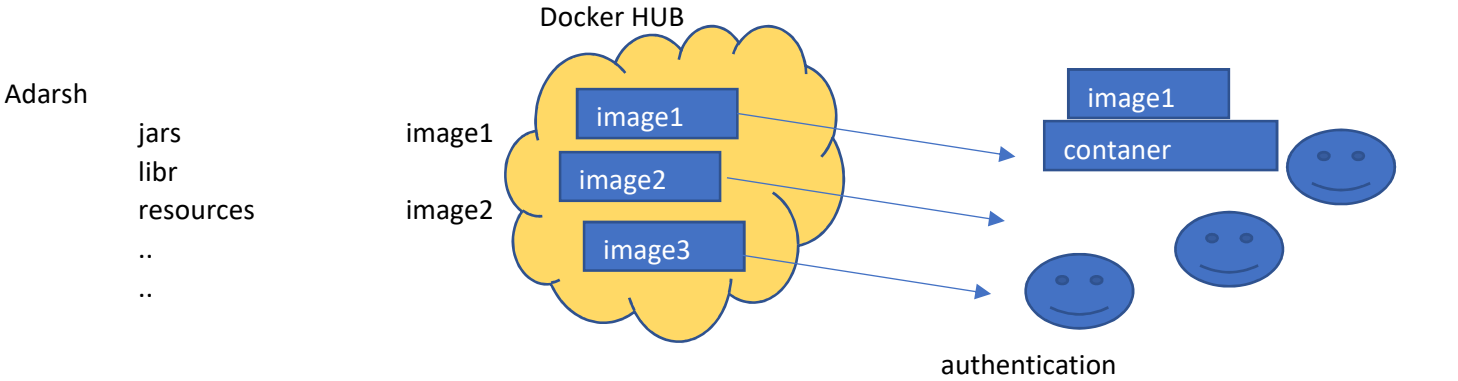
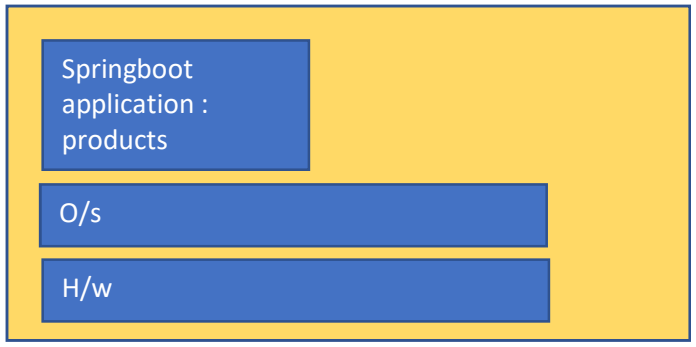
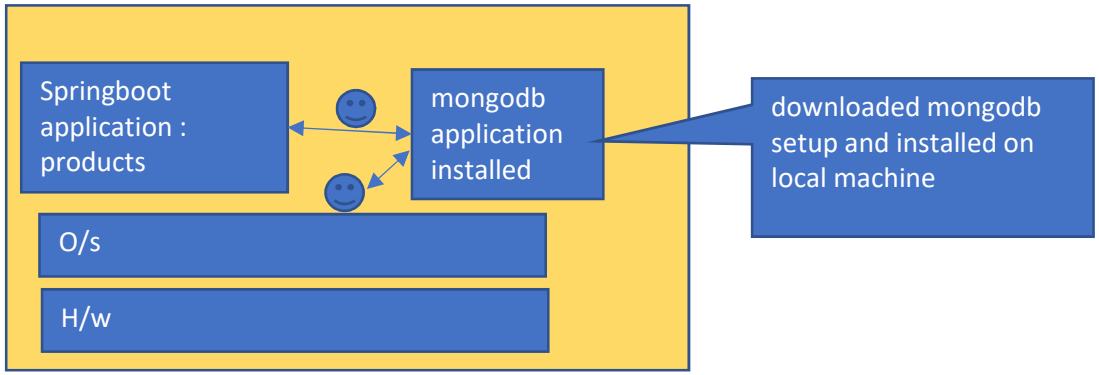


