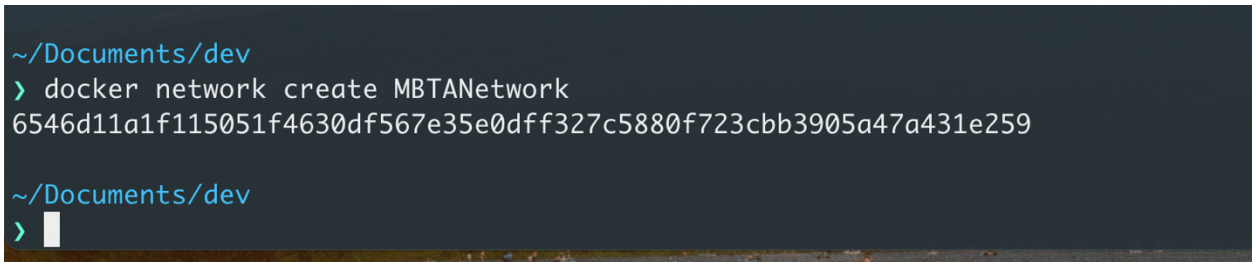
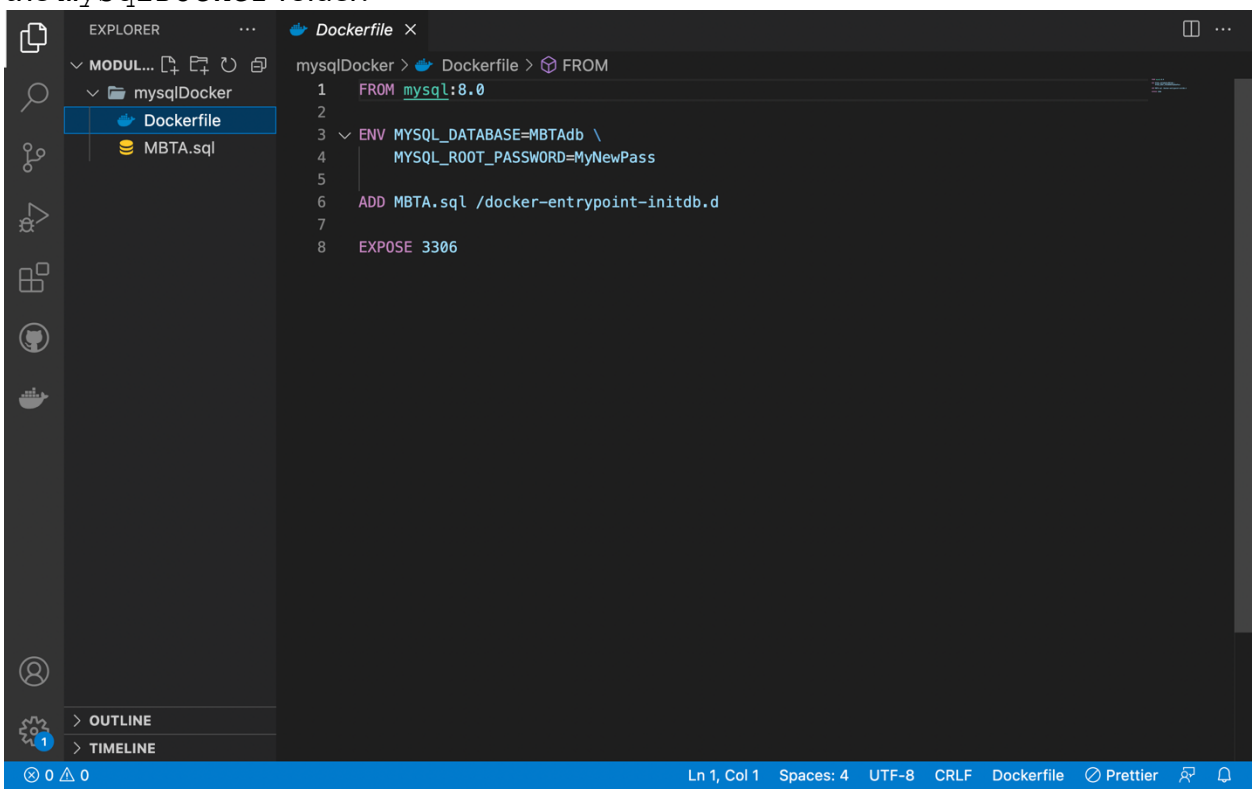
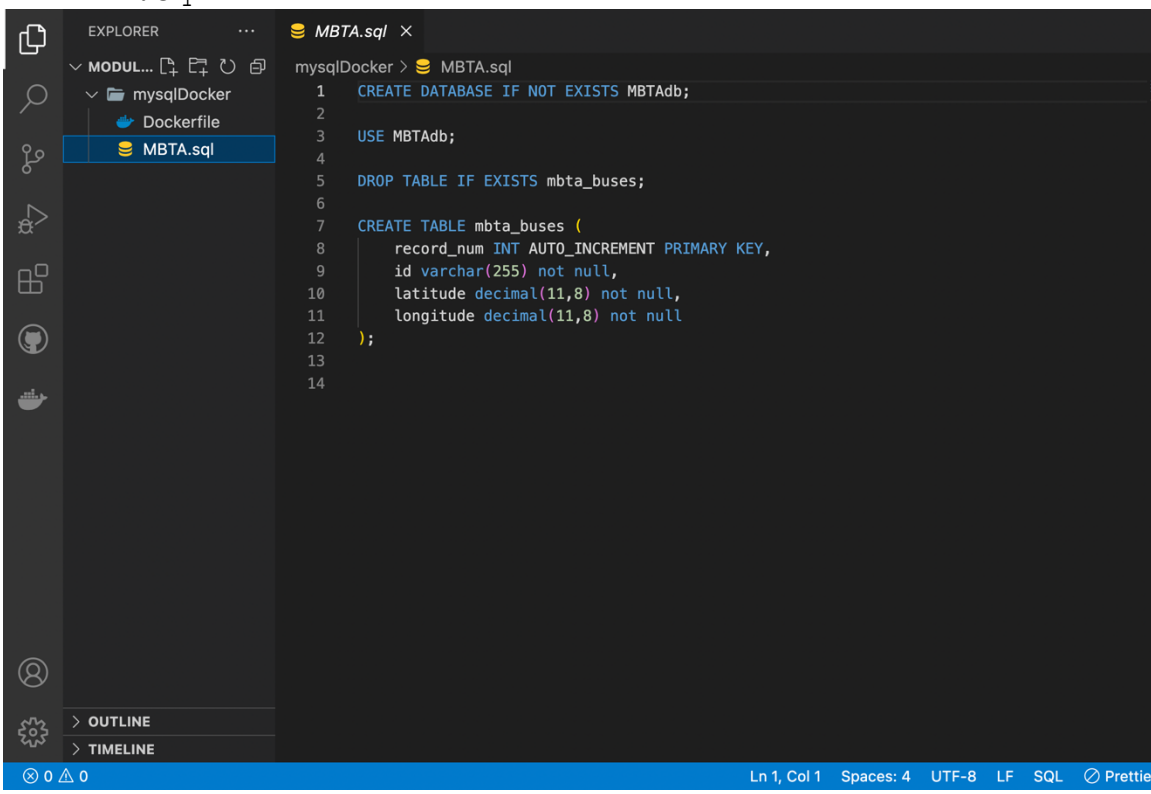
1. Provide a screenshot to show that you have successfully created the `MBTANetwork` network.



