9a95764c5fc37f409ab9e97021e7b4775d82bf6fa 3.40MB / 3.40MB                                                        1.7s
 => => sha256:34b58b4f5c6d133d97298cbaae140283dc325ff1aeffb28176f63078baeffd14 1.99kB / 1.99kB                                                        0.0s
 => => sha256:f2004135e416117cc29b9fd1a5c217b19bd25556f8f54f981f1191674080a1f2 1.90MB / 1.90MB                                                        0.8s
 => => sha256:fbf1cf5026c467c51d6532a304acb35164d5aaee73d59e12def63095f4fe895f 626B / 626B                                                            0.3s
 => => sha256:4c93a3bd8bf95412889dd84213570102176b6052d88bb828eaf449c56aca55ef 1.65kB / 1.65kB                                                        0.0s
 => => sha256:38966af6931dff98fc0ff3f63f490938a895c2739b20e819b60ad6024b6dbfe4 958B / 958B                                                            0.7s
 => => sha256:c3ee70732c61e54665d4cd10d75c2962958b72d6dbefe015e76956109d9b5313 370B / 370B                                                            1.0s
 => => sha256:7e2fd992447a7940a6090f3c4eb2dd92ad37ae1144d6a9285bf3eb08bbe9be6e 1.21kB / 1.21kB                                                        1.2s
 => => sha256:76cbc9ea6abf200d8089d7fe3c6ad19d6f0ce9eb05199736fe1d62f711a3d507 1.40kB / 1.40kB                                                        1.3s
 => => sha256:37f8bcf34db7931f3e1386852d3dde3d244cb54f28aabed22d4a69082078dc59 12.64MB / 12.64MB                                                      5.0s
 => => extracting sha256:96526aa774ef0126ad0fe9e9a95764c5fc37f409ab9e97021e7b4775d82bf6fa                                                             0.1s
 => => extracting sha256:f2004135e416117cc29b9fd1a5c217b19bd25556f8f54f981f1191674080a1f2                                                             0.2s
 => => extracting sha256:fbf1cf5026c467c51d6532a304acb35164d5aaee73d59e12def63095f4fe895f                                                             0.0s
 => => extracting sha256:38966af6931dff98fc0ff3f63f490938a895c2739b20e819b60ad6024b6dbfe4                                                             0.0s
 => => extracting sha256:c3ee70732c61e54665d4cd10d75c2962958b72d6dbefe015e76956109d9b5313                                                             0.0s
 => => extracting sha256:7e2fd992447a7940a6090f3c4eb2dd92ad37ae1144d6a9285bf3eb08bbe9be6e                                                             0.0s
 => => extracting sha256:76cbc9ea6abf200d8089d7fe3c6ad19d6f0ce9eb05199736fe1d62f711a3d507                                                             0.0s
 => => extracting sha256:37f8bcf34db7931f3e1386852d3dde3d244cb54f28aabed22d4a69082078dc59                                                             0.4s
 => [2/2] COPY . /usr/share/nginx/html                                                                                                                0.1s
 => exporting to image                                                                                                                                0.0s
 => => exporting layers                                                                                                                               0.0s
```



```
 => => writing image sha256:06db0b1483660d2f27b333f4944e7113df3aee000dd816cc522bdf38e3aaedb8                                                          0.0s
 => => naming to docker.io/library/static-web-app                                                                                                     0.0s

What's Next?
  View summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\Users\rudra\desktop\containerdemo> docker run -d -p 8080:80 static-web-app
9cb5a41f78ce9281ee54af144dd8bff422876d5f4eb268171cc9afb9e77c70b0
PS C:\Users\rudra\desktop\containerdemo>
```



About Me

I don't look like some handsome horse, but I am a real desert king. I can survive days without water.

My Career

I work as a web developer for a company that makes websites for camel businesses.

