

Docker Assignment

Objective –

- Create a simple web application (e.g., a Python Flask, DotNet, Java or Node application).
- Write a Dockerfile to containerize the application.
- Build the Docker image and run a container from it.
- Use Docker commands to list, start, stop, and remove containers.
- Inspect running containers and view logs.
- Write a docker-compose.yml file to define a multi-container application.
- Use Docker Compose to bring up the application and ensure all services are running correctly.
- Create a private Docker registry or use Docker Hub.
- Push your Docker images to the registry.
- Pull the images from the registry and run them locally.

I. WSL Installed: Ensure you have [WSL](#) installed and set up, so start the wsl using your password.

```
aryan@IN-CD1D5S3: /mnt/c/l x + v
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\aryanverma>wsl
Windows Subsystem for Linux is now available in the Microsoft Store!
You can upgrade by running 'wsl.exe --update' or by visiting https://aka.ms/wslstorepage
Installing WSL from the Microsoft Store will give you the latest WSL updates, faster.
For more information please visit https://aka.ms/wslstoreinfo

Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.10.102.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Sep 10 12:20:53 IST 2024

System load:  1.1          Processes:    8
Usage of /:   1.8% of 250.98GB  Users logged in: 0
Memory usage: 1%          IPv4 address for eth0: 172.21.230.159
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

132 updates can be applied immediately.
87 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

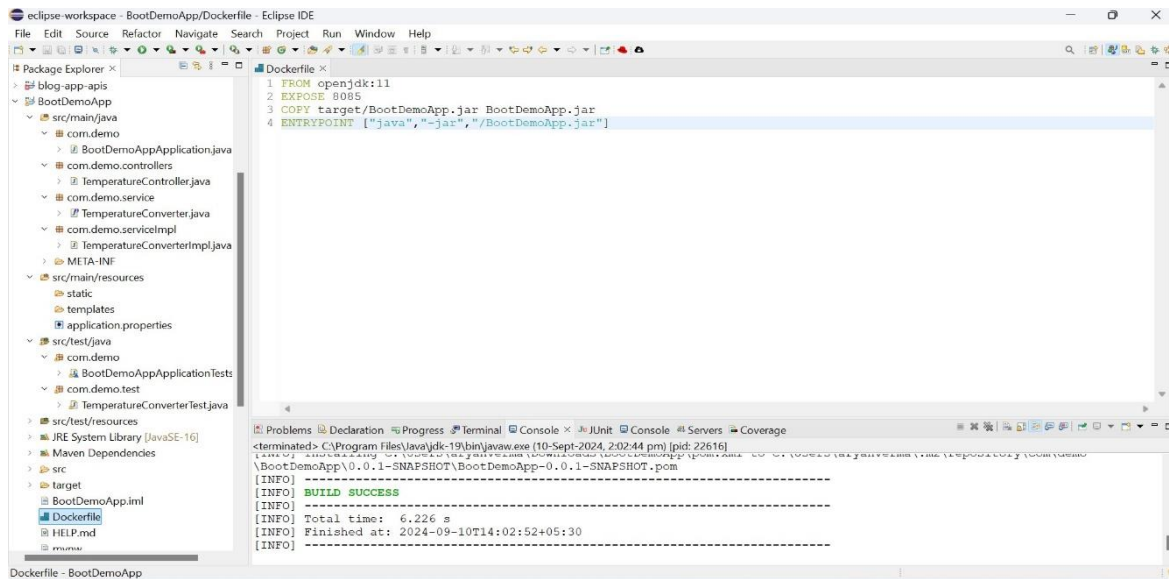
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
```

II. Docker should be installed and integrated with wsl, so start the docker using wsl and check the status.

```
This message is shown once a day. To disable it please create the
/home/aryan/.hushlogin file.
[sudo] password for aryan:
* Starting Docker: docker
aryan@IN-CD1D5S3:/mnt/c/Users/aryanverma$ sudo service docker start
* Starting Docker: docker
aryan@IN-CD1D5S3:/mnt/c/Users/aryanverma$ sudo service docker status
* Docker is running
```

