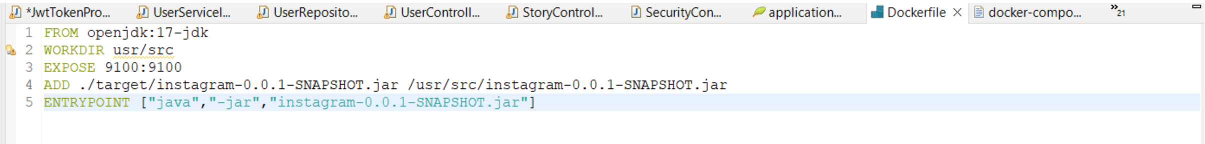
SMARTBRIDGE EXTERNSHIP: MODERN APPLICATION DEVELOPMENT: SOCIAL MEDIA CLONE: VSOCIAL

TOPIC: CREATING A DOCKER IMAGE AND RUNNING IT

APPLICATIONS NEEDED: DOCKER FOR DESKTOP

Our first step was to create a Dockerfile in the same folder where we have our pom.xml file.

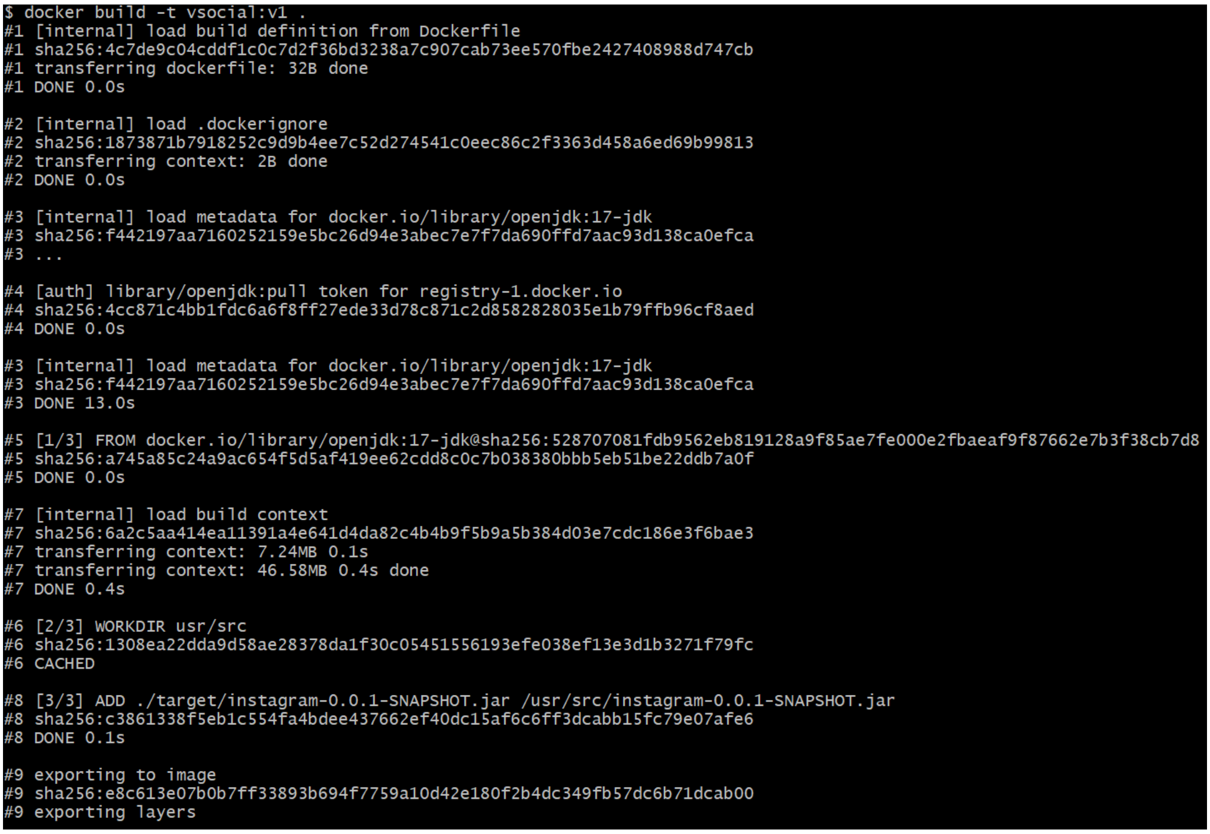


```
FROM openjdk:17-jdk
WORKDIR usr/src
EXPOSE 9100:9100
ADD ./target/instagram-0.0.1-SNAPSHOT.jar /usr/src/instagram-0.0.1-SNAPSHOT.jar
ENTRYPOINT ["java","-jar","instagram-0.0.1-SNAPSHOT.jar"]
```

