Assignment: Docker Challenge Part2 Report:

Submitted By: Darak Ajmani (Student ID: 000907926)

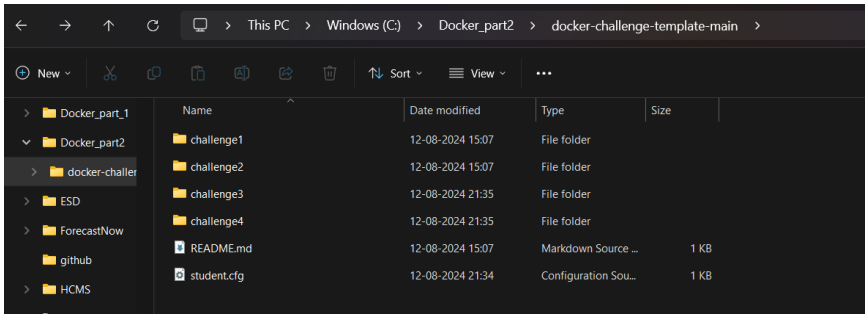
Repository Link: Darak376/Docker_part2 (github.com)

Challenge 3 – Setting up a Docker full-stack application

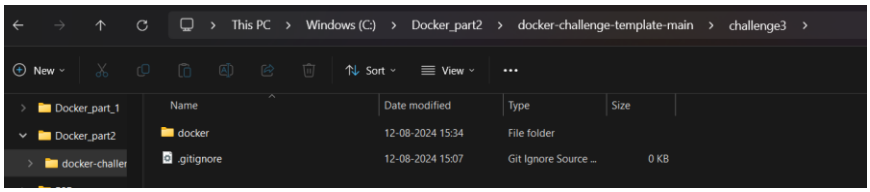
1. The first step to complete this challenge is to clone the repository or download the zip file from the repository link given below

   Github Link: https://github.com/eduluz1976/docker-challenge-template

2. Now, download the Challenge 3 zip folder from course homepage and extract it under the challenge 3 folder

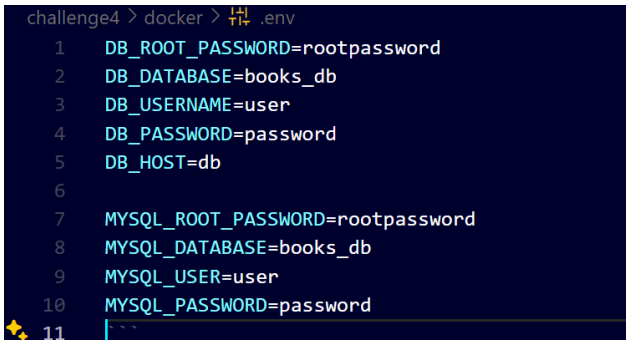


Img1: This is the image of reference for the first step



Img2: This is the image of reference for the second step

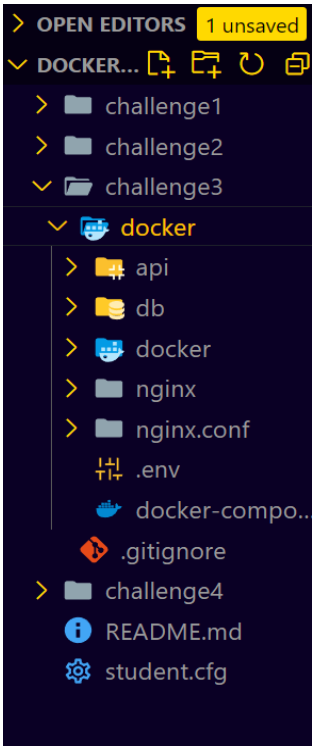
3. Create an env. file now inside the Challenge 3 folder and add the contents given below