III. Write a Dockerfile to Containerize the Application and move to the directory where the docker file is placed.



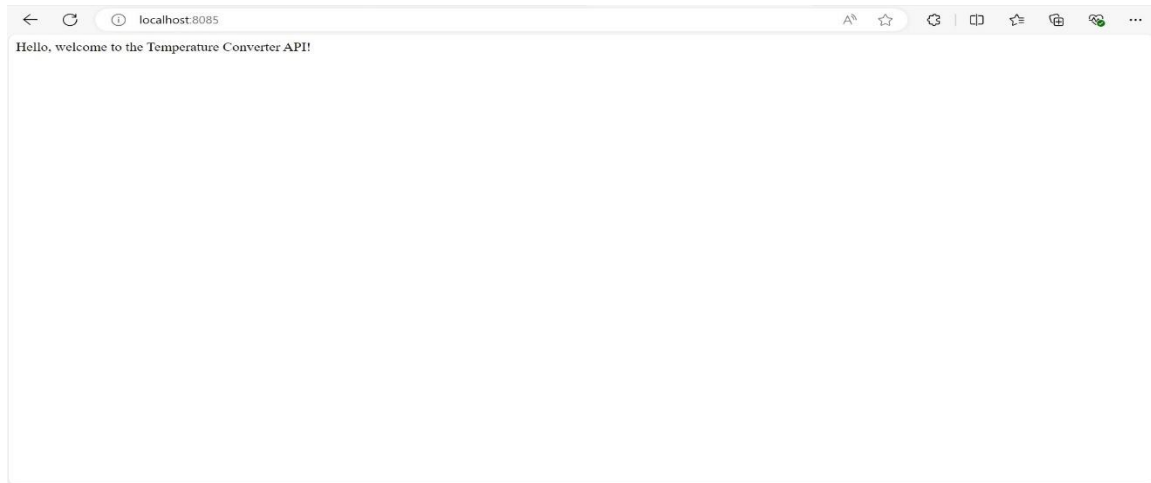
```
aryan@IN-CD1D5S3:/mnt/c/Users/aryanverma/Downloads$ cd BootDemoApp
aryan@IN-CD1D5S3:/mnt/c/Users/aryanverma/Downloads/BootDemoApp$ ls
BootDemoApp.iml Dockerfile HELP.md mvnw mvnw.cmd pom.xml src target
```

IV. Build the docker image of the docker file using below command and run the container.

```
aryan@IN-CD1D5S3:/mnt/c/Users/aryanverma/Downloads/BootDemoApp$ docker build -t.springboot-app .
[+] Building 1.6s (7/7) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile              0.1s
=> => transferring dockerfile: 157B                             0.0s
=> [internal] load metadata for docker.io/library/openjdk:11    1.0s
=> [internal] load .dockerignore                                0.1s
=> => transferring context: 2B                                    0.0s
=> [internal] load build context                                0.2s
=> => transferring context: 17.35MB                               0.2s
=> CACHED [1/2] FROM docker.io/library/openjdk:11@sha256:99bac5bf83633e3c7399aed725c8415e7b569b54e03e4599e580fc9c9db7c21ab 0.0s
=> [2/2] COPY target/BootDemoApp.jar BootDemoApp.jar           0.1s
=> exporting to image                                           0.1s
=> => exporting layers                                           0.1s
=> => writing image sha256:18737048cde22c61f30a9263269704c05b2726f1156ab2e894f9e3de2dd5b3c7 0.0s
=> => naming to docker.io/library/springboot-app                0.0s

aryan@IN-CD1D5S3:/mnt/c/Users/aryanverma/Downloads/BootDemoApp$ docker run -d -p 8085:8085.springboot-app
0e91b30c0789b11985a993e0a149dd8d22f9087604c3f0cc346b56cc1786dd31
```

- V. Use the docker commands to list the running containers and non running containers.



- VI. Use the docker commands to list the running containers and non running containers.

```
aryan@IN-CD1D5S3:/mnt/c/Users/aryanverma/Downloads/BootDemoApp$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
0e91b30c0789	springboot-app	"java -jar /BootDemo..."	11 minutes ago	Up 11 minutes	0.0.0.0:8085->8085/tcp, :::8085->8085/tcp
594e8fe641f6	magical_carson	"/hello"	2 hours ago	Exited (0) 2 hours ago	
	awesome_pascal				

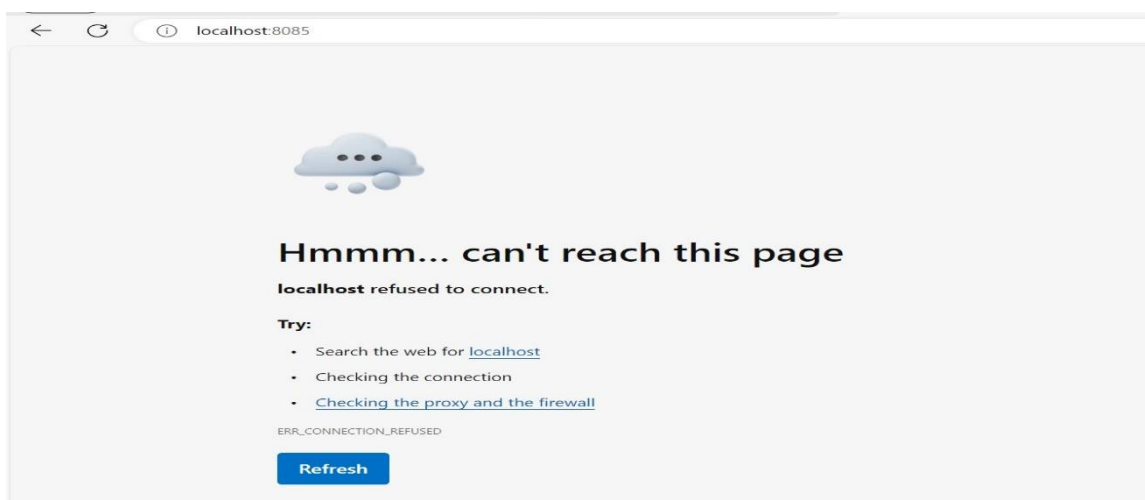
Use the docker commands to stop the running containers and check whether it is stopped or not.

