

Journal

Challenge 3

Steps taken

1. Downloaded zip file and opened on VScode.
2. Opened Docker Desktop and made sure engine was running.
3. Made a .env file for my database variables, placed inside challenge3 directory.

```
DB_HOST=db

DB_USERNAME=root

DB_PASSWORD=rootpassword

DB_DATABASE=books

MYSQL_ROOT_PASSWORD=rootpassword

MYSQL_DATABASE=books

MYSQL_USER=root

MYSQL_PASSWORD=rootpassword

MYSQL_HOST=db
```

4. In root directory still, I created a docker-compose.yml file which helps me build the docker images. There are 3 services required for the application, Nginx Service, Node.js Service and the MariaDB Database Service. Each service is isolated in its own container for security and to make it easier to manage. The service are started in the correct order which is important for applications operation. Here is the code for this file:

```
version: '3.8'
```

services:

nginx:

build: ./docker/nginx

ports:

- "8080:80"

depends_on:

- node-service

node-service:

build: ./docker/api

environment:

DB_HOST: db

DB_USERNAME: root

DB_PASSWORD: rootpassword

DB_DATABASE: books

depends_on:

- db

db:

build: ./docker/db

```
environment:
```

```
    MYSQL_ROOT_PASSWORD: rootpassword
```

```
    MYSQL_DATABASE: books
```

```
ports:
```

```
    - "3306:3306"
```

```
volumes:
```

```
    - db-data:/var/lib/mysql
```

```
volumes:
```

```
    db-data:
```

5. Navigate to project root directory in my cmd.

```
cd C:\Users\Seth\Downloads\challenge3
```

6. Then I built the docker images using:

```
docker-compose build
```

```

C:\Users\Seth> docker-compose up
[+] Building 0.0s (0/0) docker:default
2024/04/23 18:50:05 http2: server: error reading preface from client //./pipe/docker_engine: file has already been close[+] Building 0.9s (8/8)
it
-> [db internal] load build definition from Dockerfile
-> [db internal] load build definition from Dockerfile
[+] Building 1.3s (9/9)
-> [db internal] load build definition from Dockerfile
[+] Building 7.2s (28/28)
-> exporting layers
-> exporting layers
[+] Building 8.1s (29/29) FINISHED
-> [db internal] load build definition from Dockerfile
-> transferring dockerfile: 1538
-> [db internal] load metadata for docker.io/library/mariadb:latest
-> [db auth] library/mariadb:pull token for registry-1.docker.io
-> [db internal] load .dockerignore
-> transferring context: 28
-> [db internal] load build context
-> CACHED [db 2/3] RUN apt-get update && apt-get install -y mysql-client
-> [db 3/3] COPY init/init.sql /docker-entrypoint-initdb.d/
-> [db] exporting to image
-> writing image sha256:d4cabdc62ba9f7b4a85829c35e0fc25be37a32c709132e8038a928b6105cac5
-> naming to docker.io/library/challenge3-db
-> [node-service internal] load build definition from Dockerfile
-> transferring dockerfile: 4198
-> [node-service internal] load metadata for docker.io/library/node:alpine
-> [node-service auth] library/node:pull token for registry-1.docker.io
-> [node-service internal] load .dockerignore
-> transferring context: 28
-> [node-service 1/5] FROM docker.io/library/node:alpine@sha256:db8772d9f5796ac4e8c47508038c413ea1478da010568a2e48672f19a8b80
-> resolve docker.io/library/node:alpine@sha256:db8772d9f5796ac4e8c47508038c413ea1478da010568a2e48672f19a8b80
-> sha256:9914d5ae0de40864c3e75d3b6ef6d0f1c0c0876413575f445f8ac402 7.21kB / 7.21kB
-> sha256:d3f3e43882a8a0b598756f07dd6b7e0398fc98cb1d44589a26bccdad61235f 1.38MB / 1.38MB
-> sha256:af0a15da4cb2c7e3220793e3d182d0f9e5ed1a0ae483397048a1c657e4e139f 457B / 457B
-> sha256:db8772d9f5796ac4e8c47508038c413ea1478da010568a2e48672f19a8b80c02 1.43kB / 1.43kB
-> sha256:c5eaf5100a9a12e460384cd0b031701546080a42554346012f1ddc2b085c7 1.16kB / 1.16kB
-> extracting sha256:d3f3e43882a8a0b598756f07dd6b7e0398fc98cb1d44589a26bccdad61235f
-> extracting sha256:af0a15da4cb2c7e3220793e3d182d0f9e5ed1a0ae483397048a1c657e4e139f
-> [node-service internal] load build context
-> transferring context: 938
-> [node-service 2/5] WORKDIR /app
-> [node-service 3/5] COPY package*.json ./
-> [node-service 4/5] RUN npm install
-> [node-service 5/5] COPY . .
-> [node-service] exporting to image
-> exporting layers
-> writing image sha256:a4719357f6990225c51dd40c1527a8b819f48ee2f7997b46cb90452c9a697d6a
-> naming to docker.io/library/challenge3-node-service
-> [nginx internal] load build definition from Dockerfile
-> transferring dockerfile: 1218
-> [nginx internal] load metadata for docker.io/library/nginx:latest
-> [nginx auth] library/nginx:pull token for registry-1.docker.io
-> [nginx internal] load .dockerignore
-> transferring context: 28
-> [nginx 1/3] FROM docker.io/library/nginx:latest@sha256:0463a96ac74b84a8a1b27f3d1f4ae5d1a70ea823219394e131f5bf
-> [nginx internal] load build context
-> transferring context: 318
-> CACHED [nginx 2/3] RUN rm -f /etc/nginx/conf.d/default.conf
-> CACHED [nginx 3/3] COPY nginx.conf /etc/nginx/conf.d/default.conf
-> [nginx] exporting to image
-> exporting layers
-> writing image sha256:7cfa986a4c90efcd4ed37f1158bc1957737acd2be414efb525cf168700b1b12e88