4. Now add a docker-compose.yml and add the content given below

```yaml
version: "3.8"

services:
  db:
    image: mariadb:latest
    environment:
      MYSQL_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}
      MYSQL_DATABASE: ${MYSQL_DATABASE}
      MYSQL_USER: ${MYSQL_USER}
      MYSQL_PASSWORD: ${MYSQL_PASSWORD}
    volumes:
      - db_data:/var/lib/mysql
      - ./db/init/init.sql:/docker-entrypoint-initdb.d/init.sql
    networks:
      - app-network

  node-service:
    build:
      context: ./api
      dockerfile: Dockerfile
    environment:
      DB_HOST: db
      DB_DATABASE: ${MYSQL_DATABASE}
      DB_USERNAME: ${MYSQL_USER}
      DB_PASSWORD: ${MYSQL_PASSWORD}
    ports:
      - "3000:3000"
    depends_on:
      - db
    networks:
      - app-network

  nginx:
    image: nginx:latest
    ports:
      - "8080:80"
    volumes:
      - ./nginx/nginx.conf:/etc/nginx/conf.d/default.conf
    depends_on:
      - node-service
    networks:
      - app-network
```

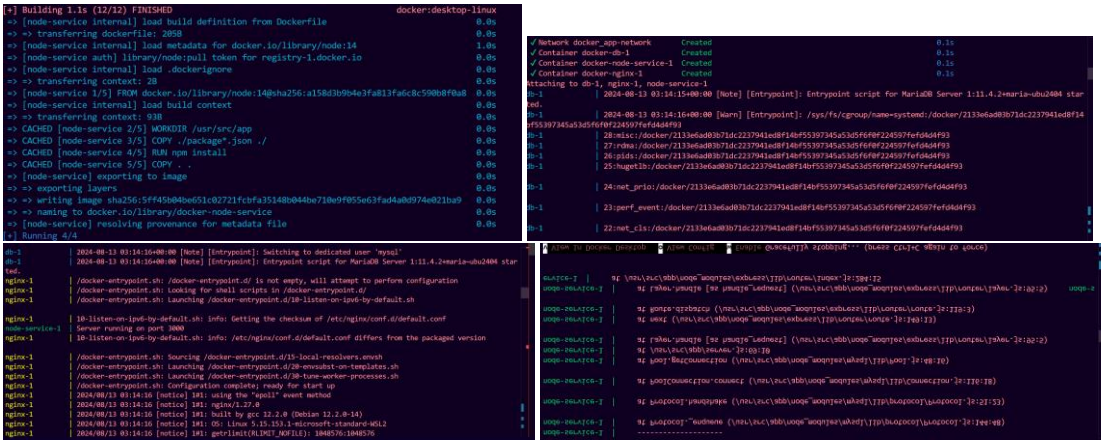
5. Below is the project directory for reference



6. Make sure this is the file format and the content is as follows too
7. Now, we will make the docker container for all three files but make sure you have the docker desktop installed and running
8. If not it will not work, so make sure it is working and installed
9. Now open the terminal and navigate to the Challenge 3 folder and then the docker folder

```
PS C:\Docker_part2\docker-challenge-template-main> cd challenge3
PS C:\Docker_part2\docker-challenge-template-main\challenge3> cd docker
PS C:\Docker_part2\docker-challenge-template-main\challenge3\docker>
```

10. Now run the “docker-compose –build” command to make the containers
11. Below is the screen that will display after running this command



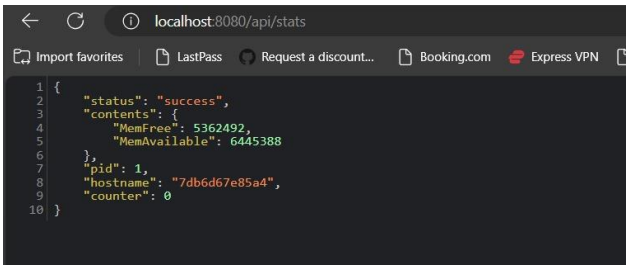
12. Now go to verify the results http://localhost:8080/api/books
13. And on this to verify specific results http://localhost:8080/api/books/1
14. Now run the “docker-compose ps“ command for the results given below

```
PS C:\Docker_part2\docker-challenge-template-main\challenge4\docker> docker-compose ps
>>
time="2024-08-12T21:37:08-06:00" level=warning msg="C:\\Docker_part2\\docker-challenge-template-main\\challenge4\\docker\\docker-
compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
NAME                    IMAGE                COMMAND                  SERVICE         CREATED          STATUS          PORTS
docker-db-1             mariadb:latest       "docker-entrypoint.s…"   db              11 seconds ago   Up 11 seconds   3306/tcp
docker-nginx-1          nginx:latest         "/docker-entrypoint.…"   nginx           11 seconds ago   Up 10 seconds   0.0.0.0:80
80->80/tcp
docker-node-service-1   docker-node-service   "docker-entrypoint.s…"   node-service   11 seconds ago   Up 10 seconds   0.0.0.0:30
00->3000/tcp
```

