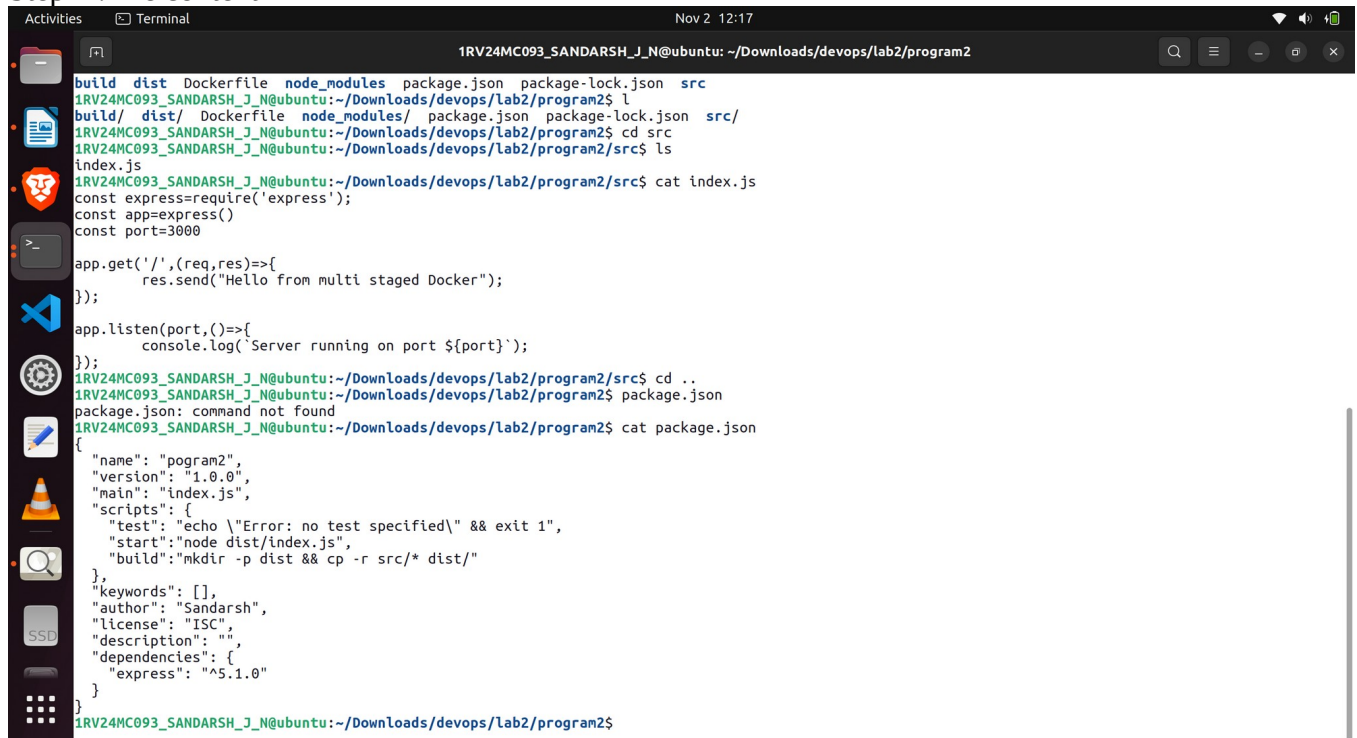


Develop a multi stage docker file fro container orchastration

step 1 : file structures

```
program2
-----build
-----src
      --index.js
-----dist
-----node_modules
-----package.json
-----package-lock.json
-----Dockerfile
```