```

7. To start docker compose in detached mode, I ran:

```
docker-compose up -d
```

```

C:\Users\Seth\Downloads\challenge3>docker-compose up -d
[+] Running 3/3
  Container challenge3-db-1           Started                                0.6s
  Container challenge3-node-service-1 Started                                10.4s
  Container challenge3-nginx-1        Started                                0.6s

```

8. To see the 3 services I ran:

```
docker-compose ps
```

```

C:\Users\Seth\Downloads\challenge3>docker-compose ps
NAME                IMAGE                COMMAND                  SERVICE    CREATED      STATUS      PORTS
challenge3-db-1     challenge3-db        "docker-entrypoint.s..." db         39 seconds ago Up 27 seconds 0.0.0.0:3306->3306/tcp
challenge3-nginx-1  challenge3-nginx    "/docker-entrypoint..." nginx      28 seconds ago Up 26 seconds 0.0.0.0:8080->80/tcp
challenge3-node-service-1 challenge3-node-service "docker-entrypoint.s..." node-service 39 seconds ago Up 27 seconds 3000/tcp

```

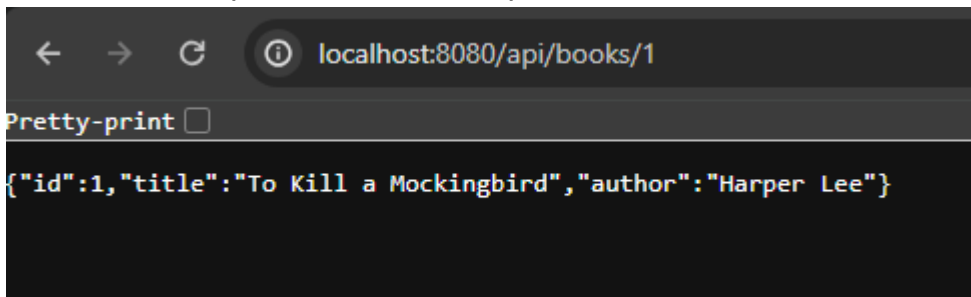
9. I then went to <http://localhost:8080/api/books> to check to see the json format for the books.

```

localhost:8080/api/books
Pretty-print
[{"id":1,"title":"To Kill a Mockingbird","author":"Harper Lee"}, {"id":2,"title":"1984","author":"George Orwell"}, {"id":3,"title":"Pride and Prejudice","author":"Jane Austen"}, {"id":4,"title":"The Great Gatsby","author":"F. Scott Fitzgerald"}]

```

10. Checked the <http://localhost:8080/api/books/1> to see the first book.



Challenge 4

Since the docker-compose.yml file is modular and scalable, the only thing to do for this challenge was to run one command.

```
docker-compose up -d --scale node-service=3
```

```
C:\Users\Seth\Downloads\challenge3>docker-compose down
time="2024-04-22T21:22:14-06:00" level=warning msg="C:\\Users\\Seth\\Downloads\\challenge3\\docker-compose.yml: `version` is obsolete"
[+] Running 6/6
  Container challenge3-nginx-1      Removed          0.0s
  Container challenge3-node-service-2 Removed          10.4s
  Container challenge3-node-service-3 Removed          0.0s
  Container challenge3-node-service-1 Removed          0.0s
  Container challenge3-db-1         Removed          0.7s
  Network challenge3_default        Removed          0.2s

C:\Users\Seth\Downloads\challenge3>docker-compose up -d --scale node-service=3
time="2024-04-22T21:22:26-06:00" level=warning msg="C:\\Users\\Seth\\Downloads\\challenge3\\docker-compose.yml: `version` is obsolete"
[+] Running 6/6
  Network challenge3_default        Created          0.0s
  Container challenge3-db-1         Started          0.1s
  Container challenge3-node-service-3 Started          0.1s
  Container challenge3-node-service-2 Started          0.1s
  Container challenge3-node-service-1 Started          0.1s
  Container challenge3-nginx-1      Started          0.1s

C:\Users\Seth\Downloads\challenge3>docker-compose ps
time="2024-04-22T21:22:31-06:00" level=warning msg="C:\\Users\\Seth\\Downloads\\challenge3\\docker-compose.yml: `version` is obsolete"
NAME                                IMAGE                                COMMAND                                SERVICE    CREATED     STATUS     PORTS
challenge3-db-1                     challenge3-db                        "docker-entrypoint.s..."          db         5 seconds ago Up 4 seconds 0.0.0.0:3306->3306/tcp
challenge3-nginx-1                  challenge3-nginx                    "/docker-entrypoint.s..."          nginx      5 seconds ago Up 3 seconds 0.0.0.0:8080->80/tcp
challenge3-node-service-1            challenge3-node-service              "docker-entrypoint.s..."          node-service 5 seconds ago Up 4 seconds 3000/tcp
challenge3-node-service-2            challenge3-node-service              "docker-entrypoint.s..."          node-service 5 seconds ago Up 4 seconds 3000/tcp
challenge3-node-service-3            challenge3-node-service              "docker-entrypoint.s..."          node-service 5 seconds ago Up 4 seconds 3000/tcp

C:\Users\Seth\Downloads\challenge3>
```

Now I just navigate to <http://localhost:8080/api/stats> and look at the host names, I visited this site 3 times and received 3 different names.

**

First Check: ("hostname":"0272f26780d0")

Second Check: ("hostname":"3b0b7e6fb795")

Third Check: ("hostname":"185c9089fa8b")

**