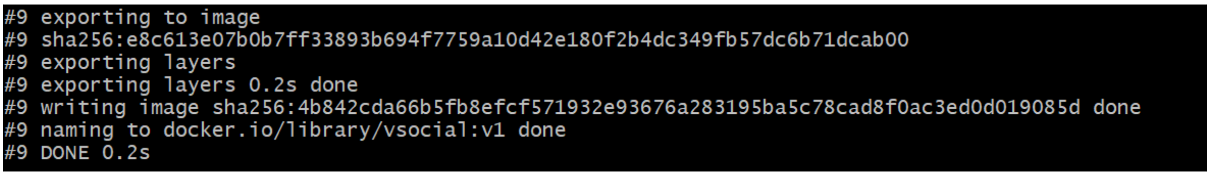
We are using openjdk:17-jdk as the base image, setting up a working directory and copying the jar file to the directory. To get the jar file, we can use the command **mvn clean install -dSkipTests=true**

After creating a Dockerfile, we created our image. For this we accessed the server through docker desktop

To build our docker image we use the command **docker build -t vsocial:v1** (vsocial: v1 is simply the name of our image)

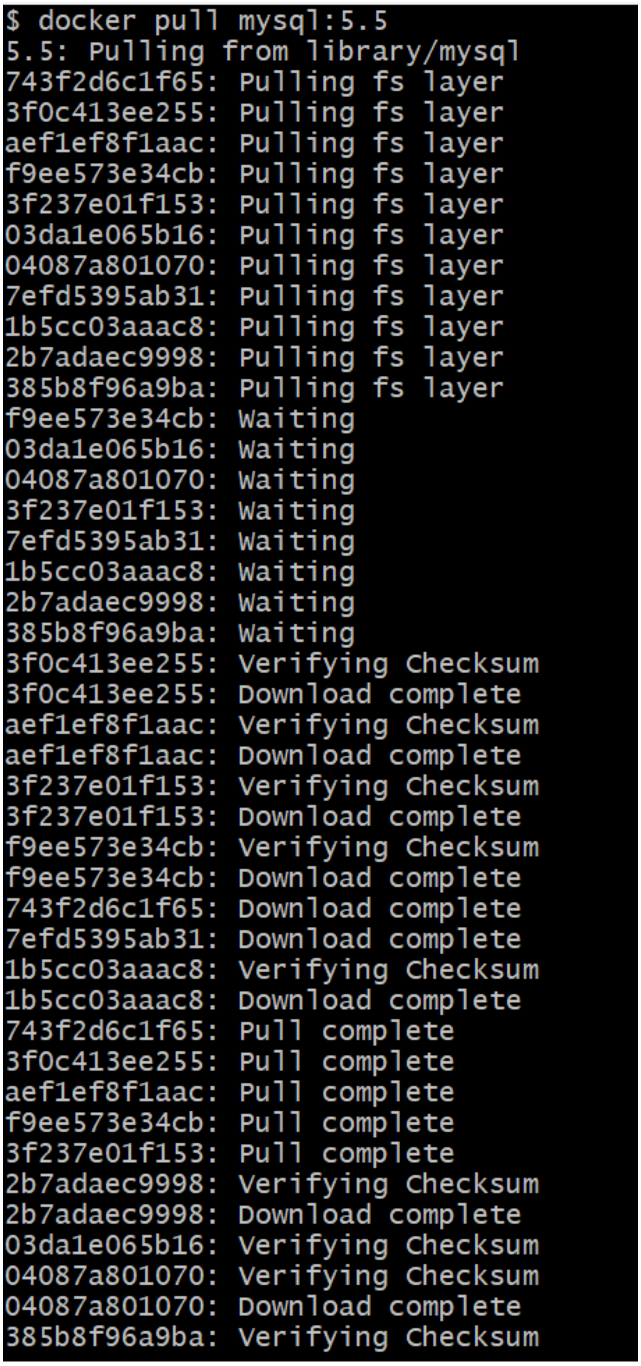


```
$ docker build -t vsocial:v1 .
#1 [internal] load build definition from Dockerfile
#1 sha256:4c7de9c04cddf1c0c7d2f36bd3238a7c907cab73ee570fbe2427408988d747cb
#1 transferring dockerfile: 32B done
#1 DONE 0.0s

#2 [internal] load .dockerignore
#2 sha256:1873871b7918252c9d9b4ee7c52d274541c0eec86c2f3363d458a6ed69b99813
#2 transferring context: 2B done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/library/openjdk:17-jdk
#3 sha256:f442197aa7160252159e5bc26d94e3abec7e7f7da690ffd7aac93d138ca0efca
#3 ...

#4 [auth] library/openjdk:pull token for registry-1.docker.io
#4 sha256:4cc871c4bb1fdc6a6f8ff27ede33d78c871c2d8582828035e1b79ffb96cf8aed
#4 DONE 0.0s

#3 [internal] load metadata for docker.io/library/openjdk:17-jdk
#3 sha256:f442197aa7160252159e5bc26d94e3abec7e7f7da690ffd7aac93d138ca0efca
#3 DONE 13.0s

#5 [1/3] FROM docker.io/library/openjdk:17-jdk@sha256:528707081fdb9562eb819128a9f85ae7fe000e2fbaeaf9f87662e7b3f38cb7d8
#5 sha256:a745a85c24a9ac654f5d5af419ee62cdd8c0c7b038380bbb5eb51be22ddb7a0f
#5 DONE 0.0s

#7 [internal] load build context
#7 sha256:6a2c5aa414ea11391a4e641d4da82c4b4b9f5b9a5b384d03e7cdc186e3f6bae3
#7 transferring context: 7.24MB 0.1s
#7 transferring context: 46.58MB 0.4s done
#7 DONE 0.4s

#6 [2/3] WORKDIR usr/src
#6 sha256:1308ea22dda9d58ae28378da1f30c05451556193efe038ef13e3d1b3271f79fc
#6 CACHED

#8 [3/3] ADD ./target/instagram-0.0.1-SNAPSHOT.jar /usr/src/instagram-0.0.1-SNAPSHOT.jar
#8 sha256:c3861338f5eb1c554fa4bdee437662ef40dc15af6c6ff3dcabb15fc79e07afe6
#8 DONE 0.1s

#9 exporting to image
#9 sha256:e8c613e07b0b7ff33893b694f7759a10d42e180f2b4dc349fb57dc6b71dcab00
#9 exporting layers
```