Step 2 : file content

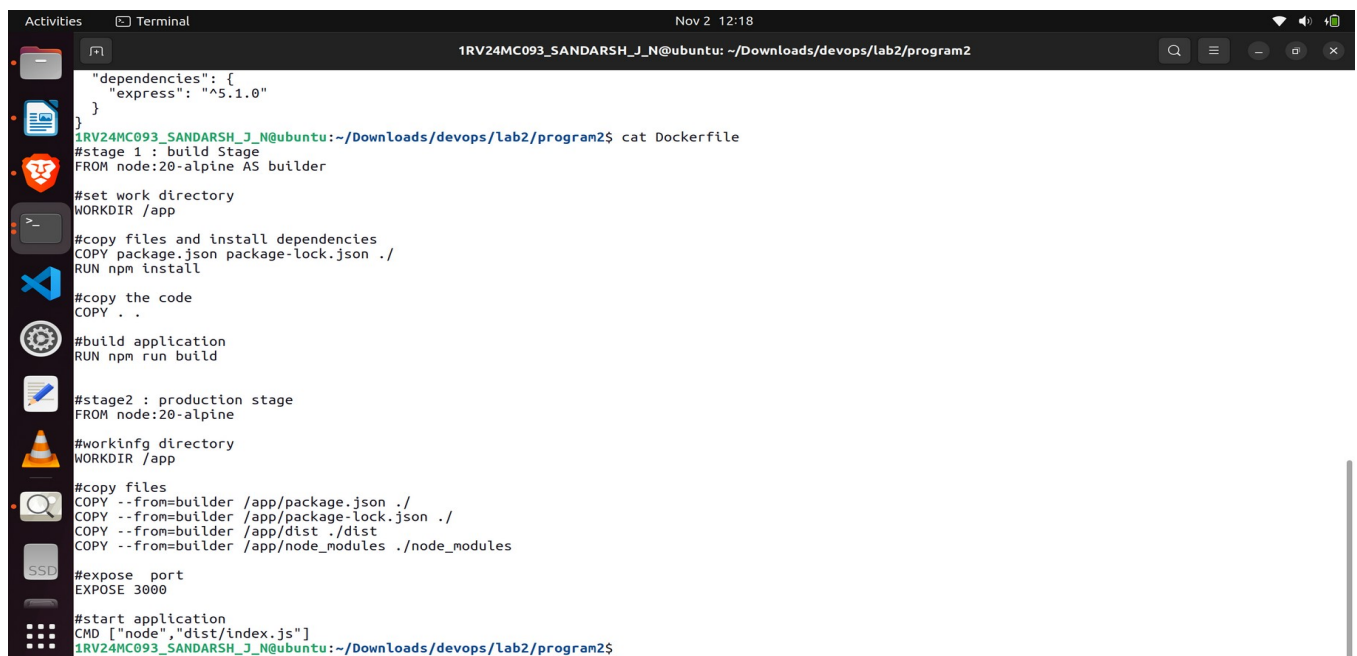


A terminal window titled "1RV24MC093_SANDARSH_J_N@ubuntu: ~/Downloads/devops/lab2/program2" showing the creation of files and the content of package.json. The user runs 'ls' to see build, dist, Dockerfile, node_modules, package.json, package-lock.json, and src. Then they run 'cd src' and 'ls' to see index.js. They view index.js which shows an Express.js server setup on port 3000. Then they run 'cd ..' and 'cat package.json' to show the package.json content.

```
1RV24MC093_SANDARSH_J_N@ubuntu:~/Downloads/devops/lab2/program2$ ls
build  dist  Dockerfile  node_modules  package.json  package-lock.json  src
1RV24MC093_SANDARSH_J_N@ubuntu:~/Downloads/devops/lab2/program2$ cd src
1RV24MC093_SANDARSH_J_N@ubuntu:~/Downloads/devops/lab2/program2/src$ ls
index.js
1RV24MC093_SANDARSH_J_N@ubuntu:~/Downloads/devops/lab2/program2/src$ cat index.js
const express=require('express');
const app=express()
const port=3000

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

app.listen(port,()=>{
    console.log(`Server running on port ${port}`);
});
1RV24MC093_SANDARSH_J_N@ubuntu:~/Downloads/devops/lab2/program2/src$ cd ..
1RV24MC093_SANDARSH_J_N@ubuntu:~/Downloads/devops/lab2/program2$ cat package.json
package.json: command not found
1RV24MC093_SANDARSH_J_N@ubuntu:~/Downloads/devops/lab2/program2$ cat package.json
{
  "name": "pogram2",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1",
    "start": "node dist/index.js",
    "build": "mkdir -p dist && cp -r src/* dist/"
  },
  "keywords": [],
  "author": "Sandarsh",
  "license": "ISC",
  "description": "",
  "dependencies": {
    "express": "^5.1.0"
  }
}
```



