2. Develop a Multi-stage Dockerfile for container Orchestration

Directory Structure:

```
P2
|---- Dockerfile
|---- package.json
|---- / src/ index.js
|---- / dist / index.js
|---- node_modules
|---- / build
```

Step-1 : Create a Dockerfile using nano in the pwd

→ Dockerfile

```
// Stage-1
FROM node:20-alpine AS builder
WORKDIR /app
COPY package.json package-lock.json ./
RUN npm install
COPY . .
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 5000
CMD ["node", "dist/index.js"]
```

Step-2: Create a express script inside src folder

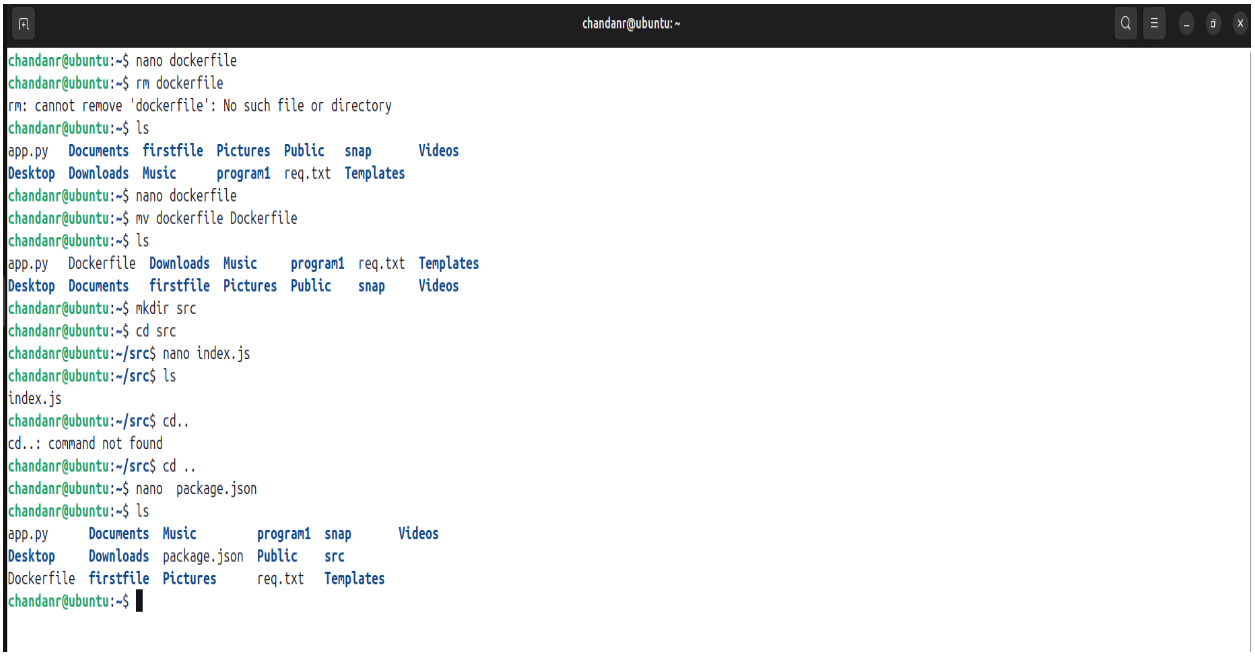
→ / src / index.js

```
const express= require('express');
const app = express();
const PORT = 5000;

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

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

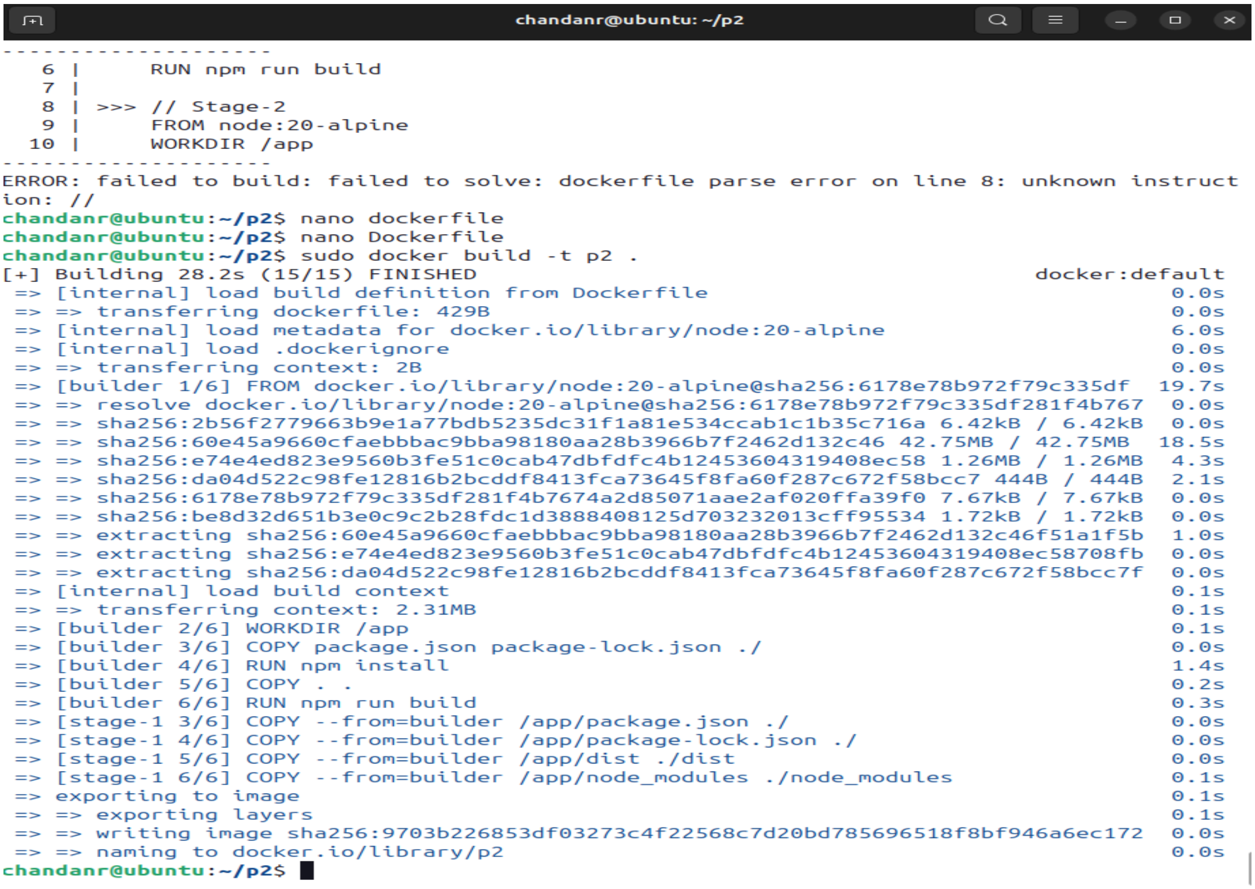
Step-3: Create a custom package.json file or you can also run npm init -y it'll generate package.json for you and you can modify the content according to you.

→**package.json**

