

## Assignment – 2

### Tasks:

Use Docker Compose to manage a multi-service application.

1. Define a multi-service application consisting of at least three services (e.g., a web app, a database, and a caching service) in a `docker-compose.yml` file.
2. Configure the services with appropriate environment variables and networking settings.
3. Build and run the Docker Compose application.
4. Test the connectivity between the services and the functionality of the application.

### Steps:

1. Here I have taken a Node-JS application which uses MongoDB as database.
2. I added Dockerfile in the root directory of folder.

```
FROM node:20.11.1
WORKDIR /usr/src/app
COPY package*.json .
RUN npm install
COPY . .
EXPOSE 8000
ENV MONGODB_URI=mongodb://mongo:27017
CMD ["node", "app.js"]
```

3. Now I added docker-compose.yml file into the root directory.

```
version: '3.8'
services:
  # Node-Js web app
  web:
    build: ./BookIT
```

ports:

- "8000:8000"

environment:

- MONGODB\_URI=mongodb://mongo:27017

- DB\_NAME=my\_db

depends\_on:

- mongo

mongo:

container\_name: db\_container

image: mongo:latest

restart: always

ports:

- "27017:27017"

volumes:

- mongo\_data:/data/db

volumes:

mongo\_data:















4. Now use command “docker compose build” to build the docker image.

```
PS C:\Users\venis\Downloads\BookIt\BookIT> docker compose build
[+] Building 0.0s (0/0)  docker:default
2024/02/27 20:21:37 http2: server: error reading preface from client //./pipe/docker_engine[+] Building 7.6s (11/11) FINISHED
docker:default
=> [web internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 495B 0.0s
=> [web internal] load metadata for docker.io/library/node:20.11.1 4.7s
=> [web auth] library/node:pull token for registry-1.docker.io 0.0s
=> [web internal] load .dockerignore 0.0s
=> => transferring context: 56B 0.0s
=> [web 1/5] FROM docker.io/library/node:20.11.1@sha256:f3299f16246c71ab8b304d6745 0.0s
=> [web internal] load build context 1.3s
=> => transferring context: 45.95MB 1.3s
=> CACHED [web 2/5] WORKDIR /usr/src/app 0.0s
=> CACHED [web 3/5] COPY package*.json . 0.0s
=> CACHED [web 4/5] RUN npm install 0.0s
=> [web 5/5] COPY . . 1.2s
=> [web] exporting to image 0.2s
=> => exporting layers 0.2s
=> => writing image sha256:4d2ba2b84603b4ec9605c93945c8f9697041c8142e3d8b871874841 0.0s
=> => naming to docker.io/library/bookit-web 0.0s
PS C:\Users\venis\Downloads\BookIt\BookIT>
```

