**Name: Meet Brijwani**

**Batch: T11**

**Roll no: 14**

**Experiment 10**

**AIM:** To learn Docker file instructions, build an image for a sample web application using DOCKERFILE.

**THEORY:**

A **Dockerfile** is a text file that contains a list of instructions for Docker to build an image. It automates the process of creating a Docker image by specifying everything your app needs to run — from base images, dependencies, to startup commands.

**Common Dockerfile Instructions:**

| **Instruction** | **Description** |
| --- | --- |
| FROM | Specifies the base image (e.g., node:18-alpine, python:3.10) |
| WORKDIR | Sets the working directory inside the container |
| COPY | Copies files from your system to the container |
| RUN | Executes commands to install dependencies or perform setup |
| CMD | Specifies the default command to run when the container starts |
| EXPOSE | Documents the port the container will listen on |
| ENV | Sets environment variables |
| ENTRYPOINT | Like CMD, but used when you want the command to always run |

**Practical: Build a Docker Image for a Sample Web App**

Let’s take a **Node.js Express** web application as an example.

**1. Project Structure:**

sample-app/

├── Dockerfile

├── package.json

├── package-lock.json

└── index.js

**2. package.json**

json

CopyEdit

{

"name": "sample-app",

"version": "1.0.0",

"main": "index.js",

"scripts": {

"start": "node index.js"

},

"dependencies": {

"express": "^4.18.2"

}

}

**3. index.js**

js

const express = require('express');

const app = express();

const PORT = 3000;

app.get('/', (req, res) => {

res.send('Hello, Docker!');

});

app.listen(PORT, () => {

console.log(`Server is running on port ${PORT}`);

});

**4. Dockerfile**

dockerfile

# Step 1: Use an official Node.js runtime as a parent image

FROM node:18-alpine

# Step 2: Set working directory

WORKDIR /app

# Step 3: Copy package.json and package-lock.json

COPY package\*.json ./

# Step 4: Install dependencies

RUN npm install

# Step 5: Copy source code

COPY . .

# Step 6: Expose the port your app runs on

EXPOSE 3000

# Step 7: Define the command to run the app

CMD ["npm", "start"]

**Build & Run the Image**

**Build the Docker Image:**

bash

docker build -t sample-node-app .