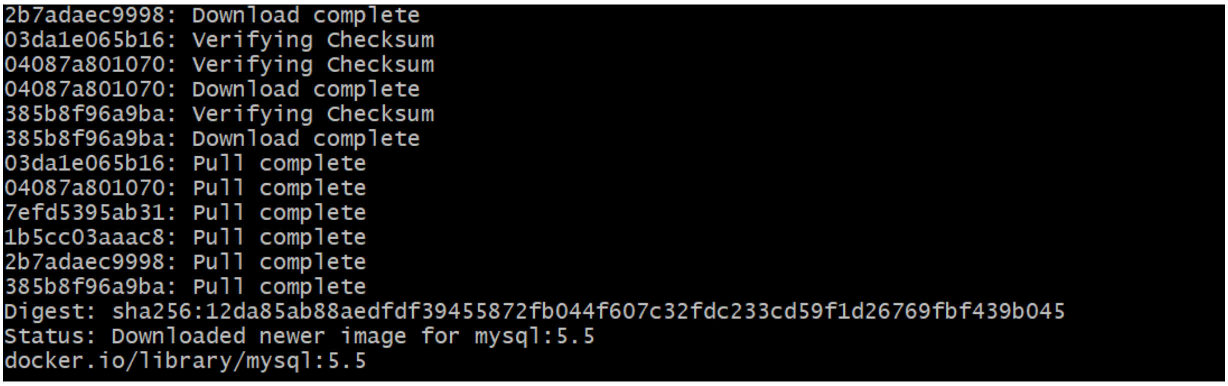


```
#9 exporting to image
#9 sha256:e8c613e07b0b7ff33893b694f7759a10d42e180f2b4dc349fb57dc6b71dcab00
#9 exporting layers
#9 exporting layers 0.2s done
#9 writing image sha256:4b842cda66b5fb8efcf571932e93676a283195ba5c78cad8f0ac3ed0d019085d done
#9 naming to docker.io/library/vsocial:v1 done
#9 DONE 0.2s
```

Upon building our image, we pull the mysql image from docker hub using the command **docker pull mysql:5.5**

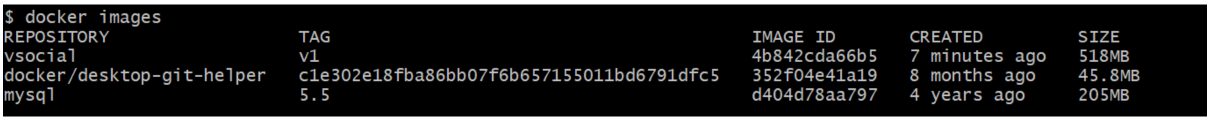


```
$ docker pull mysql:5.5
5.5: Pulling from library/mysql
743f2d6c1f65: Pulling fs layer
3f0c413ee255: Pulling fs layer
aef1ef8f1aac: Pulling fs layer
f9ee573e34cb: Pulling fs layer
3f237e01f153: Pulling fs layer
03da1e065b16: Pulling fs layer
04087a801070: Pulling fs layer
7efd5395ab31: Pulling fs layer
1b5cc03aaac8: Pulling fs layer
2b7adaec9998: Pulling fs layer
385b8f96a9ba: Pulling fs layer
f9ee573e34cb: Waiting
03da1e065b16: Waiting
04087a801070: Waiting
3f237e01f153: Waiting
7efd5395ab31: Waiting
1b5cc03aaac8: Waiting
2b7adaec9998: Waiting
385b8f96a9ba: Waiting
3f0c413ee255: Verifying Checksum
3f0c413ee255: Download complete
aef1ef8f1aac: Verifying Checksum
aef1ef8f1aac: Download complete
3f237e01f153: Verifying Checksum
3f237e01f153: Download complete
f9ee573e34cb: Verifying Checksum
f9ee573e34cb: Download complete
743f2d6c1f65: Download complete
7efd5395ab31: Download complete
1b5cc03aaac8: Verifying Checksum
1b5cc03aaac8: Download complete
743f2d6c1f65: Pull complete
3f0c413ee255: Pull complete
aef1ef8f1aac: Pull complete
f9ee573e34cb: Pull complete
3f237e01f153: Pull complete
2b7adaec9998: Verifying Checksum
2b7adaec9998: Download complete
03da1e065b16: Verifying Checksum
04087a801070: Verifying Checksum
04087a801070: Download complete
385b8f96a9ba: Verifying Checksum
```



```
2b7adaec9998: Download complete
03da1e065b16: Verifying Checksum
04087a801070: Verifying Checksum
04087a801070: Download complete
385b8f96a9ba: Verifying Checksum
385b8f96a9ba: Download complete
03da1e065b16: Pull complete
04087a801070: Pull complete
7efd5395ab31: Pull complete
1b5cc03aaac8: Pull complete
2b7adaec9998: Pull complete
385b8f96a9ba: Pull complete
Digest: sha256:12da85ab88aedfdf39455872fb044f607c32fdc233cd59f1d26769fbf439b045
Status: Downloaded newer image for mysql:5.5
docker.io/library/mysql:5.5
```