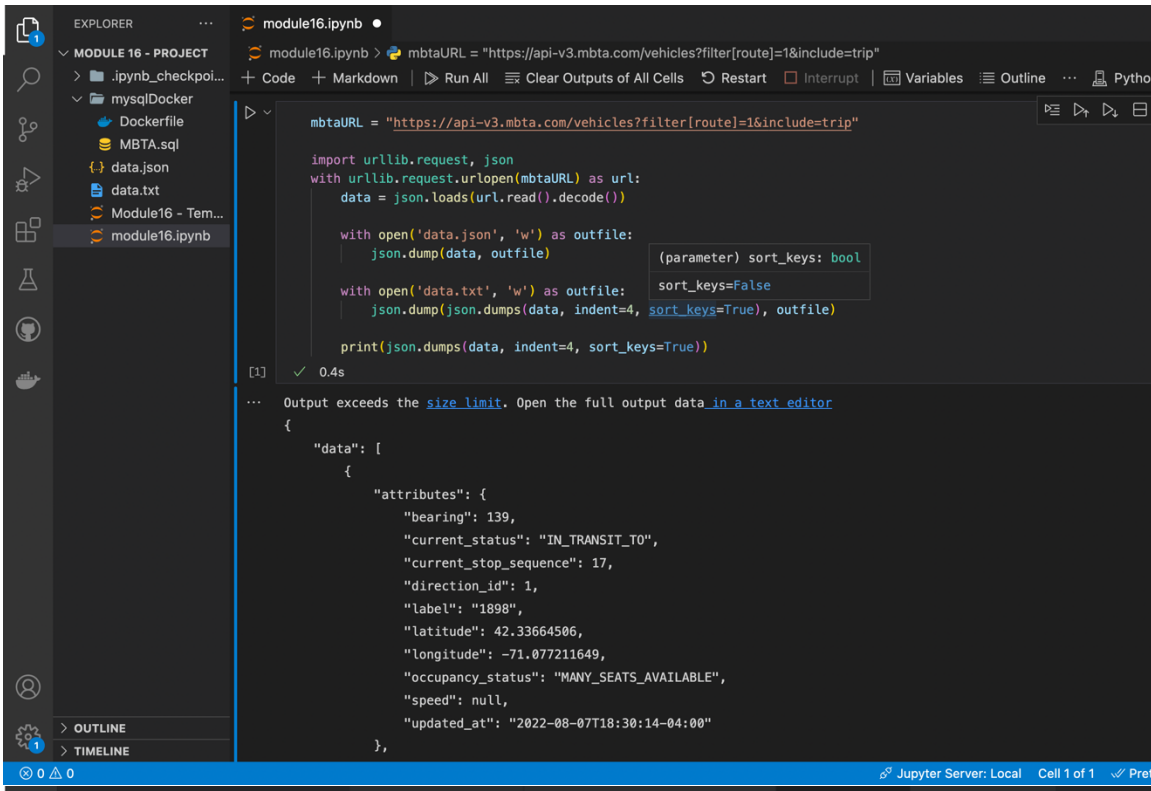
2. Provide a screenshot to show that you have successfully opened the `mysqlDocker` folder.

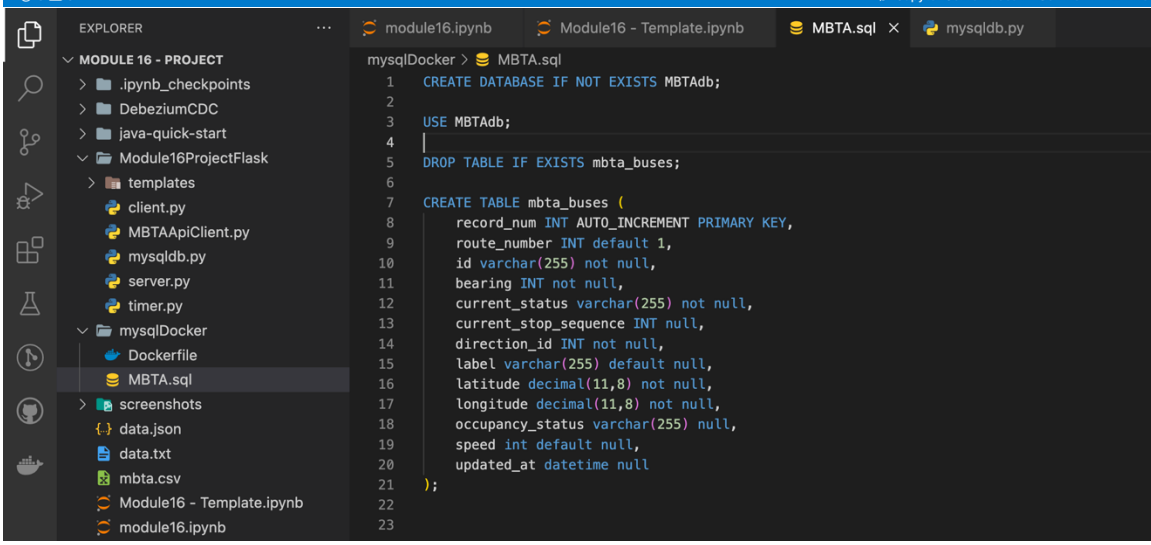


1. Provide a screenshot to show that you have successfully opened the `MBTA.sql` file.

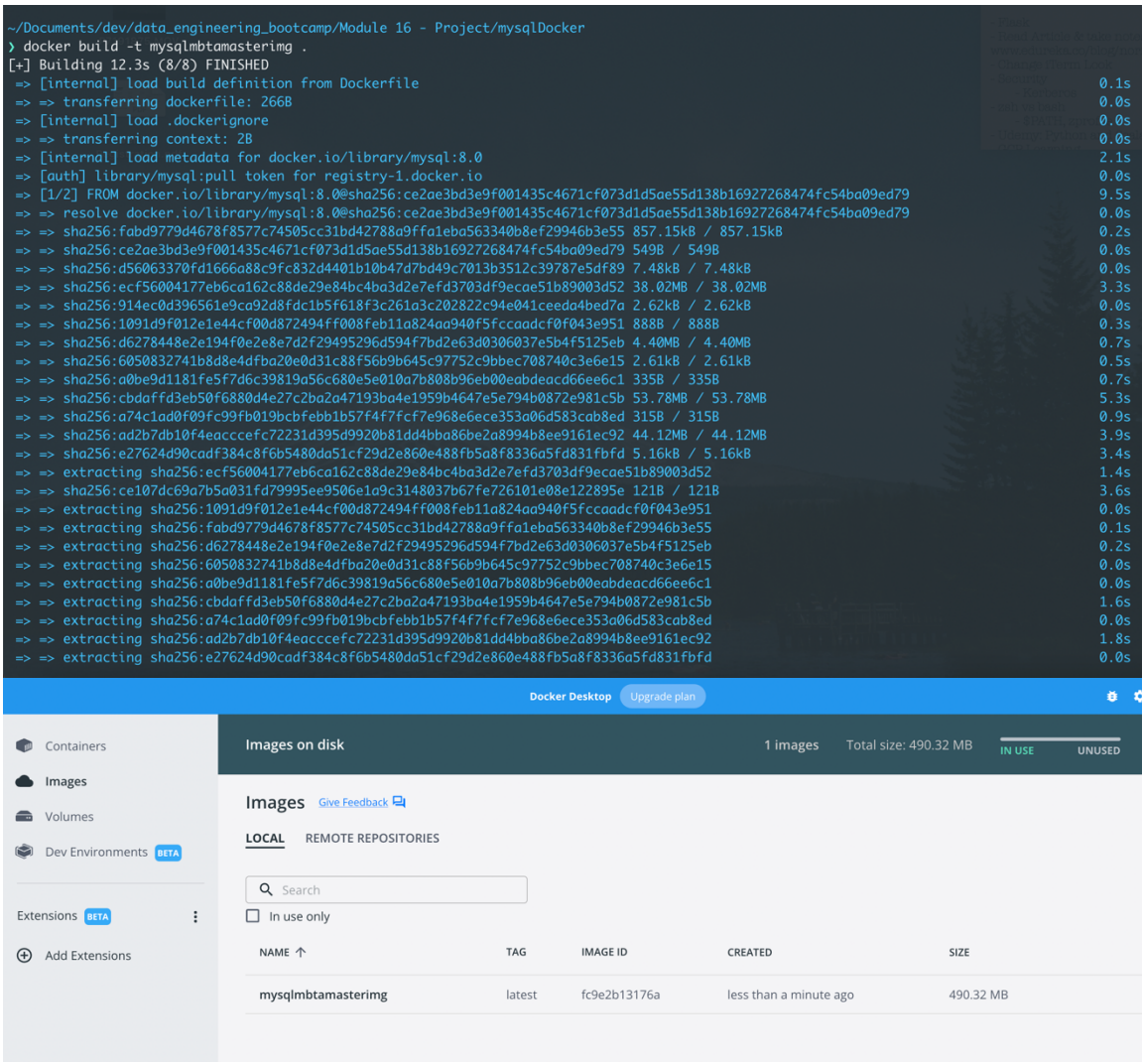


2. For this step, you will provide two screenshots. The first screenshot should show that you have successfully run the provided code in a Jupyter Notebook. The second screenshot should show that you have successfully added at least five additional fields to the `mbta_buses` table.





