

Building a Docker Image and Running it locally (For Windows)

- [Rohith Neeraje](#)

Content:

Download and install docker

Clone github repo

Create Dockerfile

Build Docker image

Run Docker Container

Verify if running or not

To Build a docker image, we first need to ensure that we have docker desktop installed and running on our system.

DOWNLOAD AND INSTALL DOCKER

<https://docs.docker.com/desktop/install/windows-install/>

Follow the steps for installing docker desktop, and ensure you have WSL2 installed in the system.

To create a docker image, follow these steps:

Step 1:

Clone the repository (<https://github.com/mrdoob/three.js.git>) using the following command in WSL command line:

```
git clone https://github.com/mrdoob/three.js.git
```

Step 2:

Navigate to the required directory:

```
cd three.js/editor
```

Step 3:

Create the Dockerfile using the following command:

Touch Dockerfile

Step 4:

To edit the dockerfile, either open the dockerfile in notepad and edit it, or if using WSL, use the following commands:

Vi Dockerfile

(press i to enter into edit mode, where we should enter the dockerfile content)

Dockerfile content:

```
# Use a lightweight web server as a base image
```

```
FROM nginx:alpine
```

```
# Set the working directory inside the container
```

WORKDIR /usr/share/nginx/html

Copy the editor files into the container

COPY . .

Expose the port on which the web server will run

EXPOSE 80

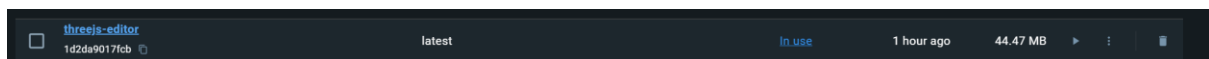
(press escape and type :wq to write and quit the vi)

Step 5:

To build the docker image, we need to run the following command to obtain a docker image:

docker build -t threejs-editor .

We can view the image in docker desktop



The command line should look like this:

```
rohith@Rohith:/mnt/c/Users/rohit/three.js/editor$ docker build -t threejs-editor .
[+] Building 2.3s (13/13) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 721B                                              0.0s
=> [internal] load metadata for docker.io/library/node:14-alpine                 1.9s
=> [auth] library/node:pull token for registry-1.docker.io                     0.0s
=> [internal] load .dockerignore                                                 0.0s
=> => transferring context: 2B                                                  0.0s
=> [1/7] FROM docker.io/library/node:14-alpine@sha256:434215b487a329c9e867202ff89e704d3a75e554822e07f3e0c0f9e606 0.0s
=> [internal] load build context                                                0.3s
=> => transferring context: 7.13kB                                              0.3s
=> CACHED [2/7] WORKDIR /app                                                    0.0s
=> CACHED [3/7] COPY package*.json ./                                           0.0s
=> CACHED [4/7] RUN npm install                                                  0.0s
=> CACHED [5/7] COPY . .                                                        0.0s
=> CACHED [6/7] COPY docker-entrypoint.sh /usr/local/bin/                     0.0s
=> CACHED [7/7] RUN chmod +x /usr/local/bin/docker-entrypoint.sh              0.0s
=> exporting to image                                                          0.0s
=> => exporting layers                                                         0.0s
=> => writing image sha256:e2311df6af35319197632778270f99c9daba00447db8cc4597f398898a69ef70 0.0s
=> => naming to docker.io/library/threejs-editor                             0.0s
```

(if you get a permission denied message, run the command with sudo before it)

Step 6:

After building the image, run the container with the following command:

docker run -p 8080:80 threejs-editor

The Docker daemon will start and should look something like this:

```

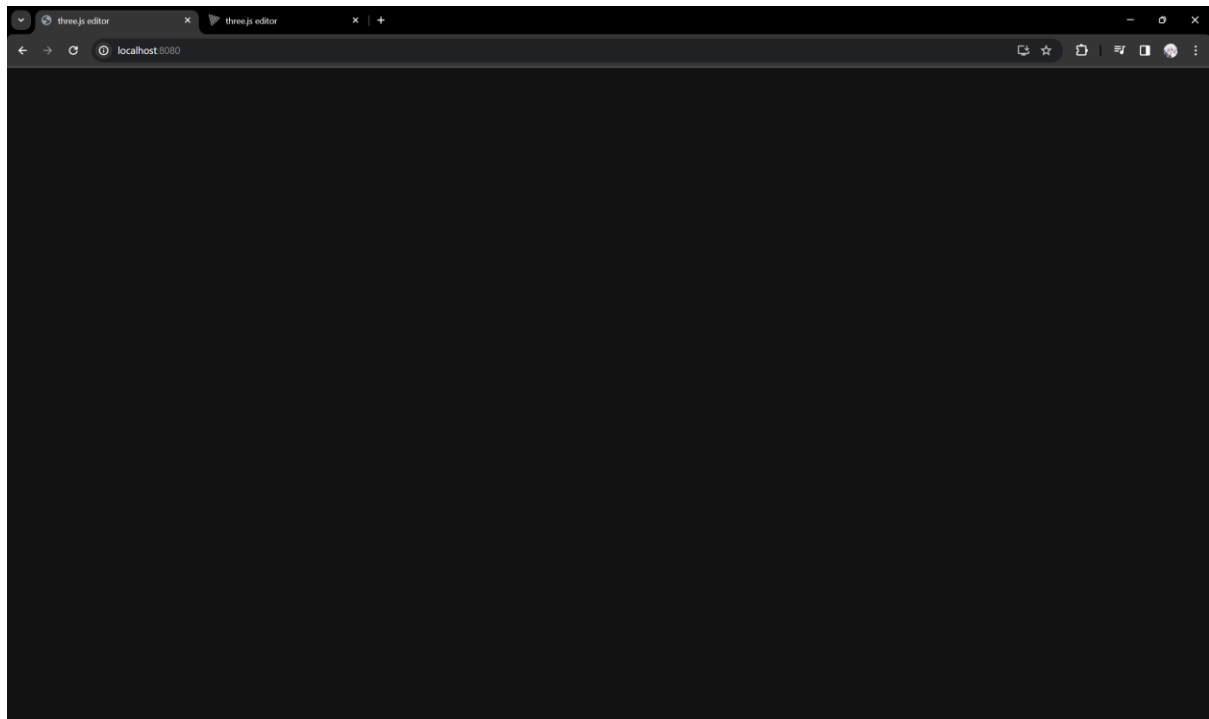
rohit@rohit:~/mnt/c/Users/rohit/three.js/editor$ docker run -p 8080:80 threejs-editor
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2024/04/08 11:10:11 [notice] #1: using the "epoll" event method
2024/04/08 11:10:11 [notice] #1: nginx/1.25.4
2024/04/08 11:10:11 [notice] #1: built by gcc 12.2.1 20220824 (Alpine 12.2.1_git20220824-r10)
2024/04/08 11:10:11 [notice] #1: OS: Linux 5.15.146.1-microsoft-standard-WSL2
2024/04/08 11:10:11 [notice] #1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/04/08 11:10:11 [notice] #1: start worker processes
2024/04/08 11:10:11 [notice] #1: start worker process 30
2024/04/08 11:10:11 [notice] #1: start worker process 31
2024/04/08 11:10:11 [notice] #1: start worker process 32
2024/04/08 11:10:11 [notice] #1: start worker process 33
2024/04/08 11:10:11 [notice] #1: start worker process 34
2024/04/08 11:10:11 [notice] #1: start worker process 35
2024/04/08 11:10:11 [notice] #1: start worker process 36
2024/04/08 11:10:11 [notice] #1: start worker process 37
2024/04/08 11:10:11 [notice] #1: start worker process 38
2024/04/08 11:10:11 [notice] #1: start worker process 39
2024/04/08 11:10:11 [notice] #1: start worker process 40
2024/04/08 11:10:11 [notice] #1: start worker process 41
2024/04/08 11:10:11 [notice] #1: start worker process 42
2024/04/08 11:10:11 [notice] #1: start worker process 43
2024/04/08 11:10:11 [notice] #1: start worker process 44
2024/04/08 11:10:11 [notice] #1: start worker process 45
2024/04/08 11:10:11 [notice] #1: start worker process 46
2024/04/08 11:10:11 [notice] #1: start worker process 47
2024/04/08 11:10:11 [notice] #1: start worker process 48
2024/04/08 11:10:11 [notice] #1: start worker process 49
2024/04/08 11:10:11 [notice] #1: start worker process 50
2024/04/08 11:10:11 [notice] #1: start worker process 51
2024/04/08 11:10:11 [notice] #1: start worker process 52
2024/04/08 11:10:11 [notice] #1: start worker process 53
172.17.0.1 -- [08/Apr/2024:11:10:16 +0000] "GET / HTTP/1.1" 200 6752 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36" "-"
172.17.0.1 -- [08/Apr/2024:11:10:16 +0000] "GET /css/main.css HTTP/1.1" 200 9472 "http://localhost:8080/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36" "-"
2024/04/08 11:10:16 [error] 31#31: *2 open() "/usr/share/nginx/html/examples/jsm/libs/draco/draco_encoder.js" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /examples/jsm/libs/draco/draco_encoder.js HTTP/1.1", host: "localhost:8080", referer: "http://localhost:8080/"
172.17.0.1 -- [08/Apr/2024:11:10:16 +0000] "GET /examples/jsm/libs/draco/draco_encoder.js HTTP/1.1" 404 555 "http://localhost:8080/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36" "-"
172.17.0.1 -- [08/Apr/2024:11:10:16 +0000] "GET /js/libs/codemirror/codemirror.css HTTP/1.1" 200 8720 "http://localhost:8080/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36" "-"

