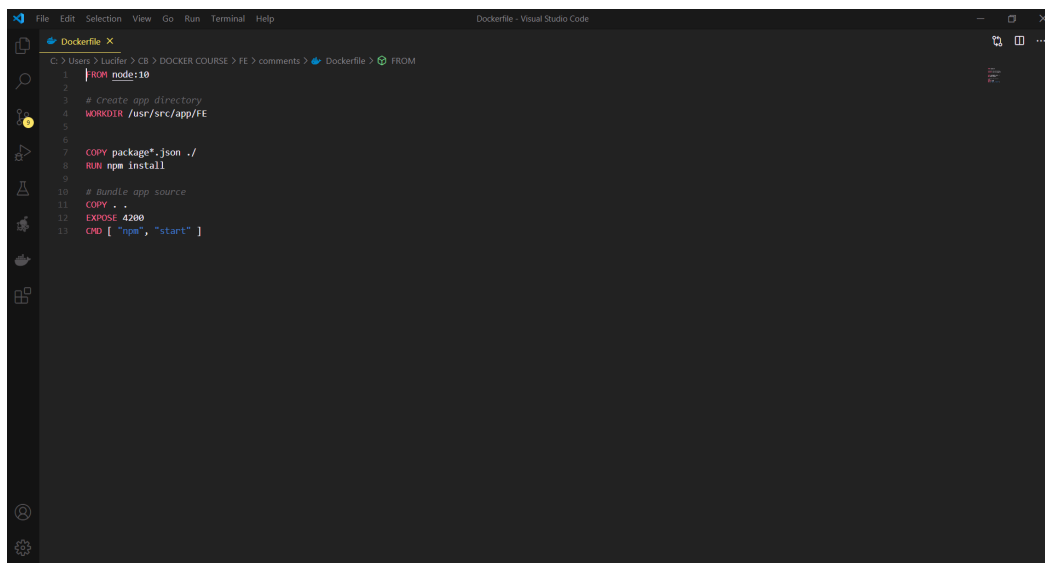


For this Docker 201 Project I have created a simple MEAN Application which takes the Comment and Username about a movie as Input and stores it in a MongoDB collection. On Application Startup, All the existing comments from the database are fetched and displayed.

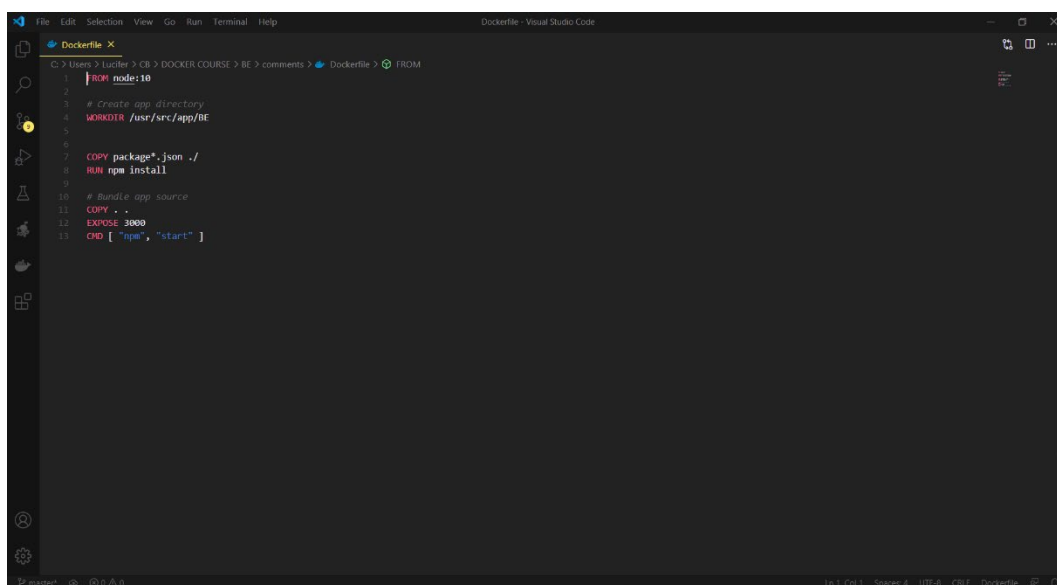
- The Application is containerized into 3 containers. One for Angular Front-End. One for NodeJS Back-End and one for MongoDB Database.
- The DockerFile to build the Front-End and Back-End images from code is shown below