3. Provide a screenshot to show that you have successfully created the `mysqlmbtamasterimg` Docker *image*.



4. Provide a screenshot to show that you have successfully created the `mysqlserver` Docker *container*.

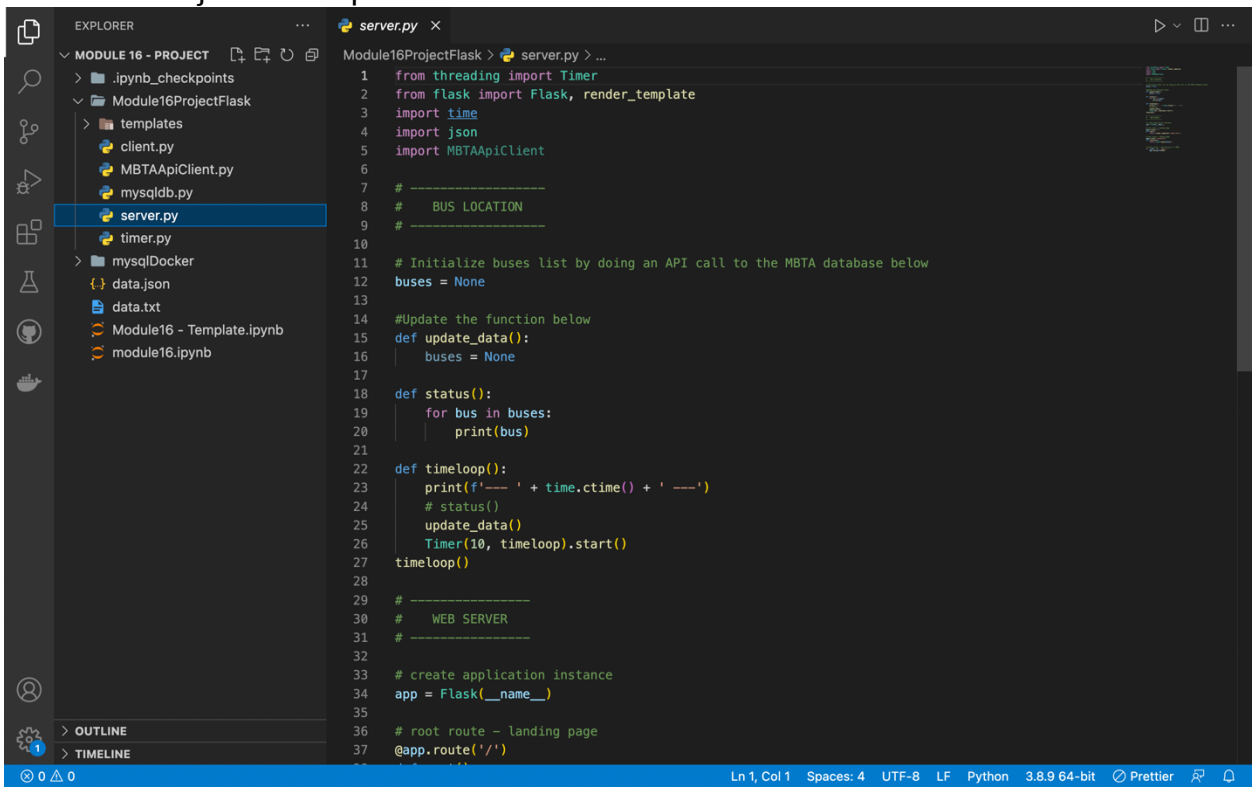
```
~/Documents/dev/data_engineering_bootcamp/Module 16 - Project/mysqlDocker 13s
❯ docker run --rm --name mysqlserver -p 3306:3306 --network MBTANetwork -d mysqlmbtamasterimg
b27116fecc98a88a1973a9b9f4fa78e27de953bccc88dd21b95f92a3faa2bde6

~/Documents/dev/data_engineering_bootcamp/Module 16 - Project/mysqlDocker
```

3. Provide a screenshot to show that you have successfully created the `some-mongo` Docker *container*.

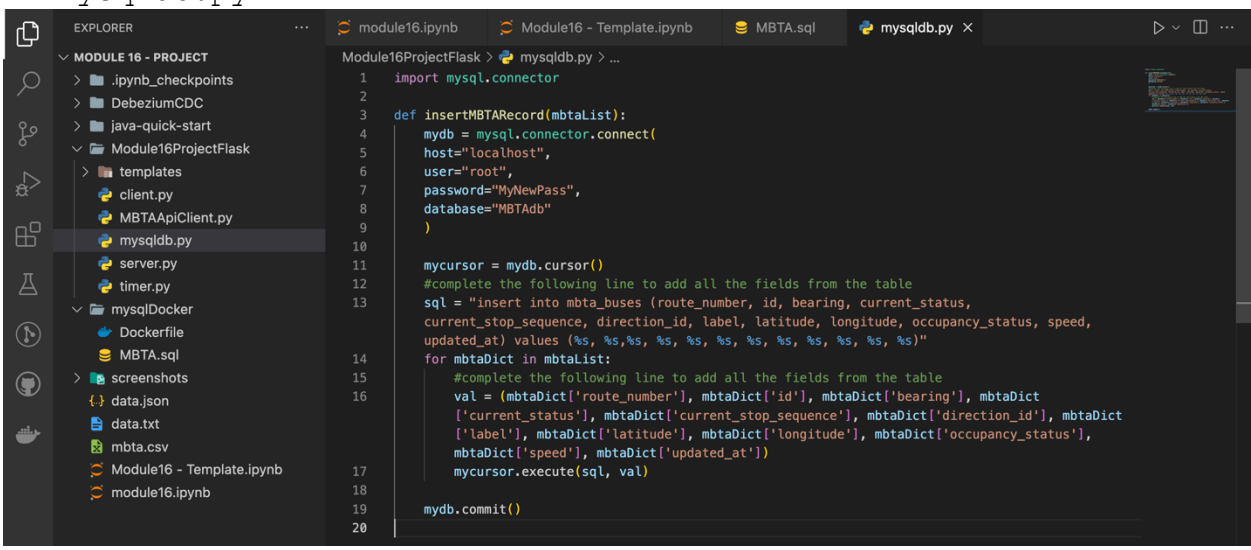
```
~/Documents/dev/data_engineering_bootcamp/Module 16 - Project/mysqlDocker
❯ docker run --name some-mongo --network MBTANetwork -p 27017:27017 -d mongo
Unable to find image 'mongo:latest' locally
latest: Pulling from library/mongo
a749a280e3e9: Pull complete
70f92e752512: Pull complete
5fdda1347325: Pull complete
c5e02463a0f4: Pull complete
ff9bc5b0968e: Pull complete
33f0d07ad9ca: Pull complete
e5c25f5ea983: Pull complete
53f9cf44c1e9: Pull complete
fbd4a874cd21: Pull complete
Digest: sha256:4200c3073389d5b303070e53ff8f5e4472efb534340d28599458ccc24f378025
Status: Downloaded newer image for mongo:latest
6cd8a288fe292d4a596e6164e068478c2d471ad049f8e2c6f79a547fb63e8d12

~/Documents/dev/data_engineering_bootcamp/Module 16 - Project/mysqlDocker 18s
```

4.

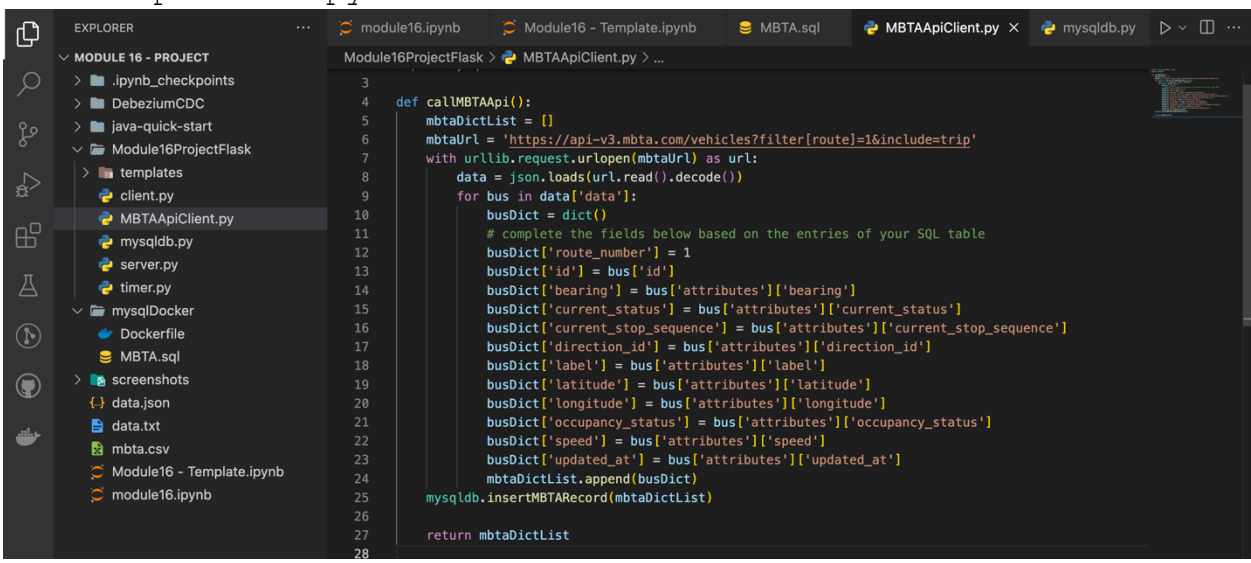
a. Provide a screenshot to show that you have successfully opened the Module16ProjectFlask.zip folder in VS Code.