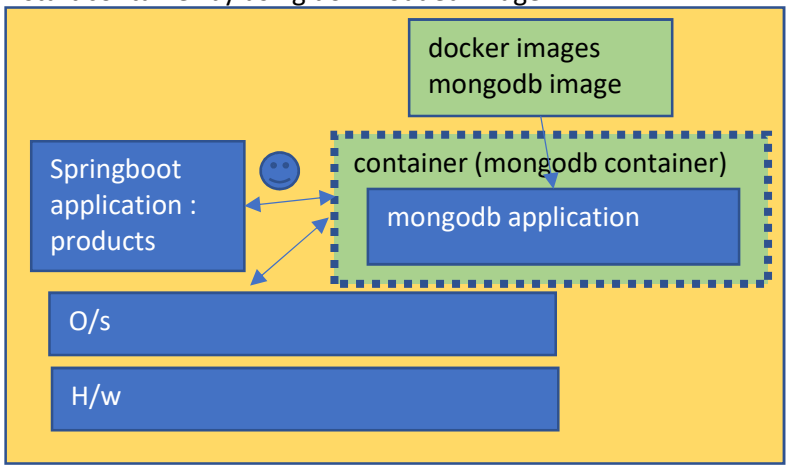


if I need mongodb application on this machine

Option 1 install mongodb application
make springboot application to connect with locally installed mongodb



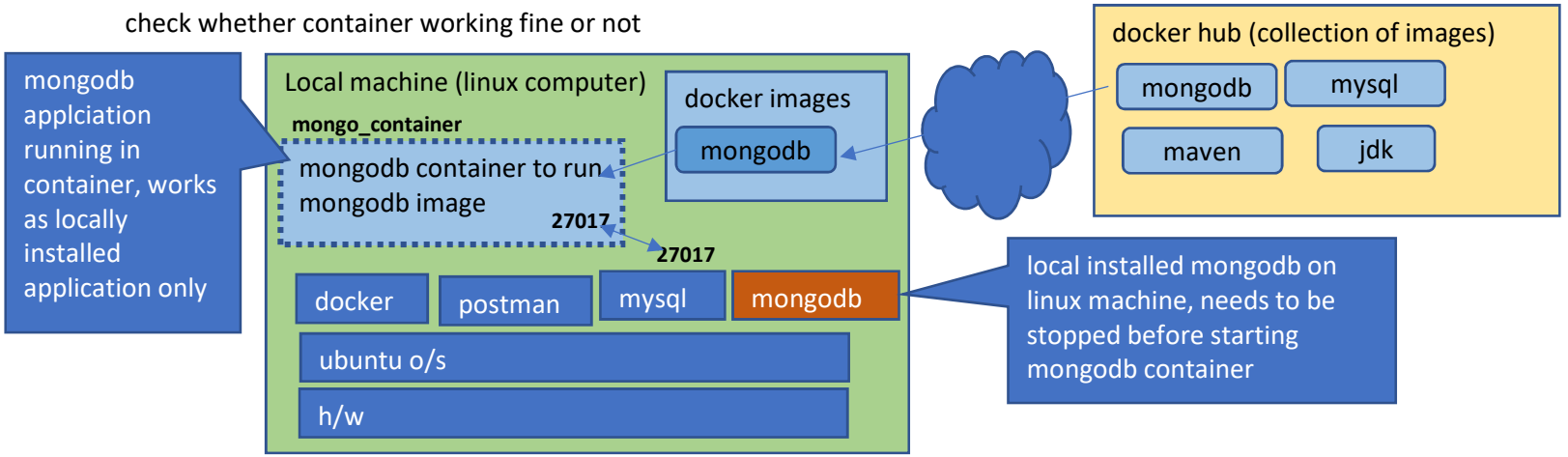
Option 2 pull mongdb image from dockerhub
start container by using downloaded image



mongodb application installed

Demo 1

Pull mongodb image from docker hub
create container with pulled mongodb image
check whether container working fine or not



Step 1 Login to VM (linux machine)

Step 2 Stop local installed and running mongodb application

sudo service mongod stop

Note: check whether service stoped or not

```
ubuntu@ip-172-31-36-21:~/Desktop$ sudo service mongod stop
ubuntu@ip-172-31-36-21:~/Desktop$ mongo
MongoDB shell version v5.0.2
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName
=mongodb
Error: couldn't connect to server 127.0.0.1:27017, connection attempt failed: So
cketException: Error connecting to 127.0.0.1:27017 :: caused by :: Connection re
fused :
connect@src/mongo/shell/mongo.js:372:17
@(connect):2:6
exception: connect failed
exiting with code 1
```

