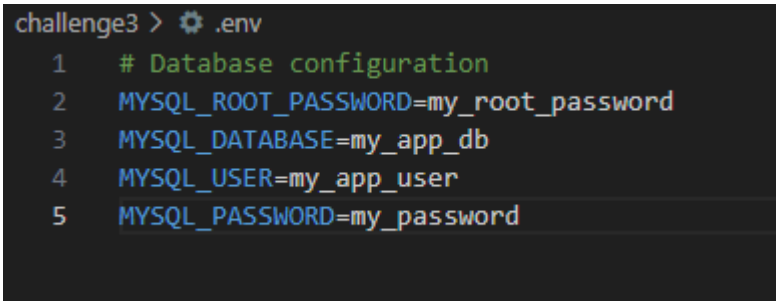
# Creating a Full-Stack App with Docker

## Challenge 3 – Full-stack application

**Basic principle:** This is a project that needs to run nginx, node service, db at the same time, you can use docker-compose inside Challenge 2, this time with more environment configuration files, define the database, database username and password, when creating the image will use the username and password defined in the environment variables to write the image, when the container is running, it will use the defined username and password to log in.

## Steps

In the root folder, .env file




And create docker-compose.yml:

```yaml
version: '3.8'

services:
  node-service:
    build:
      context: .
      dockerfile: docker/api/Dockerfile
    environment:
      DB_HOST: db
      DB_USERNAME: ${MYSQL_USER}
      DB_PASSWORD: ${MYSQL_PASSWORD}
      DB_DATABASE: ${MYSQL_DATABASE}
      DB_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}
    depends_on:
      - db

  db:
    build:
      context: .
      dockerfile: docker/db/Dockerfile
    environment:
      MARIADB_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}
      MARIADB_DATABASE: ${MYSQL_DATABASE}
      MARIADB_USER: ${MYSQL_USER}
      MARIADB_PASSWORD: ${MYSQL_PASSWORD}
      MARIADB_DB_HOST: db
    volumes:
      - db_data:/var/lib/mysql

  nginx:
    build:
      context: ./docker/nginx
      dockerfile: Dockerfile
    ports:
      - "8080:80"
    depends_on:
      - node-service

volumes:
  db_data:
```

According to the latest documentation, the “MARIADB_ prefix” will be used instead of “MYSQL_” to pass the environment variables correctly.

Then the first time you can use:

```
docker-compose up --build
```

to create all the defined images and run the corresponding containers.

```
PS D:\Home\Desktop\docker-challenge2\challenge3> docker-compose up --build
2024/04/22 22:35:50 http2: server: error reading preface from client //./pipe/docker_engine: file has already been closed
[+] Building 0.5s (2/2)                                                                                    docker:default
 => [db internal] load build definition from Dockerfile                                                             0.0s
[+] Building 1.0s (9/9)                                                                                    docker:default
 => CACHED [db 2/2] COPY docker/db/init/init.sql /docker-entrypoint-initdb.d                                        0.0s
[+] Building 1.2s (25/25) FINISHED                                                                         docker:default
 => [db internal] load build definition from Dockerfile                                                             0.0s
 => => transferring dockerfile: 108B                                                                                0.0s
 => [db internal] load metadata for docker.io/library/mariadb:latest                                                0.5s
 => [db internal] load .dockerignore                                                                                0.0s
 => => transferring context: 2B                                                                                     0.0s
 => [db internal] load build context                                                                                0.0s
 => => transferring context: 957B                                                                                   0.0s
 => [db 1/2] FROM docker.io/library/mariadb:latest@sha256:78d0c3b8c39b47e91cc32df64ef1e26f797b54b1dd762c850e7350095ed4715f  0.0s
 => CACHED [db 2/2] COPY docker/db/init/init.sql /docker-entrypoint-initdb.d                                        0.0s
 => [db] exporting to image                                                                                         0.0s
 => => exporting layers                                                                                             0.0s
 => => writing image sha256:aaddb0f38ac2cab4ba93fc027378845dd926d61b1081c65b7dc329dde793640a                        0.0s
 => => naming to docker.io/library/challenge3-db                                                                    0.0s
 => [node-service internal] load build definition from Dockerfile                                                   0.0s
 => => transferring dockerfile: 471B                                                                                0.0s
 => [node-service internal] load metadata for docker.io/library/node:alpine                                         0.4s
 => [node-service internal] load .dockerignore                                                                      0.0s
 => => transferring context: 2B                                                                                     0.0s
 => [node-service 1/5] FROM docker.io/library/node:alpine@sha256:6d0f18a1c67dc218c4af50c21256616286a53c09e500fadf025b6d342e1c90  0.0s
 => [node-service internal] load build context                                                                      0.0s
 => => transferring context: 4.31kB                                                                                 0.0s
 => CACHED [node-service 2/5] WORKDIR /app                                                                          0.0s
 => CACHED [node-service 3/5] COPY docker/api/package*.json ./                                                      0.0s
 => CACHED [node-service 4/5] RUN npm install                                                                       0.0s
 => CACHED [node-service 5/5] COPY docker/api/* .                                                                   0.0s
 => [node-service] exporting to image                                                                               0.0s
 => => exporting layers                                                                                             0.0s
 => => writing image sha256:3ca979ec5f8a1382e8b39a6b767224d5448db00d62514e01d8f885e0cd87b26a                        0.0s
 => => naming to docker.io/library/challenge3-node-service                                                          0.0s
 => [nginx internal] load build definition from Dockerfile                                                          0.0s
 => => transferring dockerfile: 132B                                                                                0.0s
 => [nginx internal] load metadata for docker.io/library/nginx:latest                                               0.0s
 => [nginx internal] load .dockerignore                                                                             0.0s
 => => transferring context: 2B                                                                                     0.0s
 => [nginx 1/3] FROM docker.io/library/nginx:latest                                                                 0.0s
 => [nginx internal] load build context                                                                             0.0s
 => => transferring context: 31B                                                                                    0.0s
 => CACHED [nginx 2/3] RUN rm /etc/nginx/conf.d/default.conf                                                        0.0s
 => CACHED [nginx 3/3] COPY nginx.conf /etc/nginx/conf.d/default.conf                                               0.0s
 => [nginx] exporting to image                                                                                      0.0s
 => => exporting layers                                                                                             0.0s
 => => writing image sha256:dd5c8acf51f55603a02d3017a56caafa8b8fb5bf069ba20c2a8d42ab901849d0                        0.0s
 => => naming to docker.io/library/challenge3-nginx                                                                 0.0s
[+] Running 3/3
 ✓ Container challenge3-db-1           Recreated                                                                    0.0s
 ✓ Container challenge3-node-service-1 Recreated                                                                    0.0s
 ✓ Container challenge3-nginx-1        Recreated                                                                    0.0s
```

