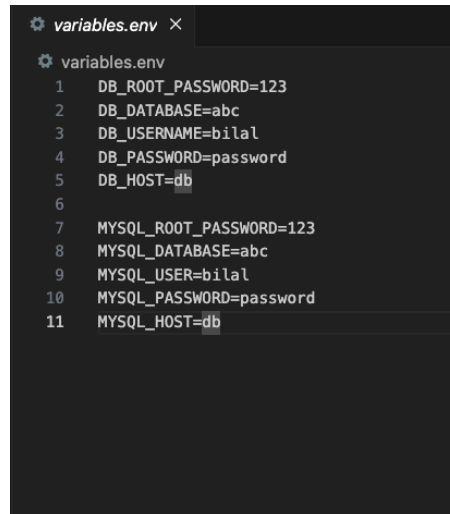


Challenge 3:

Step 1: Make a docker-compose.yml file in the root of the project directory

Step 2: Make a variables.env file at the same location as the .yml file from step 1.

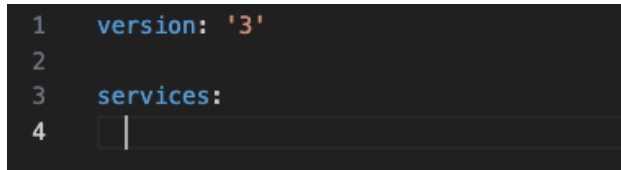
Step 3: Set environment variables for mariadb as shown below.



```
variables.env
1 DB_ROOT_PASSWORD=123
2 DB_DATABASE=abc
3 DB_USERNAME=bilal
4 DB_PASSWORD=password
5 DB_HOST=db
6
7 MYSQL_ROOT_PASSWORD=123
8 MYSQL_DATABASE=abc
9 MYSQL_USER=bilal
10 MYSQL_PASSWORD=password
11 MYSQL_HOST=db
```


Important to set DB_HOST and MYSQL_HOST to db, all other fields can be set to whatever you like but make sure DB and MYSQL respective variables are matching i.e. DB_ROOT_PASSWORD is the same as MYSQL_ROOT_PASSWORD etc.

Step 4: In the docker-compose.yml initialize the version to 3 and then begin initializing services like so



```
1 version: '3'
2
3 services:
4 |
```

Step 5: Next we want to create 3 services called nginx, node-service and db



```
1 version: '3'
2
3 services:
4   nginx:
5
6   node-service:
7
8
9   db:
10
11
```

Step 6: Build nginx service following the steps below

Image: nginx:latest

Restart: always

Ports:

- "8080:80"

Volumes:

- ./nginx:/etc/nginx/conf.d

Depends_on:

- node-service

```
docker-compose.yml
1  version: '3'
2
3  services:
4    nginx:
5      image: nginx:latest
6      restart: always
7      ports:
8        - "8080:80"
9      volumes:
10       - ./nginx:/etc/nginx/conf.d
11      depends_on:
12        - node-service
13
14  node-service:
15
16
17  db:
18
19
```

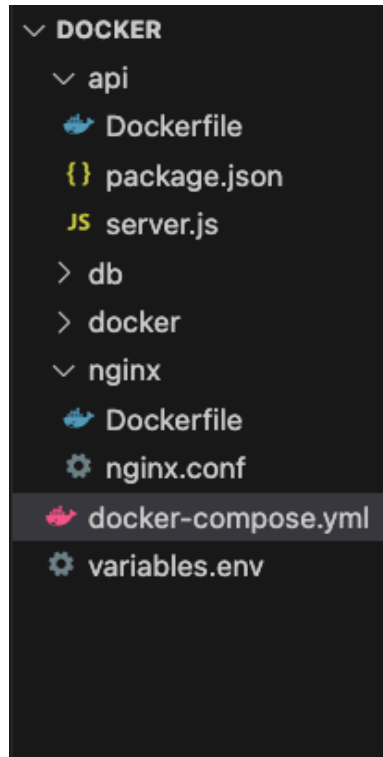
Step 7: Build node-service

```
docker-compose.yml
3  services:
4    nginx:
5      image: nginx:latest
6      restart: always
7      ports:
8        - "8080:80"
9      volumes:
10       - ./nginx:/etc/nginx/conf.d
11      depends_on:
12        - node-service
13
14    node-service:
15      build: ./api
16      restart: always
17      ports:
18        - "3000:3000"
19      env_file:
20        - ./variables.env
21
```