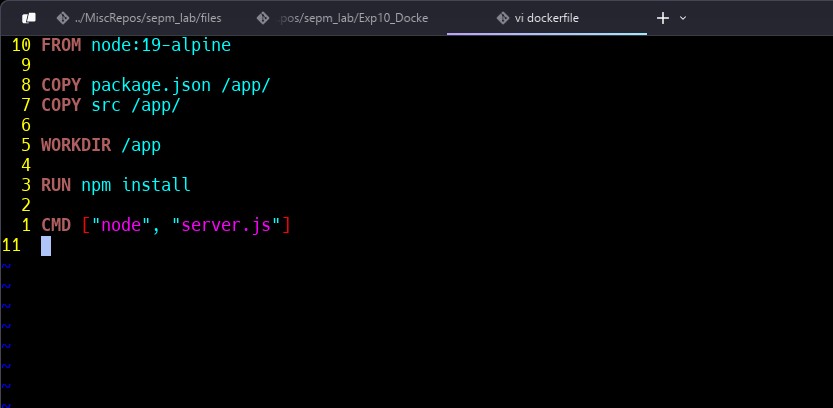
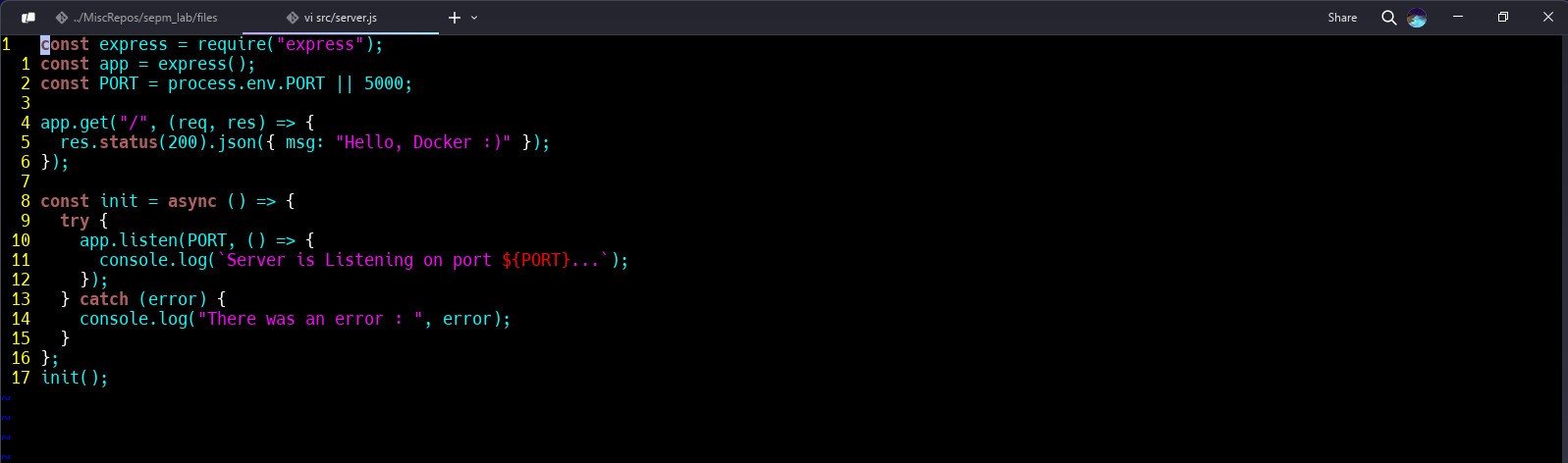
**Run the Docker Container:**