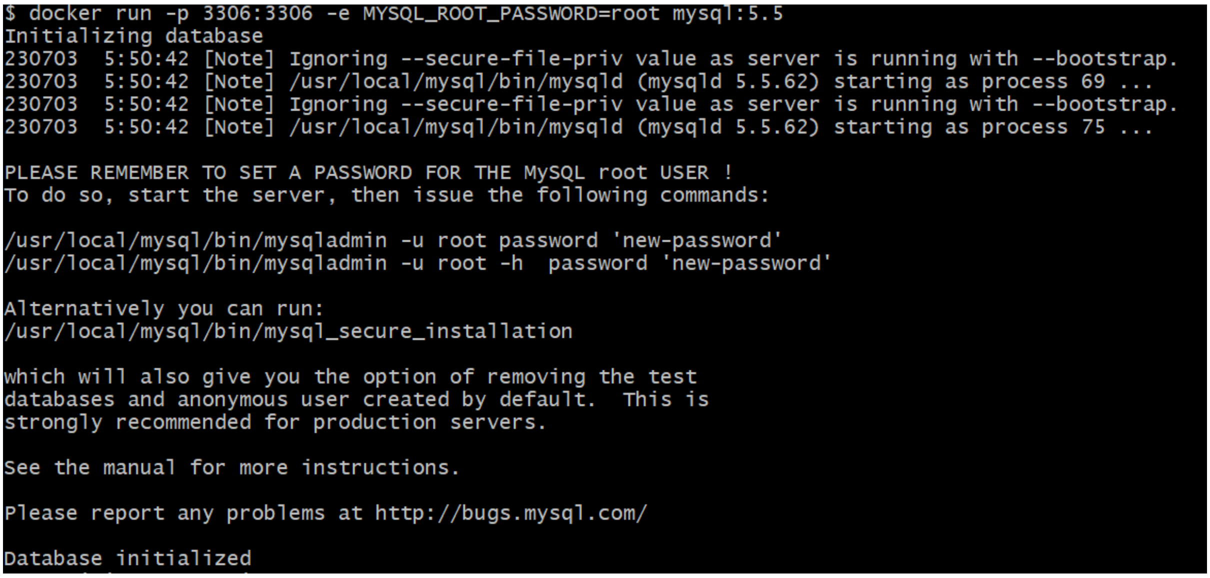
Upon pulling the mysql image, we cross check whether the images have been created and pulled successfully using the command **docker images.**



```
$ docker images
REPOSITORY                 TAG                                        IMAGE ID       CREATED         SIZE
vsocial                    v1                                         4b842cda66b5   7 minutes ago   518MB
docker/desktop-git-helper  c1e302e18fba86bb07f6b657155011bd6791dfc5   352f04e41a19   8 months ago    45.8MB
mysql                      5.5                                        d404d78aa797   4 years ago     205MB
```

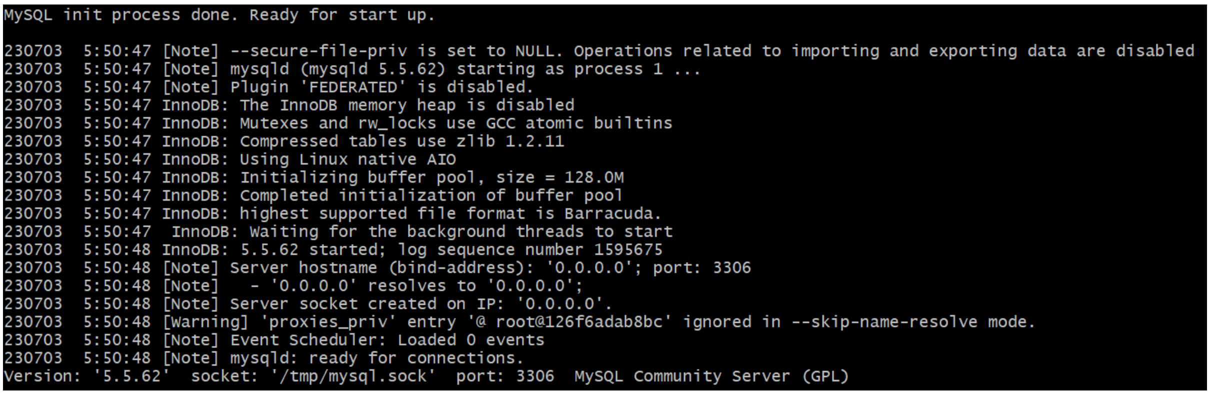
As you can see, the vsocial image was created 7 minutes ago and the MySQL image was created 4 years ago. The MySQL image has been pulled from Docker Hub.

