

# Docker Project 1

## Creating a container from a pulled Image

- 1) first pull nginx image by using `docker pull nginx` command

```
root@PUNELPT0393:/home/einfochips# docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
f11c1adaa26e: Pull complete
c6b156574604: Pull complete
ea5d7144c337: Pull complete
1bbcb9df2c93: Pull complete
537a6cfe3404: Pull complete
767bff2cc03e: Pull complete
adc73cb74f25: Pull complete
Digest: sha256:67682bda769fae1ccf5183192b8daf37b64cae99c6c3302650f6f8bf5f0f95df
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

- 2) run nginx container and verify the container is running by using `docker run --name my-nginx -d -p 8080:80 nginx` command

```
root@PUNELPT0393:/home/einfochips# docker run --name my-nginx -d -p 8080:80 nginx
77687d763263bc2672038e3ec4a179f7d00edefabb662768ab93a33d33e01870
root@PUNELPT0393:/home/einfochips# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
77687d763263	nginx	"/docker-entrypoint..."	25 seconds ago	Up 24 seconds	0.0.0.0:8080->80/tcp, :::8080->80/tcp	my-nginx

```
root@PUNELPT0393:/home/einfochips#
```

- 3) verify the container is running byr visiting in browser



## Modifying the Container and Creating a New Image

**Objective:** Modify the running Nginx container to serve a custom HTML page and create a new image from this modified container.

1) Access the Running container by using `docker exec -it my-nginx /bin/bash` command

and create HTML page by using `echo "<html><body><h1>Hello from Docker!</h1></body></html>" > /usr/share/nginx/html/index.html` command

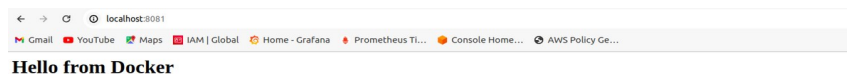
```
root@PUNELPT0393:/home/einfochips# docker exec -it my-nginx /bin/bash
root@77687d763263:/# echo "<html><body><h1>Hello from Docker</h1></body></html>" > /usr/share/nginx/html/index.html
```

2) comit the changes to create new Image by using `docker commit my-nginx custom-nginx` command

and run the container from new Image by using `docker run --name my-custom-nginx -d -p 8081:80 custom-nginx` command

```
root@77687d763263:/# exit
exit
root@PUNELPT0393:/home/einfochips# docker commit my-nginx custom-nginx
sha256:36fea635914165b114df80d69ce8712aa53a26e17a54928597a762b62e795a72
root@PUNELPT0393:/home/einfochips# docker run --name my-custom-nginx -d -p 8081:80 custom-nginx
a8379f53233f748a0243eb4adc09f3d79c1b0f82bc75472888544eb319fb188e
root@PUNELPT0393:/home/einfochips#
```

3) verify new container by visiting link in browser



## Creating Docker to build and Deploy a web Application

**Objective:** Write a Dockerfile to create an image for a simple web application and run it as a container.

1) create project directory name “ my-webapp” and move to that directory by using

```
mkdir my-webapp  
cd my-webapp
```

2) In my-webapp create “ index.html “ file to write html code .

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp# nano index.html  
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp# cat index.html  
<!DOCTYPE html>  
<html>  
<body>  
  <h1>Hello from My Web App!</h1>  
</body>  
</html>
```

3) create Dockerfile

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp# nano Dockerfile  
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp# cat Dockerfile  
# Use the official Nginx base image  
FROM nginx:latest  
  
# Copy the custom HTML file to the appropriate location  
COPY index.html /usr/share/nginx/html/  
  
# Expose port 80  
EXPOSE 80
```