```
aryan@IN-CD1D5S3:/mnt/c/Users/aryanverma/Downloads/BootDemoApp$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
0e91b30c0789	springboot-app	"java -jar /BootDemo..."	13 minutes ago	Up 13 minutes	0.0.0.0:8085->8085/tcp, :::8085->8085/tcp

```
aryan@IN-CD1D5S3:/mnt/c/Users/aryanverma/Downloads/BootDemoApp$ docker stop 0e91b30c0789
0e91b30c0789
aryan@IN-CD1D5S3:/mnt/c/Users/aryanverma/Downloads/BootDemoApp$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------



VII. Use the docker command to view the logs of running container.

```
aryan@IN-CD1D553: /mnt/c/L$ docker logs 0e91b30c0789

:: Spring Boot ::
(v2.5.4)

2024-09-10 08:34:05.518 INFO 1 --- [main] com.demo.BootDemoAppApplication : Starting BootDemoAppApplication v0.0
2024-09-10 08:34:05.523 INFO 1 --- [main] com.demo.BootDemoAppApplication : No active profile set, falling back
to default profiles: default
2024-09-10 08:34:06.657 INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 808
5 (http)
2024-09-10 08:34:06.681 INFO 1 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-09-10 08:34:06.682 INFO 1 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tom
cat/9.0.52]
2024-09-10 08:34:06.784 INFO 1 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebAppl
icationContext
2024-09-10 08:34:06.785 INFO 1 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initiali
zation completed in 1178 ms
2024-09-10 08:34:07.290 INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8085 (htt
p) with context path ''
2024-09-10 08:34:07.307 INFO 1 --- [main] com.demo.BootDemoAppApplication : Started BootDemoAppApplication in 2.
379 seconds (JVM running for 2.828)
2024-09-10 08:34:21.363 INFO 1 --- [nio-8085-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServle
t 'dispatcherServlet'
2024-09-10 08:34:21.364 INFO 1 --- [nio-8085-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServ
let'
2024-09-10 08:34:21.366 INFO 1 --- [nio-8085-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms
```

VIII. Create an account on the docker hub and login here with your respective credentials.

```
aryan@IN-CD1D553: /mnt/c/Users/aryanverma/Downloads/BootDemoApp$ docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to ht
tps://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required
for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/

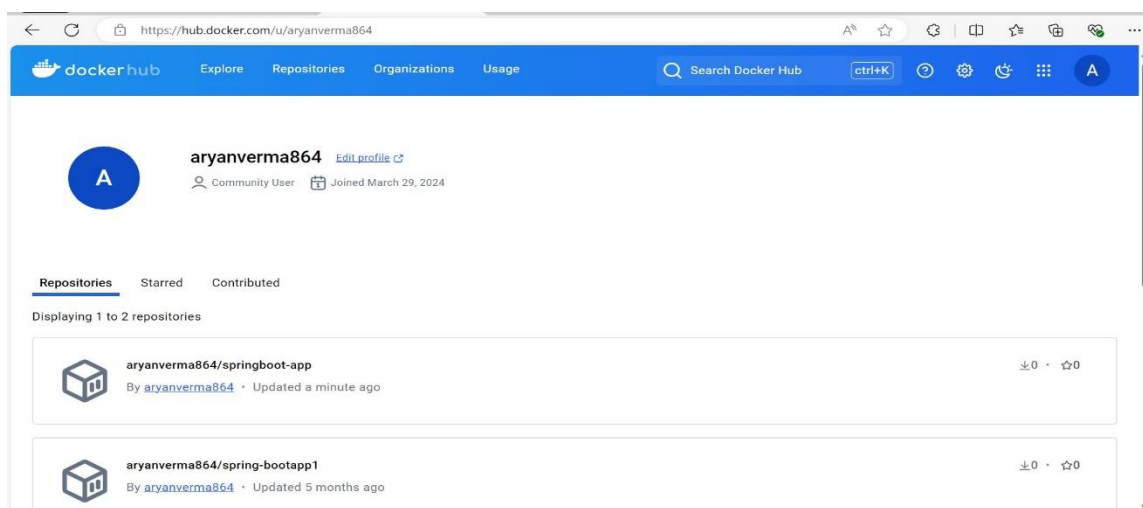
Username: aryanverma864
Password:
WARNING! Your password will be stored unencrypted in /home/aryan/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