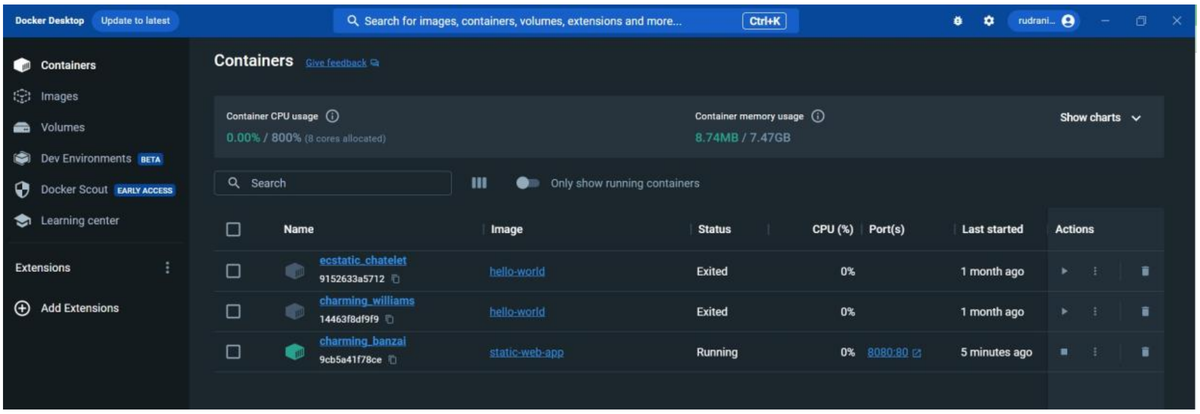
"Do something important in life. I convert green grass to code."
- Mr Camel

How Can I Help You?

SKILL 1	SKILL 2	SKILL 3
Cleaning kaktus in your backyard	Storing some fat for you	Taking you through the desert

Email:
Mobile:
Enter your message
Submit

© Copyright Mr. Camel

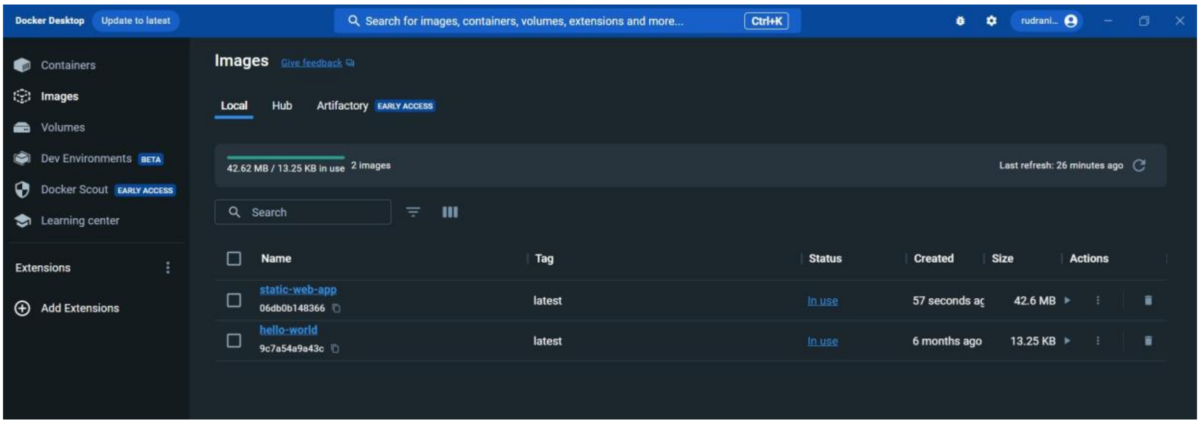


Containers

Container CPU usage 0.00% / 800% (8 cores allocated)

Container memory usage 8.74MB / 7.47GB

Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
ecstatic_chatelet 9152633a5712	hello-world	Exited	0%		1 month ago	
charming_williams 14463f8df9f9	hello-world	Exited	0%		1 month ago	
charming_banzai 9cb5a41f78ce	static-web-app	Running	0%	8080:80	5 minutes ago	



Images

42.62 MB / 13.25 KB in use 2 images

Last refresh: 26 minutes ago

Name	Tag	Status	Created	Size	Actions
static-web-app 06db0b148366	latest	In use	57 seconds ag	42.6 MB	
hello-world 9c7a54a9a43c	latest	In use	6 months ago	13.25 KB	

CONCLUSION:

In summary, deploying a static web application on Docker in Windows 11 is a straightforward process. By installing Docker Desktop, creating a Dockerfile, building an image, and running a container, you can host your web app with ease. Managing containers and cleaning up resources is also manageable, making it an efficient and scalable solution for web application deployment.