4) Build the docker Image `docker build -t my-webapp-image .`

and run the container from the build Image

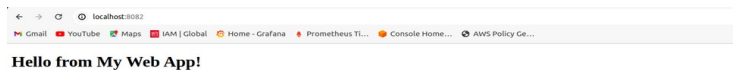
```
docker run --name my-webapp-container -d -p 8082:80 my-webapp-image
```

```

root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp# docker build -t my-webapp-image .
[+] Building 0.2s (7/7) FINISHED
docker:default
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 217B 0.0s
=> [internal] load metadata for docker.io/library/nginx:latest 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 119B 0.0s
=> [1/2] FROM docker.io/library/nginx:latest 0.1s
=> [2/2] COPY index.html /usr/share/nginx/html/ 0.0s
=> exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:e721dd479f8c22cdeb03b8870cd3f787ccb38008bc511abeb 0.0s
=> => naming to docker.io/library/my-webapp-image 0.0s
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp# docker run --name my-webapp-container -d -p 8082:80 my-webapp-image
4b133bac8a84cc30005951959dcd7f00c9eaae4107ca7ca584fd2f01cc6717d0
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp#

```

5) verify by visiting browser using URL



## Remove all related containers and images to clean up environment

1) stop , remove containers and remove Images

```
docker stop my-nginx my-custom-nginx my-webapp-container
```

```
docker rm my-nginx my-custom-nginx my-webapp-container
```

```
docker rmi nginx custom-nginx my-webapp-image
```

```

root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp# docker stop my-nginx my-custom-nginx my-webapp-container
my-nginx
my-custom-nginx
my-webapp-container
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp# docker rm my-nginx my-custom-nginx my-webapp-container
my-nginx
my-custom-nginx
my-webapp-container
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp# docker rmi nginx custom-nginx my-webapp-image
Untagged: nginx:latest
Untagged: nginx@sha256:67682bda769fae1ccf5183192b8daf37b64cae99c6c3302650f6f8bf5f0f95df
Untagged: custom-nginx:latest
Deleted: sha256:36fea635914165b114df80d69ce8712aa53a26e17a54928597a762b62e795a72
Deleted: sha256:5c170501cbf3aba09344ccc49adeb2e35020784b340d7f72094df376ab89c3b2
Deleted: sha256:fffffc90d343cbcb01a5032edac86db5998c536cd0a366514121a45c6723765c
Untagged: my-webapp-image:latest
Deleted: sha256:e721dd479f8c22cdeb03b8870cd3f787ccb38008bc511abeb7c6d1a581973271
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp# █

```

## Docker Project 2

build full-stack applicaton using docker.

Applocation consist of front-end web server (Nginx) , back-end appication server (node.js) and postgreSQL database.

For database and handle inter-container communication persistent volume is used

1) create project directory name “ fullstack-docker-app”

```

mkdir fullstack-docker-app
cd fullstack-docker-app

```

2) In Directory create subdirectories name “ frontend , backend ,database ” for each service

```

mkdir frontend backend database

```

3) create network to allow communication between containers

`docker network create fullstack-network`

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july# mkdir fullstack-docker-app
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july# cd fullstack-docker-app/
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# mkdir frontend backend database
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# ls
backend database frontend
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# docker network create fullstack-network
6323847df045d03efe778495b0177cd834c62594e6f2aae25c4937b5dbce5b0b
```

4) create volume for PostgreSQL database

`docker volume create pgdata`

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# docker volume create pgdata
pgdata
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app#
```

## set up database with docker

1) In “ database ” directory create dockerfile for database

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/database# nano Dockerfile
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/database# cat Dockerfile
FROM postgres:latest
ENV POSTGRES_USER=user
ENV POSTGRES_PASSWORD=password
ENV POSTGRES_DB=mydatabase
```



## 2) build postgresQL Image

`docker build -t my-postgres-db .`