Front-End DockerFile (Located in “DOCKER COURSE/FE/comments”)



```
1 FROM node:10
2
3 # Create app directory
4 WORKDIR /usr/src/app/FE
5
6
7 COPY package*.json ./
8 RUN npm install
9
10 # Bundle app source
11 COPY . .
12 EXPOSE 4200
13 CMD [ "npm", "start" ]
```

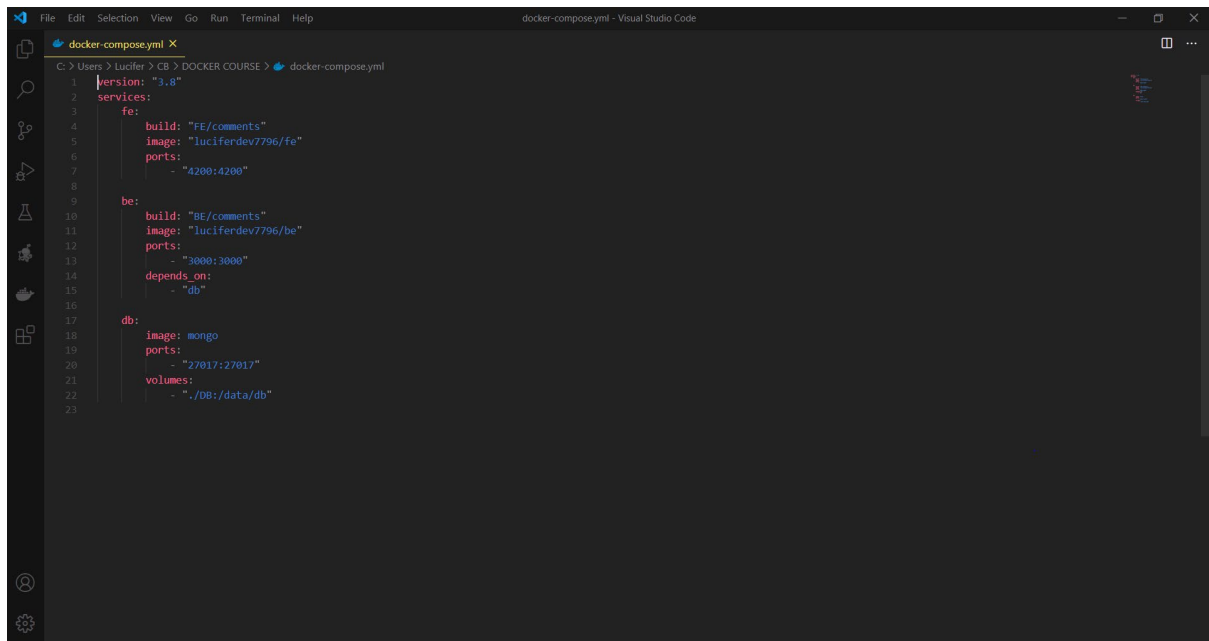
Back-End DockerFile (Located in “DOCKER COURSE/BE/comments”)



```
1 FROM node:10
2
3 # Create app directory
4 WORKDIR /usr/src/app/BE
5
6
7 COPY package*.json ./
8 RUN npm install
9
10 # Bundle app source
11 COPY . .
12 EXPOSE 3000
13 CMD [ "npm", "start" ]
```

- Both Folders contain a “.dockerignore” to avoid adding the node_modules folder to the image.

