**Lab Program 1**

**Title: Build a docker container from a custom docker file**

**Step1:**

Create a directory and navigate inside it

```
BitterSteel@victus:~$ cd Desktop/
BitterSteel@victus:~/Desktop$ mkdir my_app
BitterSteel@victus:~/Desktop$ cd my_app/
BitterSteel@victus:~/Desktop/my_app$ code .
```

**Step2:**

Initialize project and install express

```
BitterSteel@victus:~/Desktop/my_app$ npm init -y
Wrote to /home/BitterSteel/Desktop/my_app/package.json:

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

```
BitterSteel@victus:~/Desktop/my_app$ npm install express

added 68 packages, and audited 69 packages in 862ms

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

found 0 vulnerabilities
```

**Step3:**

Write app.js file

app.js > ...

```
const express = require('express');
const app = express();
const port = 3000;

app.use(express.json());

app.get('/', (req, res) => {
  res.send('Hello World');
});

app.listen(port,'0.0.0.0', () => {
  console.log(`Server is running at http://localhost:${port}`);
});
```

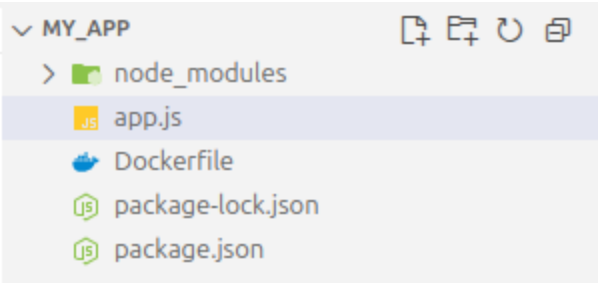
**Step4:**

Write Dockerfile

Dockerfile > ...

```
FROM node:18-alpine

WORKDIR /app

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 3000

CMD ["node", "app.js"]
```

Project Structure



**Step5:**

Create Docker image and then check if the image is installed

```
BitterSteel@victus:~/Desktop/my_app$ docker build -t app:2.0 .
[+] Building 3.7s (10/10) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 157B
 => [internal] load metadata for docker.io/library/node:18-alpine
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/5] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4
 => [internal] load build context
 => => transferring context: 43.73kB
 => CACHED [2/5] WORKDIR /app
 => CACHED [3/5] COPY package*.json ./
 => CACHED [4/5] RUN npm install
 => [5/5] COPY . .
 => exporting to image
 => => exporting layers
 => => writing image sha256:e685a731bd67c8a97876930f8f7e26fce85caaa
 => => naming to docker.io/library/app:2.0

BitterSteel@victus:~/Desktop/my_app$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
app          2.0       e685a731bd67   7 seconds ago    130MB
app          1.0       b8bbe76d2761   9 minutes ago    130MB
```

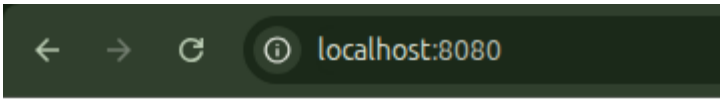
**Step6:**

Run the container by binding the port, in detached mode and then check the status of container

```
BitterSteel@victus:~/Desktop/my_app$ docker run -d -p 8080:3000 app:2.0
eb487b3a58dd425199c0fa2d16863d4851fc828cfc9c454c0595da7f0664eb21

BitterSteel@victus:~/Desktop/my_app$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS
eb487b3a58dd   app:2.0   "docker-entrypoint.s…"   53 seconds ago       Up 53 seconds
f6195cf8a23a   app:2.0   "docker-entrypoint.s…"   About a minute ago   Created
970d1b42694a   app:1.0   "docker-entrypoint.s…"   11 minutes ago       Exited (137) About a minute ago
```

**Step7:**

Check in browser http://localhost:8080





**Step8:**

Stop the container

$ docker container stop <container-id>