```
root@PUNELPT0393:/home/einfochips/Desktop/Learning/10july/fullstack-docker-app/database# docker build -t my-postgres-db .
[+] Building 105.3s (3/4)
ault
=> [internal] load build definition from Dockerfile
0.0s
=> => transferring dockerfile: 139B
0.0s
=> [internal] load metadata for docker.io/library/postgres:latest
3.1s
=> [internal] load .dockerignore
0.0s
=> => transferring context: 2B
0.0s
=> [1/1] FROM docker.io/library/postgres:latest@sha256:0aafd2ae7e6c391f39fb6b7621632d79f54068faebc726caf469e87bd1d301c0
2.1s
=> => resolve docker.io/library/postgres:latest@sha256:0aafd2ae7e6c391f39fb6b7621632d79f54068faebc726caf469e87bd1d301c0
0.0s
=> => sha256:6b7a1245fe7148b56315cd314fe4764418e6e4ec1095b4e781e3a60ca6990e64 1.45MB / 1.45MB
1.3s
=> => sha256:0aafd2ae7e6c391f39fb6b7621632d79f54068faebc726caf469e87bd1d301c0 10.26kB / 10.26kB
0.0s
=> => sha256:919ca406a058a0c62788518f39a66257c122ace2007ac6e1425e51846f37e300 4.53MB / 4.53MB
8.9s
=> => sha256:d13ef786196545cd69aff1945929fc868712196e195bc66581fb1bfe81649eaf 3.63kB / 3.63kB
0.0s
=> => sha256:76ce212b9153cf6743484a31c2967fd5f284455f8ede2277943f9d92010ffa9e 1.17kB / 1.17kB
0.4s
=> => sha256:f23dc7cd74bd7693fc164fd829b9a7fa1edf8eaaed488c117312aef2a48cafaa 10.09kB / 10.09kB
0.0s
=> => extracting sha256:76ce212b9153cf6743484a31c2967fd5f284455f8ede2277943f9d92010ffa9e
0.0s
=> => sha256:8064ffe06c65d307820b1373f7d39b16c32929f7f2c2cf3c3438077f3409a3e5 8.07MB / 8.07MB
```

```

[+] Building 349.7s (5/5) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile                0.0s2
=> => transferring dockerfile: 139B                               0.0s
=> [internal] load metadata for docker.io/library/postgres:latest 3.1s2
=> [internal] load .dockerignore                                  0.0s
=> => transferring context: 2B                                     0.0s2
=> [1/1] FROM docker.io/library/postgres:latest@sha256:0aafd2ae7e6c391f39fb6b7621632d79f54068faebc726caf469e87bd1d301c0 346.5s
=> => resolve docker.io/library/postgres:latest@sha256:0aafd2ae7e6c391f39fb6b7621632d79f54068faebc726caf469e87bd1d301c0 0.0s0
=> => sha256:6b7a1245fe7148b56315cd314fe4764418e6e4ec1095b4e781e3a60ca6990e64 1.45MB / 1.45MB 11.3s
=> => sha256:0aafd2ae7e6c391f39fb6b7621632d79f54068faebc726caf469e87bd1d301c0 10.26kB / 10.26kB 0.0s
=> => sha256:919ca406a058a0c62788518f39a66257c122ace2007ac6e1425e51846f37e300 4.53MB / 4.53MB 48.9s
=> => sha256:d13ef786196545cd69aff1945929fc868712196e195bc66581fb1bfe81649eaf 3.63kB / 3.63kB 0.0s5
=> => sha256:76ce212b9153cf6743484a31c2967fd5f284455f8ede2277943f9d92010ffa9e 1.17kB / 1.17kB 0.4s
=> => sha256:f23cd7cd74bd7693fc164fd829b9a7fa1edf8eaaed488c117312aef2a48cafaa 10.09kB / 10.09kB 0.0s