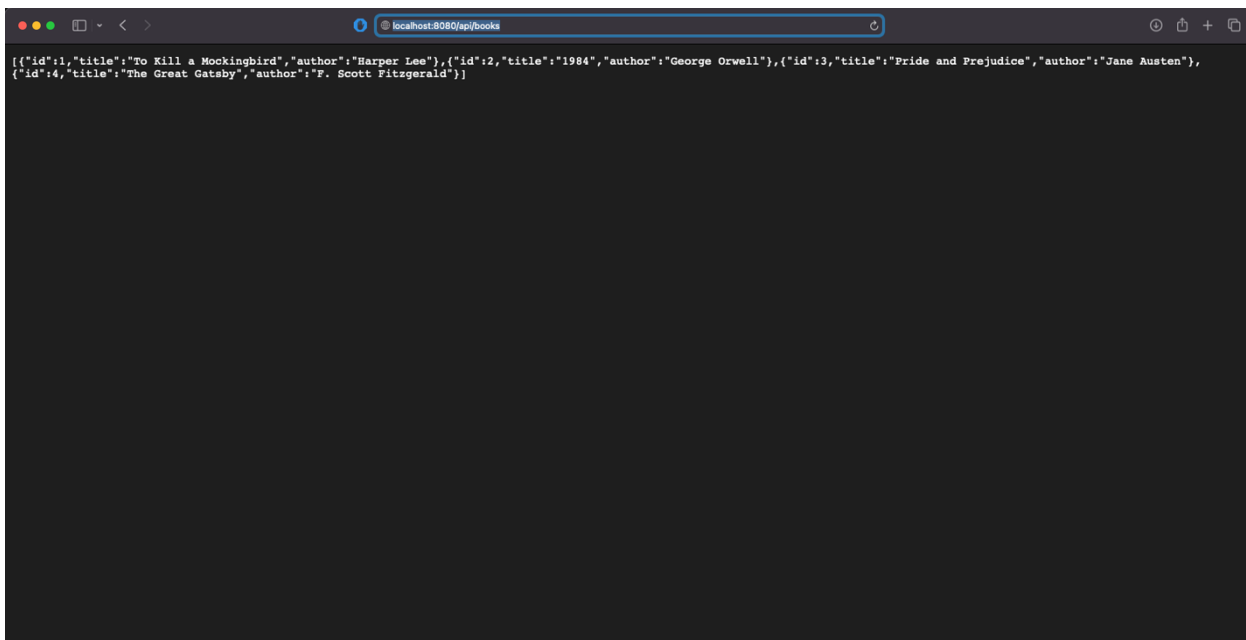
Step 8: Build db service

```
docker-compose.yml
3  services:
4    nginx:
5      image: nginx:latest
6      restart: always
7      ports:
8        - "8080:80"
9      volumes:
10       - ./nginx:/etc/nginx/conf.d
11      depends_on:
12        - node-service
13
14    node-service:
15      build: ./api
16      restart: always
17      ports:
18        - "3000:3000"
19      env_file:
20        - ./variables.env
21
22    db:
23      image: mariadb
24      env_file:
25        - ./variables.env
26      volumes:
27        - ./db/init:/docker-entrypoint-initdb.d
28
29
```

Step 9: It is important that the directory is like this



Here is the working api requests



```
localhost:8080/api/books/1
{"id":1,"title":"To Kill a Mockingbird","author":"Harper Lee"}
```

Challenge 4:

Simply a scale of 3 to node-service and comment out the ports like this

```
docker-compose.yml
1 version: '3'
2
3 services:
4   nginx:
5     image: nginx:latest
6     restart: always
7     ports:
8       - "8080:80"
9     volumes:
10      - ./nginx:/etc/nginx/conf.d
11     depends_on:
12      - node-service
13
14   node-service:
15     build: ./api
16     restart: always
17     # ports:
18     #   - "3000:3000"
19     env_file:
20      - ./variables.env
21     scale: 3
22
23   db:
24     image: mariadb
25     env_file:
26      - ./variables.env
27     volumes:
28      - ./db/init:/docker-entrypoint-initdb.d
29
30
```

```
{"status":"success","contents":{"MemFree":6505268,"MemAvailable":6969644},"pid":1,"hostname":"2dfb04ffa83a","counter":0}
```

```
{"status":"success","contents":{"MemFree":6508568,"MemAvailable":6973932},"pid":1,"hostname":"b5a4cc985063","counter":0}
```

```
{"status":"success","contents":{"MemFree":6523420,"MemAvailable":6988812},"pid":1,"hostname":"ebf11335cc9a","counter":0}
```