Now we are going to run our MySQL image with the command **docker run -p 3306:3306 -e MYSQL_ROOT_PASSWORD=root mysql: 5.5**



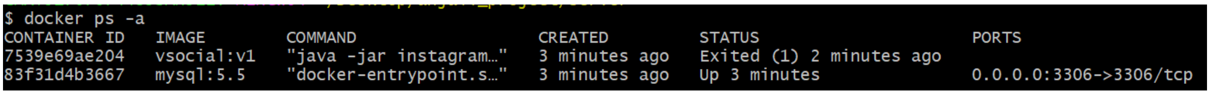
```
$ docker run -p 3306:3306 -e MYSQL_ROOT_PASSWORD=root mysql:5.5
Initializing database
230703  5:50:42 [Note] Ignoring --secure-file-priv value as server is running with --bootstrap.
230703  5:50:42 [Note] /usr/local/mysql/bin/mysqld (mysqld 5.5.62) starting as process 69 ...
230703  5:50:42 [Note] Ignoring --secure-file-priv value as server is running with --bootstrap.
230703  5:50:42 [Note] /usr/local/mysql/bin/mysqld (mysqld 5.5.62) starting as process 75 ...

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:

/usr/local/mysql/bin/mysqladmin -u root password 'new-password'
/usr/local/mysql/bin/mysqladmin -u root -h  password 'new-password'

Alternatively you can run:
/usr/local/mysql/bin/mysql_secure_installation

which will also give you the option of removing the test
databases and anonymous user created by default.  This is
strongly recommended for production servers.

See the manual for more instructions.

Please report any problems at http://bugs.mysql.com/

Database initialized
```

….



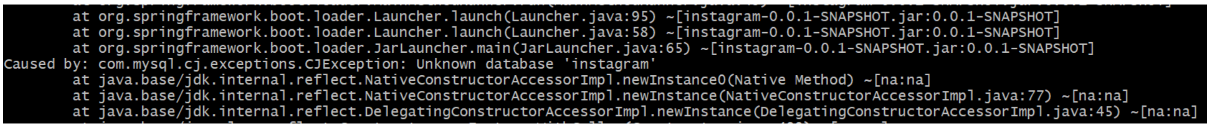
```
MySQL init process done. Ready for start up.

230703  5:50:47 [Note] --secure-file-priv is set to NULL. Operations related to importing and exporting data are disabled
230703  5:50:47 [Note] mysqld (mysqld 5.5.62) starting as process 1 ...
230703  5:50:47 [Note] Plugin 'FEDERATED' is disabled.
230703  5:50:47 InnoDB: The InnoDB memory heap is disabled
230703  5:50:47 InnoDB: Mutexes and rw_locks use GCC atomic builtins
230703  5:50:47 InnoDB: Compressed tables use zlib 1.2.11
230703  5:50:47 InnoDB: Using Linux native AIO
230703  5:50:47 InnoDB: Initializing buffer pool, size = 128.0M
230703  5:50:47 InnoDB: Completed initialization of buffer pool
230703  5:50:47 InnoDB: highest supported file format is Barracuda.
230703  5:50:47  InnoDB: Waiting for the background threads to start
230703  5:50:48 InnoDB: 5.5.62 started; log sequence number 1595675
230703  5:50:48 [Note] Server hostname (bind-address): '0.0.0.0'; port: 3306
230703  5:50:48 [Note]   - '0.0.0.0' resolves to '0.0.0.0';
230703  5:50:48 [Note] Server socket created on IP: '0.0.0.0'.
230703  5:50:48 [Warning] 'proxies_priv' entry '@ root@126f6adab8bc' ignored in --skip-name-resolve mode.
230703  5:50:48 [Note] Event Scheduler: Loaded 0 events
230703  5:50:48 [Note] mysqld: ready for connections.
Version: '5.5.62'  socket: '/tmp/mysql.sock'  port: 3306  MySQL Community Server (GPL)
```

After the database was successfully ready for connection, we tried running our vsocial image and connecting it to the database, but we ran into an error (Which was eventually rectified. Herein, I'll explain the steps and modifications we made)