=> => extracting sha256:76ce212b9153cf6743484a31c2967fd5f284455f8ede2277943f9d92010ffa9e 0.0s
=> => sha256:8064ffe06c65d307820b1373f7d39b16c32929f7f2c2cf3c3438077f3409a3e5 8.07MB / 8.07MB 59.5s5
=> => sha256:4b5c59f2d82cd85059458a41c01ebe3b5a2c96f1d584b2e24574149ff0406161 1.20MB / 1.20MB 20.1s
=> => sha256:fe72764b907045579a9993c60f23f3213caf82d2ade388988d41ccff9c184f59 116B / 116B 20.7s5
=> => sha256:6ef8e2c0f4d924192a8ac20be4b72e17ed0933c5edbf5007a295bf8e77f0e451 3.14kB / 3.14kB 21.1s
=> => sha256:e71fe9d7ff11d95e9c7e54d020428f20e5262263af563100c8df0508820b2707 109.03MB / 109.03MB 340.2s5
=> => extracting sha256:919ca406a058a0c62788518f39a66257c122ace2007ac6e1425e51846f37e300 0.2s
=> => sha256:f3225d69190dd8f447a026cdaebfe03d062d04bf51df47f33a30f2088056b921 9.92kB / 9.92kB 50.6s5
=> => extracting sha256:6b7a1245fe7148b56315cd314fe4764418e6e4ec1095b4e781e3a60ca6990e64 0.0s
=> => sha256:2bf90d17afc8b5ec96e10e71bb08c75b3f82646c388a8084b3e9de73437b02be 128B / 128B 51.2s
=> => sha256:d3aee49eb07954f0cb1c957ca0998e5c635d0617ab593f42433c97ed16312342 170B / 170B 51.6s
=> => sha256:e1e856589194f439ba266b753710f59da6a5de9715724c6249a88230a8211c7 5.42kB / 5.42kB 52.1s
=> => sha256:95c2c2ef9f02d7666e80992c98c53c9ec7b5e8ccf244d00a5c85e46bbc2820ae 184B / 184B 52.4s
=> => extracting sha256:8064ffe06c65d307820b1373f7d39b16c32929f7f2c2cf3c3438077f3409a3e5 0.9s
=> => extracting sha256:4b5c59f2d82cd85059458a41c01ebe3b5a2c96f1d584b2e24574149ff0406161 0.1s
=> => extracting sha256:fe72764b907045579a9993c60f23f3213caf82d2ade388988d41ccff9c184f59 0.0s
=> => extracting sha256:6ef8e2c0f4d924192a8ac20be4b72e17ed0933c5edbf5007a295bf8e77f0e451 0.0s
=> => extracting sha256:e71fe9d7ff11d95e9c7e54d020428f20e5262263af563100c8df0508820b2707 5.7s
=> => extracting sha256:f3225d69190dd8f447a026cdaebfe03d062d04bf51df47f33a30f2088056b921 0.0s
=> => extracting sha256:2bf90d17afc8b5ec96e10e71bb08c75b3f82646c388a8084b3e9de73437b02be 0.0s
=> => extracting sha256:d3aee49eb07954f0cb1c957ca0998e5c635d0617ab593f42433c97ed16312342 0.0s
=> => extracting sha256:e1e856589194f439ba266b753710f59da6a5de9715724c6249a88230a8211c7 0.0s
=> => extracting sha256:95c2c2ef9f02d7666e80992c98c53c9ec7b5e8ccf244d00a5c85e46bbc2820ae 0.0s
=> exporting to image                                           0.1s
=> => exporting layers                                           0.0s
=> => writing image sha256:1cfe6ff8722f48bf2889ab46a727e0e5ed0bf4046b570b920262d21e57ca2275 0.0s
=> => naming to docker.io/library/my-postgres-db                0.0s
root@PUNELPT0393:/home/einfochips/Desktop/Learning/10july/fullstack-docker-app/database#

```

### 3) Run postgresQL container