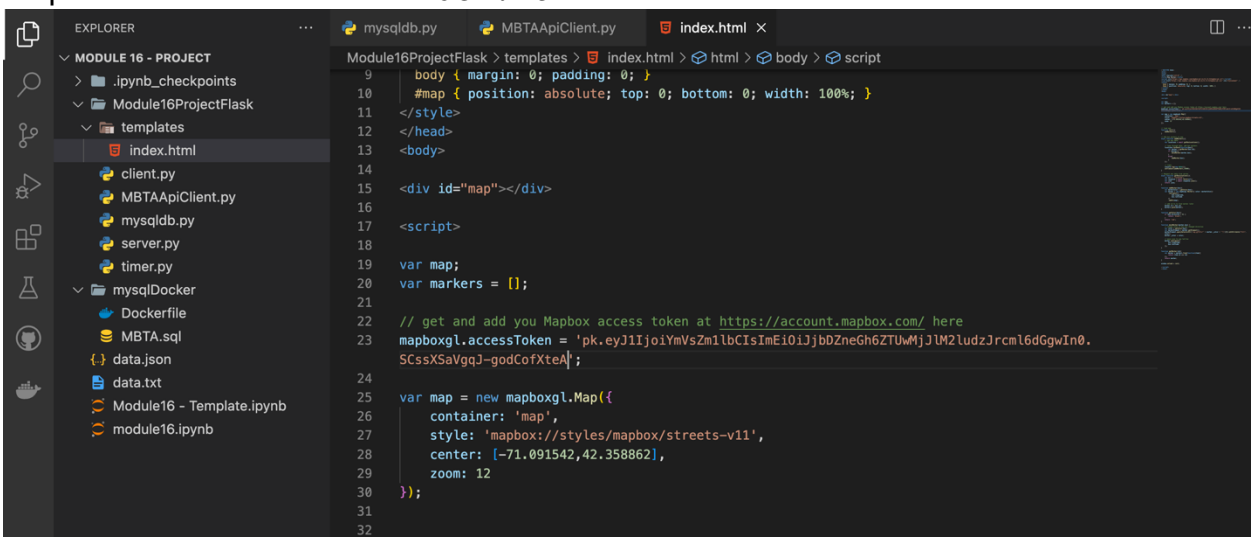
b. Provide a screenshot to show that you have successfully modified the `mysqldb.py` file.



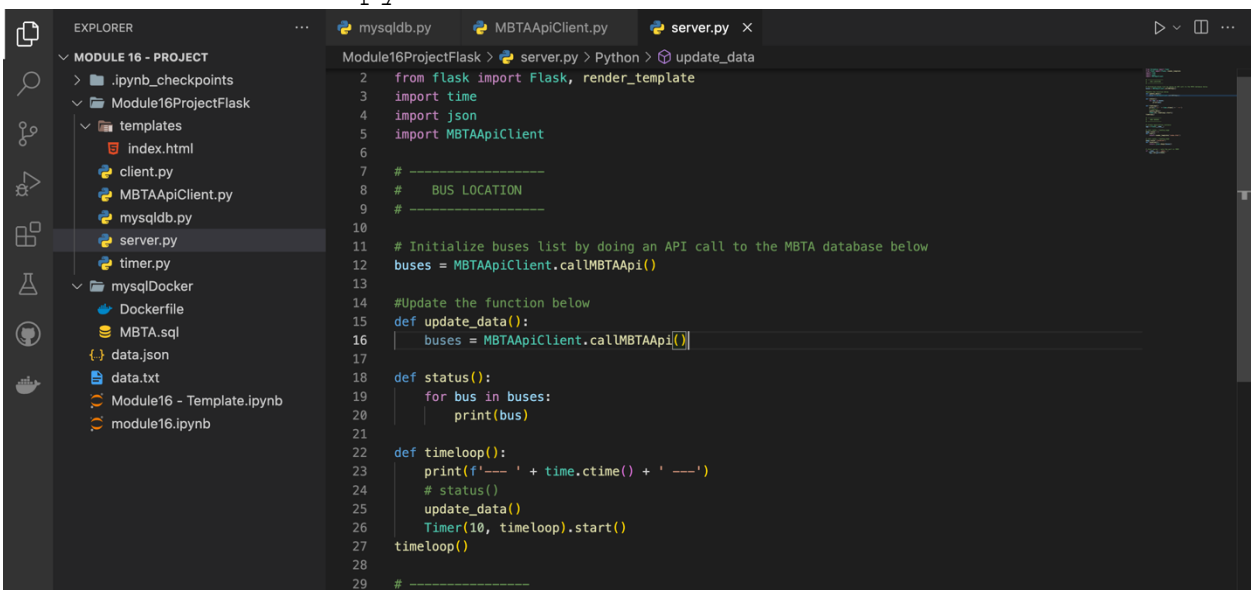
c. Provide a screenshot to show that you have successfully modified the `MBTAApiClient.py` file.



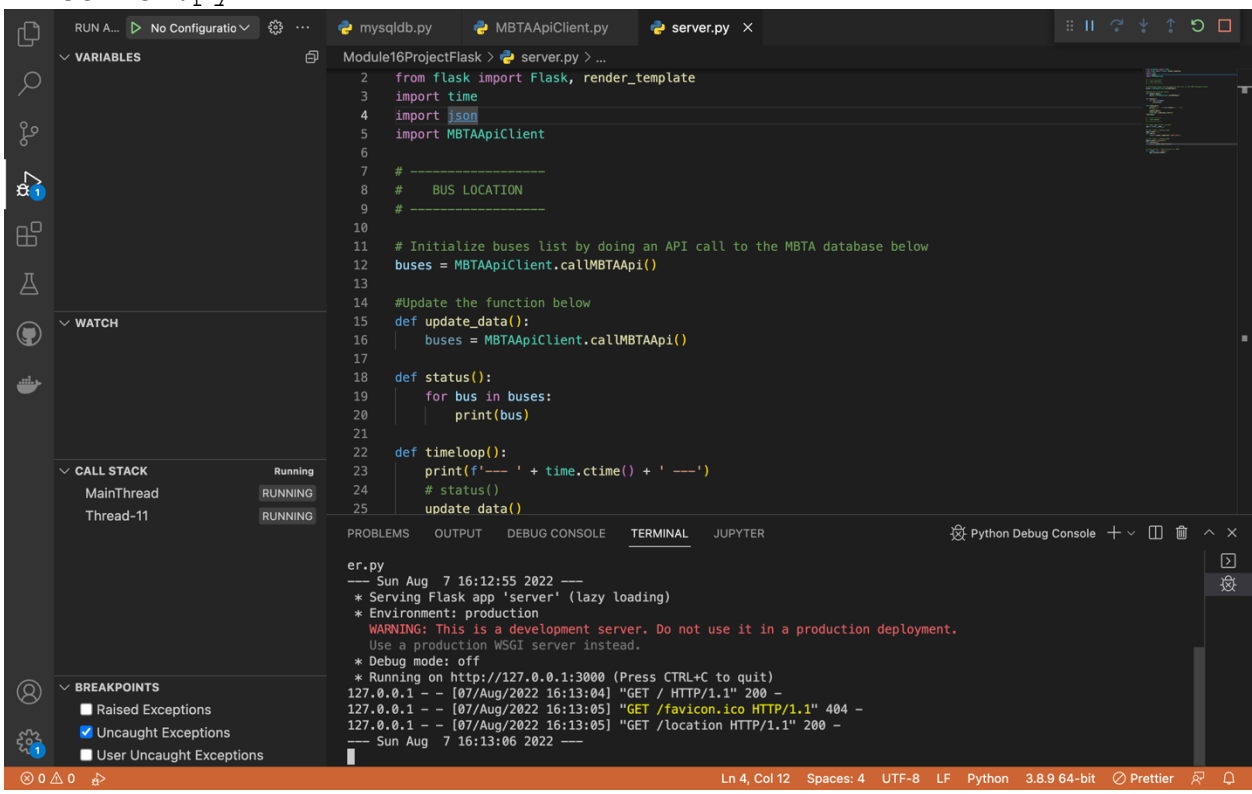
d. Provide a screenshot to show that you have successfully added your Mapbox access *token* in the `index.html` file.