Step 3 Pull mongodb image from docker hub

commands

sudo docker images

shows list of docker images available on local machine

```
ubuntu@ip-172-31-36-21:~/Desktop$ sudo docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
ubuntu@ip-172-31-36-21:~/Desktop$
```

Note: actual docker command to get docker images is '**docker images**'

but $ user does not have permissions to execute docker commands in linux environment

so, **sudo** to be used in order to execute docker commands with required permissions

sudo docker pull mongo:3.4-jessie

pulls mongodb docker image from docker hub to local machine

Note: Make sure required free-space available on linux machine

```
ubuntu@ip-172-31-36-21:~/Desktop$ sudo docker pull mongo:3.4-jessie
3.4-jessie: Pulling from library/mongo
2a639da97f77: Pull complete
073b4f52defe: Pull complete
bce37d0f5c17: Pull complete
379dc19f9963: Pull complete
e44806c61e63: Pull complete
b76faf91d209: Pull complete
dd1d9be5b26b: Pull complete
9420e1982a2f: Pull complete
9ad2432e6a03: Pull complete
a80971ca2409: Pull complete
3a0937743e17: Pull complete
Digest: sha256:b39da8a18a6a9429f964f58d0da883d726f495dce3a00e3a7e67bd89cd16b40c
Status: Downloaded newer image for mongo:3.4-jessie
```

Note: Make sure image downloaded

```
ubuntu@ip-172-31-36-21:~/Desktop$ sudo docker images
REPOSITORY   TAG          IMAGE ID       CREATED       SIZE
mongo        3.4-jessie   f97f03a006c7   3 years ago   390MB
ubuntu@ip-172-31-36-21:~/Desktop$
```

Step 4 Start container by using pulled mongo image

sudo run --name yyyyyy -p 27017:27017 DOCKERIMAGE

sudo run --name mongo_container -p 27017:27017 mongo:3.4-jessie

```
ubuntu@ip-172-31-36-21:~/Desktop$ sudo docker run --name mongo_container -p 27017:27017 mongo:3.4-jessie
2022-10-26T06:43:48.443+0000 I CONTROL  [initandlisten] MongoDB starting : pid=1 port=27017 dbpath=/data/db
 64-bit host=b35327fcebb6
2022-10-26T06:43:48.443+0000 I CONTROL  [initandlisten] db version v3.4.20
2022-10-26T06:43:48.444+0000 I CONTROL  [initandlisten] git version: 447847d93d6e0a21b018d5df45528e815c7c13
d8
2022-10-26T06:43:48.444+0000 I CONTROL  [initandlisten] OpenSSL version: OpenSSL 1.0.1t  3 May 2016
2022-10-26T06:43:48.444+0000 I CONTROL  [initandlisten] allocator: tcmalloc
```