Shown below is the docker-compose file written to build, run and start server for the 3 containers.



```
1 version: "3.8"
2 services:
3   fe:
4     build: "FE/comments"
5     image: "luciferdev7796/fe"
6     ports:
7       - "4200:4200"
8
9   be:
10    build: "BE/comments"
11    image: "luciferdev7796/be"
12    ports:
13      - "3000:3000"
14    depends_on:
15      - "db"
16
17   db:
18     image: mongo
19     ports:
20       - "27017:27017"
21     volumes:
22       - ".:/DB:/data/db"
```

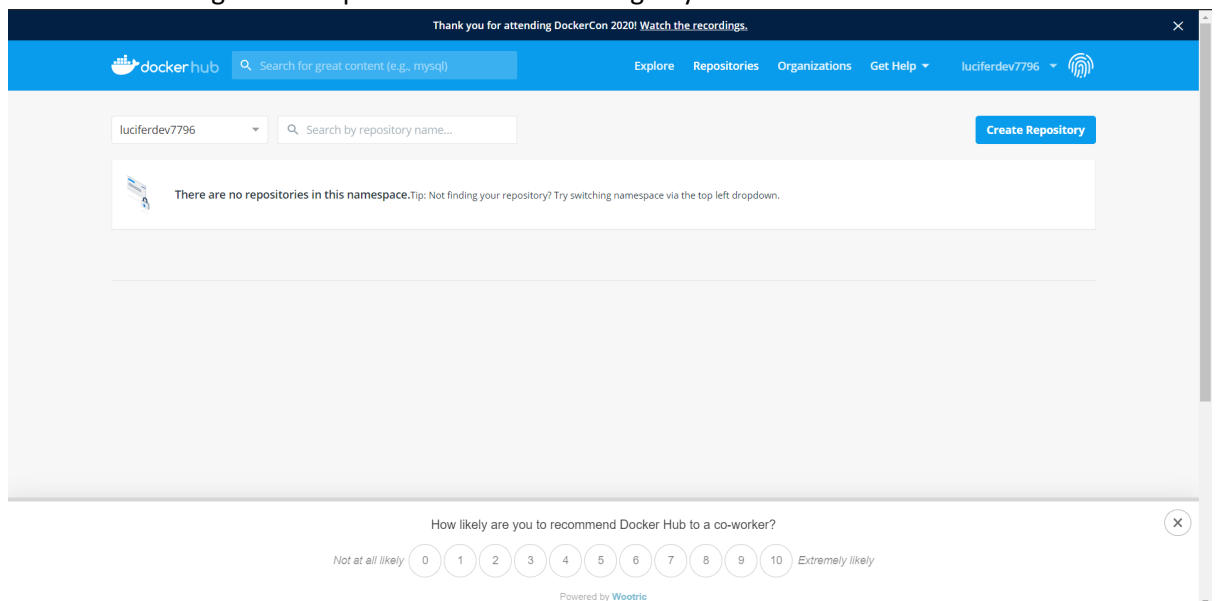
As seen above there are 3 services, one each for Front-End, Back-End and Database.

Execution:-

1. Initially there are no containers or images on the machine. if there are, they can be removed using
“docker rm <container name>” to remove container and
“docker rmi <image name>” to remove image respectively.

```
C:\Windows\System32\cmd.exe
C:\Users\Lucifer\CB\DOCKER COURSE>docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
C:\Users\Lucifer\CB\DOCKER COURSE>
C:\Users\Lucifer\CB\DOCKER COURSE>
C:\Users\Lucifer\CB\DOCKER COURSE>
C:\Users\Lucifer\CB\DOCKER COURSE>
C:\Users\Lucifer\CB\DOCKER COURSE>
C:\Users\Lucifer\CB\DOCKER COURSE>
C:\Users\Lucifer\CB\DOCKER COURSE>
C:\Users\Lucifer\CB\DOCKER COURSE>
C:\Users\Lucifer\CB\DOCKER COURSE>
C:\Users\Lucifer\CB\DOCKER COURSE>docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
C:\Users\Lucifer\CB\DOCKER COURSE>
```

2. There are no images on our personal Docker Hub Registry either.



3. Now we will open a cmd where the docker-compose file is present and run the command

“docker-compose up -d”

This will build all the images and spin up containers for the same and start the servers of the applications for us.