A terminal window titled "1RV24MC093_SANDARSH_J_N@ubuntu: ~/Downloads/devops/lab2/program2" showing the content of the Dockerfile. The user runs 'cat Dockerfile' to display the multi-stage build instructions.

```
1RV24MC093_SANDARSH_J_N@ubuntu:~/Downloads/devops/lab2/program2$ cat Dockerfile
#stage 1 : build Stage
FROM node:20-alpine AS builder

#set work directory
WORKDIR /app

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

#copy the code
COPY . .

#build application
RUN npm run build

#stage2 : production stage
FROM node:20-alpine

#workinfg directory
WORKDIR /app

#copy files
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 port
EXPOSE 3000

#start application
CMD ["node","dist/index.js"]
1RV24MC093_SANDARSH_J_N@ubuntu:~/Downloads/devops/lab2/program2$
```

step 2 : build a docker image for program2

```
1RV24MC093_SANDARSH_J_N@ubuntu: ~/Downloads/devops/lab2/program2
1RV24MC093_SANDARSH_J_N@ubuntu:~/Downloads/devops/lab2/program2$ ls
build dist Dockerfile node_modules package.json package-lock.json src
1RV24MC093_SANDARSH_J_N@ubuntu:~/Downloads/devops/lab2/program2$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
1RV24MC093_SANDARSH_J_N@ubuntu:~/Downloads/devops/lab2/program2$ docker build -t prg2 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 3.251MB
Step 1/14 : FROM node:20-alpine AS builder
20-alpine: Pulling from library/node
2d35ebdb57d9: Pulling fs layer
60e45a9660cf: Pulling fs layer
e74e4ed823e9: Pulling fs layer
da04d522c98f: Pulling fs layer
da04d522c98f: Waiting
e74e4ed823e9: Verifying Checksum
e74e4ed823e9: Download complete
2d35ebdb57d9: Download complete
2d35ebdb57d9: Pull complete
da04d522c98f: Verifying Checksum
da04d522c98f: Download complete
60e45a9660cf: Verifying Checksum
60e45a9660cf: Download complete
60e45a9660cf: Pull complete
e74e4ed823e9: Pull complete
da04d522c98f: Pull complete
Digest: sha256:6178e78b972f79c335df281f4b7674a2d85071aae2af020ffa39f0a770265435
Status: Downloaded newer image for node:20-alpine
--> 2b56f2779663
Step 2/14 : WORKDIR /app
--> Running in c14237d65295
--> Removed intermediate container c14237d65295
```

step 3 : create a running container for program 2

```
1RV24MC093_SANDARSH_J_N@ubuntu: ~/Downloads/devops/lab2/program2
--> Running in 33eee7a4a5d0
> pogram2@1.0.0 build
> mkdir -p dist && cp -r src/* dist/
--> Removed intermediate container 33eee7a4a5d0
--> a5797ec7286b
Step 7/14 : FROM node:20-alpine
--> 2b56f2779663
Step 8/14 : WORKDIR /app
--> Using cache
--> ef7c77a8977d
Step 9/14 : COPY --from=builder /app/package.json ./
--> f9a3df6993a3
Step 10/14 : COPY --from=builder /app/package-lock.json ./
--> 020bdf4bbcd
Step 11/14 : COPY --from=builder /app/dist ./dist
--> aca8bedf48b0
Step 12/14 : COPY --from=builder /app/node_modules ./node_modules
--> 0a82494ebab9
Step 13/14 : EXPOSE 3000
--> Running in 18ddc9cab69
--> Removed intermediate container 18ddc9cab69
--> f76914491d17
Step 14/14 : CMD ["node","dist/index.js"]
--> Running in 2b4f87480e33
--> Removed intermediate container 2b4f87480e33
--> edff37cb7fdb
Successfully built edff37cb7fdb
Successfully tagged prg2:latest
1RV24MC093_SANDARSH_J_N@ubuntu:~/Downloads/devops/lab2/program2$ docker run -it -p 3000:3000 --name cont2 prg2
Server running on port 3000
```

output :