Later you can use:

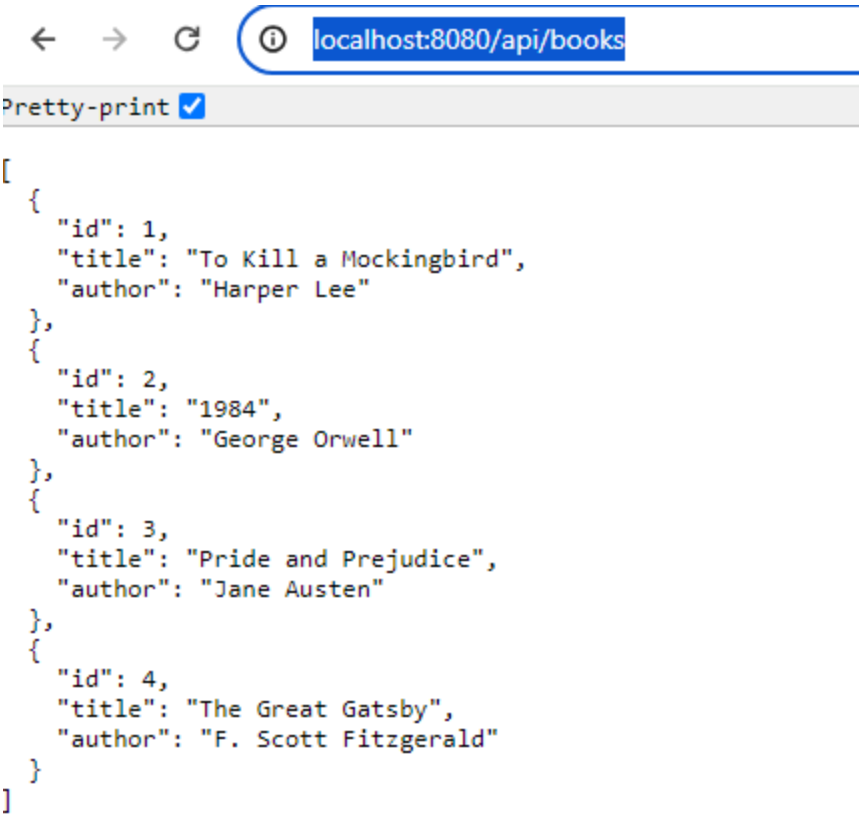
`docker-compose up / down`

to start and stop the relevant containers defined in docker-compose.yml.