Output:-

```
C:\Users\Lucifer\COURSEDocker> docker-compose up -d
Building fe
Step 1/7 : FROM node:10
10: Pulling from library/node
ef9f8191915: pull complete
ed4e6ea23d1: pull complete
c2851092c538: Pull complete
8af58c44c0b: pull complete
16369df421a: pull complete
c23295f148ff: Pull complete
6c7d46bb1290: pull complete
dc317662d73: Pull complete
sha256:b704009: pull complete
Digest: sha256:f9ae09bdacba78001920b5dbf0cc78d4cf427bd104918912cb5b
Status: Downloaded newer image for node:10 --> c2b71092d62
Step 2/7 : WORKDIR /usr/src/app/fe
--> Running in bdec4552954
Removing intermediate container bf9e94954094
--> 2dc4dd6deaa6
Step 3/7 : COPY package*.json .
--> 3e3d1bf7309
Step 4/7 : RUN npm install
--> Running in 93064c3225d
C:\Users\Lucifer\COURSEDocker> docker-compose up -d
core-j@0.6.4 postinstall /usr/src/app/FE/node_modules/core-js
node <"require".postinstall"> catch(e){}
Thank you for using core-js (<https://github.com/zloirock/core.js> for polyfilling JavaScript standard library)
the project needs your help! please consider supporting of core-js on Open Collective or Patreon:
<https://opencollective.com/core-js>
<https://www.patreon.com/zloirock>
Also, the author of core-js (<https://github.com/zloirock>) is looking for a good job :)
Bungular/C1190.0.7 postinstall /usr/src/app/FE/node_modules/bungular/c11
node ./bin/postinstall/script.js
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.3 (node_modules/watchpack-dev-server/node_modules/fsevents):
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: @tjow/dspifmduicy: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":"x64"} (current: {"os":"linux","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.3 (node_modules/watchpack-chokidar/node_modules/fsevents):
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: @tjow/dspifmduicy: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":"x64"} (current: {"os":"linux","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.3 (node_modules/fsevents):
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: @tjow/dspifmduicy: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":"x64"} (current: {"os":"linux","arch":"x64"})
added 1363 packages from 1119 contributors and audited 1368 packages in 31.61s
37 packages are looking for funding
run 'npm fund' for details
found 3 low severity vulnerabilities
run 'npm audit fix' to fix them, or 'npm audit' for details
Removing intermediate container 93064c3225d
--> d85321b4551c
Step 5/7 : CMD [ "npm", "start" ]
--> 0ee1470b1f25
Step 6/7 : EXPOSE 3000
--> Running in da99fc1111a
Removing intermediate container 48315c1111a
--> 7b3eb2b3eebe
Step 7/7 : CMD ["npm", "start"]
--> Running in 6062b102768f
Removing intermediate container 60b1b102768f
--> b9e4d40a0618
Successfully tagged luciferdev7796/fe:latest
Pulling db (mongo:)
latest: Pulling from library/mongo
07316b7320d: pull complete
11f98ae7fd4: pull complete
322ca380e408: Pull complete
ec330b01309d: pull complete
85f04ab1b59f: pull complete
8e3962a1e6d: pull complete
9144ff41cd1c: pull complete
692b8635810: pull complete
1226231705cd: pull complete
8c9c49240bc: pull complete
7763a7a65fc: pull complete
79eb099f20d: pull complete
acc04ade5db: pull complete
Digest: sha256:29805e9a6fc2a233b9633c7b01382a43d13f119522b82366a72abc03a7293
Status: Downloaded newer image for mongo:latest
Building db
Step 1/7 : FROM mongo:10
--> c2b71092d62
Step 2/7 : WORKDIR /usr/src/app/db
--> Running in ad28ba8d0821
Removing intermediate container 1bd146606871
--> 3e734b32d7ad
Step 3/7 : COPY package*.json .
--> ab3e2c04d11b
Step 4/7 : RUN npm install
--> Running in cb85631d709b
added 82 packages from 49 contributors and audited 81 packages in 1.745s
1 package is looking for funding
run 'npm fund' for details
found 0 vulnerabilities
Removing intermediate container e1005107f0f1
--> 8da98f25c79f
Step 5/7 : CMD [ "npm", "start" ]
--> 2d455020e42f
Step 6/7 : EXPOSE 3000
--> Running in ad28ba8d75e
Removing intermediate container a628a38d75e
--> 3e734b32d7ad
Step 7/7 : CMD ["npm", "start"]
--> Running in df138dd5b1c6
Removing intermediate container df138dd5b1c6
--> 67762363AC18
Successfully tagged luciferdev7796/db:latest
luciferdev7796: Image for service fe was built because it did not already exist. To rebuild this image you must use 'docker-compose build' or 'docker-compose up --build'.
Creating dockercourse_fe_1... done
Creating dockercourse_db_1... done
Creating dockercourse_fe_2... done
Creating dockercourse_db_2... done
C:\Users\Lucifer\COURSEDocker>
```