```

docker run --name postgres-container --network fullstack-network -v
pgdata:/var/lib/postgresql/data -d my-postgres-db

```

```

root@PUNELPT0393:/home/einfochips/Desktop/Learning/10july/fullstack-docker-app/database# docker run --name postgres-container --network fullstack-network -v pgdata:/var/lib/postgresql
/data -d my-postgres-db
66102c0b6d073ceec650cbe77ea8c6fcc5391f8159f5027ab20e61d6358eed0

```



## create node.js application with Express and set it up with Docker

1) Initialize the node.js Application. Before that we check version of node and npm

```
npm init -y
```

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# node --version
v10.19.0
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# npm --version
6.14.4
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# npm init -y
Wrote to /home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend/package.json:

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

2) install Express and pg . This are postgresQL client for Node.js

```
npm install express pg
```

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# npm install express pg
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN backend@1.0.0 No description
npm WARN backend@1.0.0 No repository field.

+ express@4.19.2
+ pg@8.12.0
added 78 packages from 49 contributors and audited 78 packages in 7.038s

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

found 0 vulnerabilities

root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# ls
node_modules package.json package-lock.json
```

### 3) In backend directory create file name “ index.js ”

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# nano index.js
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# code index.js
You are trying to start Visual Studio Code as a super user which isn't recommended. If this was intended, please add the argument '--no-sandbox' and specify an alternate user data directory using the '--user-data-dir' argument.
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# cat index.js
const express = require('express');
const { Pool } = require('pg');
const app = express();
const port = 3000;

const pool = new Pool({
  user: 'user',
  host: 'postgres-container',
  database: 'mydatabase',
  password: 'password',
  port: 5432,
});

app.get('/', (req, res) => {
  res.send('Hello from Node.js and Docker!');
});

app.get('/data', async (req, res) => {
  const client = await pool.connect();
  const result = await client.query('SELECT NOW()');
  client.release();
  res.send(result.rows);
});

app.listen(port, () => {
  console.log(`App running on http://localhost:${port}`);
});
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend#
```

### 4) create “ dockerfile ” in “ backend ” directory

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# nano dockerfile
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# cat dockerfile
FROM node:latest

WORKDIR /usr/src/app

COPY package*.json ./
RUN npm install

COPY . .

EXPOSE 3000
CMD ["node", "index.js"]
```

## 5) build backend image

`docker build -t my-node-app .`

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# docker build -t my-node-app .
[+] Building 1136.6s (10/10) FINISHED
=> [internal] load build definition from dockerfile
=> => transferring dockerfile: 163B
=> [internal] load metadata for docker.io/library/node:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/node:latest@sha256:c8a559f733bf1f9b3c1d05b97d9a9c7e5d3647c99abedaf5cdd3b54c9cbb8eff
=> resolve docker.io/library/node:latest@sha256:c8a559f733bf1f9b3c1d05b97d9a9c7e5d3647c99abedaf5cdd3b54c9cbb8eff
=> sha256:f5ee4e641452e29a0ed1c0dc9ee8eb90796ca1ebd1f439bb130248548ffed06a 2.49kB / 2.49kB
=> sha256:3dbed71fc5444cf6889a21b002de3e7805e810aa88f91a9ca941b4e3880246d1 64.14MB / 64.14MB
=> sha256:c8a559f733bf1f9b3c1d05b97d9a9c7e5d3647c99abedaf5cdd3b54c9cbb8eff 6.41kB / 6.41kB
=> sha256:cd86d0acabd627e36426b1259849c8ee44f2f800b45e38030eedd5033abadf1d 6.60kB / 6.60kB
=> sha256:e9aef93137af6e967e7242f3b3c8ecd8e6f571d1e6fdd9e72db0befeeae3cf13 49.55MB / 49.55MB
=> sha256:58b365fa3e8dc16e70d89fab0e91f5242feb38ae3cf6b6655e654209ea109333 24.05MB / 24.05MB
=> sha256:ae70830af8b64968596022cd9041ce522f0c77eab5419b19e169b53582c69e3f 211.23MB / 211.23MB
=> sha256:572e7a55de7f456ea9ae48f2cd37245ab1b21832ae1fa679d0f3a215cc86dab7 3.32kB / 3.32kB
=> extracting sha256:e9aef93137af6e967e7242f3b3c8ecd8e6f571d1e6fdd9e72db0befeeae3cf13
=> sha256:9f45a73683ad62471247db5bb99c0df1cdf1d2d68a03fc06f421e3d6e557f08 53.03MB / 53.03MB
=> extracting sha256:58b365fa3e8dc16e70d89fab0e91f5242feb38ae3cf6b6655e654209ea109333
=> sha256:0892d1c8f6930529b7afa2592b0887da6fc713ed26bcca50d8cc4ccf75f3691 1.25MB / 1.25MB
=> extracting sha256:3dbed71fc5444cf6889a21b002de3e7805e810aa88f91a9ca941b4e3880246d1
=> sha256:819caf31f4d073fac60e211533d396e30cfe1bc06391ea4bf10ae6d81483b831 448B / 448B
=> extracting sha256:ae70830af8b64968596022cd9041ce522f0c77eab5419b19e169b53582c69e3f
=> extracting sha256:572e7a55de7f456ea9ae48f2cd37245ab1b21832ae1fa679d0f3a215cc86dab7
=> extracting sha256:9f45a73683ad62471247db5bb99c0df1cdf1d2d68a03fc06f421e3d6e557f08
=> extracting sha256:0892d1c8f6930529b7afa2592b0887da6fc713ed26bcca50d8cc4ccf75f3691
=> extracting sha256:819caf31f4d073fac60e211533d396e30cfe1bc06391ea4bf10ae6d81483b831
=> [internal] load build context
=> => transferring context: 2.74MB
=> [2/5] WORKDIR /usr/src/app
=> [3/5] COPY package*.json ./
=> [4/5] RUN npm install
=> [5/5] COPY . .
=> exporting to image
=> exporting layers
=> => writing image sha256:b47e31cbcbf247b69718a6dafb02c9bc4e734daf6590a1de0ade305ed7301e4f
=> => naming to docker.io/library/my-node-app
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend#
```

## 6)run backend container

`docker run --name backend-container --network fullstack-network -d my-node-app`

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# docker run --name backend-container --network fullstack-network -d my-node-app
c52b28aa5cc7a5b412d6169ab6b2c972558cd3c8c45d0b91b2b6ad55d284576
```