sudo run --name mongo_container -d -p 27017:27017 mongo:3.4-jessie

container runs in detached mode, prompt will be back in the same terminal

Make sure container started and running

open other terminal window if required

sudo docker ps -a

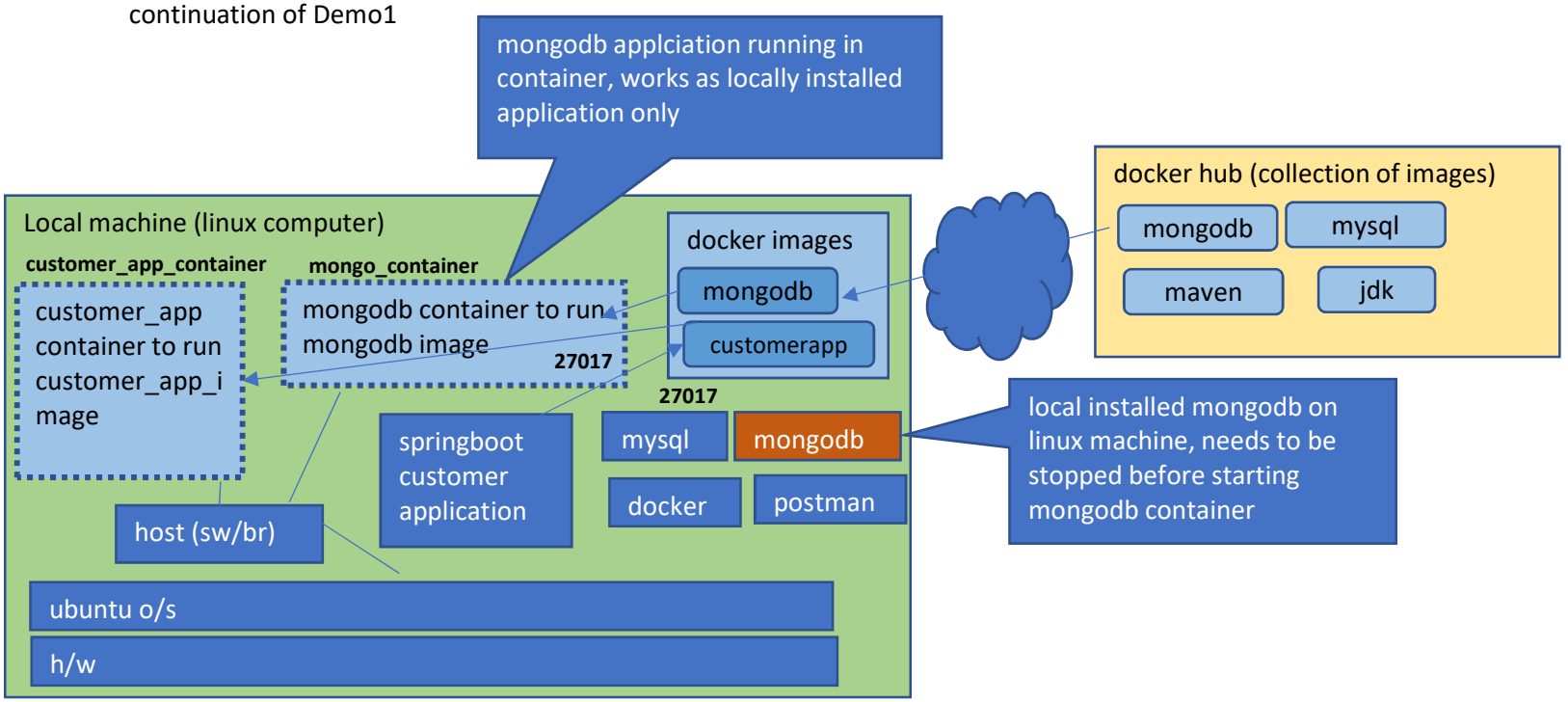
```
ubuntu@ip-172-31-36-21:~/Desktop$ sudo docker ps -a
CONTAINER ID   IMAGE            COMMAND                  CREATED         STATUS         PORTS                                                NAMES
b35327fcebb6   mongo:3.4-jessie   "docker-entrypoint.s…"   3 minutes ago   Up 3 minutes   0.0.0.0:27017->27017/tcp, :::27017->27017/tcp   mongo_container
```

Step 5 check mongo container by logging into mongo

```
ubuntu@ip-172-31-36-21:~/Desktop$ mongo
MongoDB shell version v5.0.2
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServi
WARNING: No implicit session: Logical Sessions are only supported on serve
Implicit session: dummy session
MongoDB server version: 3.4.20
WARNING: shell and server versions do not match
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has
an upcoming release.
We recommend you begin using "mongosh".
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
2022-10-26T06:43:48.450+0000 I STORAGE  [initandlisten]
2022-10-26T06:43:48.451+0000 I STORAGE  [initandlisten] ** WARNING: Using
2022-10-26T06:43:48.451+0000 I STORAGE  [initandlisten] **          See ht
2022-10-26T06:43:48.520+0000 I CONTROL  [initandlisten]
2022-10-26T06:43:48.520+0000 I CONTROL  [initandlisten] ** WARNING: Access
2022-10-26T06:43:48.521+0000 I CONTROL  [initandlisten] **          Read a
2022-10-26T06:43:48.521+0000 I CONTROL  [initandlisten]
---
> show dbs;
admin  0.000GB
local  0.000GB
```