5. Now use command “docker compose up” to run that docker image.

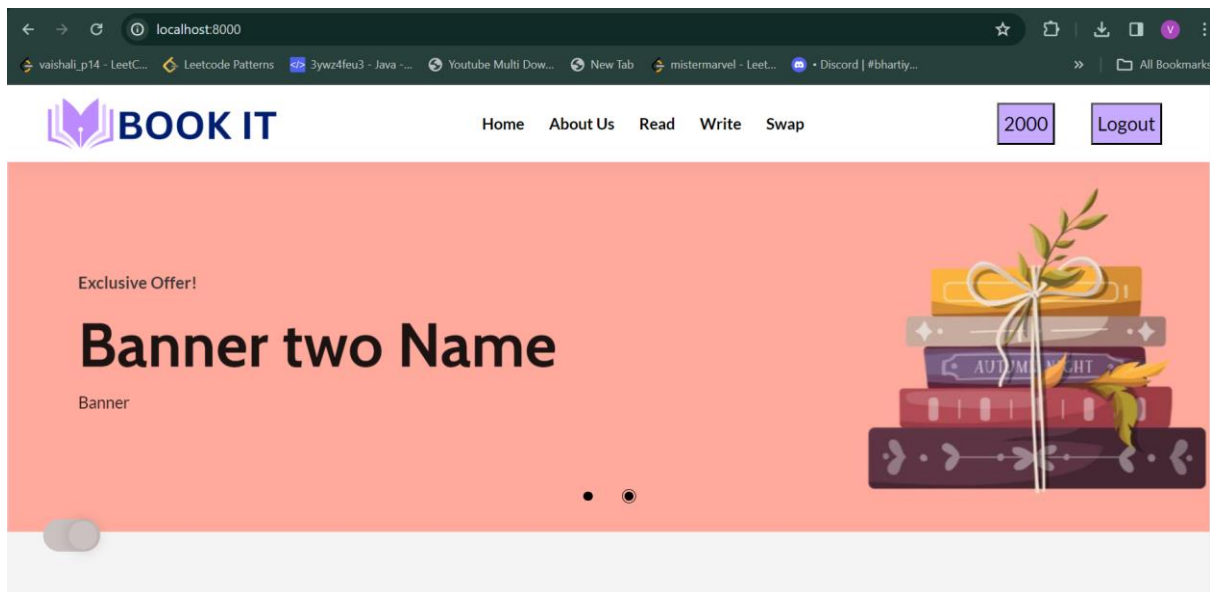
```
PS C:\Users\venis\Downloads\BookIT\BookIT> docker compose up
[+] Running 3/3
 ✓ Network bookit_default Created                                0.1s
 ✓ Container db_container Created                               0.1s
 ✓ Container bookit-web-1 Created                               0.1s
Attaching to web-1, db_container
db_container | {"t":{"$date":"2024-02-27T14:54:02.268+00:00"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"main","msg":"Initializ
ed wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":21},"incomingInternalClient":{"
minWireVersion":0,"maxWireVersion":21},"outgoing":{"minWireVersion":6,"maxWireVersion":21},"isInternalClient":true}}}
db_container | {"t":{"$date":"2024-02-27T14:54:02.271+00:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"main","msg":"Automatic
ally disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
db_container | {"t":{"$date":"2024-02-27T14:54:02.274+00:00"},"s":"I", "c":"NETWORK", "id":4648601, "ctx":"main","msg":"Implicit
TCP FastOpen unavailable. If TCP FastOpen is required, set tcpFastOpenServer, tcpFastOpenClient, and tcpFastOpenQueueSize."}
db_container | {"t":{"$date":"2024-02-27T14:54:02.280+00:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"main","msg":"Successfu
lly registered PrimaryOnlyService","attr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDonors"}}
db_container | {"t":{"$date":"2024-02-27T14:54:02.280+00:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"main","msg":"Successfu
lly registered PrimaryOnlyService","attr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigrationRecipients
"}}
db_container | {"t":{"$date":"2024-02-27T14:54:02.280+00:00"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"main","msg":"Multi thr
eading initialized"}
db_container | {"t":{"$date":"2024-02-27T14:54:02.281+00:00"},"s":"I", "c":"TENANT_M", "id":7091600, "ctx":"main","msg":"Starting
TenantMigrationAccessBlockerRegistry"}
db_container | {"t":{"$date":"2024-02-27T14:54:02.281+00:00"},"s":"I", "c":"CONTROL", "id":4615611, "ctx":"initandlisten","msg":"
MongoDB starting","attr":{"pid":1,"port":27017,"dbPath":"/data/db","architecture":"64-bit","host":"e7349721b79d"}}
db_container | {"t":{"$date":"2024-02-27T14:54:02.281+00:00"},"s":"I", "c":"CONTROL", "id":23403, "ctx":"initandlisten","msg":"
Build Info", "attr":{"buildInfo":{"version":"7.0.5","gitVersion":"7809d71e84e314b497f282ea8aa06d7ded3eb205","opensslVersion":"OpenSSL
3.0.2 15 Mar 2022","modules":[],"allocator":"tcmalloc","environment":{"distmod":"ubuntu2204","distarch":"x86_64","target_arch":"x86
64"}}}}
```

6. Here in docker desktop we can see that our container is running.

<input type="checkbox"/>	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	 bookit		Running (2/2)	0.58%		15 seconds ago	  
<input type="checkbox"/>	 web-1	bookit-web	Running	0%	8000:8000 	15 seconds ago	  
<input type="checkbox"/>	 db_conta	mongo:latest	Running	0.58%	27017:27017 	16 seconds ago	  

Showing 6 items

7. Now go to “localhost:8000” and here we can see that our application is running.



**BOOK IT**

Sign In

START FOR FREE

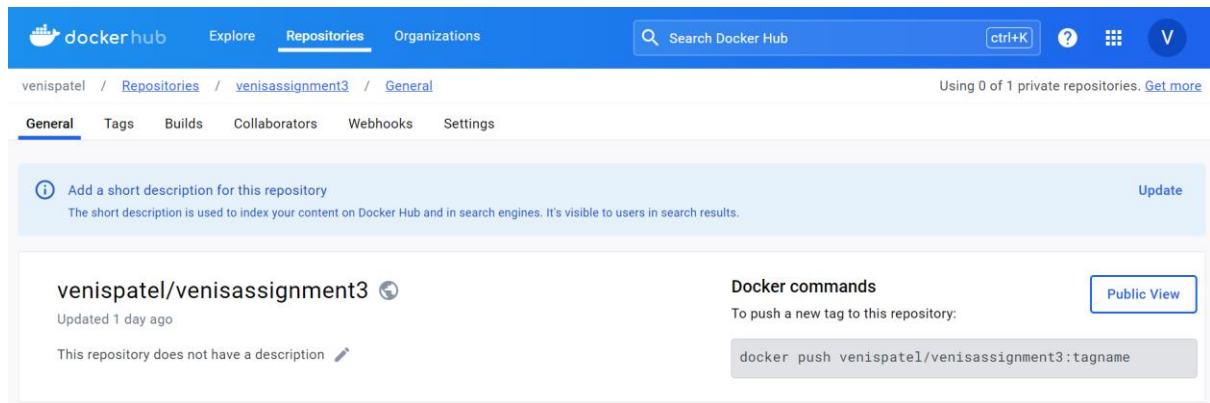
Create an account

Sign Up

Sign up with Google

Already have an account ? SIGNIN

8. Now push it into docker hub using command: docker push



9. Now to pull this docker image , run following command :

```
docker pull venispatel/venisassignment3
```

```
docker run -d -p 8000:8000 venispatel/venisassignment3
```