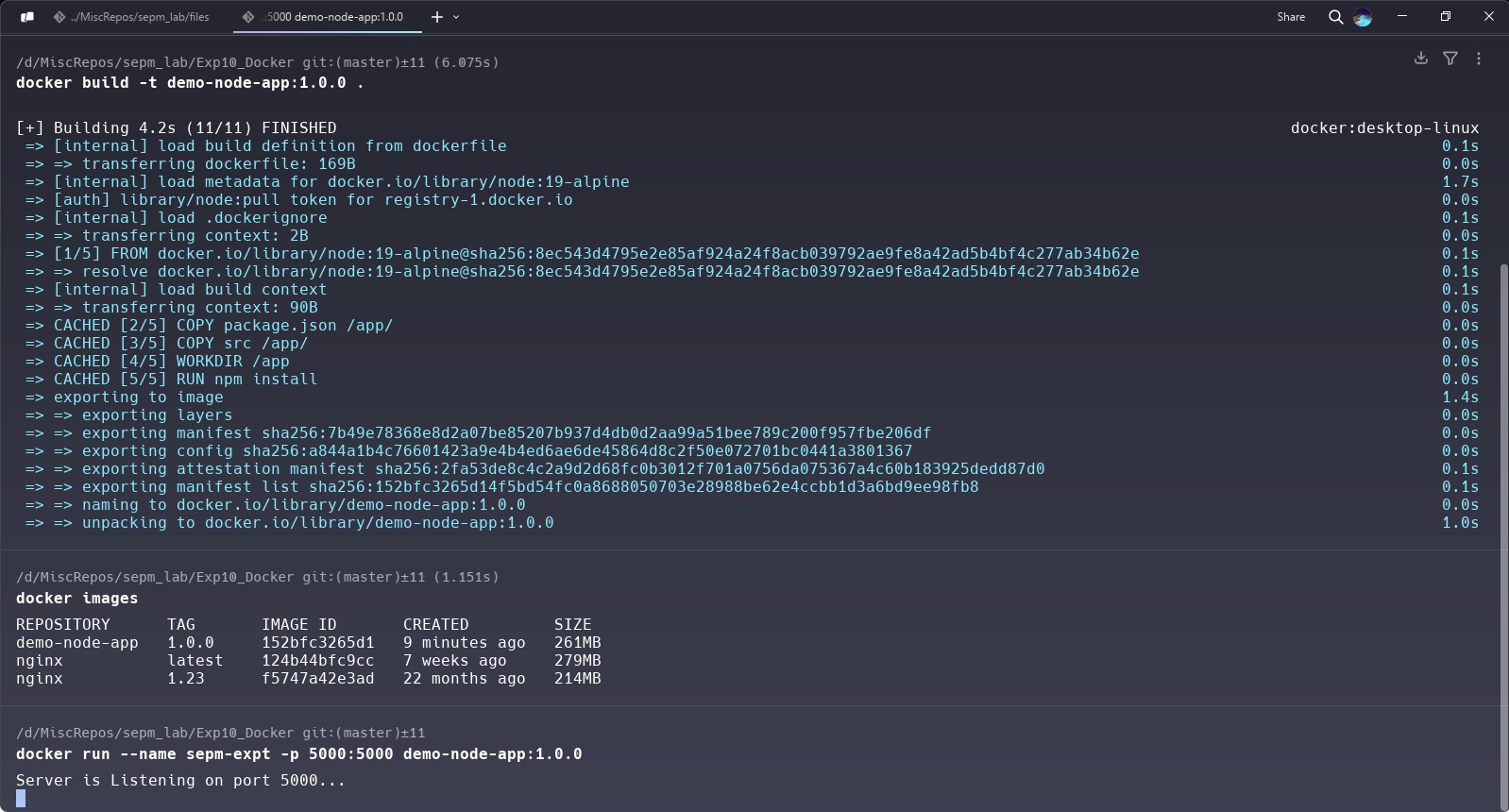
bash

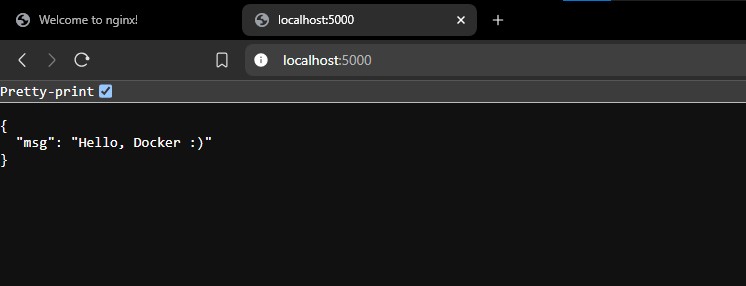
docker run -p 3000:3000 sample-node-app

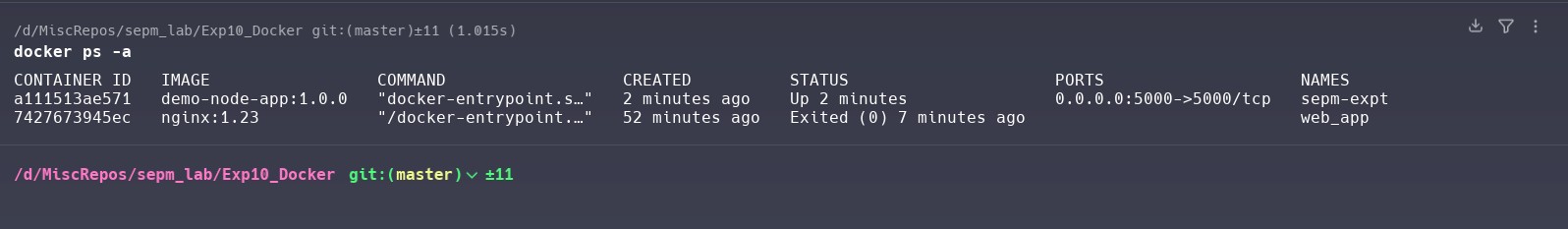
Now, visit http://localhost:3000 and you’ll see **"Hello, Docker!"**

**SCREENSHOTS:**









**CONCLUSION:**

Hence, we have learnt Docker file instructions, build an image for a sample web application using DOCKERFILE.