PROGRAM-2

TITLE: Develop a multi-stage Dockerfile for container orchestration

<u>Structure of the Program:</u>

Program-2:

Dockerfile

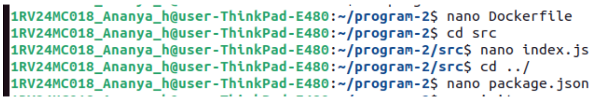
Src/index.js

package.json

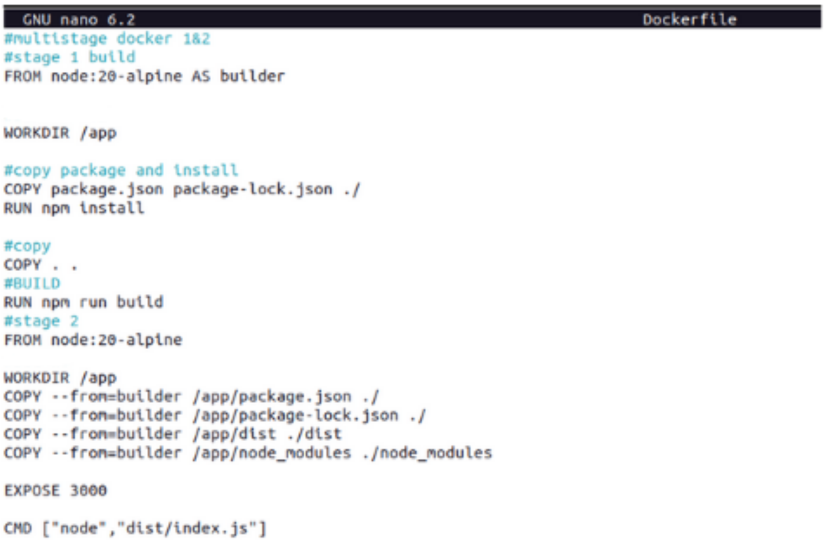
package-lock.json

node_modules

Step-1: Create a Dockerfile

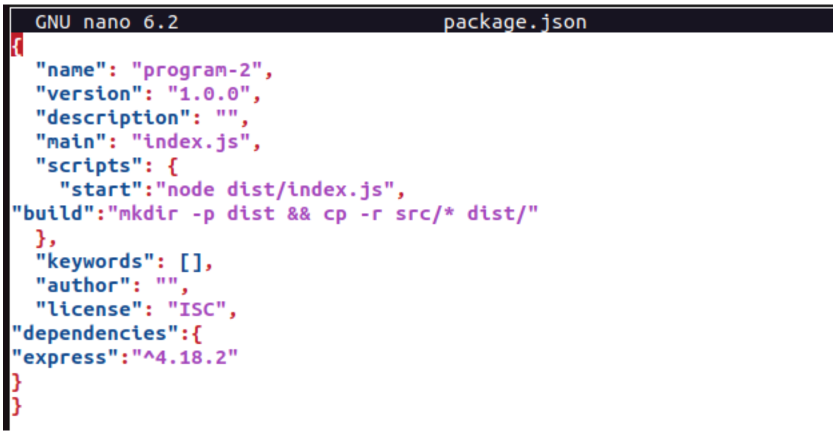


```
1RV24MC018_Ananya_h@user-ThinkPad-E480:~/program-2$ nano Dockerfile
1RV24MC018_Ananya_h@user-ThinkPad-E480:~/program-2$ cd src
1RV24MC018_Ananya_h@user-ThinkPad-E480:~/program-2/src$ nano index.js
1RV24MC018_Ananya_h@user-ThinkPad-E480:~/program-2/src$ cd ../
1RV24MC018_Ananya_h@user-ThinkPad-E480:~/program-2$ nano package.json
```