4. Now if we do
“docker ps”
We should see our 3 containers running.

```
Command Prompt
C:\Users\Lucifer\CB\DOCKER COURSE>docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS                               NAMES
5d412c806c5f        luciferdev7796/be  "docker-entrypoint.s..."  5 seconds ago       Up 4 seconds       0.0.0.0:3000->3000/tcp             dockercourse_be_1
63f37fd669e6        mongo              "docker-entrypoint.s..."  7 seconds ago       Up 6 seconds       0.0.0.0:27017->27017/tcp           dockercourse_db_1
57570bfb3b36        luciferdev7796/fe  "docker-entrypoint.s..."  7 seconds ago       Up 6 seconds       0.0.0.0:4200->4200/tcp             dockercourse_fe_1
C:\Users\Lucifer\CB\DOCKER COURSE>
```

5. We can access our Application at the Shown ports
(4200 for Front-End, 3000 for Back-End and 27017 for database)
6. Opening localhost:4200 gives us the app

Add Comment

Comment:

User Name:

Good Movie
-John Wick

Didn't Like It
-Santino D'antonio

Okish
-Winston

Nice Movie
-Gianna D'antonio

7. Now we'll add a new comment

Add Comment

Comment:

Amazing Movie

User Name:

Aurelio

Submit

Good Movie

-John Wick

Didn't Like It

-Santino D'antonio

Okish

-Waston

Nice Movie

-Gianna D'antonio

8. The comment gets saved below in the list of comments

Add Comment

Comment:

User Name:

Submit

Good Movie

-John Wick

Didn't Like It

-Santino D'antonio

Okish

-Waston

Nice Movie

-Gianna D'antonio

Amazing Movie

-Aurelio

9. To stop all the containers we do
“docker-compose down”

```
Command Prompt
C:\Users\Lucifer\CB\DOCKER COURSE>docker-compose down
Stopping dockercourse_be_1 ... done
Stopping dockercourse_fe_1 ... done
Stopping dockercourse_db_1 ... done
Removing dockercourse_be_1 ... done
Removing dockercourse_fe_1 ... done
Removing dockercourse_db_1 ... done
Removing network dockercourse_default
C:\Users\Lucifer\CB\DOCKER COURSE>
```