Demo 2

continuation of Demo1



Make sure MongoDB container running behind and working

Steps to create container for our own spring boot application

Step 1 Make sure spring boot application is running on linux machine

Use slack to copy project from windows machine to linux machine

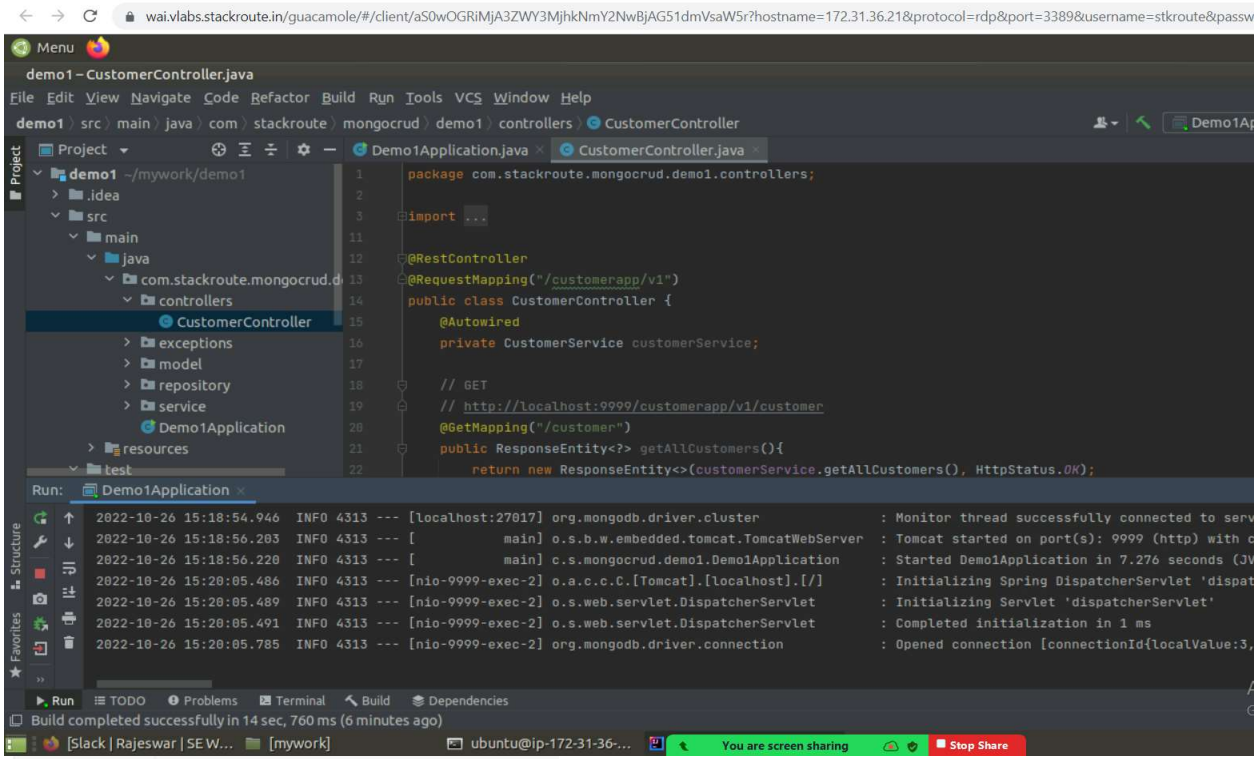
download project in linux machine using slack in linux machine

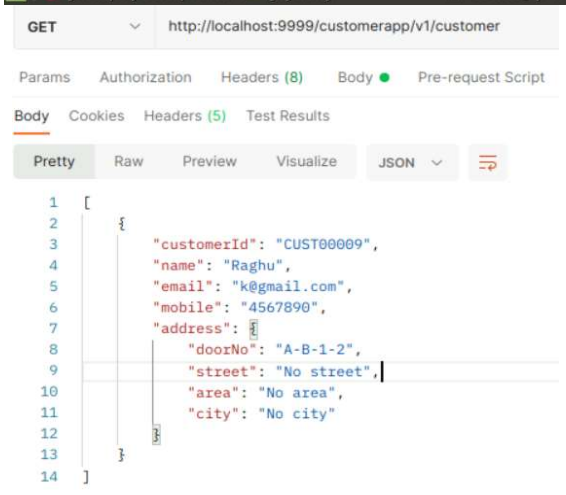
Open downloaded project in intellij on linux machine