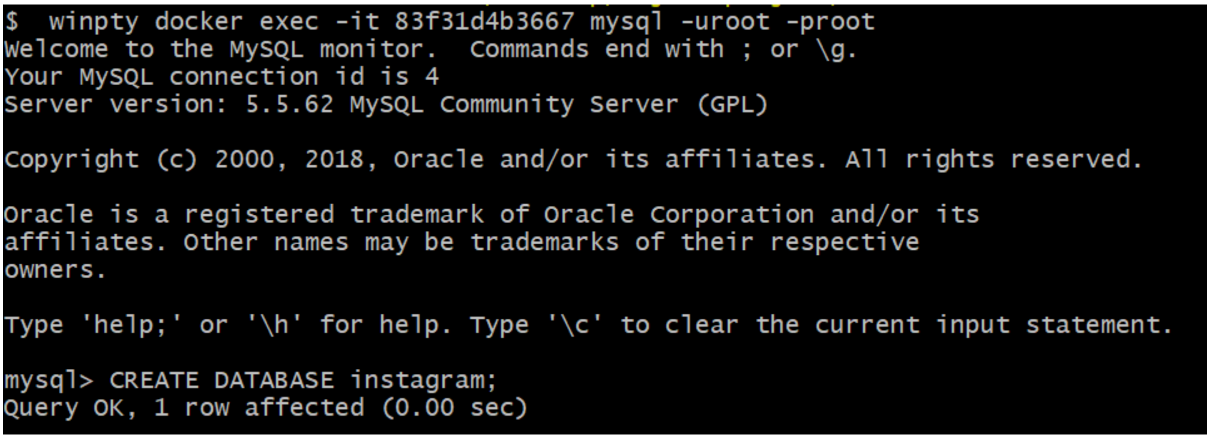
```
$ docker ps -a
CONTAINER ID   IMAGE        COMMAND                  CREATED         STATUS                      PORTS
7539e69ae204   vsocial:v1   "java -jar instagram…"   3 minutes ago   Exited (1) 2 minutes ago
83f31d4b3667   mysql:5.5    "docker-entrypoint.s…"   3 minutes ago   Up 3 minutes                0.0.0.0:3306->3306/tcp
```

In our application properties file, we replaced the local host name with the IP address of the local system. The error was also caused by the database not being recognized.



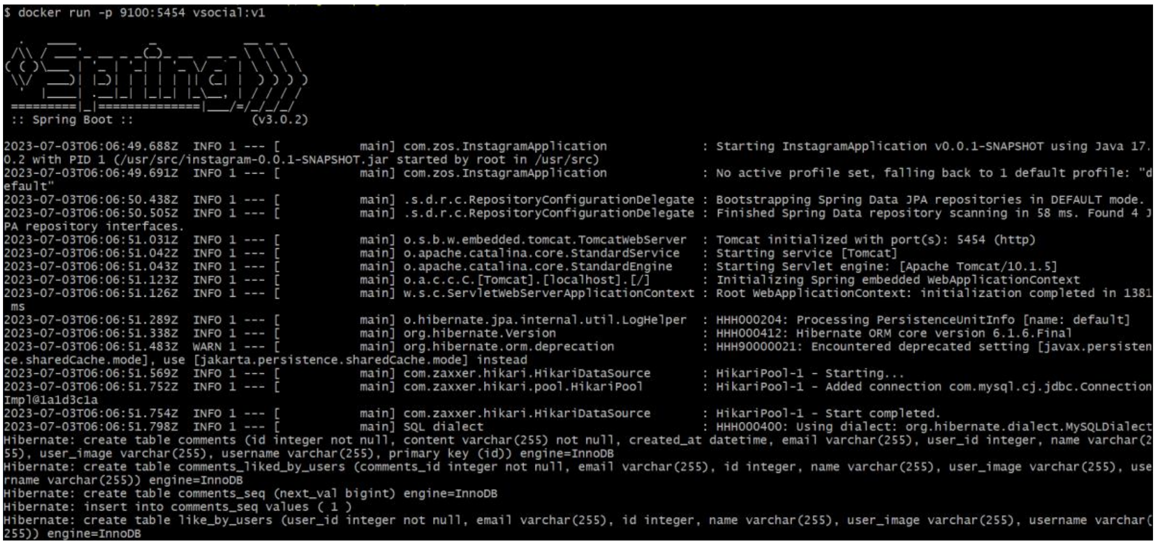
```
        at org.springframework.boot.loader.Launcher.launch(Launcher.java:95) ~[instagram-0.0.1-SNAPSHOT.jar:0.0.1-SNAPSHOT]
        at org.springframework.boot.loader.Launcher.launch(Launcher.java:58) ~[instagram-0.0.1-SNAPSHOT.jar:0.0.1-SNAPSHOT]
        at org.springframework.boot.loader.JarLauncher.main(JarLauncher.java:65) ~[instagram-0.0.1-SNAPSHOT.jar:0.0.1-SNAPSHOT]
Caused by: com.mysql.cj.exceptions.CJException: Unknown database 'instagram'
        at java.base/jdk.internal.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method) ~[na:na]
        at java.base/jdk.internal.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:77) ~[na:na]
        at java.base/jdk.internal.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45) ~[na:na]
```

To rectify this, we entered the container via the **winpty docker exec -it** command and created the database