```
{
  "name": "p2",
  "version": "1.0.0",
  "description": "A Node js app with multi-staged Dockerfile",
  "main": "dist/index.js",
  "scripts": {
    "start": "node dist/index.js",
    "build": "mkdir -p dist && cp -r src/* dist/"
},
  "keywords": [],
  "author": "Niranjan",
  "license": "No-Licence-yet",
  "dependencies":{
  "express":"^4.18.2"
  }
}
```



```
chandanr@ubuntu:~$ nano dockerfile
chandanr@ubuntu:~$ rm dockerfile
rm: cannot remove 'dockerfile': No such file or directory
chandanr@ubuntu:~$ ls
app.py   Documents  firstfile  Pictures  Public  snap      Videos
Desktop  Downloads  Music      program1  req.txt  Templates
chandanr@ubuntu:~$ nano dockerfile
chandanr@ubuntu:~$ mv dockerfile Dockerfile
chandanr@ubuntu:~$ ls
app.py   Dockerfile  Downloads  Music     program1  req.txt  Templates
Desktop  Documents   firstfile  Pictures  Public    snap     Videos
chandanr@ubuntu:~$ mkdir src
chandanr@ubuntu:~$ cd src
chandanr@ubuntu:~/src$ nano index.js
chandanr@ubuntu:~/src$ ls
index.js
chandanr@ubuntu:~/src$ cd..
cd..: command not found
chandanr@ubuntu:~/src$ cd ..
chandanr@ubuntu:~$ nano  package.json
chandanr@ubuntu:~$ ls
app.py      Documents  Music         program1  snap      Videos
Desktop     Downloads  package.json  Public    src
Dockerfile  firstfile  Pictures      req.txt   Templates
chandanr@ubuntu:~$ 
```



