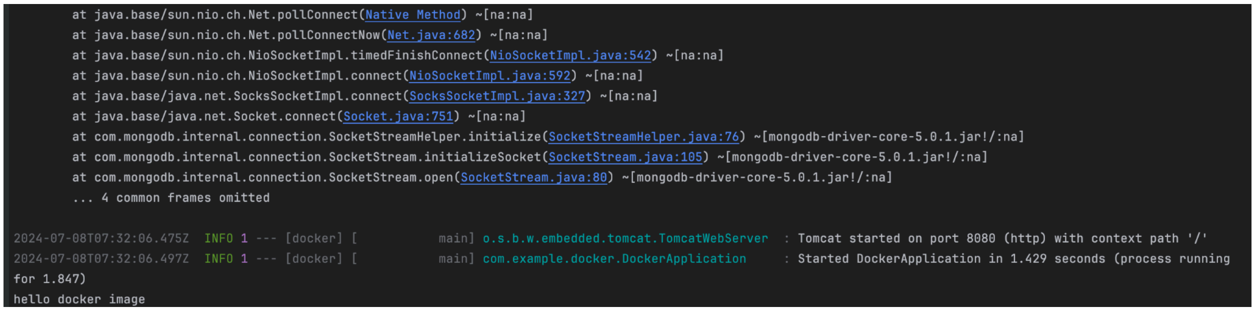
Carry out the following exercises.

- Install Docker
- Create a new Java project with Maven
- Create a main class and print “Hello docker example”



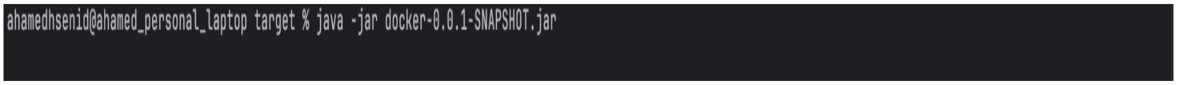
```
        at java.base/sun.nio.ch.Net.pollConnect(Native Method) ~[na:na]
        at java.base/sun.nio.ch.Net.pollConnectNow(Net.java:682) ~[na:na]
        at java.base/sun.nio.ch.NioSocketImpl.timedFinishConnect(NioSocketImpl.java:542) ~[na:na]
        at java.base/sun.nio.ch.NioSocketImpl.connect(NioSocketImpl.java:592) ~[na:na]
        at java.base/java.net.SocksSocketImpl.connect(SocksSocketImpl.java:327) ~[na:na]
        at java.base/java.net.Socket.connect(Socket.java:751) ~[na:na]
        at com.mongodb.internal.connection.SocketStreamHelper.initialize(SocketStreamHelper.java:76) ~[mongodb-driver-core-5.0.1.jar!/:na]
        at com.mongodb.internal.connection.SocketStream.initializeSocket(SocketStream.java:105) ~[mongodb-driver-core-5.0.1.jar!/:na]
        at com.mongodb.internal.connection.SocketStream.open(SocketStream.java:80) ~[mongodb-driver-core-5.0.1.jar!/:na]
        ... 4 common frames omitted

2024-07-08T07:32:06.475Z  INFO 1 --- [docker] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8080 (http) with context path '/'
2024-07-08T07:32:06.497Z  INFO 1 --- [docker] [           main] com.example.docker.DockerApplication     : Started DockerApplication in 1.429 seconds (process running
for 1.847)
hello docker image
```

- Create a jar file for the project (inside target directory)

  mvn package

  (The JAR file will be created in the `target` directory (`docker-java-app/target/docker-java-app.jar`)

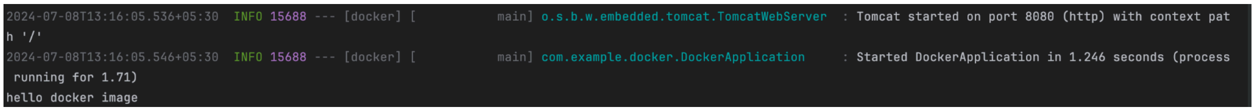
- Run the generated jar file inside target directory with command line



```
ahamedhsenid@ahamed_personal_laptop target % java -jar docker-0.0.1-SNAPSHOT.jar
```