15. Now we will move to the final step, save all files and post this on github.
16. And post the repository link below :
    Darak376/Docker_part2 (github.com)

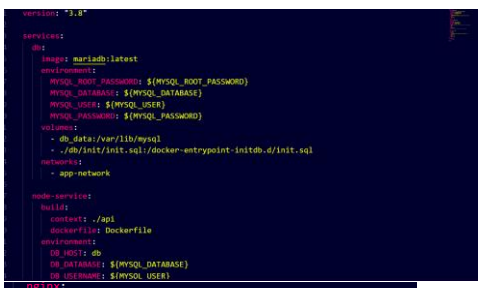
Challenge 4 – Scaling up a Docker full-stack application

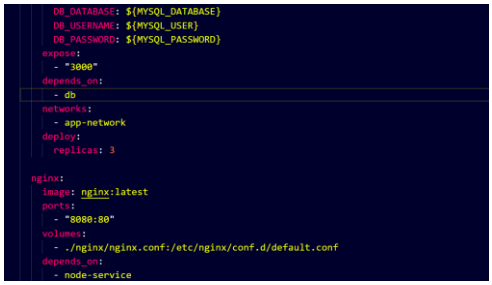
1. Now for this first copy all the files from the Challenge 3 folder.
2. Now paste it all in the Challenge 4 folder
3. First we will go to the localhost:8080/api/stats to check the status also record the output



4. For scaling it up we will start editing the docker-compose.yml for this challenge

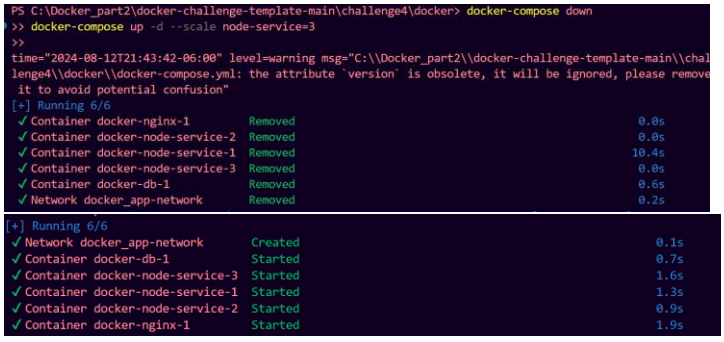
5. The edited content is given below





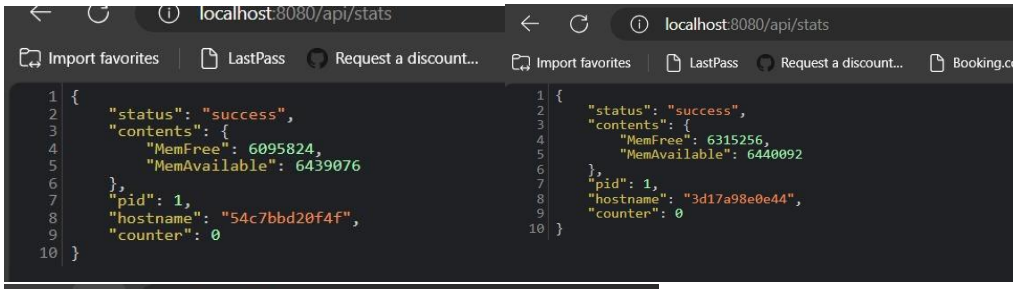
```yaml
  nginx:
    image: nginx:latest
    ports:
      - "8080:80"
    volumes:
      - ./nginx/nginx.conf:/etc/nginx/conf.d/default.conf
    depends_on:
      - node-service
    networks:
      - app-network

networks:
  app-network:
    driver: bridge

volumes:
  db_data:
```