```
--------------------
   6 |      RUN npm run build
   7 |
   8 | >>> // Stage-2
   9 |     FROM node:20-alpine
  10 |     WORKDIR /app
--------------------
ERROR: failed to build: failed to solve: dockerfile parse error on line 8: unknown instruct
ion: //
chandanr@ubuntu:~/p2$ nano dockerfile
chandanr@ubuntu:~/p2$ nano Dockerfile
chandanr@ubuntu:~/p2$ sudo docker build -t p2 .
[+] Building 28.2s (15/15) FINISHED                                         docker:default
 => [internal] load build definition from Dockerfile                                  0.0s
 => => transferring dockerfile: 429B                                                  0.0s
 => [internal] load metadata for docker.io/library/node:20-alpine                     6.0s
 => [internal] load .dockerignore                                                     0.0s
 => => transferring context: 2B                                                       0.0s
 => [builder 1/6] FROM docker.io/library/node:20-alpine@sha256:6178e78b972f79c335df  19.7s
 => => resolve docker.io/library/node:20-alpine@sha256:6178e78b972f79c335df281f4b767  0.0s
 => => sha256:2b56f2779663b9e1a77bdb5235dc31f1a81e534ccab1c1b35c716a 6.42kB / 6.42kB  0.0s
 => => sha256:60e45a9660cfaebbbac9bba98180aa28b3966b7f2462d132c46 42.75MB / 42.75MB  18.5s
 => => sha256:e74e4ed823e9560b3fe51c0cab47dbfdfc4b12453604319408ec58 1.26MB / 1.26MB  4.3s
 => => sha256:da04d522c98fe12816b2bcddf8413fca73645f8fa60f287c672f58bcc7 444B / 444B  2.1s
 => => sha256:6178e78b972f79c335df281f4b7674a2d85071aae2af020ffa39f0 7.67kB / 7.67kB  0.0s
 => => sha256:be8d32d651b3e0c9c2b28fdc1d3888408125d703232013cff95534 1.72kB / 1.72kB  0.0s
 => => extracting sha256:60e45a9660cfaebbbac9bba98180aa28b3966b7f2462d132c46f51a1f5b  1.0s