## create static front-end and set it up with docker

1) in “ frontend ” directory create “ index.html “ file

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# cd ..
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# cd frontend/
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/frontend# nano index.html
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/frontend# cat index.html
<!DOCTYPE html>
<html>
<body>
  <h1>Hello from Nginx and Docker!</h1>
  <p>This is a simple static front-end served by Nginx.</p>
</body>
</html>
```

2) in “ frontend ” directory create “ dockerfile ”

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/frontend# nano dockerfile
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/frontend# cat dockerfile
FROM nginx:latest
COPY index.html /usr/share/nginx/html/index.html
```

3) build frontend Image and Run frontend container

`docker build -t my-nginx-app .`

`cd ..`

`docker run --name frontend-container --network fullstack-network -p 8080:80 -d my-nginx-app`

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/frontend# docker build -t my-nginx-app .
[+] Building 3.6s (7/7) FINISHED
                                docker:default
=> [internal] load build definition from dockerfile                                0.0s
=> => transferring dockerfile: 104B                                              0.0s
=> [internal] load metadata for docker.io/library/nginx:latest                  3.4s
=> [internal] load .dockerignore                                                 0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load build context                                                0.1s
=> => transferring context: 189B                                               0.1s
=> CACHED [1/2] FROM docker.io/library/nginx:latest@sha256:67682bda769fae1ccf5183192b8daf 0.0s
=> => resolve docker.io/library/nginx:latest@sha256:67682bda769fae1ccf5183192b8daf37b64ca 0.0s
=> [2/2] COPY index.html /usr/share/nginx/html/index.html                      0.0s
=> exporting to image                                                            0.0s
=> => exporting layers                                                            0.0s
=> => writing image sha256:6d586e9f57be8fe3bc0fac35a91acf12bd82587ab4afc82c357d73eb457289 0.0s
=> => naming to docker.io/library/my-nginx-app                                0.0s
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/frontend# cd..
cd..: command not found
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/frontend# cd ..
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# docker run --name frontend-container --network fullstack-network -p 8080:80 -d my-nginx-app
6b28fb4cfb0c34482073a97f35a5abdecfb7a0df4fde00ca6cfa6c5fedb926b
```

## Connecting the Backend and Database

Backend can communicate with database and handle data requests

1) Access backend container

```
docker exec -it backend-container /bin/bash
```

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# docker exec -it backend-container /bin/bash
```