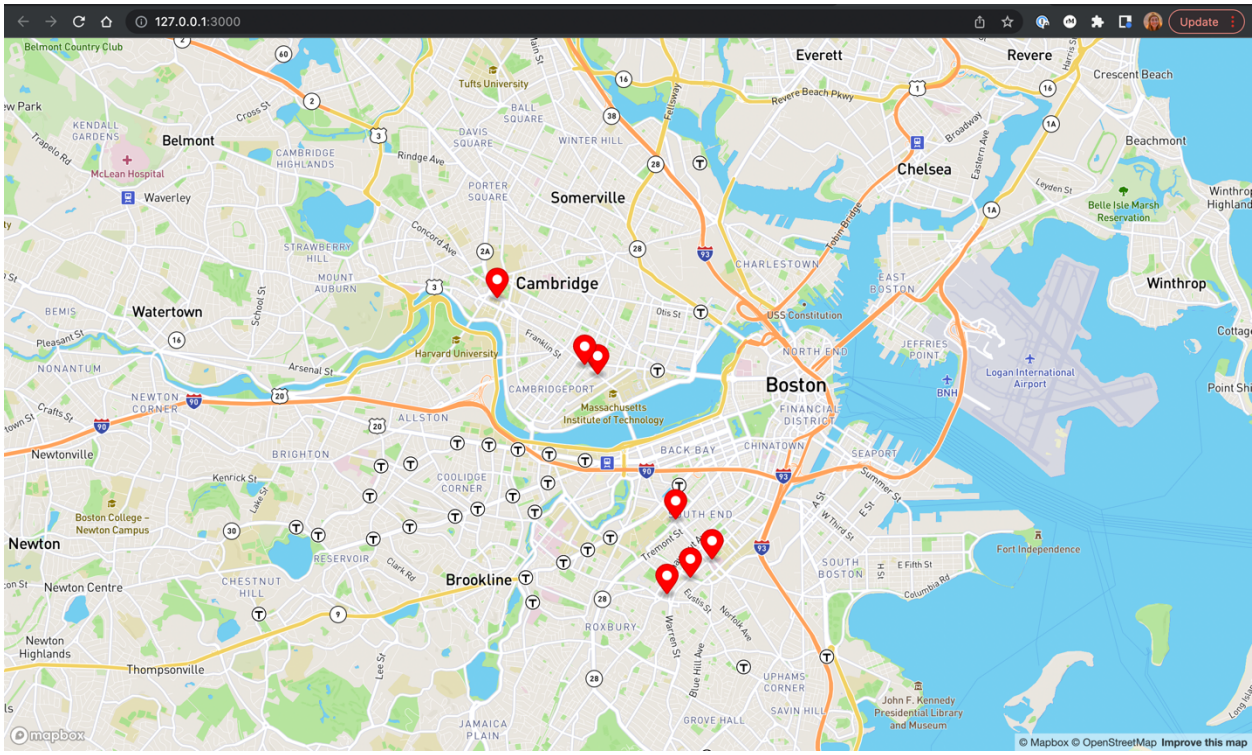
e. Provide a screenshot to show that you have successfully initialized the buses *list* in the `server.py` file.



f. Provide a screenshot to show that you have successfully run the `server.py` file in VS Code.

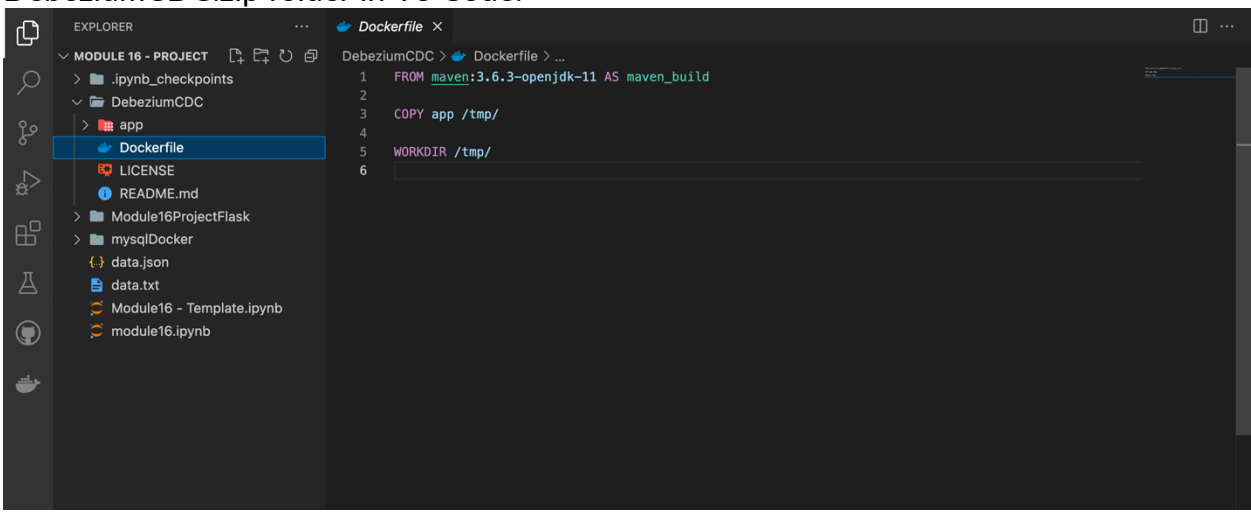


g. Provide a screenshot to show that you have successfully navigated to localhost:3000.

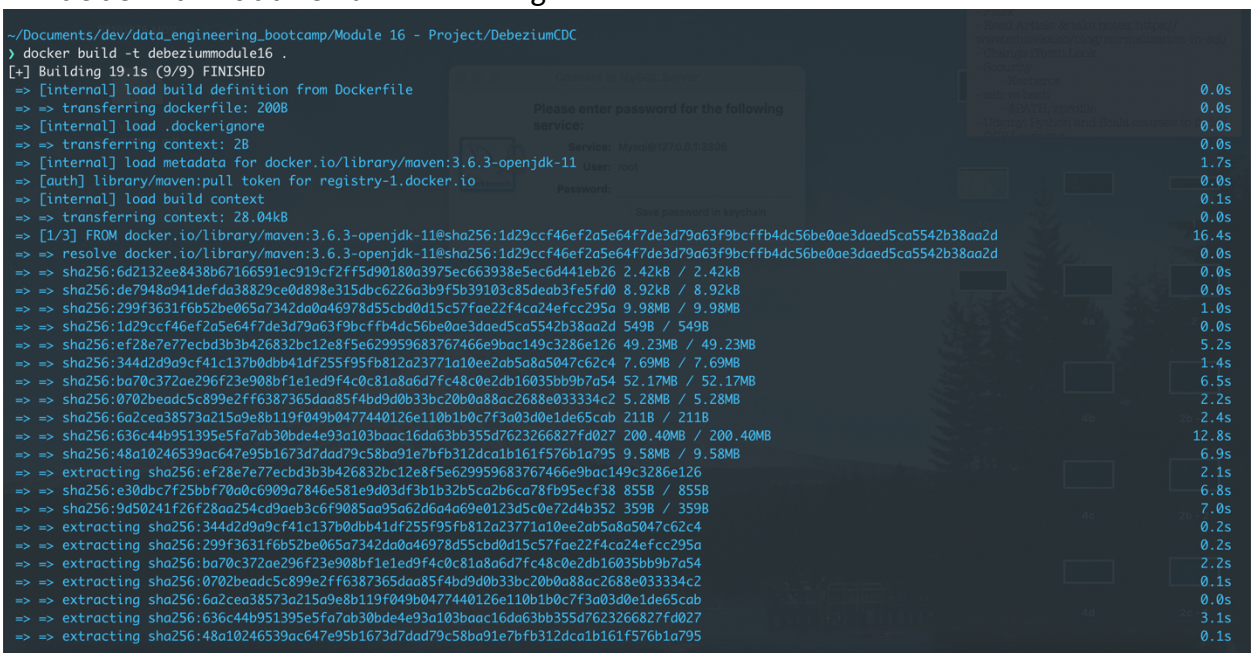


5.

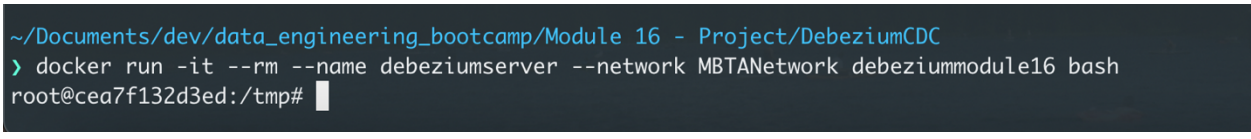
a. Provide a screenshot to show that you have successfully opened the DebeziumCDC.zip folder in VS Code.