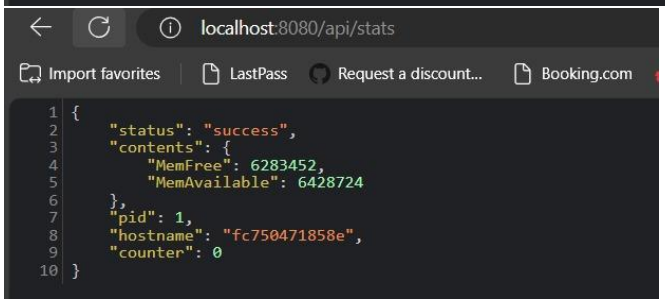
6. Now in the next step we will build the container again
7. For that first we have to remove the container and rebuild it



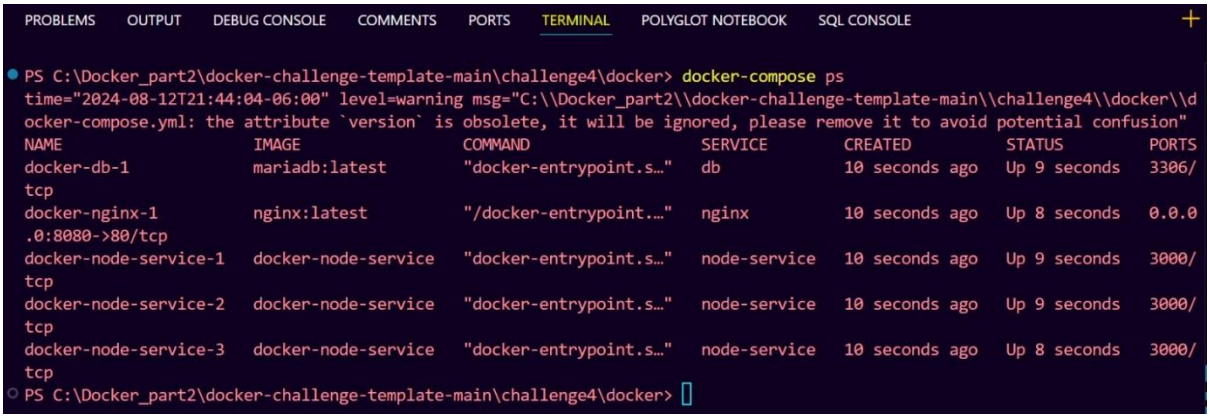
```
PS C:\Docker_part2\docker-challenge-template-main\challenge4\docker> docker-compose down
>> docker-compose up -d --scale node-service=3
>>
time="2024-08-12T21:43:42-06:00" level=warning msg="C:\\Docker_part2\\docker-challenge-template-main\\challenge4\\docker\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 6/6
 ✓ Container docker-nginx-1          Removed                    0.0s
 ✓ Container docker-node-service-2   Removed                    0.0s
 ✓ Container docker-node-service-1   Removed                   10.4s
 ✓ Container docker-node-service-3   Removed                    0.0s
 ✓ Container docker-db-1             Removed                    0.6s
 ✓ Network docker_app-network        Removed                    0.2s
[+] Running 6/6
 ✓ Network docker_app-network        Created                    0.1s
 ✓ Container docker-db-1             Started                    0.7s
 ✓ Container docker-node-service-3   Started                    1.6s
 ✓ Container docker-node-service-1   Started                    1.3s
 ✓ Container docker-node-service-2   Started                    0.9s
 ✓ Container docker-nginx-1          Started                    1.9s
```

8. Now go back and visit the same link visited in the previous steps and you will notice that the user name is different also run it around 1-3 times below is the example output





9. Now we will run the “docker-compose ps “ command again and record the results



10. You will notice that now it has 5 containers or processes running instead of 3 that were running in part3
11. Now again for the final step post this code on the repository created in the steps before
12. At, last after following every step you will get the desired results.
13. The link is given below
    Darak376/Docker_part2 (github.com)