Add JDK and Maven tools to project if required

Make sure spring boot application is working fine and connecting to mongodb container

Use postman to test springboot application


wai.vlabs.stackroute.in/guacamole/#/client/aS0wOGRiMjA3ZWY3MjhkNmY2NwBjAG51dmVsaW5r?hostname=172.31.36.21&protocol=rdp&port=3389&username=stkroute&passw
Menu
demo1 – CustomerController.java
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
demo1 src main java com stackroute mongocrud demo1 controllers CustomerController
Project
demo1 ~/mywork/demo1
.idea
src
main
java
com.stackroute.mongocrud.d
controllers
CustomerController
exceptions
model
repository
service
Demo1Application
resources
test
Demo1Application.java
CustomerController.java
package com.stackroute.mongocrud.demo1.controllers;
import ...
@RestController
@RequestMapping("/customerapp/v1")
public class CustomerController {
@Autowired
private CustomerService customerService;
// GET
// http://localhost:9999/customerapp/v1/customer
@GetMapping("/customer")
public ResponseEntity<?> getAllCustomers(){
return new ResponseEntity<>(customerService.getAllCustomers(), HttpStatus.OK);
Run: Demo1Application
2022-10-26 15:18:54.946 INFO 4313 --- [localhost:27017] org.mongodb.driver.cluster : Monitor thread successfully connected to serv
2022-10-26 15:18:56.203 INFO 4313 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 9999 (http) with c
2022-10-26 15:18:56.220 INFO 4313 --- [main] c.s.mongocrud.demo1.Demo1Application : Started Demo1Application in 7.276 seconds (JV
2022-10-26 15:20:05.486 INFO 4313 --- [nio-9999-exec-2] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispat
2022-10-26 15:20:05.489 INFO 4313 --- [nio-9999-exec-2] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2022-10-26 15:20:05.491 INFO 4313 --- [nio-9999-exec-2] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
2022-10-26 15:20:05.785 INFO 4313 --- [nio-9999-exec-2] org.mongodb.driver.connection : Opened connection [connectionId{localValue:3,
Run TODO Problems Terminal Build Dependencies
Build completed successfully in 14 sec, 760 ms (6 minutes ago)
[Slack | Rajeswar | SE W... [mywork] ubuntu@ip-172-31-36-...
You are screen sharing
Stop Share


GET
http://localhost:9999/customerapp/v1/customer
Params Authorization Headers (8) Body Pre-request Script
Body Cookies Headers (5) Test Results
Pretty Raw Preview Visualize JSON
[
{
"customerId": "CUST00009",
"name": "Raghu",
"email": "k@gmail.com",
"mobile": "4567890",
"address": {
"doorNo": "A-B-1-2",
"street": "No street",
"area": "No area",
"city": "No city"
}
}
]

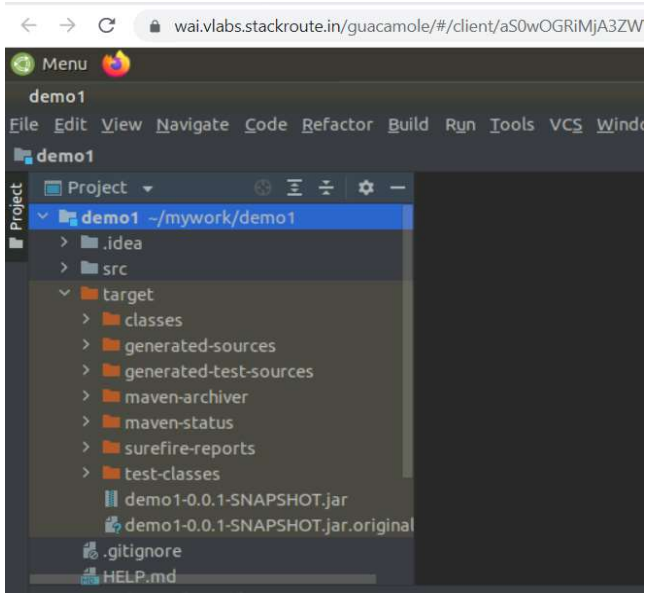

> db.customer.find().pretty();
{
"_id" : "CUST00009",
"name" : "Raghu",
"email" : "k@gmail.com",
"mobile" : "4567890",
"address" : {
"doorNo" : "A-B-1-2",
"street" : "No street",
"area" : "No area",
"city" : "No city"
},
"_class" : "com.stackroute.mongocrud.demo1.model.Customer"
}
{
"_id" : "CUST00005",
"name" : "Raghu1",
"email" : "k@gmail.com1",
"mobile" : "45678901",
"address" : {
"doorNo" : "A-B-1-2-1",
"street" : "No street1",
"area" : "No area1",
"city" : "No city1"
},
"_class" : "com.stackroute.mongocrud.demo1.model.Customer"
}