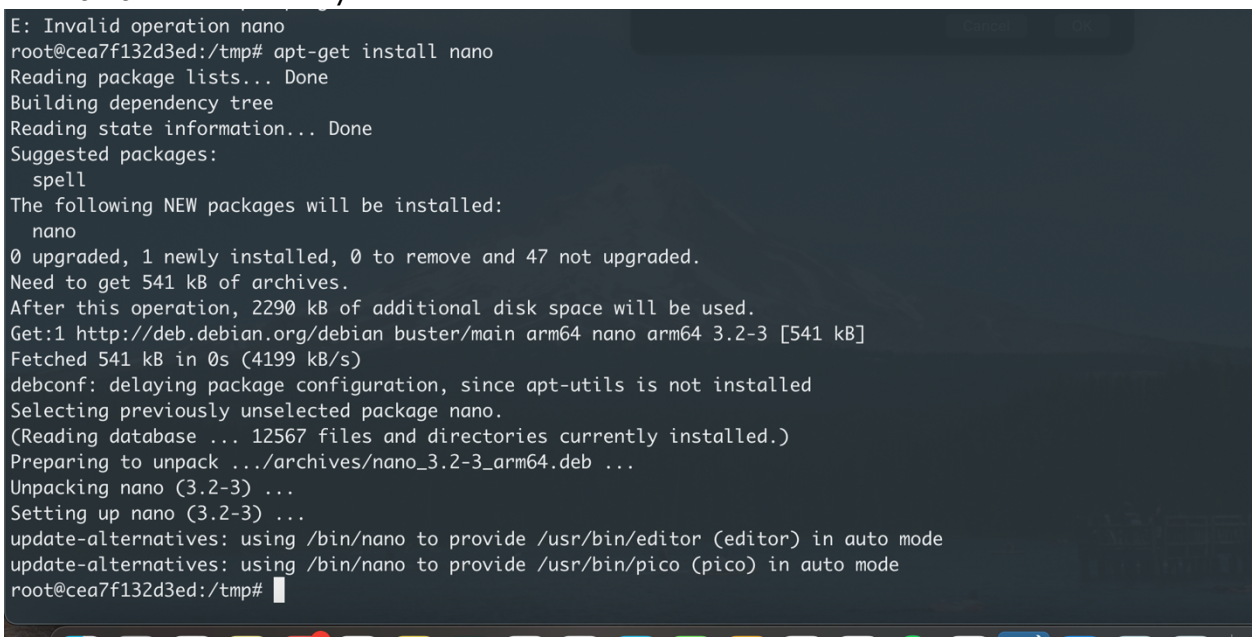
b. Provide a screenshot to show that you have successfully created the `debeziummodule16` Docker *image*.



c. Provide a screenshot to show that you have successfully created the Docker *container* and associated it with the `MBTANetwork` network.

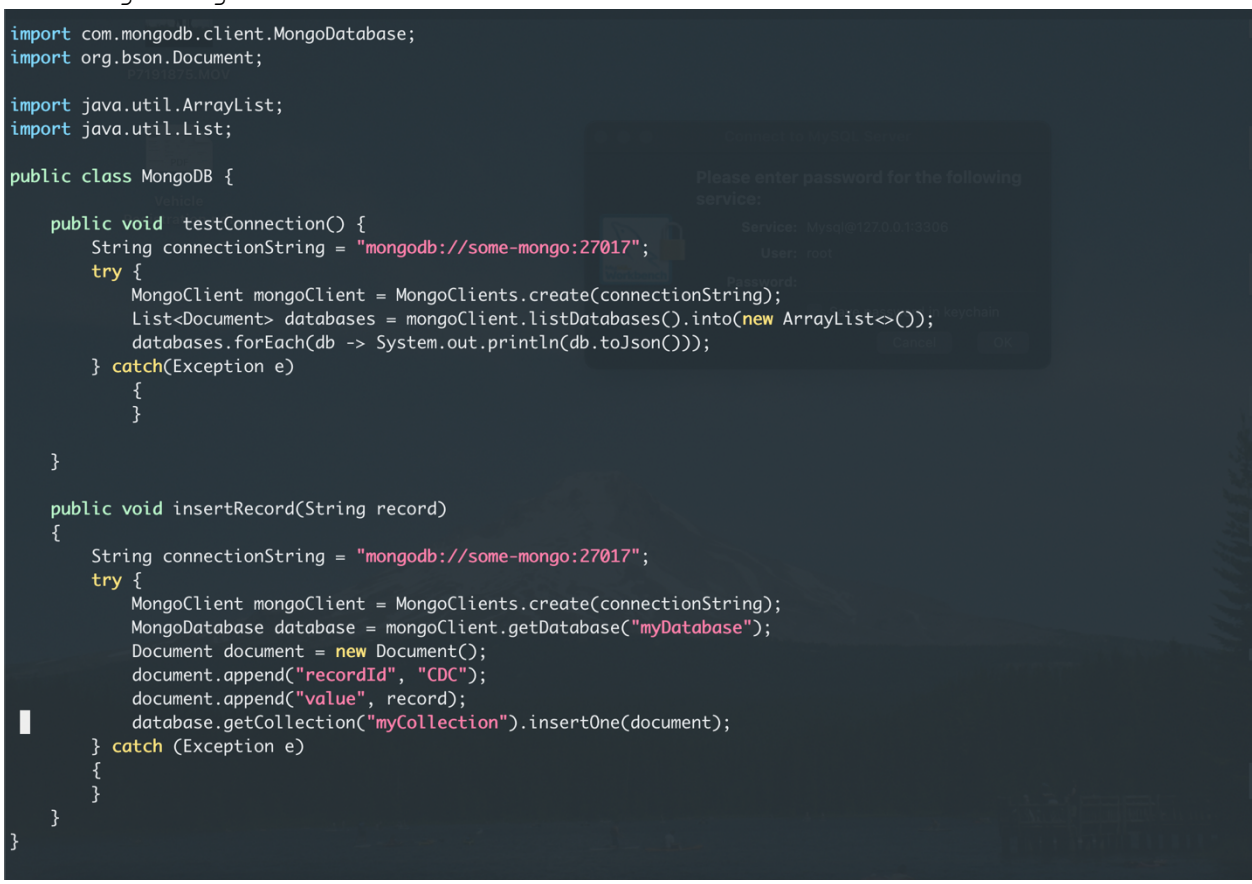


d. Provide a screenshot to show that you have successfully installed the `nano` text editor in your shell.



```
E: Invalid operation nano
root@cea7f132d3ed:/tmp# apt-get install nano
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  spell
The following NEW packages will be installed:
  nano
0 upgraded, 1 newly installed, 0 to remove and 47 not upgraded.
Need to get 541 kB of archives.
After this operation, 2290 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian buster/main arm64 nano arm64 3.2-3 [541 kB]
Fetched 541 kB in 0s (4199 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package nano.
(Reading database ... 12567 files and directories currently installed.)
Preparing to unpack .../archives/nano_3.2-3_arm64.deb ...
Unpacking nano (3.2-3) ...
Setting up nano (3.2-3) ...
update-alternatives: using /bin/nano to provide /usr/bin/editor (editor) in auto mode
update-alternatives: using /bin/nano to provide /usr/bin/pico (pico) in auto mode
root@cea7f132d3ed:/tmp#
```

e. Provide a screenshot to show that you have successfully modified the `MongoDB.java` *class*.



```
import com.mongodb.client.MongoDatabase;
import org.bson.Document;

import java.util.ArrayList;
import java.util.List;

public class MongoDB {

    public void  testConnection() {
        String connectionString = "mongodb://some-mongo:27017";
        try {
            MongoClient mongoClient = MongoClients.create(connectionString);
            List<Document> databases = mongoClient.listDatabases().into(new ArrayList<>());
            databases.forEach(db -> System.out.println(db.toJson()));
        } catch(Exception e)
            {
            }

    }

    public void insertRecord(String record)
    {
        String connectionString = "mongodb://some-mongo:27017";
        try {
            MongoClient mongoClient = MongoClients.create(connectionString);
            MongoDatabase database = mongoClient.getDatabase("myDatabase");
            Document document = new Document();
            document.append("recordId", "CDC");
            document.append("value", record);
            database.getCollection("myCollection").insertOne(document);
        } catch (Exception e)
        {
        }
    }
}
```

f. Provide a screenshot to show that you have successfully modified the `handleChangeEvent` *method*.

