**DevOps Lab**

**LAB 10**

Shashwat Tripathi

Batch C

D15A Roll No: 64

**Aim -** To learn Dockerfile instructions, build an image for a sample web application using Dockerfile.

**Output-**

**Index.js**

// index.js

const http = require('http');

const port = 3000;

const server = http.createServer((req, res) => {

res.writeHead(200, { 'Content-Type': 'text/plain' });

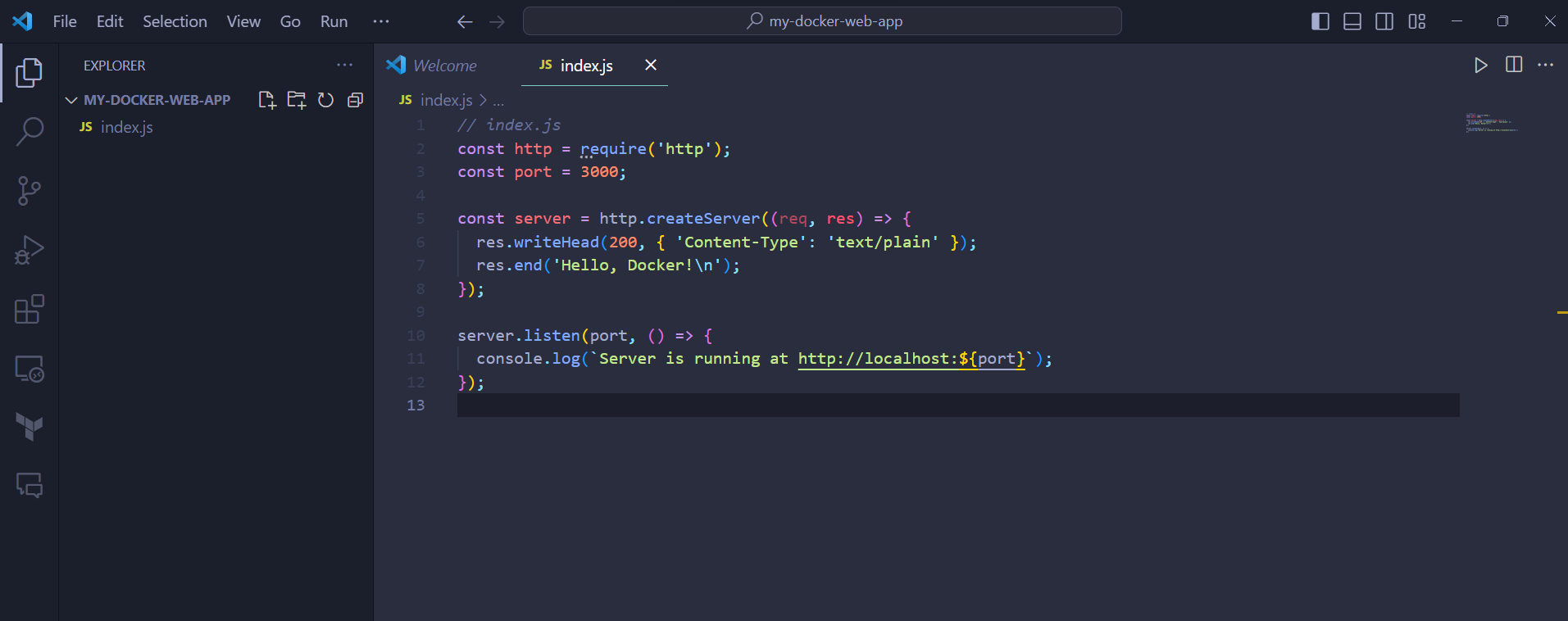
res.end('Hello, Docker!\n');

});

server.listen(port, () => {

console.log(`Server is running at http://localhost:${port}`);

});



**package.json**

{

"name": "docker-web-app",

"version": "1.0.0",

"description": "A simple Node.js web app for Docker",

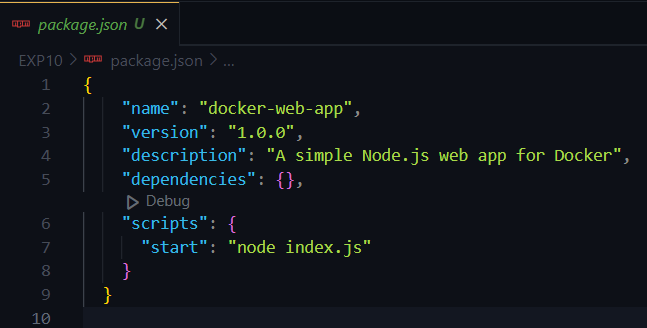
"dependencies": {},

"scripts": {

"start": "node index.js"

}

}



**Dockerfile**

# Use an official Node.js runtime as a parent image

FROM node:14

# Set the working directory in the container

WORKDIR /app

# Copy package.json and package-lock.json to the container

COPY package\*.json ./

# Install application dependencies

RUN npm install

# Copy your application's source code into the container

COPY . .

# Expose the port your application will run on

EXPOSE 3000

# Define the command to start your application

CMD [ "npm", "start" ]

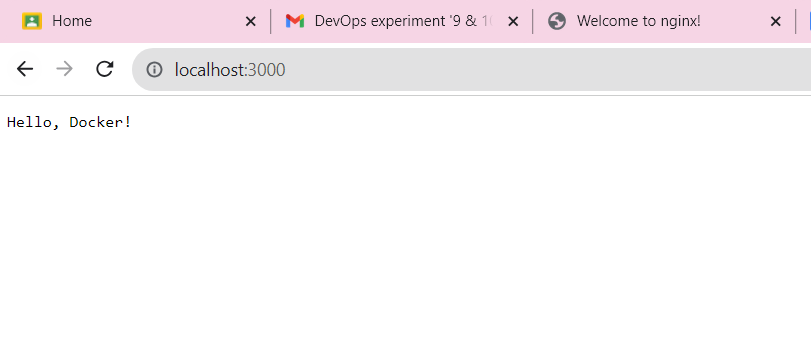
**Build the Docker Image:**

docker build -t dockexp .



**Run a Container from the Image:**

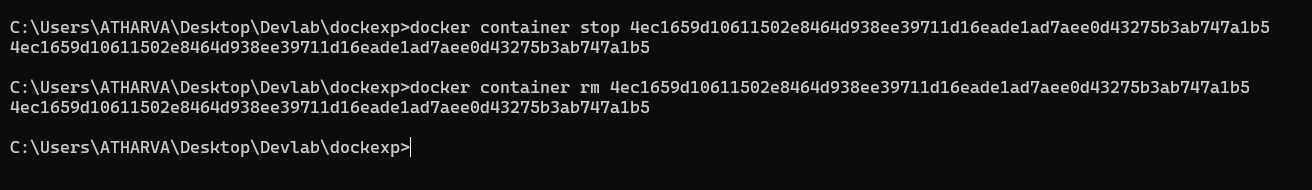
docker run -p 3000:3000 dockexp

****

**Cleanup:**

To stop the running container, press Ctrl+C in the terminal. To remove the container and the image, you can use the following commands:

To get the docker id cmd docker ps



**Conclusion:** Thus, we have built an image for a sample web application using Dockerfile.