2) Test the connection to the database using psql

```
apt-get update && apt-get install -y postgresql-client
```

```
psql -h postgres-container -U user -d mydatabase -c "SELECT NOW();" 
```

```
root@c52b28aa5ccc:/usr/src/app# apt-get update && apt-get install -y postgresql-client
Hit:1 http://deb.debian.org/debian bookworm InRelease
Hit:2 http://deb.debian.org/debian bookworm-updates InRelease
Hit:3 http://deb.debian.org/debian-security bookworm-security InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  postgresql-client-15 postgresql-client-common
Suggested packages:
  postgresql-15 postgresql-doc-15
The following NEW packages will be installed:
  postgresql-client postgresql-client-15 postgresql-client-common
0 upgraded, 3 newly installed, 0 to remove and 10 not upgraded.
Need to get 1744 kB of archives.
After this operation, 8244 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bookworm/main amd64 postgresql-client-common all 248 [35.1 kB]
Get:2 http://deb.debian.org/debian bookworm/main amd64 postgresql-client-15 amd64 15.7-0+deb12u1 [1699 kB]
Get:3 http://deb.debian.org/debian bookworm/main amd64 postgresql-client all 15+248 [10.1 kB]
Fetched 1744 kB in 5s (363 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package postgresql-client-common.
(Reading database ... 23243 files and directories currently installed.)
Preparing to unpack .../postgresql-client-common_248_all.deb ...
Unpacking postgresql-client-common (248) ...
Selecting previously unselected package postgresql-client-15.
Preparing to unpack .../postgresql-client-15_15.7-0+deb12u1_amd64.deb ...
Unpacking postgresql-client-15 (15.7-0+deb12u1) ...
Selecting previously unselected package postgresql-client.
Preparing to unpack .../postgresql-client_15+248_all.deb ...
Unpacking postgresql-client (15+248) ...
Setting up postgresql-client-common (248) ...
Setting up postgresql-client-15 (15.7-0+deb12u1) ...
update-alternatives: using /usr/share/postgresql/15/man/man1/psql.1.gz to provide /usr/share/man/man1/psql.1.gz (psql.1.gz) in auto mode
Setting up postgresql-client (15+248) ...
```

```

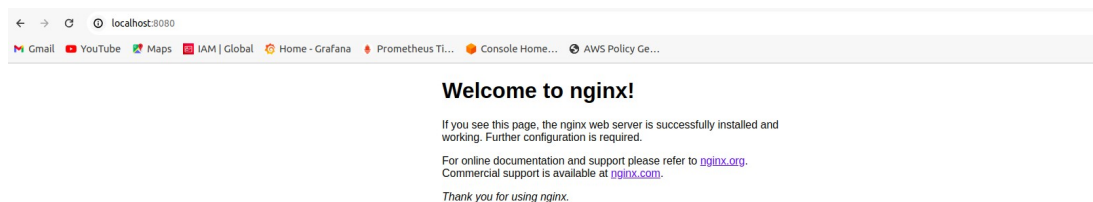
root@c52b28aa5ccc:/usr/src/app# psql -h postgres-container -U user -d mydatabase -c "SELECT NOW();"
Password for user user:
      now
-----
2024-07-11 09:35:58.857331+00
(1 row)

root@c52b28aa5ccc:/usr/src/app# exit
exit
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app#

```

## Final Integration and Testing

1) visit the browser use localhost:8080 to see the Nginx welcome page



2) On “ frontend ” branch update “ index.html ” file to include backend link . Using this we will verify full Integration

```

root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# cd frontend/
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/frontend# ls
dockerfile index.html
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/frontend# nano index.html
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/frontend# cat index.html
<!DOCTYPE html>
<html>
<body>
  <h1>Hello from Nginx and Docker!</h1>
  <p>This is a simple static front-end served by Nginx.</p>
  <a href="http://localhost:3000/data">Fetch Data from Backend</a>
</body>
</html>

```



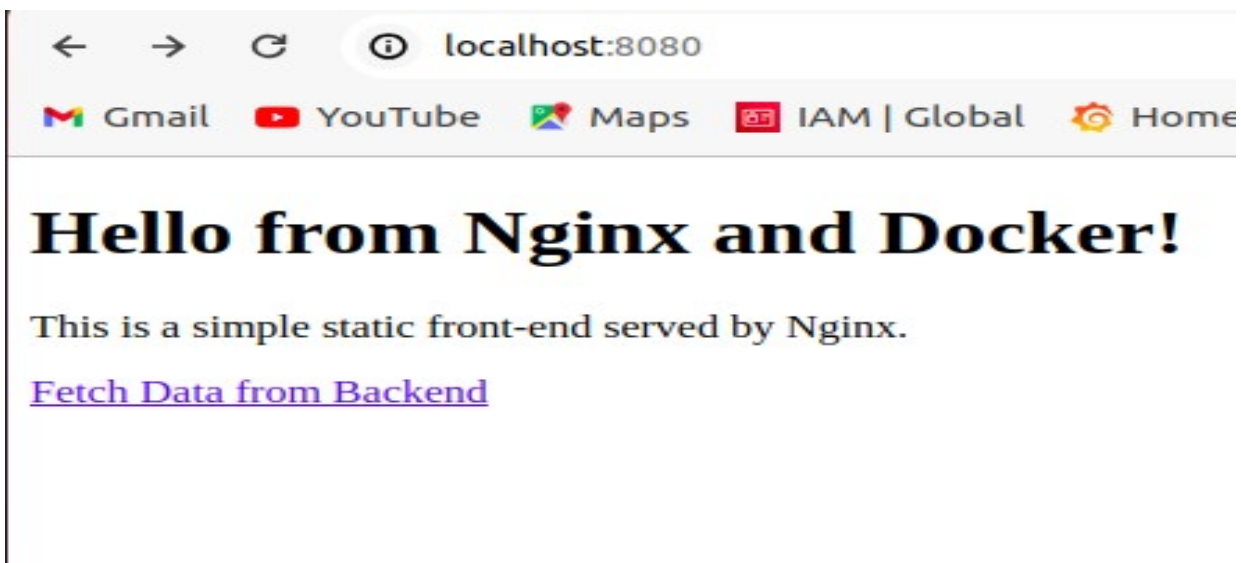
### 3) Rebuild and run the updated Frontend Container

```
docker build -t my-nginx-app .
```

```
docker run --name frontend-container --network fullstack-network -p 8080:80 -d my-nginx-app
```