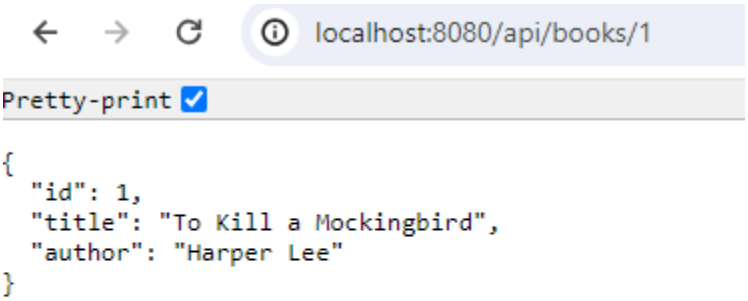
```
PS D:\Home\Desktop\docker-challenge2\challenge3> docker-compose down
[+] Running 4/4
 ✓ Container challenge3-nginx-1         Removed                                                                    0.4s
 ✓ Container challenge3-node-service-1  Removed                                                                   10.3s
 ✓ Container challenge3-db-1            Removed                                                                    0.5s
 ✓ Network challenge3_default           Removed                                                                    0.2s
```

The API is running:

http://localhost:8080/api/books/




http://localhost:8080/api/books/1



## docker-compose ps

```
PS D:\Home\Desktop\docker-challenge2\challenge3> docker-compose ps
NAME                          IMAGE                     COMMAND                  SERVICE        CREATED          STATUS          PORTS
challenge3-db-1               challenge3-db             "docker-entrypoint.s…"   db             7 seconds ago    Up 6 seconds    3306/tcp
challenge3-nginx-1            challenge3-nginx          "/docker-entrypoint.…"   nginx          7 seconds ago    Up 6 seconds    0.0.0.0:8080->80/tcp
challenge3-node-service-1     challenge3-node-service   "docker-entrypoint.s…"   node-service   7 seconds ago    Up 6 seconds    3000/tcp
PS D:\Home\Desktop\docker-challenge2\challenge3>
```

# Challenge 4 – Scaling up an application

## Goals

- Scale up the node-service to have 3 running instances.
- Understand the benefits of having multiple instances, such as increased availability and load distribution.

## Steps

Modify the Docker Compose Configuration

Everything is the same except that at the end of the node-service configuration:

```
deploy: replicas: 3

    replicas: 3 # This tells Docker Compose to start 3 instances of node-service
```

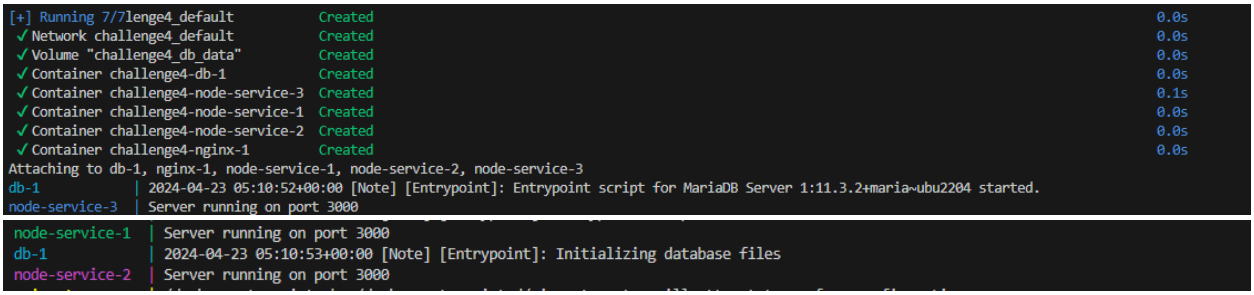
```yaml
  node-service:
    build:
      context: .
      dockerfile: docker/api/Dockerfile
    environment:
      DB_HOST: db
      DB_USERNAME: ${MYSQL_USER}
      DB_PASSWORD: ${MYSQL_PASSWORD}
      DB_DATABASE: ${MYSQL_DATABASE}
      DB_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}
    depends_on:
      - db
    deploy:
      replicas: 3  # This tells Docker Compose to start 3 instances of node-service
```