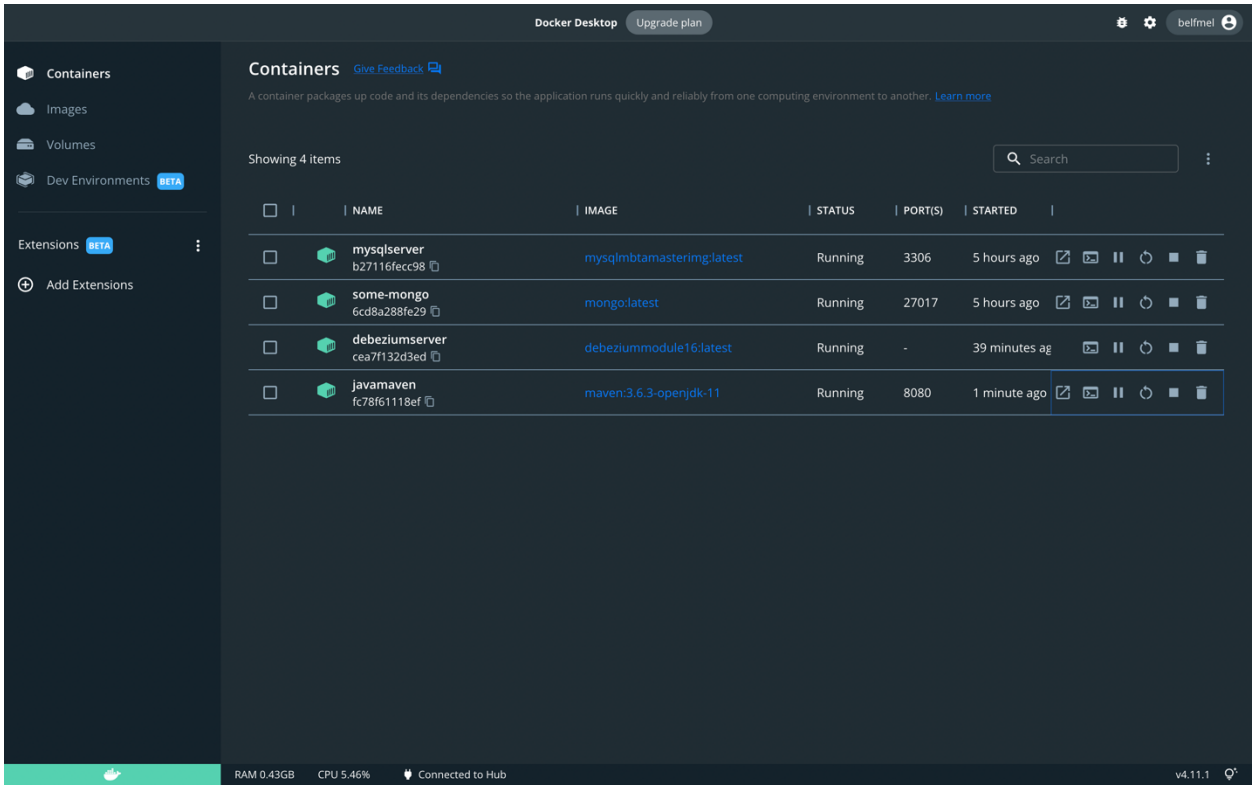
```
public class DebeziumListener {

    private final Executor executor = Executors.newSingleThreadExecutor();
    private final DebeziumEngine<RecordChangeEvent<SourceRecord>> debeziumEngine;

    public DebeziumListener(Configuration customerConnectorConfiguration) {

        this.debeziumEngine = DebeziumEngine.create(ChangeEventFormat.of(Connect.class))
            .using(customerConnectorConfiguration.asProperties())
            .notifying(this::handleChangeEvent)
            .build();

        // this.customerService = customerService;
    }

    private void handleChangeEvent(RecordChangeEvent<SourceRecord> sourceRecordRecordChangeEvent) {
        SourceRecord sourceRecord = sourceRecordRecordChangeEvent.record();


        MongoDB mongoDB = new MongoDB();
        mongoDB.testConnection();
        mongoDB.insertRecord(sourceRecord.value().toString());

        System.out.println("Key = '" + sourceRecord.key() + "' value = '" + sourceRecord.value() + "'");
    }

    @PostConstruct
    private void start() {
        this.executor.execute(debeziumEngine);
    }

    @PreDestroy
    private void stop() throws IOException {
```

g. Provide a screenshot to show that you have successfully run the Maven SpringBoot application.

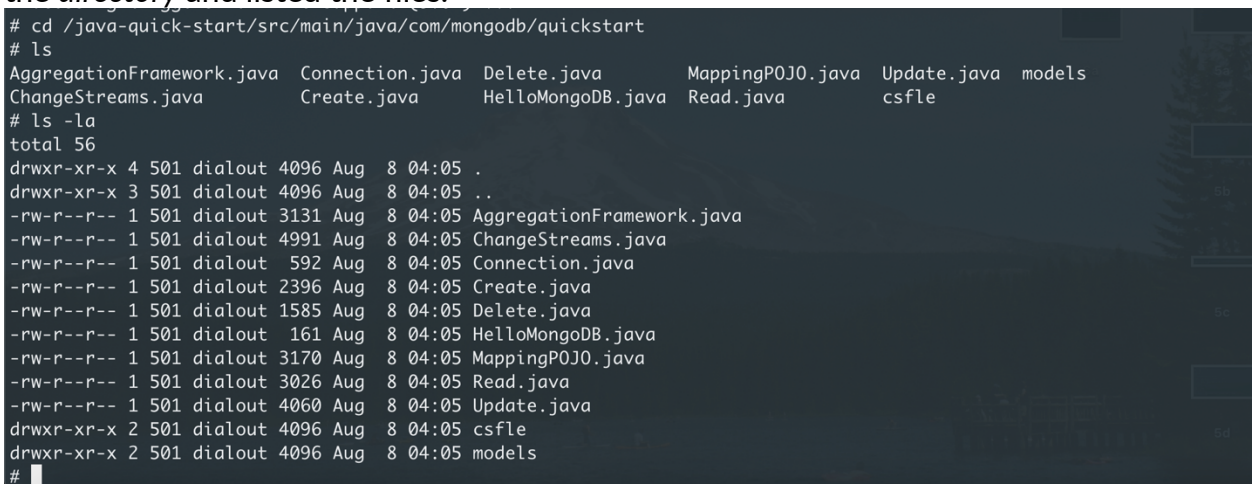
```
⌥⌘1                    DebeziumCDC: docker run -it --rm --name debeziumserver --network MBTANetwork bash

ANSIT_TO,current_stop_sequence=1,direction_id=0,occupancy_status=MANY_SEATS_AVAILABLE},source=Struct{version=1.5.2.Final,connector=mysql,name=localhost_MBTAdb
,ts_ms=1659930816000,db=MBTAdb,table=mbta_buses,server_id=1,file=binlog.000002,pos=19551,row=0},op=c,ts_ms=1659930816742}'
2022-08-08 03:53:36.868  INFO 501 --- [pool-1-thread-1] org.mongodb.driver.cluster               : Cluster created with settings {hosts=[some-mongo:27017], mo
de=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms'}
2022-08-08 03:53:36.869  INFO 501 --- [pool-1-thread-1] org.mongodb.driver.cluster               : Cluster description not yet available. Waiting for 30000 ms
 before timing out
2022-08-08 03:53:36.869  INFO 501 --- [ome-mongo:27017] org.mongodb.driver.connection            : Opened connection [connectionId{localValue:535, serverValue
:535}] to some-mongo:27017
2022-08-08 03:53:36.870  INFO 501 --- [ome-mongo:27017] org.mongodb.driver.cluster               : Monitor thread successfully connected to server with descri
ption ServerDescription{address=some-mongo:27017, type=STANDALONE, state=CONNECTED, ok=true, minWireVersion=0, maxWireVersion=13, maxDocumentSize=16777216, lo
gicalSessionTimeoutMinutes=30, roundTripTimeNanos=774708}
2022-08-08 03:53:36.870  INFO 501 --- [ome-mongo:27017] org.mongodb.driver.connection            : Opened connection [connectionId{localValue:536, serverValue
:536}] to some-mongo:27017
2022-08-08 03:53:36.870  INFO 501 --- [pool-1-thread-1] org.mongodb.driver.connection            : Opened connection [connectionId{localValue:537, serverValue
:537}] to some-mongo:27017
{"name": "admin", "sizeOnDisk": 40960, "empty": false}
{"name": "config", "sizeOnDisk": 12288, "empty": false}
{"name": "local", "sizeOnDisk": 40960, "empty": false}
{"name": "myDatabase", "sizeOnDisk": 40960, "empty": false}
2022-08-08 03:53:36.872  INFO 501 --- [pool-1-thread-1] org.mongodb.driver.cluster               : Cluster created with settings {hosts=[some-mongo:27017], mo
de=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms'}
2022-08-08 03:53:36.872  INFO 501 --- [pool-1-thread-1] org.mongodb.driver.cluster               : Cluster description not yet available. Waiting for 30000 ms
 before timing out
2022-08-08 03:53:36.873  INFO 501 --- [ome-mongo:27017] org.mongodb.driver.connection            : Opened connection [connectionId{localValue:538, serverValue
:538}] to some-mongo:27017
2022-08-08 03:53:36.873  INFO 501 --- [ome-mongo:27017] org.mongodb.driver.cluster               : Monitor thread successfully connected to server with descri
ption ServerDescription{address=some-mongo:27017, type=STANDALONE, state=CONNECTED, ok=true, minWireVersion=0, maxWireVersion=13, maxDocumentSize=16777216, lo
gicalSessionTimeoutMinutes=30, roundTripTimeNanos=450833}
2022-08-08 03:53:36.874  INFO 501 --- [ome-mongo:27017] org.mongodb.driver.connection            : Opened connection [connectionId{localValue:539, serverValue
:539}] to some-mongo:27017
2022-08-08 03:53:36.874  INFO 501 --- [pool-1-thread-1] org.mongodb.driver.connection            : Opened connection [connectionId{localValue:540, serverValue
:540}] to some-mongo:27017
Key = 'Struct{record_num=90}' value = 'Struct{after=Struct{record_num=90,id=y1786,latitude=42.32981000,longitude=-71.08353000,bearing=90,current_status=IN_TRA
NSIT_TO,current_stop_sequence=1,direction_id=1},source=Struct{version=1.5.2.Final,connector=mysql,name=localhost_MBTAdb,ts_ms=1659930816000,db=MBTAdb,table=mb
ta_buses,server_id=1,file=binlog.000002,pos=19743,row=0},op=c,ts_ms=1659930816742}'
```