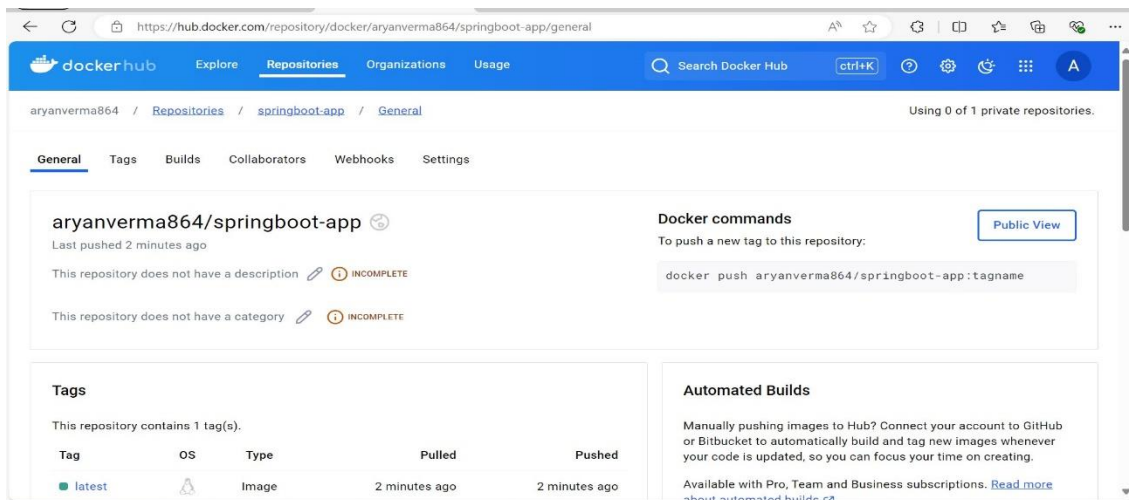
Login Succeeded
aryan@IN-CD1D553: /mnt/c/Users/aryanverma/Downloads/BootDemoApp$ |
```

IX. Adding tag for your application in which the docker username should be included and pushing it to the docker hub.

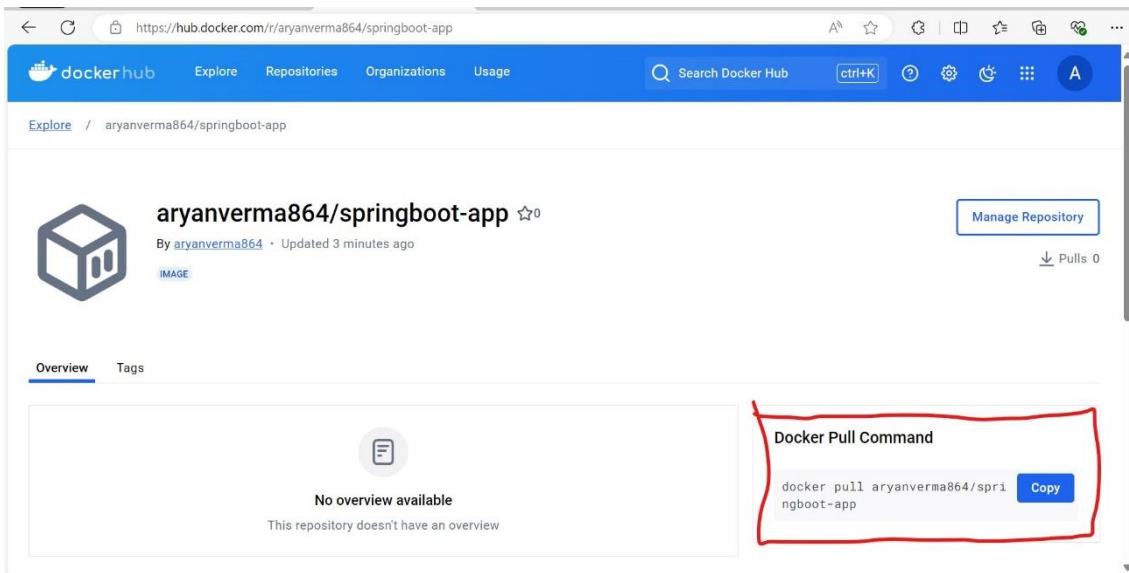
```
aryan@IN-CD1D553: /mnt/c/Users/aryanverma/Downloads/BootDemoApp$ docker tag springboot-app aryanverma864/springboot-app:latest
aryan@IN-CD1D553: /mnt/c/Users/aryanverma/Downloads/BootDemoApp$ docker push aryanverma864/springboot-app:latest
The push refers to repository [docker.io/aryanverma864/springboot-app]
425b5ae89f9e: Pushed
7b7f3078e1db: Pushed
826c3ddb29c: Pushed
b626401ef603: Pushed
9b55156abf26: Pushed
293d5db30c9f: Pushed
03127cdb479b: Pushed
9c742cdc7a5: Pushed
latest: digest: sha256:421d5c79e6c04c8136c7290de16f44c0a1c39a384c0d1d23505c41f30c463dc2 size: 2007
aryan@IN-CD1D553: /mnt/c/Users/aryanverma/Downloads/BootDemoApp$ |
```

X. Verify on the docker hub whether the application deployed successfully or not.





XI. Now Copy the pull command of the repository from the docker hub.



XII. Paste the docker pull command to your command prompt.