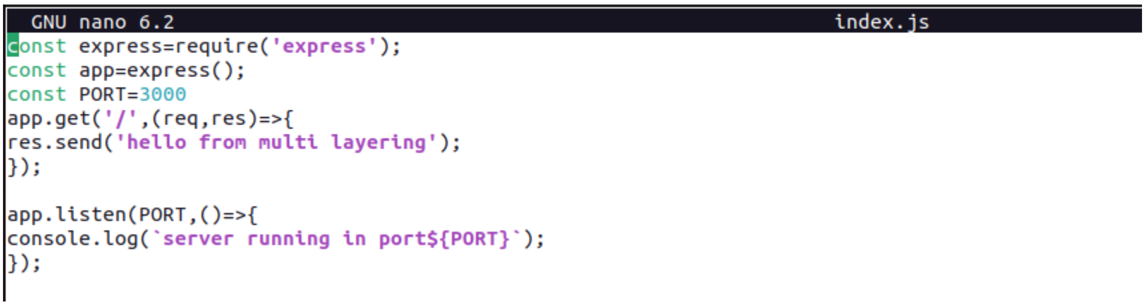
```
  GNU nano 6.2                                                   Dockerfile
#multistage docker 1&2
#stage 1 build
FROM node:20-alpine AS builder


WORKDIR /app

#copy package and install
COPY package.json package-lock.json ./
RUN npm install

#copy
COPY . .
#BUILD
RUN npm run build
#stage 2
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"]
```

Step-2: You can write package.json or create it using npm init-y.



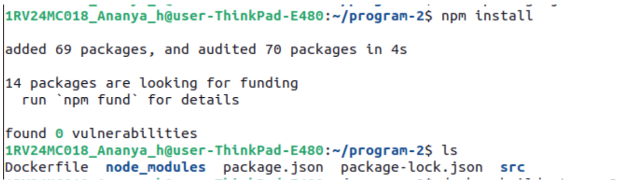
```
  GNU nano 6.2                              package.json
{
  "name": "program-2",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start":"node dist/index.js",
"build":"mkdir -p dist && cp -r src/* dist/"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
"dependencies":{
"express":"^4.18.2"
}
}
```

Step-3: Create a src folder and write index.js file inside it



```
  GNU nano 6.2                                                       index.js
const express=require('express');
const app=express();
const PORT=3000
app.get('/',(req,res)=>{
res.send('hello from multi layering');
});

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

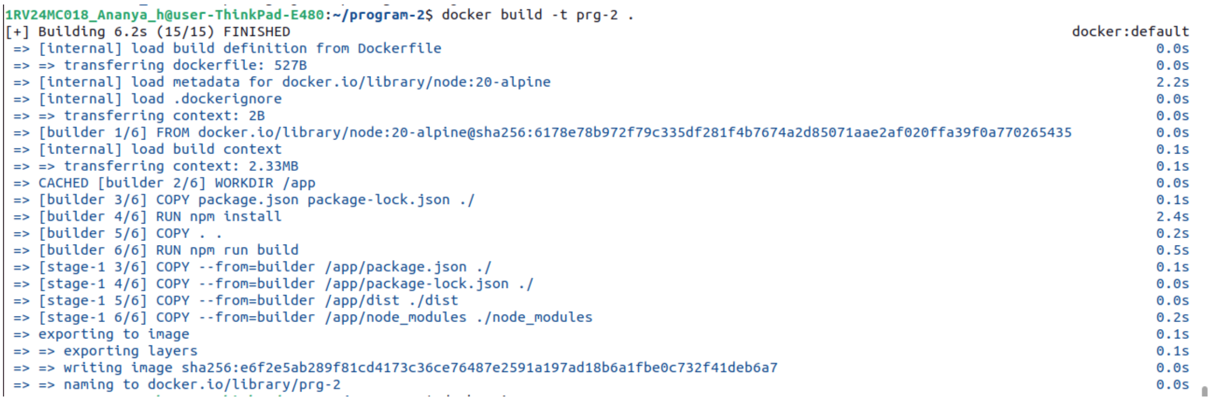
Step-4: install node dependencies