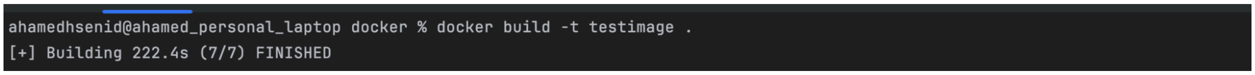
java -jar docker-0.0.1-SNAPSHOT.jar

- Display the output



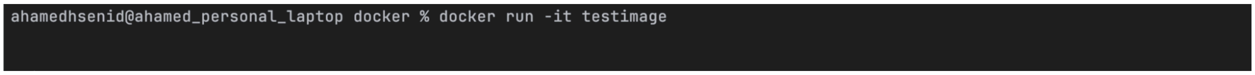
```
2024-07-08T13:16:05.536+05:30  INFO 15688 --- [docker] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8080 (http) with context pat
h '/'
2024-07-08T13:16:05.546+05:30  INFO 15688 --- [docker] [           main] com.example.docker.DockerApplication     : Started DockerApplication in 1.246 seconds (process
 running for 1.71)
hello docker image
```

- Create a docker image for the java project.



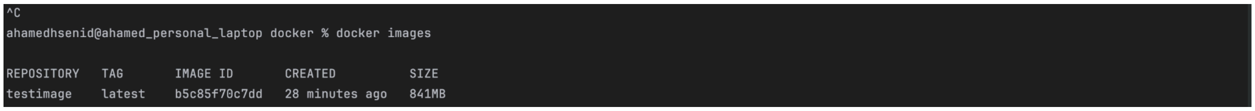
```
ahamedhsenid@ahamed_personal_laptop docker % docker build -t testimage .
[+] Building 222.4s (7/7) FINISHED
```

- Run the created docker image.



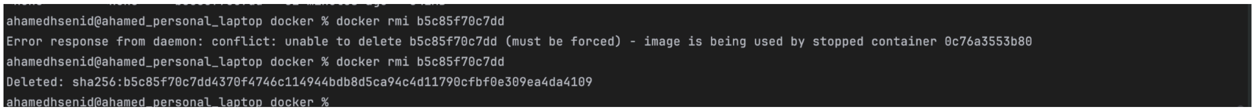
```
ahamedhsenid@ahamed_personal_laptop docker % docker run -it testimage
```

- List all the docker images and show output



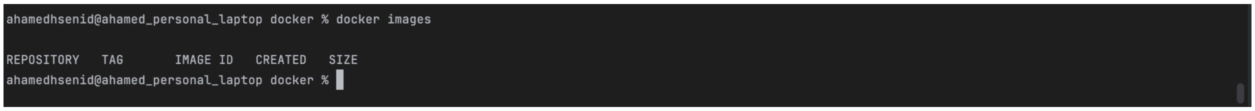
```
^C
ahamedhsenid@ahamed_personal_laptop docker % docker images

REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
testimage    latest    b5c85f70c7dd   28 minutes ago   841MB
```

- Remove the docker image.



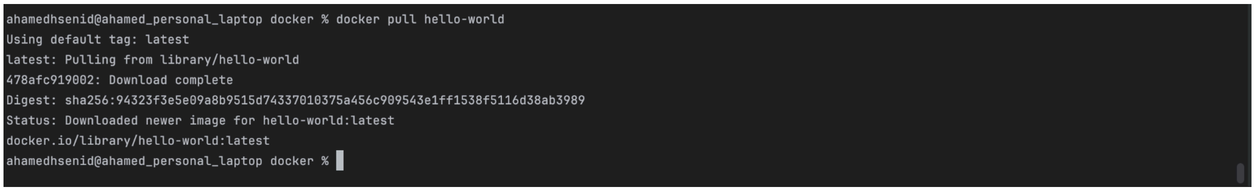
```
ahamedhsenid@ahamed_personal_laptop docker % docker rmi b5c85f70c7dd
Error response from daemon: conflict: unable to delete b5c85f70c7dd (must be forced) - image is being used by stopped container 0c76a3553b80
ahamedhsenid@ahamed_personal_laptop docker % docker rmi b5c85f70c7dd
Deleted: sha256:b5c85f70c7dd4370f4746c114944bdb8d5ca94c4d11790cfbf0e309ea4da4109
ahamedhsenid@ahamed_personal_laptop docker %
```