```

aryan@IN-CD1D5S3:/mnt/c/Users/aryanverma/Downloads/BootDemoApp$ docker pull aryanverma864/springboot-app
Using default tag: latest
latest: Pulling from aryanverma864/springboot-app
Digest: sha256:421d5c79e6c04c0136c7290de16f44c0a1c39a384c0dd1d23505c41f30c463dc2
Status: Image is up to date for aryanverma864/springboot-app:latest
docker.io/aryanverma864/springboot-app:latest
aryan@IN-CD1D5S3:/mnt/c/Users/aryanverma/Downloads/BootDemoApp$ docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS
19612f562a42  .springboot-app      "java -jar /BootDemo..." 23 minutes ago Up 23 minutes 0.0.0.0:8085->8085/tcp, :::8085->8085/tcp
nervous_swanson
aryan@IN-CD1D5S3:/mnt/c/Users/aryanverma/Downloads/BootDemoApp$ docker stop 19
19
aryan@IN-CD1D5S3:/mnt/c/Users/aryanverma/Downloads/BootDemoApp$ docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS        NAMES

```

XIII. Now run the pulled image of the application from the docker hub locally.

```

aryan@IN-CD1D5S3:/mnt/c/Users/aryanverma/Downloads/BootDemoApp$ docker run -d -p 8085:8085 aryanverma864/springboot-app
6c738a0183e19a04666feeafd3abf923a4cc3859a0a6c561607b5cddb0a50a52
aryan@IN-CD1D5S3:/mnt/c/Users/aryanverma/Downloads/BootDemoApp$ docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS
6c738a0183e1   aryanverma864/springboot-app      "java -jar /BootDemo..." 26 seconds ago Up 26 seconds 0.0.0.0:8085->8085/tcp, :::8085->8085/tcp
priceless_maxwell
aryan@IN-CD1D5S3:/mnt/c/Users/aryanverma/Downloads/BootDemoApp$

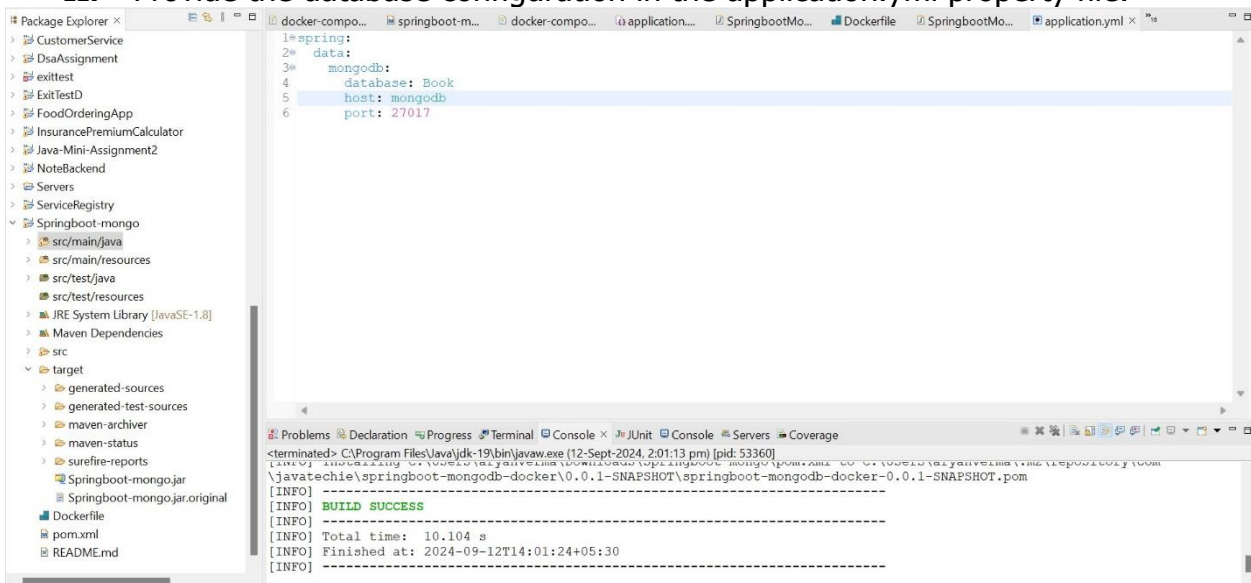
```


Docker-compose Deployment steps

- I. Pull the image of MongoDB from the docker hub and run it on the default port 27017 and check whether the image created or not.