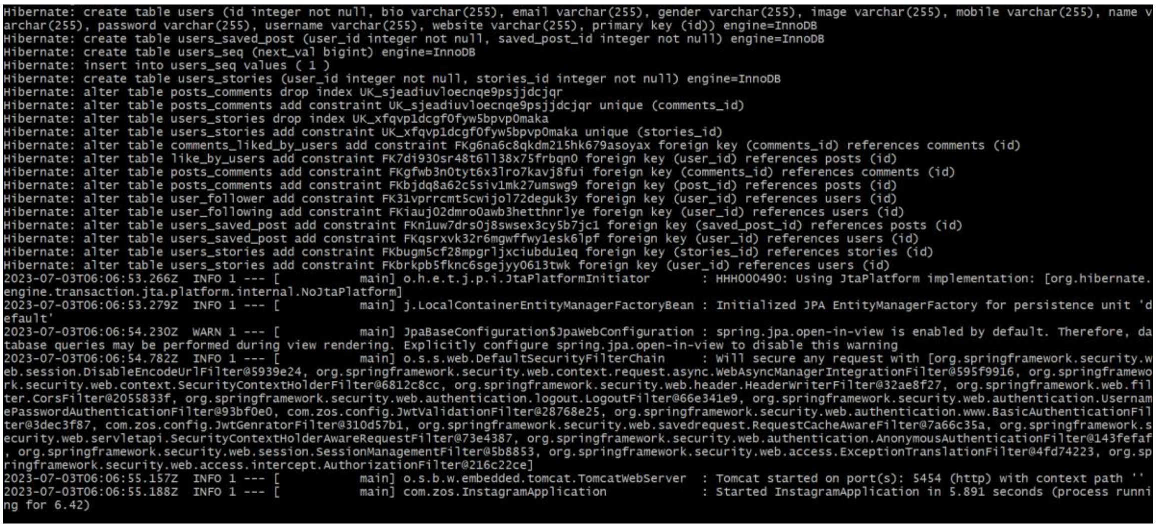


```
$  winpty docker exec -it 83f31d4b3667 mysql -uroot -proot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.5.62 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE instagram;
Query OK, 1 row affected (0.00 sec)
```

After this, we tried running our vsocial:v1 again. This time the connection was successful.

