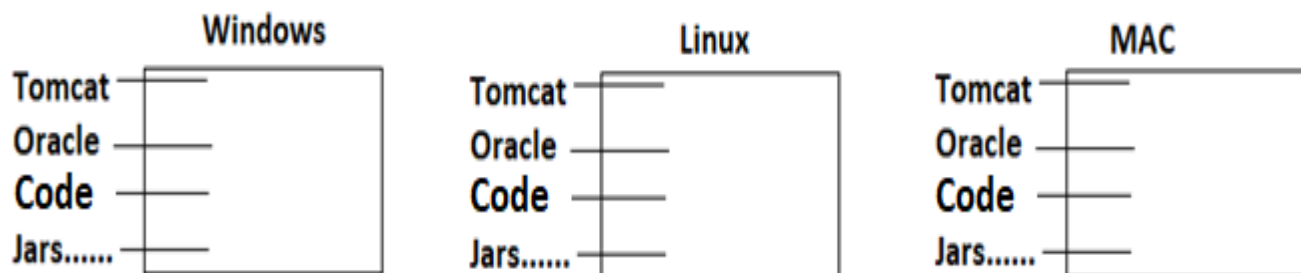


Docker

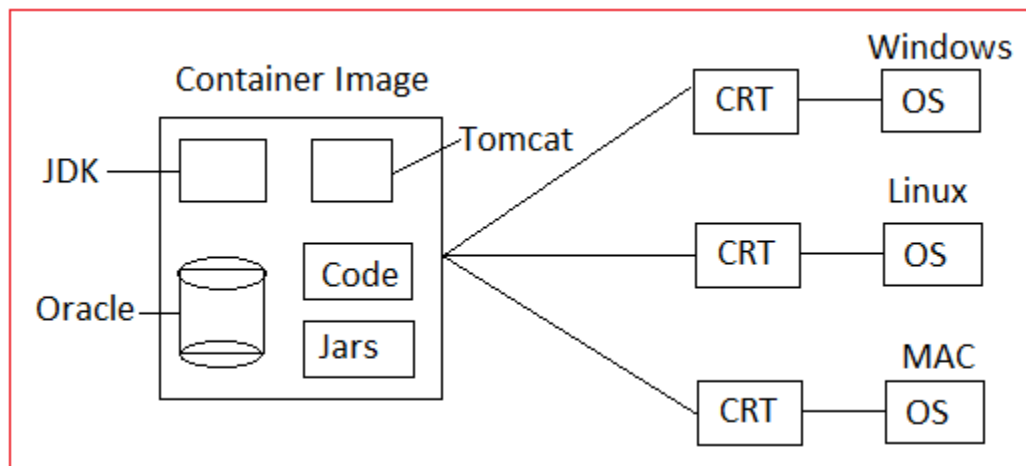
Docker:

- Docker is “**CONTAINER SYSTEM**” which includes all software’s a unit to run application, On any Platform (Windows, Linux, Mac...).
- Docker supports running application on cloud servers also.
- Docker supports follow of “**CROSS-OS**”. It means docker behaves a middleware between our runtime software and actual operating System.
- Docker tool is used for Application Deployment (Running Application).

Before Using Container System:

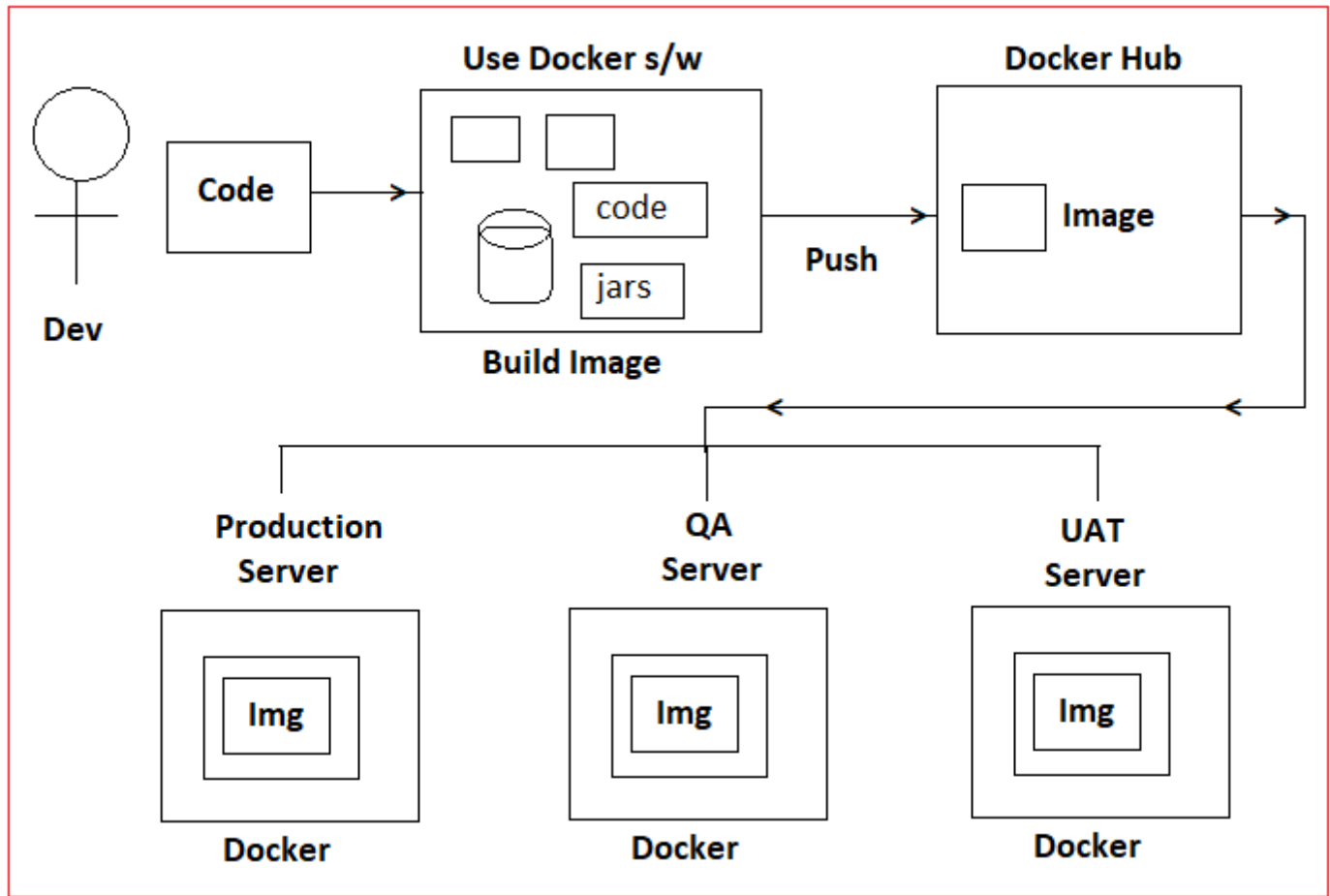


After Container System (Docker):



CRT=Container Run Time

Docker Workflow:

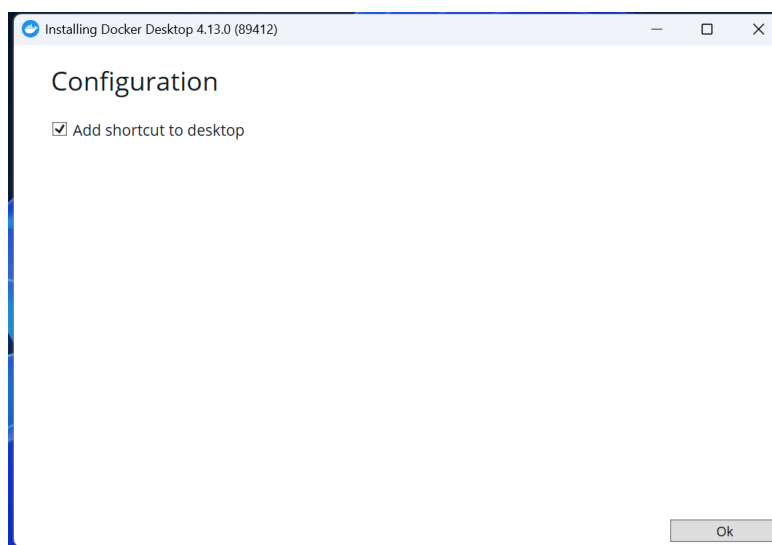
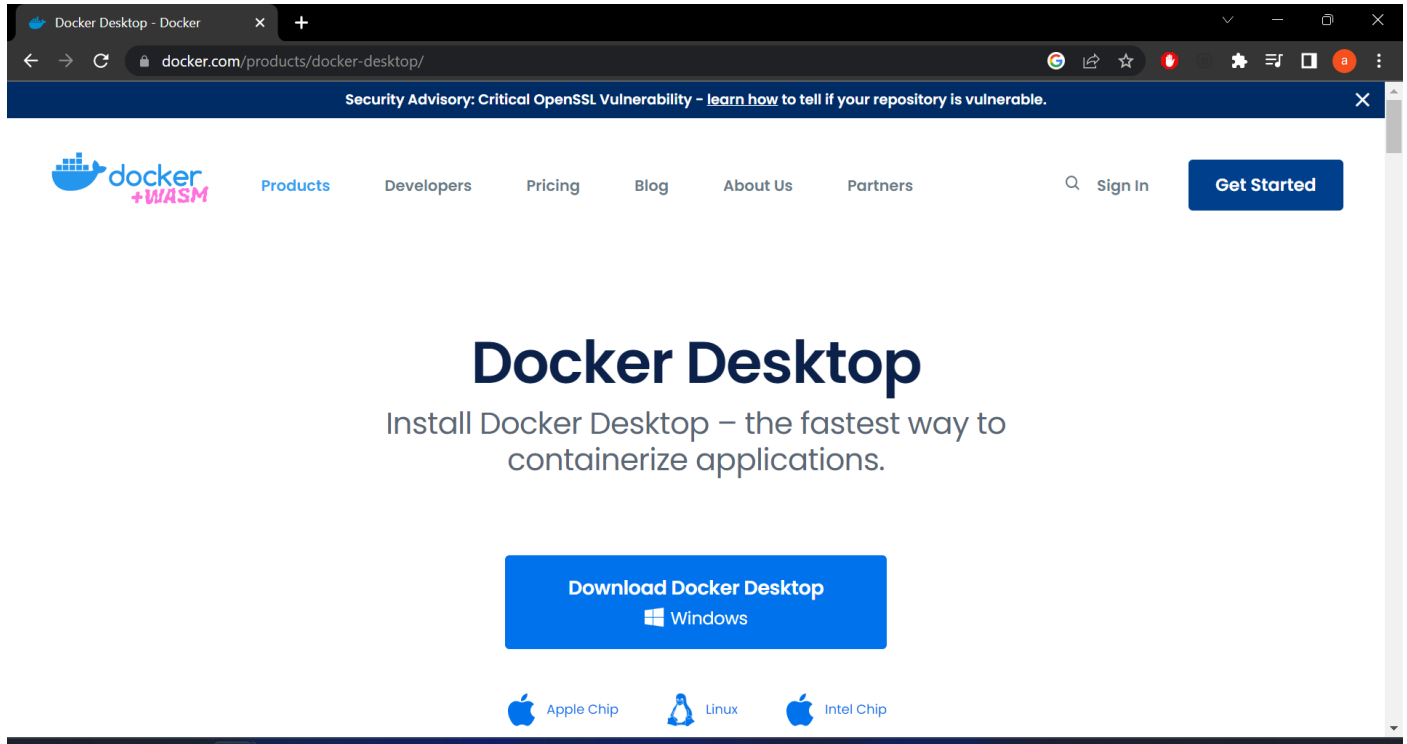