```
aryan@IN-CD1D553: /mnt/c/U x  aryan@IN-CD1D553: /mnt/c/l x  + v
* Starting Docker: docker
aryan@IN-CD1D553: /mnt/c/Users/aryanverma$ sudo service docker status
* Docker is running
aryan@IN-CD1D553: /mnt/c/Users/aryanverma$ docker pull mongo:latest
latest: Pulling from library/mongo
857cc8cb19c0: Pull complete
a54f12bd5819: Pull complete
f95b02a6236d: Pull complete
0d20d29fe9ca: Pull complete
2382733f40de: Pull complete
c1458145b657: Pull complete
fee77be41765: Pull complete
da4d4cbb623f: Pull complete
Digest: sha256:1ade6afda762cb6a68ba65e83ef305660ef0517d6d4140627211970e85f4588a
Status: Downloaded newer image for mongo:latest
docker.io/library/mongo:latest
aryan@IN-CD1D553: /mnt/c/Users/aryanverma$ docker run -d -p 27017:27017 --name mongodb mongo:latest
5feb0fd675d713ebaa396fd6be1bc6ada43afdb7852f9639bc2d74c2acb2974d
aryan@IN-CD1D553: /mnt/c/Users/aryanverma$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
5feb0fd675d7   mongo:latest   "docker-entrypoint.s..." 27 seconds ago Up 26 seconds 0.0.0.0:27017->27017/tcp, :::27017->27017/tcp
mongodb
aryan@IN-CD1D553: /mnt/c/Users/aryanverma$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
bootdemoapp   latest    f083780aff4a   42 minutes ago 672MB
aryanverma864/springboot-app  latest    670f5620cadd   About an hour ago 672MB
springboot-app  latest    670f5620cadd   About an hour ago 672MB
mongo          latest    81a05b728352   2 weeks ago    782MB
sonarqube      lts-community  1473eb950a6a   5 months ago   603MB
hello-world    latest     d2c94e258dcb   16 months ago 13.3kB
openjdk        11         47a932d998b7   2 years ago    654MB
aryan@IN-CD1D553: /mnt/c/Users/aryanverma$
```

- II. Provide the database configuration in the application.yml property file.



The screenshot shows an IDE with a Package Explorer on the left and a Console window at the bottom. The Package Explorer shows a project structure for 'Springboot-mongo' with folders for 'src/main/java', 'src/main/resources', 'src/test/java', 'src/test/resources', 'JRE System Library [JavaSE-1.8]', 'Maven Dependencies', 'src', 'target', 'generated-sources', 'generated-test-sources', 'maven-archiver', 'maven-status', 'surefire-reports', 'Springboot-mongo.jar', 'Springboot-mongo.jar.original', 'Dockerfile', 'pom.xml', and 'README.md'.

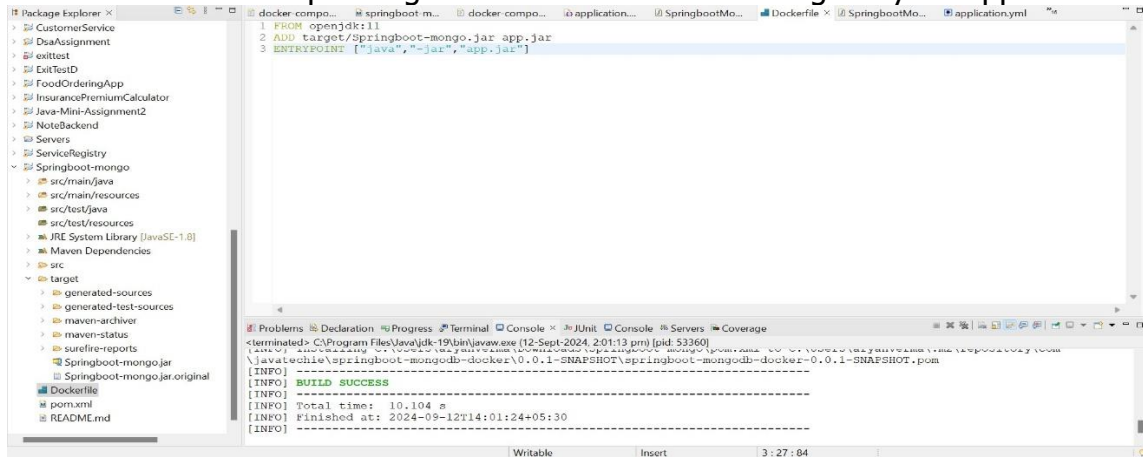
The 'application.yml' file is open in the editor, showing the following configuration:

```
1*spring:
2#  data:
3#    mongodb:
4#      database: Book
5#      host: mongodb
6#      port: 27017
```

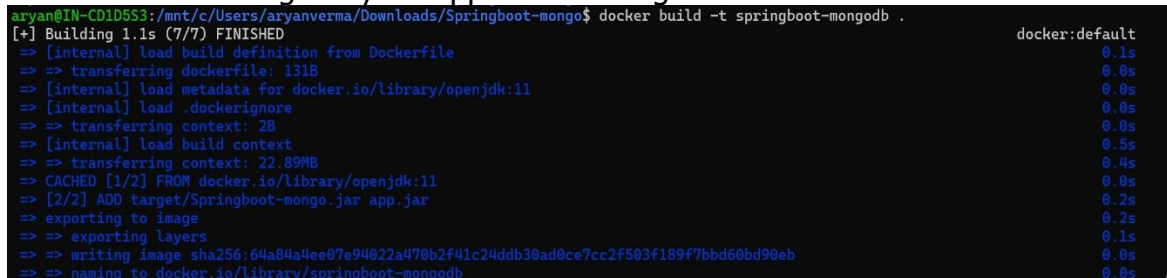
The Console window shows the output of a build process, including the following lines:

```
<terminated> CAProgram Files\Java\jdk-19\bin\javaw.exe (12-Sept-2024, 2:01:13 pm) [pid: 53360]
[INFO] BUILD SUCCESS
[INFO] Total time: 10.104 s
[INFO] Finished at: 2024-09-12T14:01:24+05:30
[INFO]
```

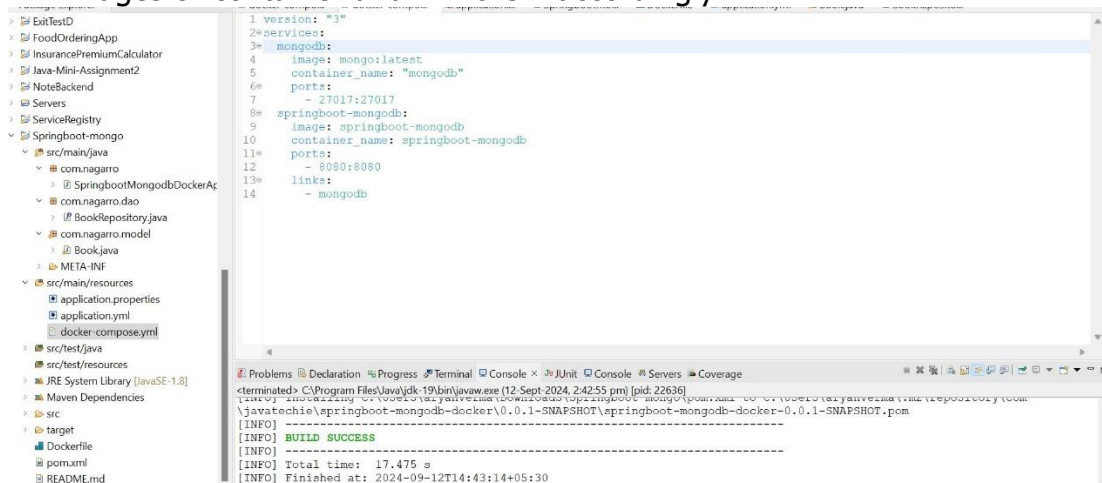
III. Write the corresponding docker file to build the image of your application.



IV. Build the image of your application using the docker file.



V. Write the docker-compose.yml file having all the configuration of both the images or container and link them accordingly.



VI. Use Docker Compose to bring up the application and ensure all services are running correctly.

