Docker Project Documentation: From Structure to Docker Hub

Project Structure

Here's the initial project structure:

docker/

├── Dockerfile

└── src/

└── html/

└── index.html

Path: cd C:\Users\USER\Desktop\learning\Aws\docker

Index.html :  
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Welcome to Nginx</title>

</head>

<body>

<h1>Welcome to Nginx!</h1>

</body>

</html>

**docker/ :**# Use the official Nginx image from the Docker Hub

FROM nginx:1.10.1-alpine

COPY src/html /usr/share/nginx/html

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]

**Build the Docker Image**:

* Open a terminal and navigate to the **docker/** directory.
* Build the Docker image using the following command:

docker build -t my-nginx-image .

**Run the Docker Container**:

* Run the Docker container and map port 80 of the container to port 8080 of the host:

docker run -d -p 8080:80 my-nginx-image

You can check if the container is running using:

docker ps

Access the Application:

Open a web browser and navigate to *http://localhost:8080*. You should see the "Welcome to Nginx!" page.

**Log in to Docker Hub:**

Log in to your Docker Hub account:

docker login

Enter your Docker Hub username and password when prompted.

Tag your Docker image with your Docker Hub username and repository name:

docker tag my-nginx-image meshackkimutai/my-nginx-image:latest

**Push the Docker Image:**

Push the tagged image to Docker Hub:

docker push meshackkimutai/my-nginx-image:latest

**Verify the Image on Docker Hub:**

* Go to Docker Hub.
* Log in to your account.
* Navigate to your profile and find the repository my-nginx-image.

**Additional Docker Commands**

Here are some additional Docker commands you might find useful:

List all Docker images:

docker images

Remove a Docker image:

docker rmi <image\_id>

Stop a running container:

docker stop <container\_id>

Remove a stopped container:

docker rm <container\_id>

List all Docker containers (including stopped ones):

docker ps -a

Display detailed information about a container:

docker inspect <container\_id>

Display real-time log output of a container:

docker logs -f <container\_id>

Enter a running container (for debugging purposes):

docker exec -it <container\_id> /bin/sh

Summary

This guide has taken you through the steps from creating your project structure, writing a Dockerfile, building and running a Docker image, to pushing it to Docker Hub. Additionally, it has provided some useful Docker commands for managing images and containers.