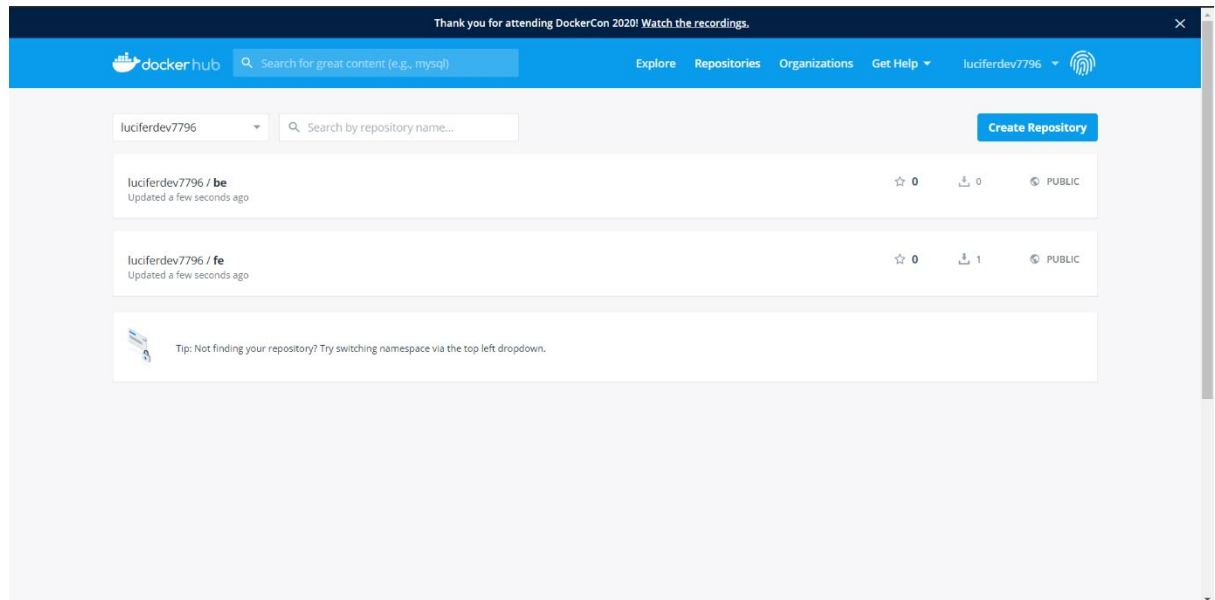
10. Now the containers are stopped. If we again do “docker-compose up -d” and open the app again, We see that the data is not lost and is preserved because in the docker-compose.yml, We have specified the volume configuration which maps a host folder to the container folder so the database is stored in the “DOCKER COURSE\DB” folder rather than in the container preventing loss on restart.

11. Finally we can push our Front-End and Back-End images to our registry on docker hub using the command

“docker-compose push”

```
Command Prompt
C:\Users\Lucifer\CB\DOCKER COURSE>docker-compose push
Pushing fe (luciferdev7796/fe:latest)...
The push refers to repository [docker.io/luciferdev7796/fe]
c02b23deee40: Pushed
7719327c9f25: Pushed
13fa80bc3ad: Pushed
66c5b26a0fbb: Pushed
d1e551c519b5: Mounted from library/node
70a3a00c2b44: Mounted from library/node
918a6facd0de: Mounted from library/node
bc17cd405095: Mounted from library/node
ee854067fbbd: Mounted from library/node
740ffea5d5c3: Mounted from library/node
eac9ead92b24: Mounted from library/node
23bca356262f: Mounted from library/node
8354d5896557: Mounted from library/node
latest: digest: sha256:7f3b7b81d830bf87771282043573b3ef6a02142c2cee19bf1d629b890059de3 size: 3055
Pushing be (luciferdev7796/be:latest)...
The push refers to repository [docker.io/luciferdev7796/be]
0b2c0a6df5cf: Pushed
9abb50360779: Pushed
42fa7533cca8: Pushed
f2363f3f6acf: Pushed
d1e551c519b5: Mounted from luciferdev7796/fe
70a3a00c2b44: Mounted from luciferdev7796/fe
918a6facd0de: Mounted from luciferdev7796/fe
bc17cd405095: Mounted from luciferdev7796/fe
ee854067fbbd: Mounted from luciferdev7796/fe
740ffea5d5c3: Mounted from luciferdev7796/fe
eac9ead92b24: Mounted from luciferdev7796/fe
23bca356262f: Mounted from luciferdev7796/fe
8354d5896557: Mounted from luciferdev7796/fe
latest: digest: sha256:5d3f0d9a3422b7c42a2128a0b1a600f8a0cc509564a38912642a084480dffa45 size: 3050
C:\Users\Lucifer\CB\DOCKER COURSE>
```

12. If we browse to our personal Docker Hub Registry on the web, we can find our newly pushed images there.



13. "docker-compose push" is equivalent to running the "docker push" command on both the images independently one after the other

NOTE: We do not need to push the database image as we ourselves have pulled the mongo image from docker hub (done by the docker-compose file for us). So no point pushing an image we had pulled in the first place as it's already present in the Registry.