```
aryan@IN-CD10SS3:/mnt/c/Users/aryanverma/Downloads/Springboot-mongo/src/main/resources$ docker-compose up
Creating network "resources_default" with the default driver
Creating mongodb ... done
Creating springboot-mongodb ... done
Attaching to mongodb, springboot-mongodb
mongodb | {"t":{"$date":"2024-09-12T09:33:58.588+00:00"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"main","msg":"In
mongodb | initialized wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":21},"incomingInternalClie
mongodb | nt":{"minWireVersion":0,"maxWireVersion":21},"outgoing":{"minWireVersion":6,"maxWireVersion":21},"isInternalClient":true}}}
mongodb | {"t":{"$date":"2024-09-12T09:33:58.592+00:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"main","msg":"Au
mongodb | tomatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"
mongodb | {"t":{"$date":"2024-09-12T09:33:58.602+00:00"},"s":"I", "c":"NETWORK", "id":4648601, "ctx":"main","msg":"Im
mongodb | plicit TCP FastOpen is required, set tcpFastOpenServer, tcpFastOpenClient, and tcpFastOpenQueueSize."}
mongodb | {"t":{"$date":"2024-09-12T09:33:58.609+00:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"main","msg":"Su
mongodb | ccessfully registered PrimaryOnlyService","attr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDonors"}
mongodb | {"t":{"$date":"2024-09-12T09:33:58.609+00:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"main","msg":"Su
mongodb | ccessfully registered PrimaryOnlyService","attr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigrationReci
mongodb | pients"}}
mongodb | {"t":{"$date":"2024-09-12T09:33:58.609+00:00"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"main","msg":"Mu
mongodb | lti threading initialized"}
mongodb | {"t":{"$date":"2024-09-12T09:33:58.609+00:00"},"s":"I", "c":"TENANT_M", "id":7091600, "ctx":"main","msg":"St
mongodb | arting TenantMigrationAccessBlockerRegistry"}
mongodb | {"t":{"$date":"2024-09-12T09:33:58.611+00:00"},"s":"I", "c":"CONTROL", "id":4615611, "ctx":"initandlisten",
mongodb | "msg":"MongoDB starting","attr":{"pid":1,"port":27017,"dbPath":"/data/db","architecture":"64-bit","host":"decfa0d08e44"}}
mongodb | {"t":{"$date":"2024-09-12T09:33:58.611+00:00"},"s":"I", "c":"CONTROL", "id":23403, "ctx":"initandlisten",
mongodb | "msg":"Build Info","attr":{"buildInfo":{"version":"7.0.14","gitVersion":"ce59cfc6a3c5e5c067dca0d30697edd68d4f5188","openSSLVersion":"
mongodb | OpenSSL 3.0.2 15 Mar 2022"},"modules":[],"allocator":"tcmalloc","environment":{"distmod":"ubuntu2204","distarch":"x86_64","target_arch
mongodb | ":"x86_64"}}}
mongodb | {"t":{"$date":"2024-09-12T09:33:58.611+00:00"},"s":"I", "c":"CONTROL", "id":51765, "ctx":"initandlisten",
mongodb | "msg":"Operating System","attr":{"os":{"name":"Ubuntu","version":"22.04"}}}
mongodb | {"t":{"$date":"2024-09-12T09:33:58.611+00:00"},"s":"I", "c":"CONTROL", "id":21951, "ctx":"initandlisten",
mongodb | "msg":"Options set by command line","attr":{"options":{"net":{"bindIp":"*"}}}}
mongodb | {"t":{"$date":"2024-09-12T09:33:58.617+00:00"},"s":"I", "c":"STORAGE", "id":22297, "ctx":"initandlisten",
```

The screenshot shows the Postman interface with a workspace named 'My Workspace'. A collection named 'mongo docker' is expanded, showing a 'POST Add books' request. The request is a POST to 'http://localhost:8080/book' with a JSON body:

```
{  "id": 1,  "name": "aryan",  "authorName": "aizman"}
```

. The response is a 200 OK status with a 239 ms response time and 208 B of data. The response body is a JSON object:

```
{  "id": 1,  "name": "aryan",  "authorName": "aizman"}
```

The screenshot shows the Postman interface with the same workspace and collection. A 'GET get book' request is selected, which is a GET to 'http://localhost:8080/book'. The response is a 200 OK status with a 32 ms response time and 210 B of data. The response body is a JSON array:

```
[  {    "id": 1,    "name": "aryan",    "authorName": "aizman"  }]
```