```

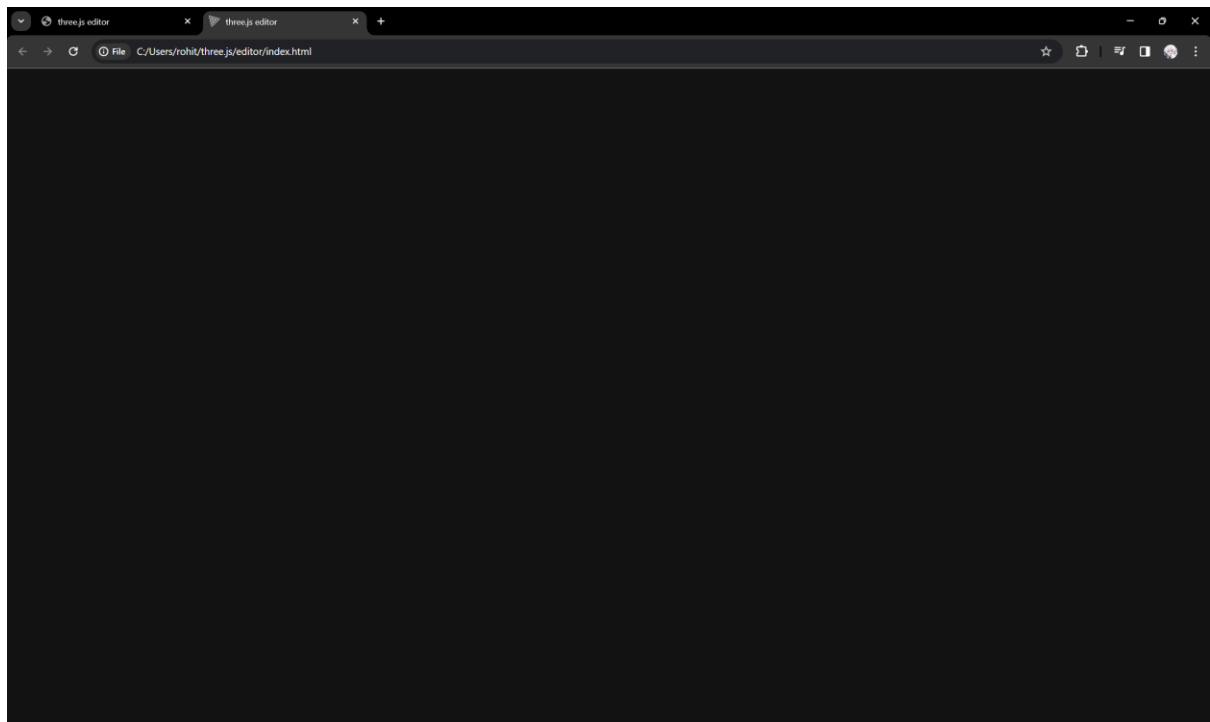
(This command prevents you from using the command line. If you don't want that, use the command line: `docker run -d -p 8080:80 threejs-editor` to run the docker container in a detached mode)

Step 7:

In a web browser, type in `localhost:8080` to check whether or not the application is running or not.



This is the screenshot of the docker image running, and in the next screenshot we can see its mirroring the html file present locally in the editor directory.



Hence, we have understood how to create a dockerfile, build a docker image and run the docker container in a local machine.