- List all the docker images and show output



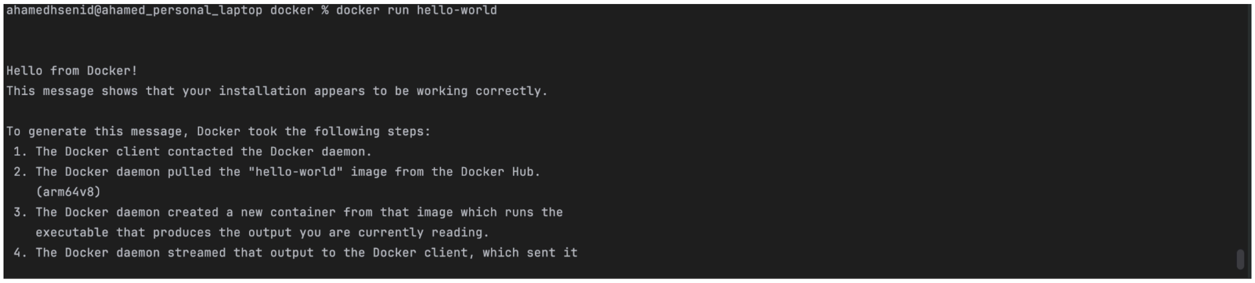
```
ahamedhsenid@ahamed_personal_laptop docker % docker images

REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
ahamedhsenid@ahamed_personal_laptop docker %
```

- Pull hello-world image from docker hub



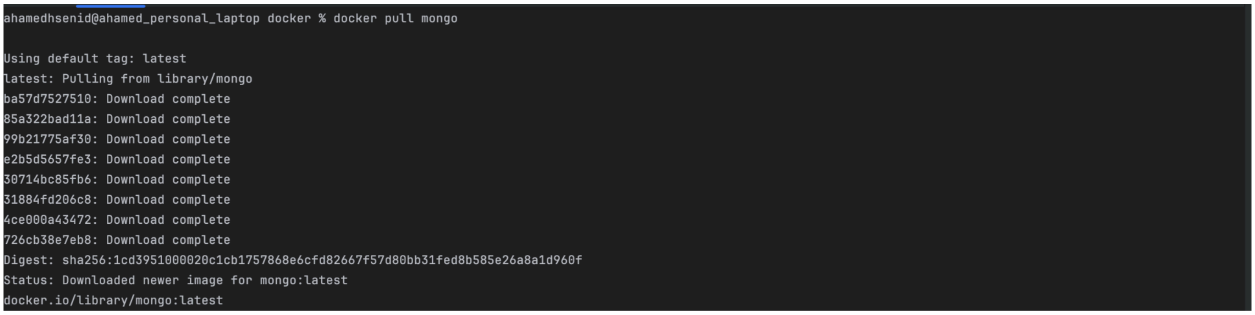
```
ahamedhsenid@ahamed_personal_laptop docker % docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
478afc919002: Download complete
Digest: sha256:94323f3e5e09a8b9515d74337010375a456c909543e1ff1538f5116d38ab3989
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
ahamedhsenid@ahamed_personal_laptop docker %
```

- Run hello-world image and show output

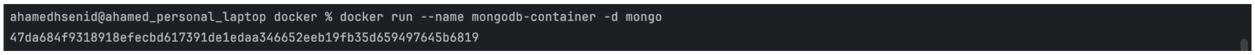


```
ahamedhsenid@ahamed_personal_laptop docker % docker run hello-world


Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (arm64v8)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
```

