

Part A: Network Creation

- Create a network in Docker by running the following command:

```
docker network create mongo
```

- List the networks using command

```
docker network ls
```

Part B: Running DB image with network

```
docker pull mongodb/mongodb-community-server:latest
```

```
docker run --network mongo --name mongodb -p 27017:27017 -d mongodb/mongodb-community-server:latest
```

- Confirm the container is running

```
docker ps
```

- Once running, check mongo db with mongosh command:

```
docker exec -it mongodb bash
mongosh
```

Part C: Creating Image of Backend Application:

- Please comment below property in application.properties file:

```
#spring.data.mongodb.uri=mongodb://localhost:27017
```

- Make sure to create application-production.properties file as follows:

NOTE: Do check the highlighted part.

```
spring.application.name=springbootdemo
```

```
server.port=8081
```

```
spring.data.mongodb.host=mongodb
```

```
spring.data.mongodb.port=27017
```

```
spring.data.mongodb.database=telecomdb
```

```
logging.level.org.springframework.data.mongodb.core.MongoTemplate=DEBUG
```

```
logging.level.org.mongodb.driver.protocol.command=DEBUG
```

```
# https://www.devglan.com/spring-boot/spring-boot-mongodb-crud
```

```
spring.output.ansi.enabled=ALWAYS
```

```
message=Default Profile
```

```
logging.file.name=logs/app.log
```

```
management.endpoints.web.exposure.include=*
```

```
management.endpoints.web.base-path=/manage
```

```
management.endpoint.health.show-details=ALWAYS
```

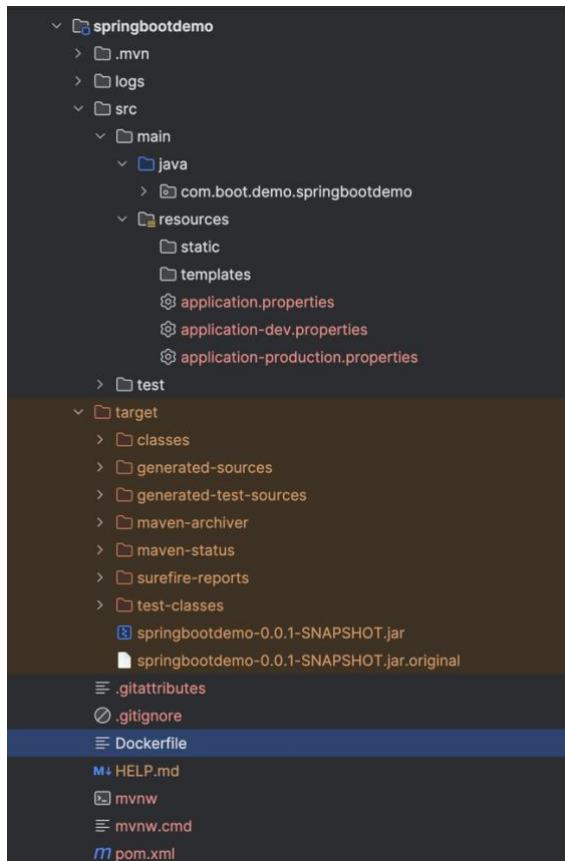
```
# Redis configuration
```

```
spring.data.redis.host=localhost  
spring.data.redis.port=6379
```

- Execute **maven install** for **springbootdemo** project
- Create Dockerfile as follows:

```
FROM amazoncorretto:17  
ADD target/springbootdemo-0.0.1-SNAPSHOT.jar app.jar  
EXPOSE 8080  
CMD ["java", "-jar", "-Dspring.profiles.active=production", "app.jar"]
```

NOTE: Do create Dockerfile exactly within the root of project. Project structure looks as below:



- Create image as follows:

```
docker image build -t springboot .
```

Note: springboot is our image name. Period (.) at the end of command is important

```
docker image ls
```

- **Visit the swagger page** in your browser. You can test the application is running - you will need to use the IP address / machine-name of your system. The url should be:

```
docker run -p8100:8081 --network mongo springboot
```

<http://localhost:8100/swagger-ui/index.html>