

in previous sprint

mongo container started manually

after that

customerapp container started manually

customerapp container
started/running

dependent container

mongo container
started/running

dependency container

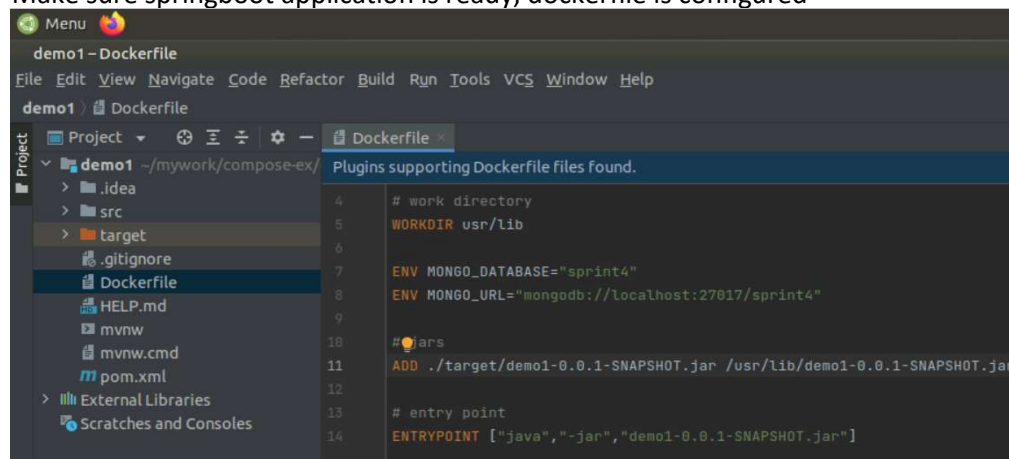
In sprint3, both containers to be started manually

if dependency container not started, dependent container wont work smooth

in Sprint 4, these both containers can be grouped

steps to add docker compose to existing application

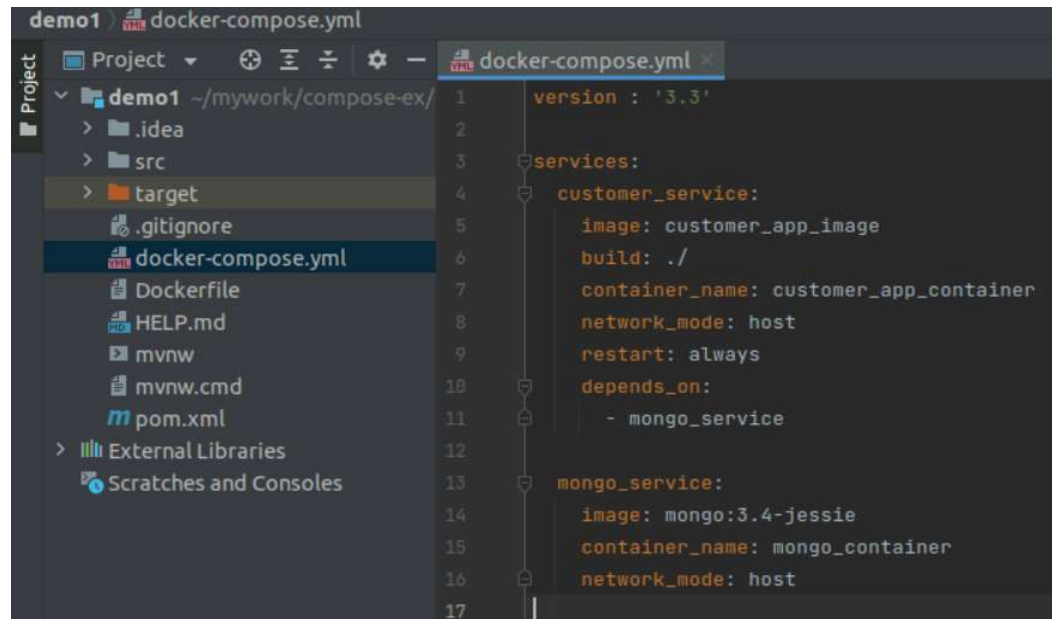
Step 1 Make sure springboot application is ready, dockerfile is configured



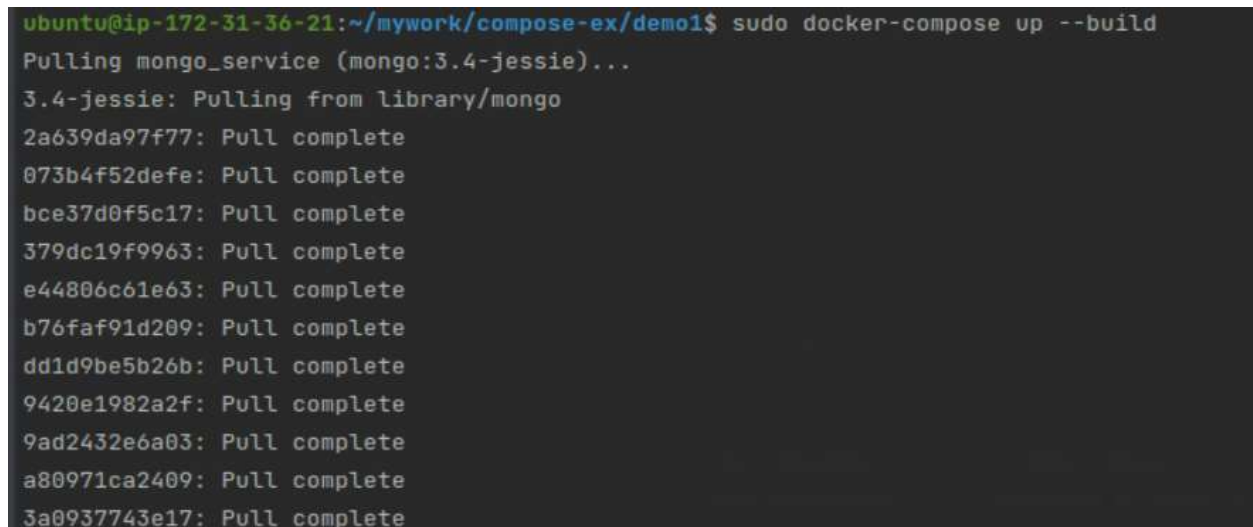
The screenshot shows an IDE window titled "demo1 - Dockerfile". The left sidebar displays a project tree with folders like ".idea", "src", "target", and files like ".gitignore", "HELP.md", "mvnw", "mvnw.cmd", and "pom.xml". The "Dockerfile" file is selected. The main editor area shows the Dockerfile content with line numbers 4 through 14. The Dockerfile includes instructions for setting the work directory, environment variables for MongoDB, adding the application JAR file, and defining the entry point.

```
4 # work directory
5 WORKDIR usr/lib
6
7 ENV MONGO_DATABASE="sprint4"
8 ENV MONGO_URL="mongodb://localhost:27017/sprint4"
9
10 # Jars
11 ADD ./target/demo1-0.0.1-SNAPSHOT.jar /usr/lib/demo1-0.0.1-SNAPSHOT.jar
12
13 # entry point
14 ENTRYPOINT ["java", "-jar", "demo1-0.0.1-SNAPSHOT.jar"]
```

Step 2 create docker-compose file in root folder



Step 3 run docker compose command
sudo docker-compose up --build



```
ubuntu@ip-172-31-36-21:~/mywork/compose-ex/demo1$ sudo docker-compose down
Stopping customer_app_container ... done
Stopping mongo_container        ... done
Removing customer_app_container ... done
Removing mongo_container        ... done
```