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/frontend# docker build -t my-nginx-app .
[+] Building 3.6s (7/7) FINISHED                                docker:default
=> [internal] load build definition from dockerfile             0.0s
=> => transferring dockerfile: 104B                             0.0s
=> [internal] load metadata for docker.io/library/nginx:latest 3.4s
=> [internal] load .dockerignore                               0.0s
=> => transferring context: 2B                                    0.0s
=> [internal] load build context                                0.1s
=> => transferring context: 189B                                  0.1s
=> CACHED [1/2] FROM docker.io/library/nginx:latest@sha256:67682bda769fae1ccf5183192b8daf 0.0s
=> => resolve docker.io/library/nginx:latest@sha256:67682bda769fae1ccf5183192b8daf37b64ca 0.0s
=> [2/2] COPY index.html /usr/share/nginx/html/index.html     0.0s
=> exporting to image                                           0.0s
=> => exporting layers                                           0.0s
=> => writing image sha256:6d586e9f57be8fe3bc0fac35a91acf12bd82587ab4afc82c357d73eb457289 0.0s
=> => naming to docker.io/library/my-nginx-app                 0.0s
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/frontend# cd..
cd..: command not found
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/frontend# cd ..
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# docker run --name frontend-container --network fullstack-network -p 8080:80 -d my-nginx-app
6b28fb4cfb0c34482073a97f35a5abdecba7e0df4fde00ca6cfa6c5fedb926b
```

### 4) to verify visit browser by using localhost:8080



## **Remove all created containers, Images , networks and volumes to cleanup environment**

1) stop and remove the container

```
docker stop frontend-container backend-container postgres-container
```

```
docker rm frontend-container backend-container postgres-container
```

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# docker stop frontend-container backend-container postgres-container
frontend-container
backend-container
postgres-container
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# docker rm frontend-container backend-container postgres-container
frontend-container
backend-container
postgres-container
```

2) Remove the images :

```
docker rmi my-nginx-app my-node-app my-postgres-db
```

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# docker rmi my-nginx-app my-node-app my-postgres-db
Untagged: my-nginx-app:latest
Deleted: sha256:b0619b2ccb0653a4ceb2606bf2bfb9662fe0bb824a8cab5109eda9056bab6613
Untagged: my-node-app:latest
Deleted: sha256:b47e31cbcbf247b69718a6dafb02c9bc4e734daf6590a1de0ade305ed7301e4f
Untagged: my-postgres-db:latest
Deleted: sha256:1cfe6ff8722f48bf2889ab46a727e0e5ed0bf4046b570b920262d21e57ca2275
```

3) remove the network and volume

docker network rm fullstack-network

docker volume rm pgdata

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# docker network rm fullstack-network
fullstack-network
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# docker volume rm pgdata
pgdata
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# cd..
cd..: command not found
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# cd ..
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july# ls
10july 'Docker ' fullstack-docker-app my-webapp
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july#
```