Checkpoint

Step 2 Create target folder and jar files

mvn clean

mvn install

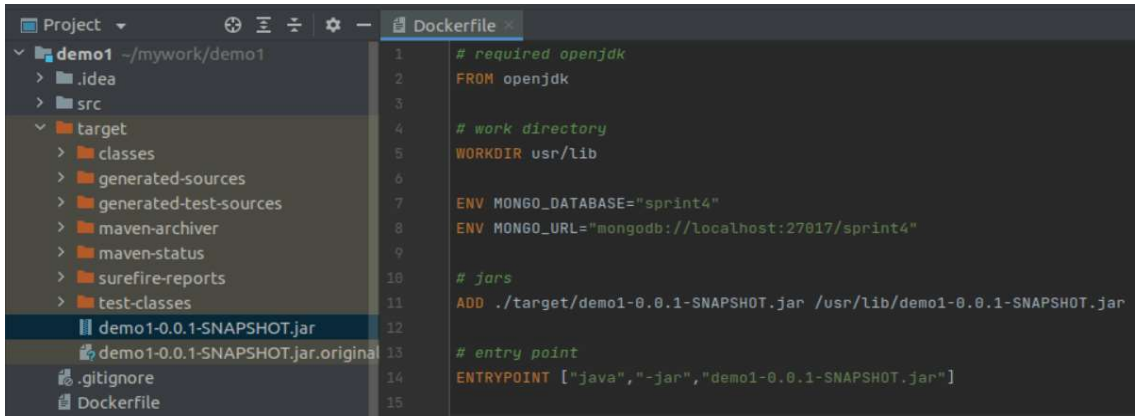


Step 3 Create docker file in project root folder

Dockerfile

plain text file

holds configuration to create docker image for our springboot application



Step 4 execute docker build command in same location where Dockerfile created

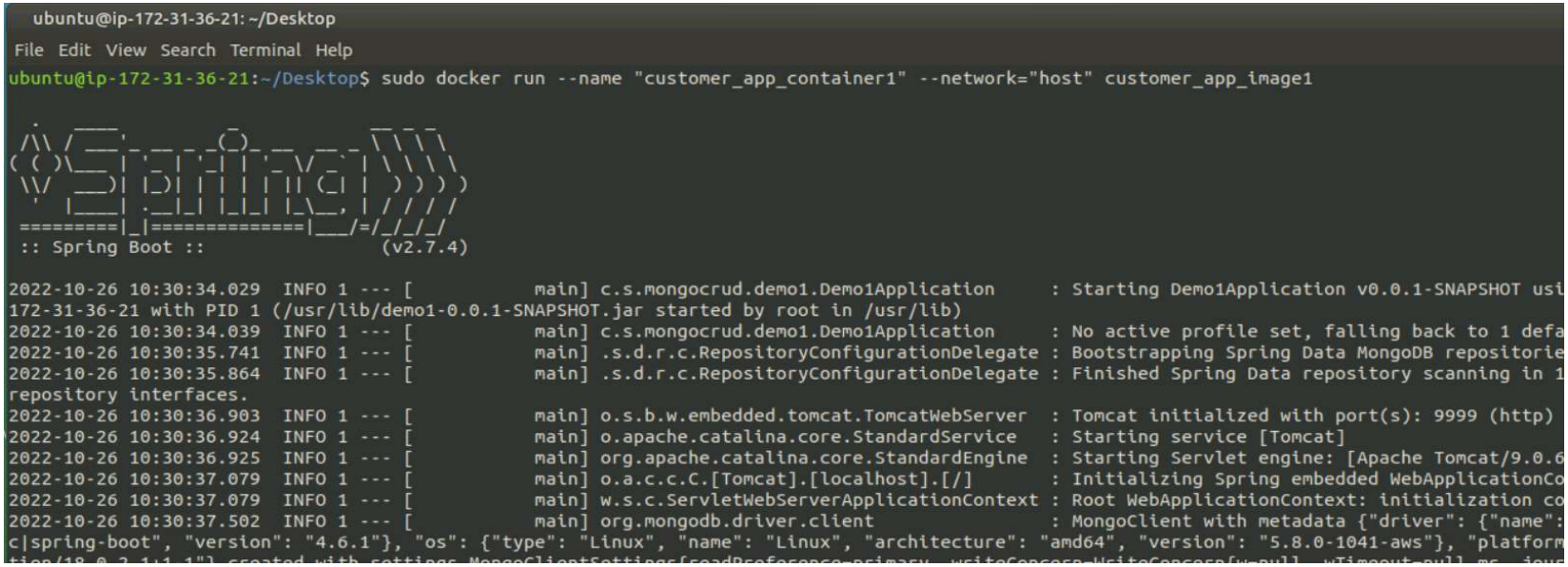
sudo docker build -t "customer_app_image1" .

```
Terminal:  Local  +
ubuntu@ip-172-31-36-21:~/mywork/demo1$ sudo docker build -t "customer_app_image1" .
Sending build context to Docker daemon   23.9MB
Step 1/6 : FROM openjdk
latest: Pulling from library/openjdk
50cbc88660a5: Pull complete
3f15da7b20d8: Pull complete
812b9f471c4d: Pull complete
Digest: sha256:448d8240b4e40a51ebe9710cf032d512457182233c4646681a58262efb15fd2d
Status: Downloaded newer image for openjdk:latest
 ---> d3df331637f8
Step 2/6 : WORKDIR usr/lib
 ---> Running in 4a193bf383fa
Removing intermediate container 4a193bf383fa
 ---> 61a5dbecd207
Step 3/6 : ENV MONGO_DATABASE="sprint4"
 ---> Running in ab8eb0dd6f2e
Removing intermediate container ab8eb0dd6f2e
 ---> 2eea6a8cb809