6.

a) Provide a screenshot of your Docker desktop to show the `javamaven` *container* running.

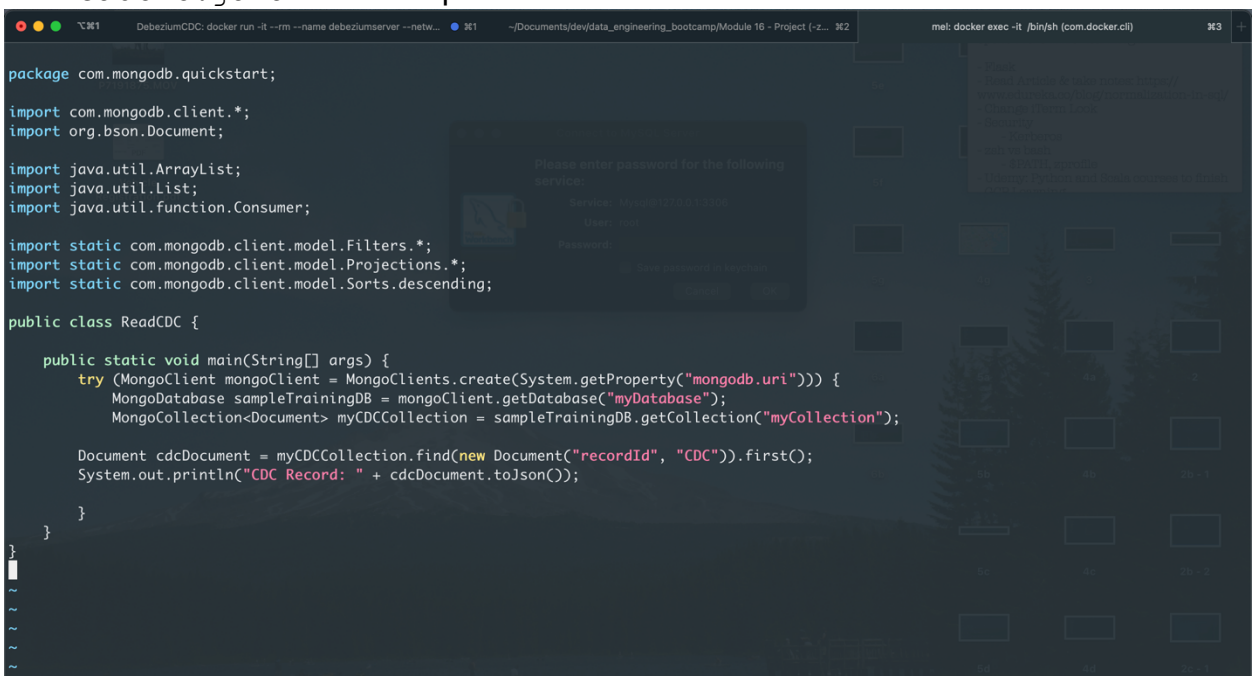


b) Provide a screenshot to show that you successfully navigated to the *directory* and listed the files.



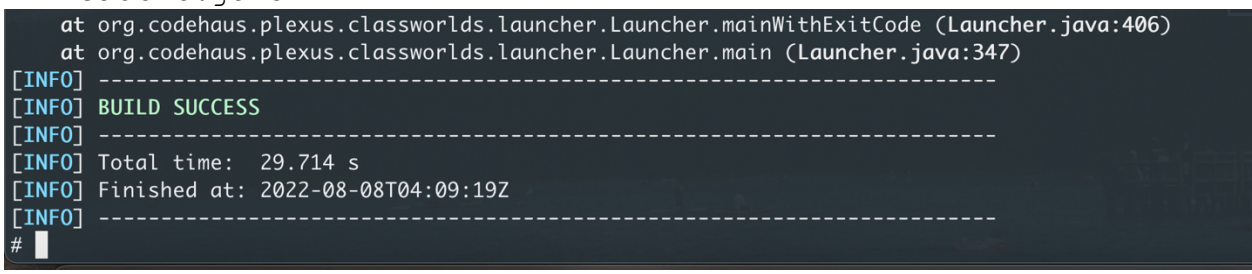
```
# cd /java-quick-start/src/main/java/com/mongodb/quickstart
# ls
AggregationFramework.java  Connection.java  Delete.java       MappingPOJO.java  Update.java  models
ChangeStreams.java         Create.java      HelloMongoDB.java  Read.java         csfle
# ls -la
total 56
drwxr-xr-x 4 501 dialout 4096 Aug  8 04:05 .
drwxr-xr-x 3 501 dialout 4096 Aug  8 04:05 ..
-rw-r--r-- 1 501 dialout 3131 Aug  8 04:05 AggregationFramework.java
-rw-r--r-- 1 501 dialout 4991 Aug  8 04:05 ChangeStreams.java
-rw-r--r-- 1 501 dialout  592 Aug  8 04:05 Connection.java
-rw-r--r-- 1 501 dialout 2396 Aug  8 04:05 Create.java
-rw-r--r-- 1 501 dialout 1585 Aug  8 04:05 Delete.java
-rw-r--r-- 1 501 dialout  161 Aug  8 04:05 HelloMongoDB.java
-rw-r--r-- 1 501 dialout 3170 Aug  8 04:05 MappingPOJO.java
-rw-r--r-- 1 501 dialout 3026 Aug  8 04:05 Read.java
-rw-r--r-- 1 501 dialout 4060 Aug  8 04:05 Update.java
drwxr-xr-x 2 501 dialout 4096 Aug  8 04:05 csfle
drwxr-xr-x 2 501 dialout 4096 Aug  8 04:05 models
#
```

c) Provide a screenshot to show that you successfully created the `ReadCDC.java` file and copied the code.



```
package com.mongodb.quickstart;

import com.mongodb.client.*;
import org.bson.Document;

import java.util.ArrayList;
import java.util.List;
import java.util.function.Consumer;

import static com.mongodb.client.model.Filters.*;
import static com.mongodb.client.model.Projections.*;
import static com.mongodb.client.model.Sorts.descending;

public class ReadCDC {

    public static void main(String[] args) {
        try (MongoClient mongoClient = MongoClients.create(System.getProperty("mongodb.uri"))) {
            MongoDatabase sampleTrainingDB = mongoClient.getDatabase("myDatabase");
            MongoCollection<Document> myCDCCollection = sampleTrainingDB.getCollection("myCollection");

        Document cdcDocument = myCDCCollection.find(new Document("recordId", "CDC")).first();
        System.out.println("CDC Record: " + cdcDocument.toJson());

        }
    }
}
```

d) Provide a screenshot to show the results of the `bash` command to execute the `ReadCDC.java` *class*.



```
    at org.codehaus.plexus.classworlds.launcher.Launcher.mainWithExitCode (Launcher.java:406)
    at org.codehaus.plexus.classworlds.launcher.Launcher.main (Launcher.java:347)
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  29.714 s
[INFO] Finished at: 2022-08-08T04:09:19Z
[INFO] ------------------------------------------------------------------------
#
```