Déploiement d'une Application Web avec Frontend et Backend avec Docker

Partie 1

1 .Mkdir mywebapp

```
root@UBUNTU:~# cd mywebapp
root@UBUNTU:~/mywebapp# ng new frontend
```

2.

```
root@UBUNTU:~/mywebapp# cd frontend
root@UBUNTU:~/mywebapp/frontend# nano Dockerfile
```

Frontend Dockerfile

```
# my-web-app/frontend/Dockerfile
FROM node:16

WORKDIR /usr/src/app

COPY package*.json ./

RUN npm install -g @angular/cli@16

RUN npm install

COPY . .

EXPOSE 4200

CMD ["ng", "serve", "--host", "0.0.0.0"]
```

Backend

```
root@UBUNTU:~/mywebapp# mkdir backend
root@UBUNTU:~/mywebapp# cd backend
root@UBUNTU:~/mywebapp/backend# npm init -y
Wrote to /root/mywebapp/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"
}


root@UBUNTU:~/mywebapp/backend# npm install express --save
```

App.js

```
  GNU nano 6.2                                app.js
// backend/app.js
const express = require('express');
const app = express();
const port = 8080;

app.get('/', (req, res) => {
  res.send('Hello from the backend!');
});

app.listen(port, () => {
  console.log("Server is running");
});
```

Dockerfile backend

```
  GNU nano 6.2                          Dockerfile
# backend/Dockerfile
FROM node:16

WORKDIR /usr/src/app

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 8080

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

Partie2

```
root@UBUNTU:~/mywebapp# nano docker-compose.yml
root@UBUNTU:~/mywebapp# docker-compose up -d
Building frontend
DEPRECATED: The legacy builder is deprecated and will be removed in a f
ease.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  361.4MB
Step 1/8 : FROM node:16
16: Pulling from library/node
311da6c465ea: Pull complete
7e9bf114588c: Pull complete
ffd9397e94b7: Pull complete
513d77925604: Downloading  122.6MB/191.9MB
ae3b95bbaa61: Download complete
0e421f66aff4: Download complete
ca266fd61921: Download complete
```

docker-compose.yml

```
  GNU nano 6.2                    docker-compose.yml
# my-web-app/docker-compose.yml
version: '3'

services:
  frontend:
    build:
      context: ./frontend
    ports:
      - "80:80"
    networks:
      - app-network

  backend:
    build:
      context: ./backend
    ports:
      - "8080:8080"
    networks:
      - app-network

networks:
  app-network:
```

```
frontend_1  | ✔ Compiled successfully.
backend_1   | Server is running
root@UBUNTU:~/mywebapp# docker-compose ps
       Name                     Command               State            Ports
-------------------------------------------------------------------------------
mywebapp_backend_1    docker-entrypoint.sh             Up      0.0.0.0:8080-
                      node  ...                               >8080/tcp,:::8080-
                                                              >8080/tcp
mywebapp_frontend_1   docker-entrypoint.sh ng          Up      4200/tcp, 0.0.0.0:80-
                      se ...                                  >80/tcp,:::80->80/tcp
```

Le resultat de ce command : docker-compose up -d --scale frontend=3

```
root@UBUNTU:~/mywebapp# docker-compose up -d --scale frontend_1=3
mywebapp_backend_1 is up-to-date
Stopping and removing mywebapp_frontend_2 ... done
Stopping and removing mywebapp_frontend_3 ... done
```

L’affichage