- Pull and run mongodb as docker container

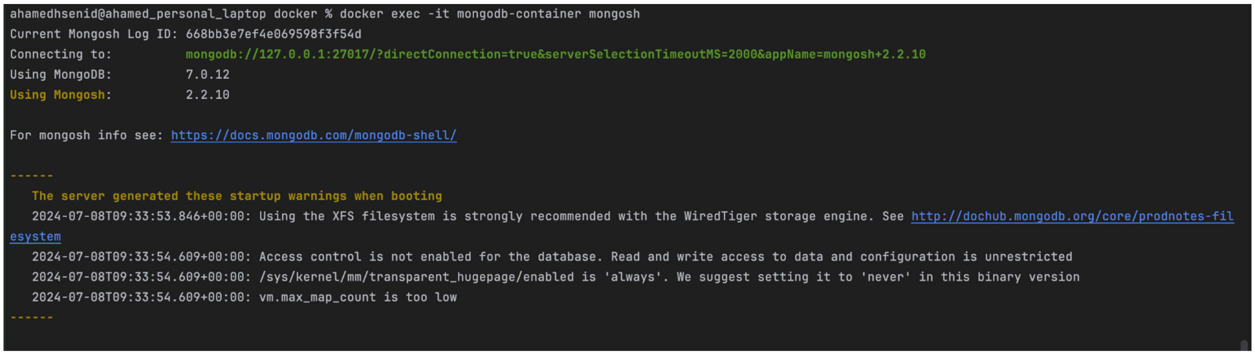


```
ahamedhsenid@ahamed_personal_laptop docker % docker pull mongo

Using default tag: latest
latest: Pulling from library/mongo
ba57d7527510: Download complete
85a322bad11a: Download complete
99b21775af30: Download complete
e2b5d5657fe3: Download complete
30714bc85fb6: Download complete
31884fd206c8: Download complete
4ce000a43472: Download complete
726cb38e7eb8: Download complete
Digest: sha256:1cd3951000020c1cb1757868e6cfd82667f57d80bb31fed8b585e26a8a1d960f
Status: Downloaded newer image for mongo:latest
docker.io/library/mongo:latest
```



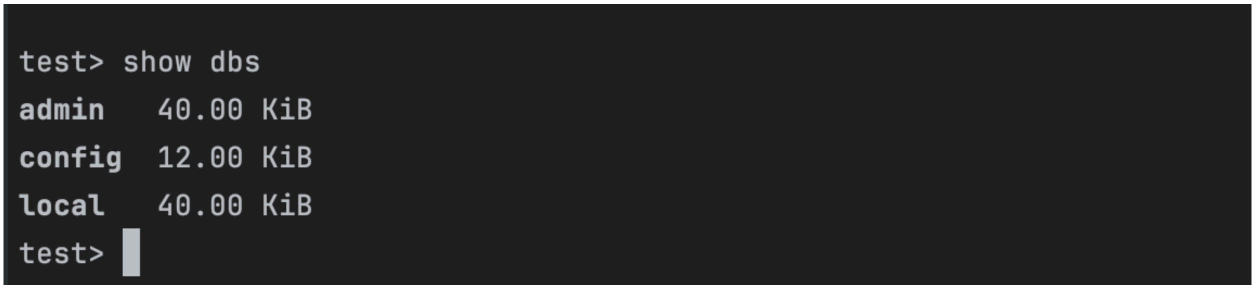
```
ahamedhsenid@ahamed_personal_laptop docker % docker run --name mongodb-container -d mongo
47da684f9318918efecbd617391de1edaa346652eeb19fb35d659497645b6819
```

- Open mongo shell



```
ahamedhsenid@ahamed_personal_laptop docker % docker exec -it mongodb-container mongosh
Current Mongosh Log ID: 668bb3e7ef4e069598f3f54d
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.10
Using MongoDB:          7.0.12
Using Mongosh:          2.2.10

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2024-07-08T09:33:53.846+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-fil
esystem
   2024-07-08T09:33:54.609+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
   2024-07-08T09:33:54.609+00:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting it to 'never' in this binary version
   2024-07-08T09:33:54.609+00:00: vm.max_map_count is too low
------
```

- List mongodb databases



```
test> show dbs
admin   40.00 KiB
config  12.00 KiB
local   40.00 KiB
test>
```