And then the same thing

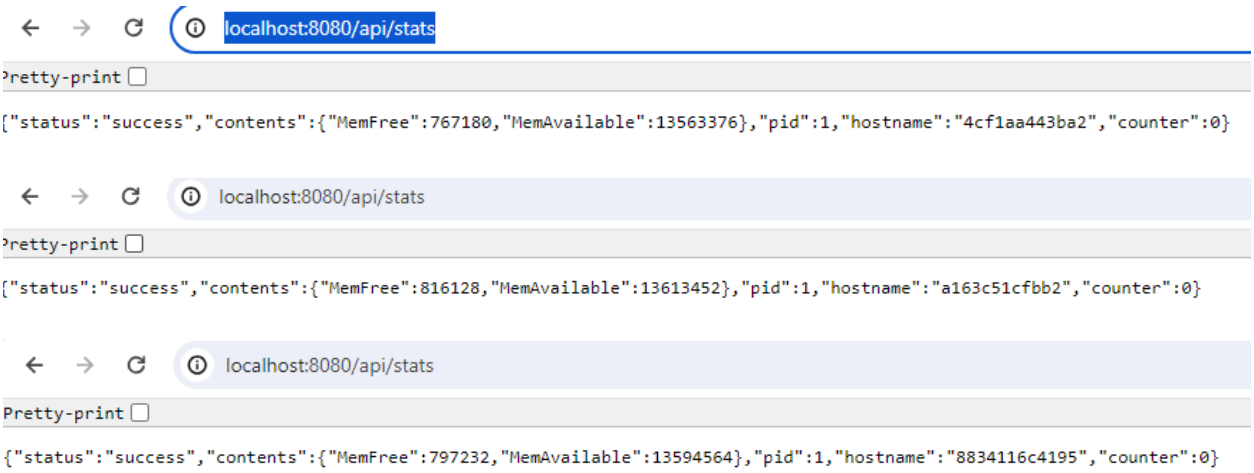
```
docker-compose up –build
```

```
PS D:\Home\Desktop\docker-challenge2\challenge4> docker-compose up --build
2024/04/22 23:10:50 http2: server: error reading preface from client //./pipe/docker_engine: file has already been closed
[+] Building 0.8s (3/3)                                                                                    docker:default
 => [db internal] load build definition from Dockerfile                                                             0.0s
[+] Building 1.8s (11/11)                                                                                  docker:default
 => => writing image sha256:97cc15b66e62ba61b1d7c6f69f5e8fc9de7815ecd5be4c6ac3c100cf2bbc5af0                        0.0s
[+] Building 1.9s (27/27) FINISHED                                                                         docker:default
```

When the container starts running, we see that 3 node-services have been started




Not surprisingly when we test the http://localhost:8080/api/stats :




we will see a different hostname, and in the docker logs we will see a different node-service responding with.

```
node-service-3  | {
node-service-3  |   status: 'success',
node-service-3  |   contents: { MemFree: 796324, MemAvailable: 13587564 },
node-service-3  |   pid: 1,
node-service-3  |   hostname: 'a163c51cfbb2',
node-service-3  |   counter: 0
node-service-3  | }
nginx-1         | 172.30.0.1 - - [23/Apr/2024:05:11:17 +0000] "GET /api/stats HTTP/1.1" 200 120 "-"
Gecko) Chrome/124.0.0.0 Safari/537.36" "-"
node-service-2  | {
node-service-2  |   status: 'success',
node-service-2  |   contents: { MemFree: 796324, MemAvailable: 13587564 },
node-service-2  |   pid: 1,
node-service-2  |   hostname: '4cf1aa443ba2',
node-service-2  |   counter: 0
node-service-2  | }
nginx-1         | 172.30.0.1 - - [23/Apr/2024:05:11:19 +0000] "GET /api/stats HTTP/1.1" 200 120 "-"
Gecko) Chrome/124.0.0.0 Safari/537.36" "-"
node-service-1  | {
node-service-1  |   status: 'success',
node-service-1  |   contents: { MemFree: 795848, MemAvailable: 13587088 },
node-service-1  |   pid: 1,
node-service-1  |   hostname: '8834116c4195',
node-service-1  |   counter: 0
node-service-1  | }
nginx-1         | 172.30.0.1 - - [23/Apr/2024:05:11:21 +0000] "GET /api/stats HTTP/1.1" 200 120 "-"
```

At the same time

```
docker-compose ps
```

will also see the 3 node-services running:

```
PS D:\Home\Desktop\docker-challenge2\challenge4> docker-compose ps
NAME                          IMAGE                         COMMAND                  SERVICE         CREATED          STATUS          PORTS
challenge4-db-1               challenge4-db                 "docker-entrypoint.s…"   db              7 minutes ago    Up 7 minutes    3306/tcp
challenge4-nginx-1            challenge4-nginx              "/docker-entrypoint.…"   nginx           7 minutes ago    Up 7 minutes    0.0.0.0:8080->80/tcp
challenge4-node-service-1     challenge4-node-service       "docker-entrypoint.s…"   node-service    7 minutes ago    Up 7 minutes    3000/tcp
challenge4-node-service-2     challenge4-node-service       "docker-entrypoint.s…"   node-service    7 minutes ago    Up 7 minutes    3000/tcp
challenge4-node-service-3     challenge4-node-service       "docker-entrypoint.s…"   node-service    7 minutes ago    Up 7 minutes    3000/tcp
```

Thanks for learning another new thing!