

## Devops lab program – 2

### Develop a Multi-Stage Dockerfile for Container Orchestration

#### Project structure

**program-2**

```
|__node_modules  
|__package.json  
|__package-lock.json  
|__build|  
|__src  
|   |__index.js  
|__dist  
|   |__index.js  
|__Dockerfile
```

#### Step1:create a project folder

**mkdir program2**

**cd program2**



```
1rv24mc074_poornimat@lenovo:/home/1rv24mc074_poornimat/Program.3/program2
```

#### Step2:initialize npm – it will create package.json

**npm init -y**



```
1rv24mc074_poornimat@lenovo:/home/1rv24mc074_poornimat/Program.3/program2
```

```
1rv24mc074_poornimat@lenovo:/home/1rv24mc074_poornimat/Program.3$ mkdir program2  
1rv24mc074_poornimat@lenovo:/home/1rv24mc074_poornimat/Program.3$ cd program2  
1rv24mc074_poornimat@lenovo:/home/1rv24mc074_poornimat/Program.3/program2$ npm init -y  
Wrote to /home/1rv24mc074_poornimat/Program.3/program2/package.json:
```

```
{  
  "name": "program2",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \\\"Error: no test specified\\\" && exit 1"  
  },  
  "keywords": [],  
  "author": "",  
  "license": "ISC"  
}
```

```
1rv24mc074_poornimat@lenovo:/home/1rv24mc074_poornimat/Program.3/program2$ ls  
package.json  
1rv24mc074_poornimat@lenovo:/home/1rv24mc074_poornimat/Program.3/program2$ 
```

### **Step3: create a src folder and create index.js file inside this folder**

**mkdir src**

**cd src**

**nano index.js**



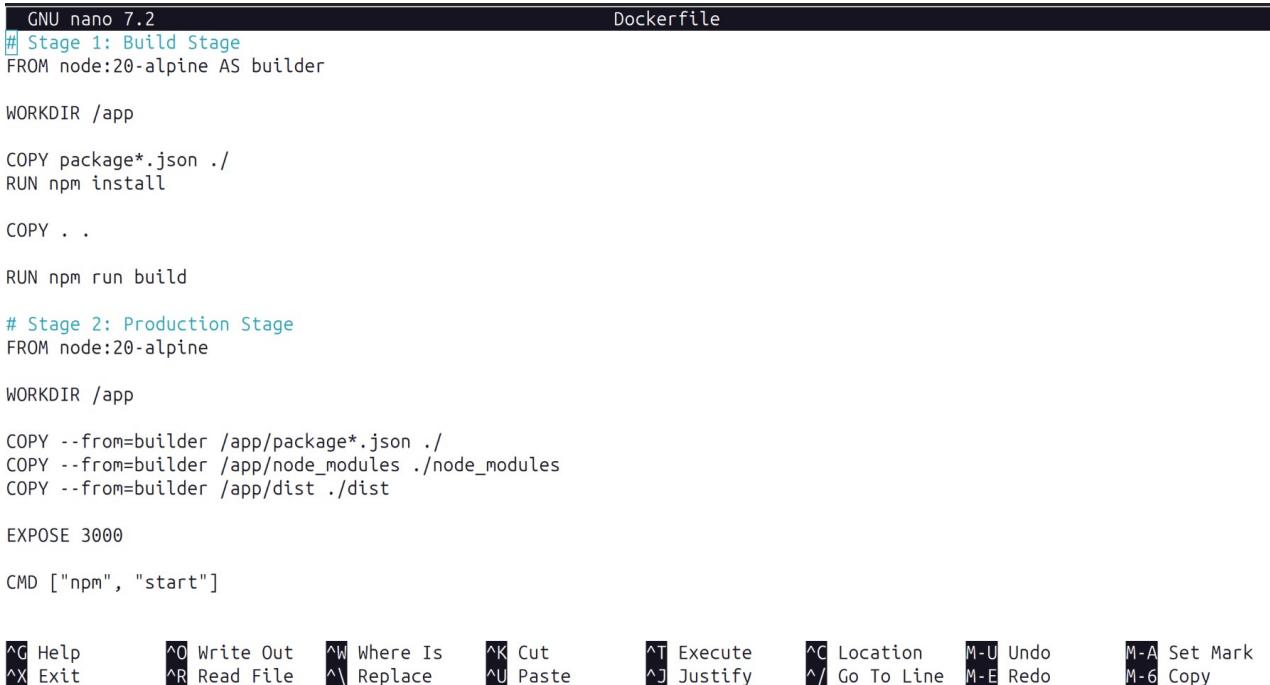
```
GNU nano 7.2                                         index.js
const express = require('express');
const app = express();
const PORT = 3000;

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

app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});
```

### **Step4:create a docker file**

**nano Dockerfile**



```
GNU nano 7.2                                         Dockerfile
# Stage 1: Build Stage
FROM node:20-alpine AS builder

WORKDIR /app

COPY package*.json ./
RUN npm install

COPY .

RUN npm run build

# Stage 2: Production Stage
FROM node:20-alpine

WORKDIR /app

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

EXPOSE 3000

CMD ["npm", "start"]
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo  
 ^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo  
 M-A Set Mark M-6 Copy

### **Step5: build docker image**

**sudo docker build -t p .**

```

1rv24mc074_poornimat@lenovo:/home/1rv24mc074_poornimat/Program_2$ sudo docker build -t p .
[sudo] password for 1rv24mc074_poornimat:
[+] Building 6.6s (14/14) FINISHED
  => [internal] load build definition from Dockerfile
  => => transferring dockerfile: 406B
  => [internal] load metadata for docker.io/library/node:20-alpine
  => [internal] load .dockerignore
  => => transferring context: 2B
  => [internal] load build context
  => => transferring context: 2.70MB
  => [builder 1/6] FROM docker.io/library/node:20-alpine@sha256:6178e78b972f79c335df281f4b7674a2d85071aae2af020ffa
  => CACHED [builder 2/6] WORKDIR /app
  => [builder 3/6] COPY package*.json .
  => [builder 4/6] RUN npm install
  => [builder 5/6] COPY . .
  => [builder 6/6] RUN npm run build
  => [stage-1 3/5] COPY --from=builder /app/package*.json .
  => [stage-1 4/5] COPY --from=builder /app/node_modules ./node_modules
  => [stage-1 5/5] COPY --from=builder /app/dist ./dist
  => exporting to image
  => => exporting layers
  => => writing image sha256:36e5a307e3b4edeeaa20fae0275390c4a261c1169eb3c70cc4bee1e8744e8f392
  => => naming to docker.io/library/p
1rv24mc074_poornimat@lenovo:/home/1rv24mc074_poornimat/Program_2$ nano Dockerfile
1rv24mc074_poornimat@lenovo:/home/1rv24mc074_poornimat/Program_2$ 

```

## Step6:run the container

**sudo docker run -p 3000:3000 p**

```

1rv24mc074_poornimat@lenovo:~/Program_2/src$ sudo docker run -p 3000:3000 p
> program_201.0.0 start
> node dist/index.js
Server running on port 3000

```

## Step7: verify the application open the browser and go to <http://localhost:3000>

