

Name: Akanksha Singh

USN: 1RV24MC012

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

### Project Structure:

Program-2/

- Dockerfile
- package.json
- package-lock.json
- src/ index.js
- node\_modules

STEP 1: Create the **Dockerfile** and add the content:

```
GNU nano 7.2
FROM node:20-alpine AS builder
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN mkdir -p dist && cp -r src/* dist/
FROM node:20-alpine
WORKDIR /app
COPY --from=builder /app/package*.json ./
COPY --from=builder /app/dist ./dist
COPY --from=builder /app/node_modules ./node_modules
EXPOSE 3000
CMD ["node","dist/index.js"]
```

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

```
GNU nano 7.2
const express=require('express');
const app=express();
const PORT=3000;

app.get('/',(req,res)=>{
res.send('Hello from multi-stage docker container');
});

app.listen(PORT,()=>{
console.log('The app is running on port {PORT}');
});
```

STEP 3: Create a file → **package.json**

Initialize it with:

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

Then edit it.

```
1rv24mc012_akanksha@akankshaPC:~/Program-2$ npm init -y
Wrote to /home/1rv24mc012_akanksha/Program-2/package.json:

{
  "name": "program-2",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

STEP 4: Install node modules with the following command

`npm install`

## STEP 5: Execute the Docker build command to build image

```
1rv24mc012_akanksha@akanshasPC:~$ mkdir Program-2
1rv24mc012_akanksha@akanshasPC:~$ cd Program-2
1rv24mc012_akanksha@akanshasPC:~/Program-2$ nano Dockerfile
1rv24mc012_akanksha@akanshasPC:~/Program-2$ mkdir src
1rv24mc012_akanksha@akanshasPC:~/Program-2$ cd src
1rv24mc012_akanksha@akanshasPC:~/Program-2/src$ nano index.js
1rv24mc012_akanksha@akanshasPC:~/Program-2/src$ cd ..
1rv24mc012_akanksha@akanshasPC:~/Program-2$ npm init -y
Wrote to /home/1rv24mc012_akanksha/Program-2/package.json:

{
  "name": "program-2",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

1rv24mc012_akanksha@akanshasPC:~/Program-2$ nano package.json
1rv24mc012_akanksha@akanshasPC:~/Program-2$ npm install

added 69 packages, and audited 70 packages in 9s

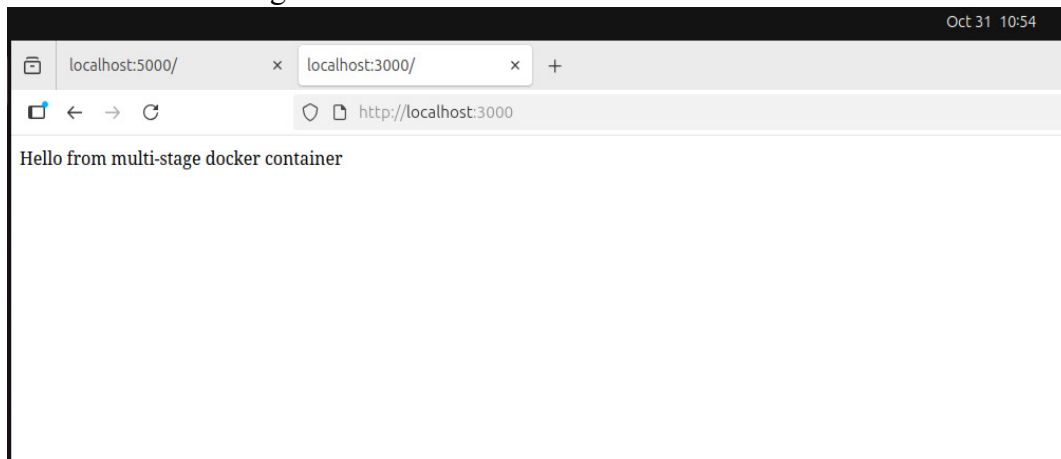
14 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
1rv24mc012_akanksha@akanshasPC:~/Program-2$ sudo docker build -t program-2 .
[sudo] password for 1rv24mc012_akanksha:
[+] Building 12.2s (14/14) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile                 0.0s
=> => transferring dockerfile: 377B                                0.0s
=> [internal] load metadata for docker.io/library/node:20-alpine    5.2s
=> [internal] load .dockerignore                                     0.0s
=> => transferring context: 2B                                       0.0s
```

## STEP 6: Run the Docker Container specifying the port number

```
1rv24mc012_akanksha@akanshasPC:~/Program-2$ sudo docker run -it -p 3000:3000 program-2
The app is running on port {PORT}
1rv24mc012_akanksha@akanshasPC:~/Program-2$ sudo docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
6f267a250be4   program-2   "docker-entrypoint.s..." 26 seconds ago Up 26 seconds  0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp  stupefied_meninsky
b4354514a564   program-1   "flask run"              11 minutes ago Exited (1) 11 minutes ago                flask-container
3a668c9ae3d3   14d3c9d119d6 "docker-entrypoint.s..." 4 hours ago    Exited (1) 4 hours ago                sharp_dhawan
1rv24mc012_akanksha@akanshasPC:~/Program-2$
```

STEP 7: Test the Container by verifying the localhost details on web browser, the text →  
Hello from multi-stage docker container



STEP 8: Stop the container with the container number, remove the docker container and delete the Docker Image with the name program-1

```
docker container stop <Container-number>
```

```
docker container rm <Container-number>
```

```
docker image rm program-1
```