Name: Jayesh Kumar R

USN: 1RV24MC048

Program 2: Develop a Multi-Stage Dockerfile for Container Orchestration

Project Structure:

Program-2/

```
1rv24mc048_Jayesh@Jayesh:~/Desktop/DEVOPS_Lab/lab2$ ls -1
build
dist
Dockerfile
node_modules
package.json
package-lock.json
src
1rv24mc048_Jayesh@Jayesh:~/Desktop/DEVOPS_Lab/lab2$ ls dist/ -1
1rv24mc048_Jayesh@Jayesh:~/Desktop/DEVOPS_Lab/lab2$ ls src/ -1
index.js
1rv24mc048_Jayesh@Jayesh:~/Desktop/DEVOPS_Lab/lab2$ ls build/ -1
1rv24mc048_Jayesh@Jayesh:~/Desktop/DEVOPS_Lab/lab2$
```

STEP 1: Create the Dockerfile and add the content:

```
1rv24mc048_Jayesh@Jayesh:~/Desktop/DEVOPS_Lab/lab2$ cat Dockerfile
FROM node:20-alpine AS builder

WORKDIR /app

COPY package.json package-lock.json ./
RUN npm install

COPY . .
RUN npm run build

FROM node:20-alpine

WORKDIR /app

COPY --from=builder /app/package.json ./
COPY --from=builder /app/package-lock.json ./
COPY --from=builder /app/dist ./dist
COPY --from=builder /app/node_modules ./node_modules

EXPOSE 3000
CMD ["node", "dist/index.js"]1rv24mc048_Jayesh@Jayesh:~/Desktop/DEVOPS_Lab/lab2$
```

STEP 2: Make a folder→ src and create a file→ index.js and add the following content:

```
1rv24mc048_Jayesh@Jayesh:~/Desktop/DEVOPS_Lab/lab2$ cat src/index.js
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
  res.send('Hello Deepika Maam');;
});

app.listen(port, () => {
  console.log(`Example app listening at http://localhost:${port}`);
});
1rv24mc048_Jayesh@Jayesh:~/Desktop/DEVOPS_Lab/lab2$
```

STEP 3: Create a file→ package.json

Initialize it with:

npm init -y (also creates the file package-lock.json)

Then edit it.

```
1rv24mc048_Jayesh@Jayesh:~/Desktop/DEVOPS_Lab/lab2$ cat package.json
{
  "name": "lab2",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start" : "node dist/index.js",
    "build" : "mkdir -p dist && cp src/* dist/",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^5.1.0"
  }
}
1rv24mc048_Jayesh@Jayesh:~/Desktop/DEVOPS_Lab/lab2$
```

STEP 4: Install node modules with the following command

npm install

STEP 5: Execute the Docker build command to build image

```
Terminal
1rv24mc048_Jayesh@Jayesh:~/Desktop/DEVOPS_Lab/lab2$ dockerscript
Enter the Name of the Image you want to create:
lab2
Image lab2 does not exist. Building it now...
[+] Building 8.3s (15/15) FINISHED                                                                  docker:default
 => [internal] load build definition from Dockerfile                                                          0.1s
 => => transferring dockerfile: 423B                                                                          0.0s
 => [internal] load metadata for docker.io/library/node:20-alpine                                             3.2s
 => [internal] load .dockerignore                                                                             0.0s
 => => transferring context: 2B                                                                               0.0s
 => [internal] load build context                                                                             0.4s
 => => transferring context: 219.84kB                                                                         0.4s
 => [builder 1/6] FROM docker.io/library/node:20-alpine@sha256:6178e78b972f79c335df281f4b7674a2d85071aa        0.0s
 => CACHED [builder 2/6] WORKDIR /app                                                                         0.0s
 => CACHED [builder 3/6] COPY package.json package-lock.json ./                                               0.0s
 => CACHED [builder 4/6] RUN npm install                                                                      0.0s
 => [builder 5/6] COPY . .                                                                                    1.5s
 => [builder 6/6] RUN npm run build                                                                           2.3s
 => CACHED [stage-1 3/6] COPY --from=builder /app/package.json ./                                             0.0s
 => CACHED [stage-1 4/6] COPY --from=builder /app/package-lock.json ./                                        0.0s
 => CACHED [stage-1 5/6] COPY --from=builder /app/dist ./dist                                                 0.0s
 => CACHED [stage-1 6/6] COPY --from=builder /app/node_modules ./node_modules                                 0.0s
 => exporting to image                                                                                        0.0s
 => => exporting layers                                                                                       0.0s
 => => writing image sha256:8c05496aafa469d0eaa7b9004a4b28fed9a875cf19fd6ea470a75397862499c0                  0.0s
 => => naming to docker.io/library/lab2                                                                       0.0s
```

STEP 6: Run the Docker Container specifying the port number

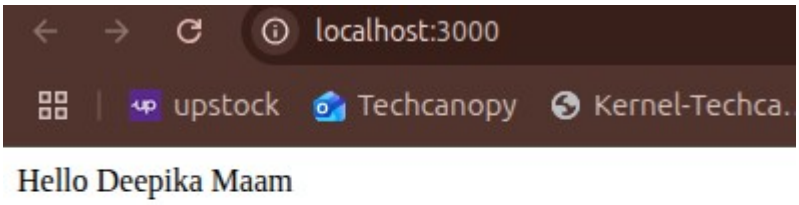
```
Do you want to run the Docker container now? (y/n):
y
Enter the number of flags you want to add to 'docker run':
2
Enter flag #1 (e.g., -p 8080:80 or --name mycontainer):
-it
Enter flag #2 (e.g., -p 8080:80 or --name mycontainer):
-p 3000:3000

Enter the container name (optional, press Enter to skip):
lab2_cont

Running command: sudo docker run  -it -p 3000:3000 --name lab2_cont lab2
Example app listening at http://localhost:3000
```

STEP 7: Test the Container by verifying the localhost details on web browser, the text

Hello Deepika Ma’am



STEP 8: Stop the container with the container number, remove the docker container and delete the Docker Image with the name lab2

docker container stop lab2_cont

docker container rm lab2_cont

docker image rmi lab2

DockerScript – a script to build and run

```
echo "Enter the Name of the Image you want to create: "
read -r image

# Check if image exists
if ! docker images | grep -q "$image"; then
    echo "Image $image does not exist. Building it now..."
    sudo docker build -t "$image" .
else
    echo "Image $image already exists."
    echo "Re-running the script..."
fi

echo
echo "Do you want to run the Docker container now? (y/n): "
read -r run_choice

if [[ "$run_choice" =~ ^[Yy]$ ]]; then
    echo "Enter the number of flags you want to add to 'docker run': "
    read -r num_flags

    flags=""

    for ((i=1; i<=num_flags; i++)); do
        echo "Enter flag #$i (e.g., -p 8080:80 or --name mycontainer): "
        read -r flag
        flags="$flags $flag"
    done

    echo
    echo "Enter the container name (optional, press Enter to skip): "
    read -r container_name

    if [[ -n "$container_name" ]]; then
        run_cmd="sudo docker run $flags --name $container_name $image"
    else
        run_cmd="sudo docker run $flags $image"
    fi

    echo
    echo "Running command: $run_cmd"
    eval "$run_cmd"
else
    echo "Exiting without running container."
fi
```