```
1RV24MC018_Ananya_h@user-ThinkPad-E480:~/program-2$ npm install

added 69 packages, and audited 70 packages in 4s

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

found 0 vulnerabilities
1RV24MC018_Ananya_h@user-ThinkPad-E480:~/program-2$ ls
Dockerfile  node_modules  package.json  package-lock.json  src
```

Step-5: Create a docker image

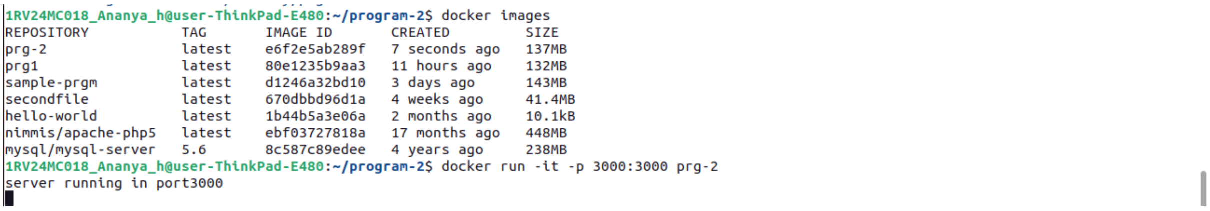
- ➢ Sudo docker build -t prg-2 .



```
1RV24MC018_Ananya_h@user-ThinkPad-E480:~/program-2$ docker build -t prg-2 .
[+] Building 6.2s (15/15) FINISHED                                                                           docker:default
 => [internal] load build definition from Dockerfile                                                                   0.0s
 => => transferring dockerfile: 527B                                                                                   0.0s
 => [internal] load metadata for docker.io/library/node:20-alpine                                                      2.2s
 => [internal] load .dockerignore                                                                                      0.0s
 => => transferring context: 2B                                                                                        0.0s
 => [builder 1/6] FROM docker.io/library/node:20-alpine@sha256:6178e78b972f79c335df281f4b7674a2d85071aae2af020ffa39f0a770265435  0.0s
 => [internal] load build context                                                                                      0.1s
 => => transferring context: 2.33MB                                                                                    0.1s
 => CACHED [builder 2/6] WORKDIR /app                                                                                  0.0s
 => [builder 3/6] COPY package.json package-lock.json ./                                                               0.1s
 => [builder 4/6] RUN npm install                                                                                      2.4s
 => [builder 5/6] COPY . .                                                                                             0.2s
 => [builder 6/6] RUN npm run build                                                                                    0.5s
 => [stage-1 3/6] COPY --from=builder /app/package.json ./                                                             0.1s
 => [stage-1 4/6] COPY --from=builder /app/package-lock.json ./                                                        0.0s
 => [stage-1 5/6] COPY --from=builder /app/dist ./dist                                                                 0.0s
 => [stage-1 6/6] COPY --from=builder /app/node_modules ./node_modules                                                 0.2s
 => exporting to image                                                                                                 0.1s
 => => exporting layers                                                                                                0.1s
 => => writing image sha256:e6f2e5ab289f81cd4173c36ce76487e2591a197ad18b6a1fbe0c732f41deb6a7                           0.0s
 => => naming to docker.io/library/prg-2                                                                               0.0s
```

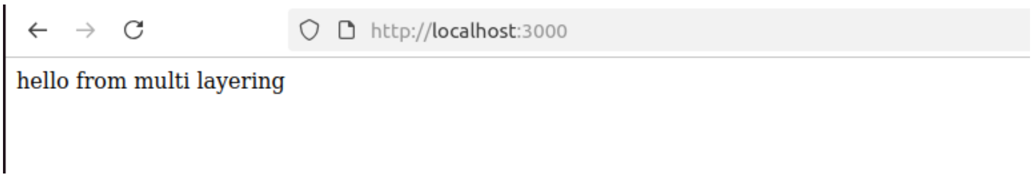
Step-6: Run the container

- Sudo docker run -it -p 3000:3000 prg-2



```
1RV24MC018_Ananya_h@user-ThinkPad-E480:~/program-2$ docker images
REPOSITORY           TAG       IMAGE ID       CREATED          SIZE
prg-2                latest    e6f2e5ab289f   7 seconds ago    137MB
prg1                 latest    80e1235b9aa3   11 hours ago     132MB
sample-prgm          latest    d1246a32bd10   3 days ago       143MB
secondfile           latest    670dbbd96d1a   4 weeks ago      41.4MB
hello-world          latest    1b44b5a3e06a   2 months ago     10.1kB
nimmis/apache-php5   latest    ebf03727818a   17 months ago    448MB
mysql/mysql-server   5.6       8c587c89edee   4 years ago      238MB
1RV24MC018_Ananya_h@user-ThinkPad-E480:~/program-2$ docker run -it -p 3000:3000 prg-2
server running in port3000
```

Step-7: go to browser and type http://localhost:3000



hello from multi layering