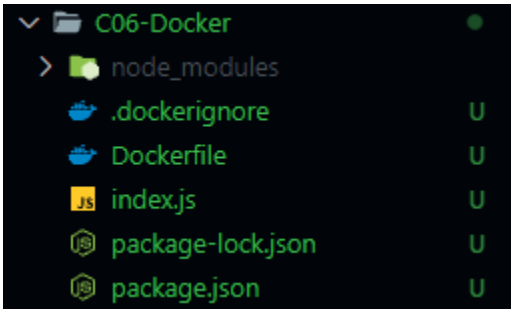
Difference between Docker and Podman

Docker uses a daemon, which is the program in charge to run all the CLI commands such as create images and run containers. By other hand podman doesn't use something like a daemon, it just run the commands by itself using a different technology called conmon

Container using Dockerfile

- First of all we need to work on the project we want to run using the docker container so the project folder structure would look similar to this



- First step is to setup a simple express project using node

✓ npm init-y
✓ npm i express

```
Cristian@DESKTOP-PM304DL MINGW64 ~/Desktop/JalaBootcamp/DevOps/C06-Docker (main)
$ npm i express

up to date, audited 51 packages in 1s

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

found 0 vulnerabilities
```

- Work on the index.js, for this example I created a simple express server

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

app.get('/', ( request, response ) => {
    response.send('Up using docker container');
});

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

- Create a docker file which will execute needed commands in order to create a docker image

```
FROM node:12

COPY [".","/usr/src"]

WORKDIR /usr/src

RUN npm install

EXPOSE 3000

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

- ✓ **FROM** indicates base image which is Node version 12
- ✓ **COPY** moves everything in the current folder to indicated container folder
- ✓ **WORKDIR** establish the working directory for the container
- ✓ **RUN** indicates commands that will be executed during build process
- ✓ **EXPOSE** indicates container to listen in port 3000
- ✓ **CMD** executes the command “node index.js” inside the container

- Create the docker image using the dockerfile

✓ docker build -t dockernode .

First time execution would take a few minutes

```
Cristian@DESKTOP-PM304DL MINGW64 ~/Desktop/JalaBootcamp/DevOps/C06-Docker (main)
$ docker build -t dockernode
[+] Building 1.3s (10/10) FINISHED
 => [internal] load build definition from Dockerfile                                                                0.0s
 => => transferring dockerfile: 32B                                                                                 0.0s
 => [internal] load .dockerignore                                                                                   0.0s
 => => transferring context: 34B                                                                                    0.0s
 => [internal] load metadata for docker.io/library/node:12                                                          1.2s
 => [auth] library/node:pull token for registry-1.docker.io                                                         0.0s
 => [internal] load build context                                                                                   0.0s
 => => transferring context: 32.04kB                                                                                0.0s
 => [1/4] FROM docker.io/library/node:12@sha256:8b4c721e613da8cfeda2f481668648ad9a4ec582faff9f43f1cfcc133dd3cba8     0.0s
 => CACHED [2/4] COPY [.,/usr/src]                                                                                  0.0s
 => CACHED [3/4] WORKDIR /usr/src                                                                                   0.0s
 => CACHED [4/4] RUN npm install                                                                                    0.0s
 => exporting to image                                                                                              0.0s
 => => exporting layers                                                                                             0.0s
 => => writing image sha256:27235534140e0d1a8031caa4ef1f06084l0001f1a7b9975cab5aba6bf60aleeb                        0.0s
 => => naming to docker.io/library/dockernode                                                                       0.0s
```

✓ docker image ls

```
Cristian@DESKTOP-PM304DL MINGW64 ~/Desktop/JalaBootcamp/DevOps/C06-Docker (main)
$ docker image ls
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
dockernode   latest    272355534140e   8 hours ago   920MB
```

- Once the image is created just left to run it as a container

Publish parameter (-p) must map the computer designated PORT (3000) with containers listening PORT (3000)

Iterative parameter -it runs the docker image and attach the logs in the current console

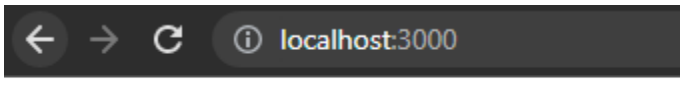
✓ docker run -it --name dockernode -p 3000:3000 dockernode

```
Cristian@DESKTOP-PM304DL MINGW64 ~/Desktop/JalaBootcamp/DevOps/C06-Docker (main)
$ docker run -it --name dockernode -p 3000:3000 dockernode
Server listening on 3000
```

- Verify that container is running listing the docker processes and opening localhost at port 3000

✓ docker ps -a (In a new terminal as the iterative mode blocks the previous one until the container is stopped)

```
Cristian@DESKTOP-PM304DL MINGW64 ~/Desktop/JalaBootcamp/DevOps (main)
$ docker ps -a
CONTAINER ID   IMAGE        COMMAND                  CREATED         STATUS         PORTS                    NAMES
bddca7e20d42   dockernode   "docker-entrypoint.s…"   2 minutes ago   Up 2 minutes   0.0.0.0:3000->3000/tcp   dockernode
```



Up using docker container