Step 4/6 : ENV MONGO_URL="mongodb://localhost:27017/sprint4"
 ---> Running in b2e0ec96c15c
Removing intermediate container b2e0ec96c15c
 ---> 4610d79b414d
Step 5/6 : ADD ./target/demo1-0.0.1-SNAPSHOT.jar /usr/lib/demo1-0.0.1-SNAPSHOT.jar
 ---> ec26a9221d83
Step 6/6 : ENTRYPOINT ["java","-jar","demo1-0.0.1-SNAPSHOT.jar"]
 ---> Running in c66636892924
Removing intermediate container c66636892924
 ---> a092f56acd88
Successfully built a092f56acd88
Successfully tagged customer_app_image1:latest
ubuntu@ip-172-31-36-21:~/mywork/demo1$
```

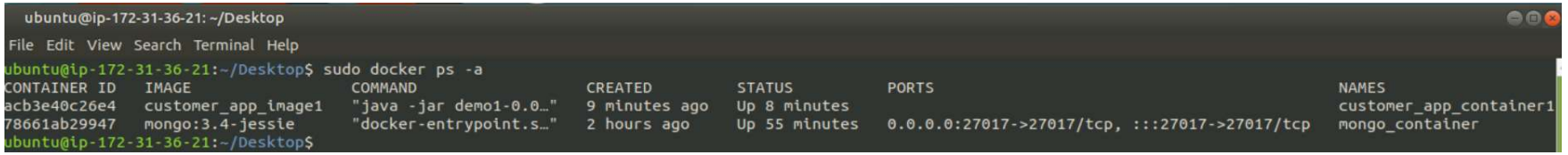
```
ubuntu@ip-172-31-36-21:~/mywork/demo1$ sudo docker images
REPOSITORY          TAG         IMAGE ID       CREATED             SIZE
customer_app_image1 latest      a092f56acd88   About a minute ago  488MB
openjdk             latest      d3df331637f8   4 days ago          464MB
mongo               3.4-jessie  f97f03a006c7   3 years ago         390MB
```

Step 5 Create and run container using above created image

sudo docker run --name "customer_app_container1" --network="host" customer_app_image1



Make sure both containers running (mongo,customerapp)



use postman to test few transactions