 => => extracting sha256:e74e4ed823e9560b3fe51c0cab47dbfdfc4b12453604319408ec58708fb  0.0s
 => => extracting sha256:da04d522c98fe12816b2bcddf8413fca73645f8fa60f287c672f58bcc7f  0.0s
 => [internal] load build context                                                     0.1s
 => => transferring context: 2.31MB                                                   0.1s
 => [builder 2/6] WORKDIR /app                                                        0.1s
 => [builder 3/6] COPY package.json package-lock.json ./                              0.0s
 => [builder 4/6] RUN npm install                                                     1.4s
 => [builder 5/6] COPY . .                                                            0.2s
 => [builder 6/6] RUN npm run build                                                   0.3s
 => [stage-1 3/6] COPY --from=builder /app/package.json ./                            0.0s
 => [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                0.1s
 => exporting to image                                                                0.1s
 => => exporting layers                                                               0.1s
 => => writing image sha256:9703b226853df03273c4f22568c7d20bd785696518f8bf946a6ec172  0.0s
 => => naming to docker.io/library/p2                                                 0.0s
chandanr@ubuntu:~/p2$ 
```

Step-4 : install Node modules using below command

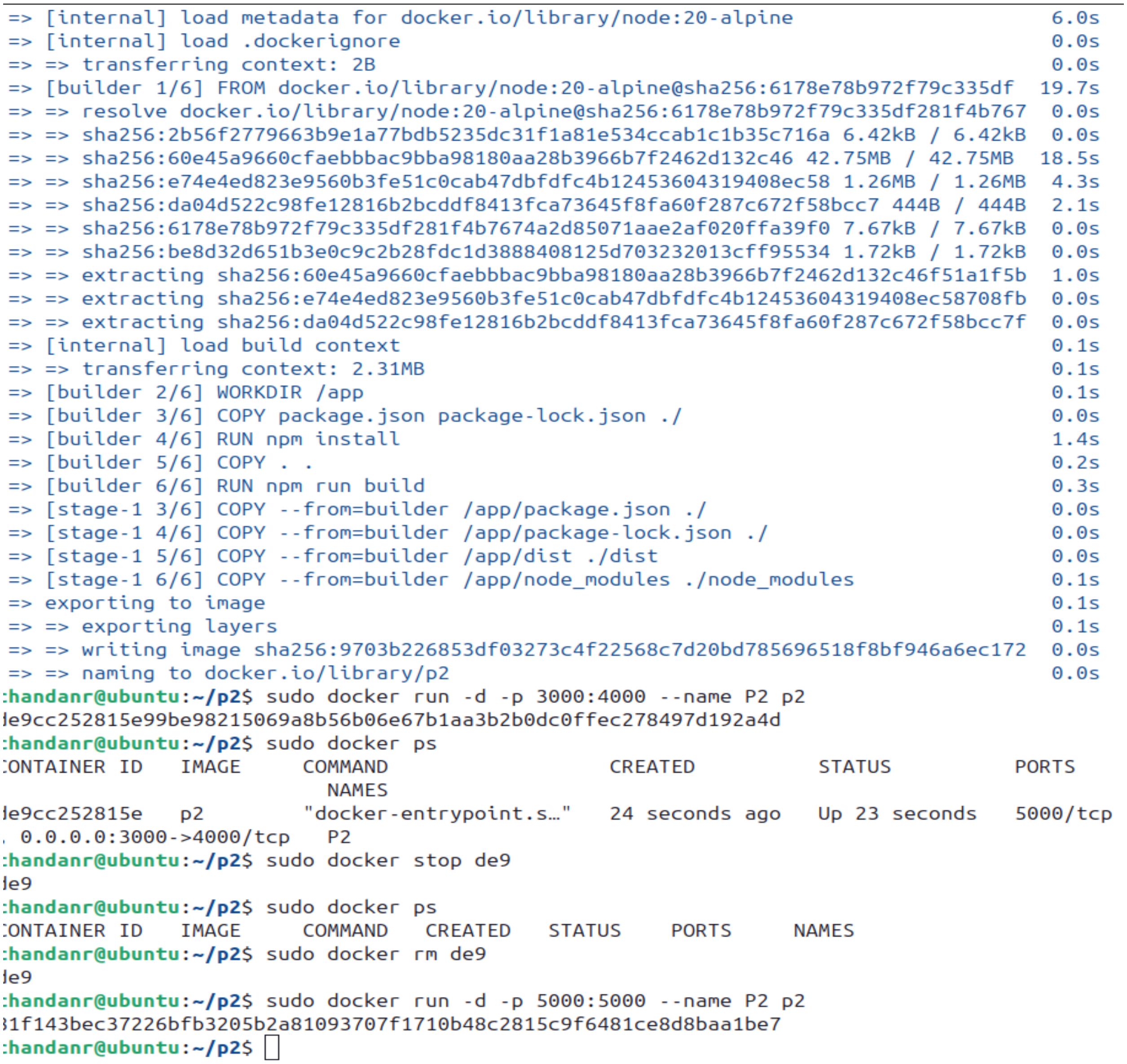
npm install

Step-5 : Build a Docker image using

Sudo docker build -t program2 .

Step-6 : Run the image to build a container

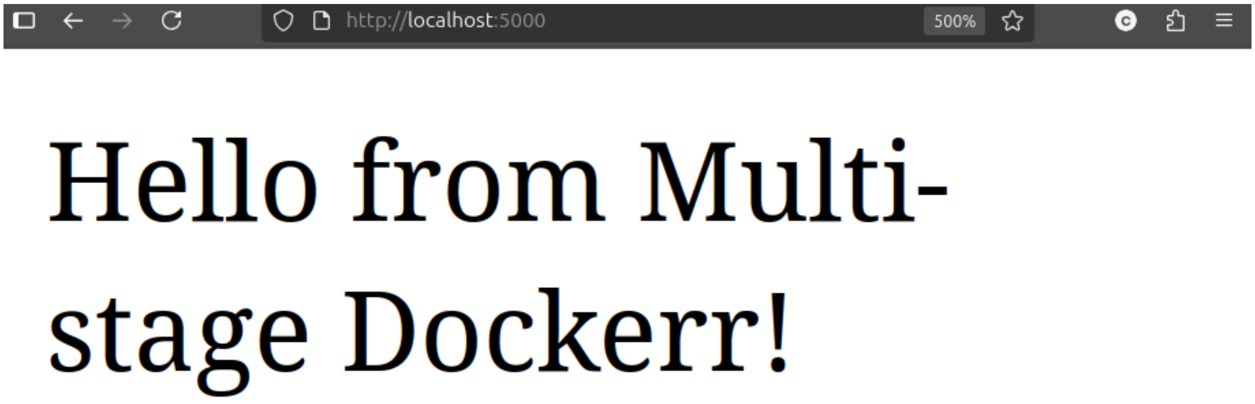
sudo docker run -t -p 3000:3000 program2