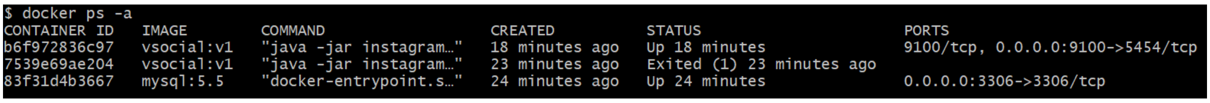


```
$ docker run -p 9100:5454 vsocial:v1

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v3.0.2)

2023-07-03T06:06:49.688Z  INFO 1 --- [           main] com.zos.InstagramApplication             : Starting InstagramApplication v0.0.1-SNAPSHOT using Java 17.
0.2 with PID 1 (/usr/src/instagram-0.0.1-SNAPSHOT.jar started by root in /usr/src)
2023-07-03T06:06:49.691Z  INFO 1 --- [           main] com.zos.InstagramApplication             : No active profile set, falling back to 1 default profile: "d
efault"
2023-07-03T06:06:50.438Z  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2023-07-03T06:06:50.505Z  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 58 ms. Found 4 J
PA repository interfaces.
2023-07-03T06:06:51.031Z  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 5454 (http)
2023-07-03T06:06:51.042Z  INFO 1 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2023-07-03T06:06:51.043Z  INFO 1 --- [           main] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.5]
2023-07-03T06:06:51.123Z  INFO 1 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2023-07-03T06:06:51.126Z  INFO 1 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1381
 ms
2023-07-03T06:06:51.289Z  INFO 1 --- [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [name: default]
2023-07-03T06:06:51.338Z  INFO 1 --- [           main] org.hibernate.Version                    : HHH000412: Hibernate ORM core version 6.1.6.Final
2023-07-03T06:06:51.483Z  WARN 1 --- [           main] org.hibernate.orm.deprecation            : HHH90000021: Encountered deprecated setting [javax.persisten
ce.sharedCache.mode], use [jakarta.persistence.sharedCache.mode] instead
2023-07-03T06:06:51.569Z  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2023-07-03T06:06:51.752Z  INFO 1 --- [           main] com.zaxxer.hikari.pool.HikariPool        : HikariPool-1 - Added connection com.mysql.cj.jdbc.Connection
Impl@1a1d3c1a
2023-07-03T06:06:51.754Z  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2023-07-03T06:06:51.798Z  INFO 1 --- [           main] SQL dialect                              : HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect
Hibernate: create table comments (id integer not null, content varchar(255) not null, created_at datetime, email varchar(255), user_id integer, name varchar(2
55), user_image varchar(255), username varchar(255), primary key (id)) engine=InnoDB
Hibernate: create table comments_liked_by_users (comments_id integer not null, email varchar(255), id integer, name varchar(255), user_image varchar(255), use
rname varchar(255)) engine=InnoDB
Hibernate: create table comments_seq (next_val bigint) engine=InnoDB
Hibernate: insert into comments_seq values ( 1 )
Hibernate: create table like_by_users (user_id integer not null, email varchar(255), id integer, name varchar(255), user_image varchar(255), username varchar(
255)) engine=InnoDB
```



```
Hibernate: create table users (id integer not null, bio varchar(255), email varchar(255), gender varchar(255), image varchar(255), mobile varchar(255), name v
archar(255), password varchar(255), username varchar(255), website varchar(255), primary key (id)) engine=InnoDB
Hibernate: create table users_saved_post (user_id integer not null, saved_post_id integer not null) engine=InnoDB
Hibernate: create table users_seq (next_val bigint) engine=InnoDB
Hibernate: insert into users_seq values ( 1 )
Hibernate: create table users_stories (user_id integer not null, stories_id integer not null) engine=InnoDB
Hibernate: alter table posts_comments drop index UK_sjeadiuvloecnqe9psjjdcjqr
Hibernate: alter table posts_comments add constraint UK_sjeadiuvloecnqe9psjjdcjqr unique (comments_id)
Hibernate: alter table users_stories drop index UK_xfqvp1dcgf0fyw5bpvp0maka
Hibernate: alter table users_stories add constraint UK_xfqvp1dcgf0fyw5bpvp0maka unique (stories_id)
Hibernate: alter table comments_liked_by_users add constraint FKg6na6c8qkdm215hk679asoyax foreign key (comments_id) references comments (id)
Hibernate: alter table like_by_users add constraint FK7di930sr48t6l138x75frbqn0 foreign key (user_id) references posts (id)
Hibernate: alter table posts_comments add constraint FKgfwb3n0tyt6x3lro7kavj8fui foreign key (comments_id) references comments (id)
Hibernate: alter table posts_comments add constraint FKbjdq8a62c5siv1mk27umswg9 foreign key (post_id) references posts (id)
Hibernate: alter table user_follower add constraint FK31vprrcmt5cwijol72deguk3y foreign key (user_id) references users (id)
Hibernate: alter table user_following add constraint FKiauj02dmro0awb3hetthnrlye foreign key (user_id) references users (id)
Hibernate: alter table users_saved_post add constraint FKn1uw7drs0j8swsex3cy5b7jc1 foreign key (saved_post_id) references posts (id)
Hibernate: alter table users_saved_post add constraint FKqsrxvk32r6mgwffwy1esk6lpf foreign key (user_id) references users (id)
Hibernate: alter table users_stories add constraint FKbugm5cf28mpgrljxciubdu1eq foreign key (stories_id) references stories (id)
Hibernate: alter table users_stories add constraint FKbrkpb5fknc6sgejyy0613twk foreign key (user_id) references users (id)
2023-07-03T06:06:53.266Z  INFO 1 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000490: Using JtaPlatform implementation: [org.hibernate.
engine.transaction.jta.platform.internal.NoJtaPlatform]
2023-07-03T06:06:53.279Z  INFO 1 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'd
efault'
2023-07-03T06:06:54.230Z  WARN 1 --- [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, da
tabase queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2023-07-03T06:06:54.782Z  INFO 1 --- [           main] o.s.s.web.DefaultSecurityFilterChain     : Will secure any request with [org.springframework.security.w
eb.session.DisableEncodeUrlFilter@5939e24, org.springframework.security.web.context.request.async.WebAsyncManagerIntegrationFilter@595f9916, org.springframewo
rk.security.web.context.SecurityContextHolderFilter@6812c8cc, org.springframework.security.web.header.HeaderWriterFilter@32ae8f27, org.springframework.web.fil
ter.CorsFilter@2055833f, org.springframework.security.web.authentication.logout.LogoutFilter@66e341e9, org.springframework.security.web.authentication.Usernam
ePasswordAuthenticationFilter@93bf0e0, com.zos.config.JwtValidationFilter@28768e25, org.springframework.security.web.authentication.www.BasicAuthenticationFil
ter@3dec3f87, com.zos.config.JwtGenratorFilter@310d57b1, org.springframework.security.web.savedrequest.RequestCacheAwareFilter@7a66c35a, org.springframework.s
ecurity.web.servletapi.SecurityContextHolderAwareRequestFilter@73e4387, org.springframework.security.web.authentication.AnonymousAuthenticationFilter@143fefaf
, org.springframework.security.web.session.SessionManagementFilter@5b8853, org.springframework.security.web.access.ExceptionTranslationFilter@4fd74223, org.sp
ringframework.security.web.access.intercept.AuthorizationFilter@216c22ce]
2023-07-03T06:06:55.157Z  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 5454 (http) with context path ''
2023-07-03T06:06:55.188Z  INFO 1 --- [           main] com.zos.InstagramApplication             : Started InstagramApplication in 5.891 seconds (process runni
ng for 6.42)
```

The second last line indicates that the application started in 5.891 seconds, indicating that our docker image successfully ran. To cross check the same, we made use of the **docker ps -a** command to check all existing processes.



```
$ docker ps -a
CONTAINER ID   IMAGE        COMMAND                  CREATED          STATUS                      PORTS
b6f972836c97   vsocial:v1   "java -jar instagram…"   18 minutes ago   Up 18 minutes               9100/tcp, 0.0.0.0:9100->5454/tcp
7539e69ae204   vsocial:v1   "java -jar instagram…"   23 minutes ago   Exited (1) 23 minutes ago
83f31d4b3667   mysql:5.5    "docker-entrypoint.s…"   24 minutes ago   Up 24 minutes               0.0.0.0:3306->3306/tcp
```

As seen above, our mysql:5.5 and vsocial:v1 image successfully ran.

This is how we created docker images and ran it.