```
=> [internal] load metadata for docker.io/library/node:20-alpine                 6.0s
=> [internal] load .dockerignore                                                 0.0s
=> => transferring context: 2B                                                   0.0s
=> [builder 1/6] FROM docker.io/library/node:20-alpine@sha256:6178e78b972f79c335df  19.7s
=> => resolve docker.io/library/node:20-alpine@sha256:6178e78b972f79c335df281f4b767  0.0s
=> => sha256:2b56f2779663b9e1a77bdb5235dc31f1a81e534ccab1c1b35c716a 6.42kB / 6.42kB  0.0s
=> => sha256:60e45a9660cfaebbbac9bba98180aa28b3966b7f2462d132c46 42.75MB / 42.75MB  18.5s
=> => sha256:e74e4ed823e9560b3fe51c0cab47dbfdfc4b12453604319408ec58 1.26MB / 1.26MB  4.3s
=> => sha256:da04d522c98fe12816b2bcddf8413fca73645f8fa60f287c672f58bcc7 444B / 444B  2.1s
=> => sha256:6178e78b972f79c335df281f4b7674a2d85071aae2af020ffa39f0 7.67kB / 7.67kB  0.0s
=> => sha256:be8d32d651b3e0c9c2b28fdc1d3888408125d703232013cff95534 1.72kB / 1.72kB  0.0s
=> => extracting sha256:60e45a9660cfaebbbac9bba98180aa28b3966b7f2462d132c46f51a1f5b  1.0s
=> => extracting sha256:e74e4ed823e9560b3fe51c0cab47dbfdfc4b12453604319408ec58708fb  0.0s
=> => extracting sha256:da04d522c98fe12816b2bcddf8413fca73645f8fa60f287c672f58bcc7f  0.0s
=> [internal] load build context                                                 0.1s
=> => transferring context: 2.31MB                                               0.1s
=> [builder 2/6] WORKDIR /app                                                    0.1s
=> [builder 3/6] COPY package.json package-lock.json ./                          0.0s
=> [builder 4/6] RUN npm install                                                 1.4s
=> [builder 5/6] COPY . .                                                        0.2s
=> [builder 6/6] RUN npm run build                                               0.3s
=> [stage-1 3/6] COPY --from=builder /app/package.json ./                        0.0s
=> [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            0.1s
=> exporting to image                                                            0.1s
=> => exporting layers                                                           0.1s
=> => writing image sha256:9703b226853df03273c4f22568c7d20bd785696518f8bf946a6ec172  0.0s
=> => naming to docker.io/library/p2                                             0.0s
:handanr@ubuntu:~/p2$ sudo docker run -d -p 3000:4000 --name P2 p2
le9cc252815e99be98215069a8b56b06e67b1aa3b2b0dc0ffec278497d192a4d
:handanr@ubuntu:~/p2$ sudo docker ps
:ONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS
                           NAMES
le9cc252815e   p2        "docker-entrypoint.s…"   24 seconds ago   Up 23 seconds   5000/tcp
, 0.0.0.0:3000->4000/tcp   P2
:handanr@ubuntu:~/p2$ sudo docker stop de9
le9
:handanr@ubuntu:~/p2$ sudo docker ps
:ONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
:handanr@ubuntu:~/p2$ sudo docker rm de9
le9
:handanr@ubuntu:~/p2$ sudo docker run -d -p 5000:5000 --name P2 p2
31f143bec37226bfb3205b2a81093707f1710b48c2815c9f6481ce8d8baa1be7
:handanr@ubuntu:~/p2$ 
```

Step-7 : check the status of the container using

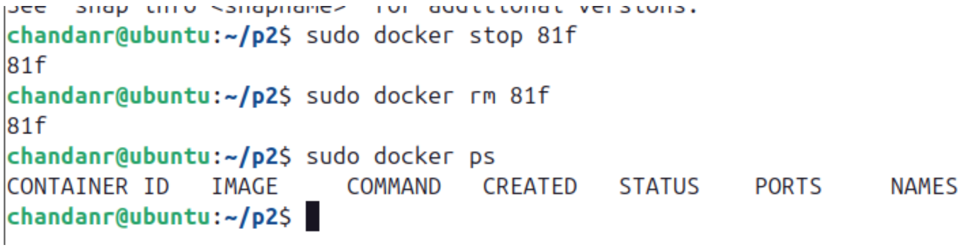
Sudo docker ps
http://localhost:5000 on any browser



Hello from Multi-stage Dockerr!

Step-8 : Stop the container , remove the container and delete the image

sudo docker container ls -a or docker ps -a
sudo docker container stop <container-id>
sudo docker container rm <container–id>
sudo docker image rm <image-id> or docker rmi <image-id>



```
See  snap info <snapname>  for additional versions.
chandanr@ubuntu:~/p2$ sudo docker stop 81f
81f
chandanr@ubuntu:~/p2$ sudo docker rm 81f
81f
chandanr@ubuntu:~/p2$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
chandanr@ubuntu:~/p2$ 
```