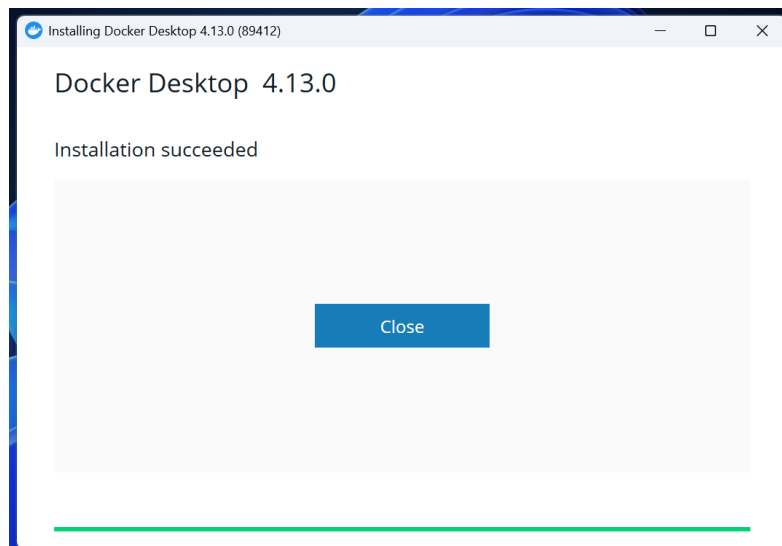
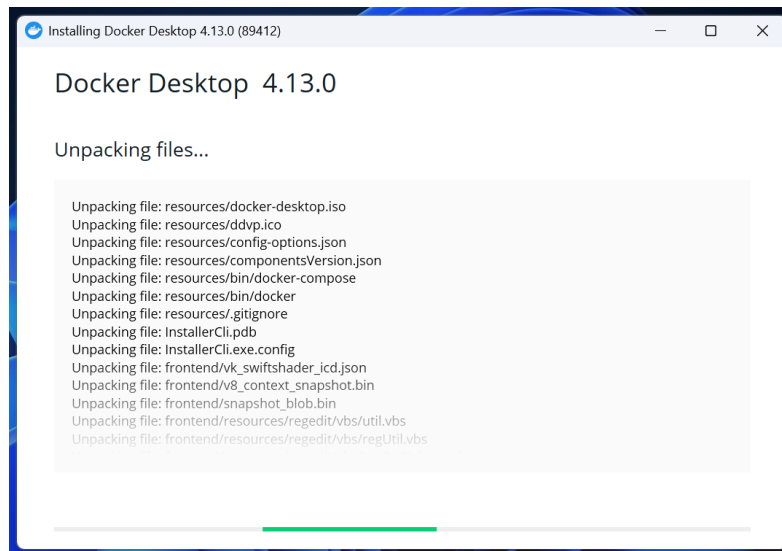
Installation of Docker Desktop

Steps:

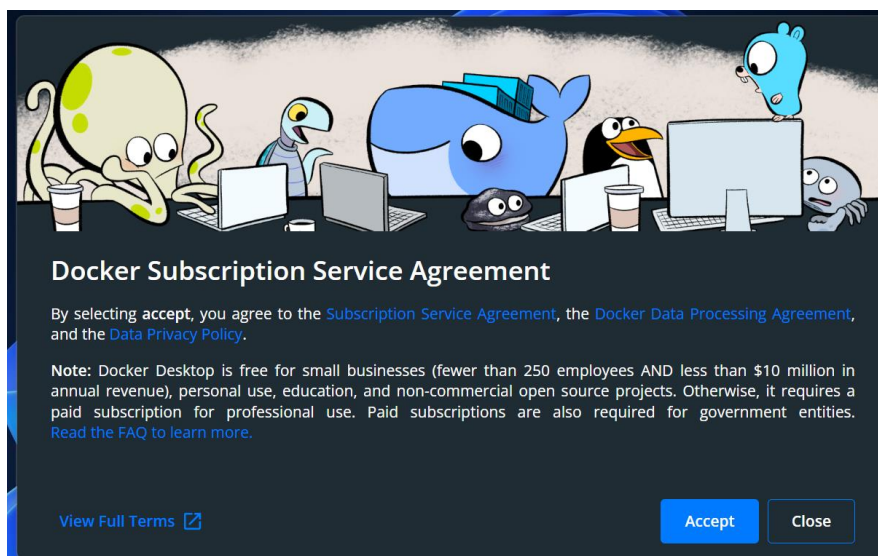
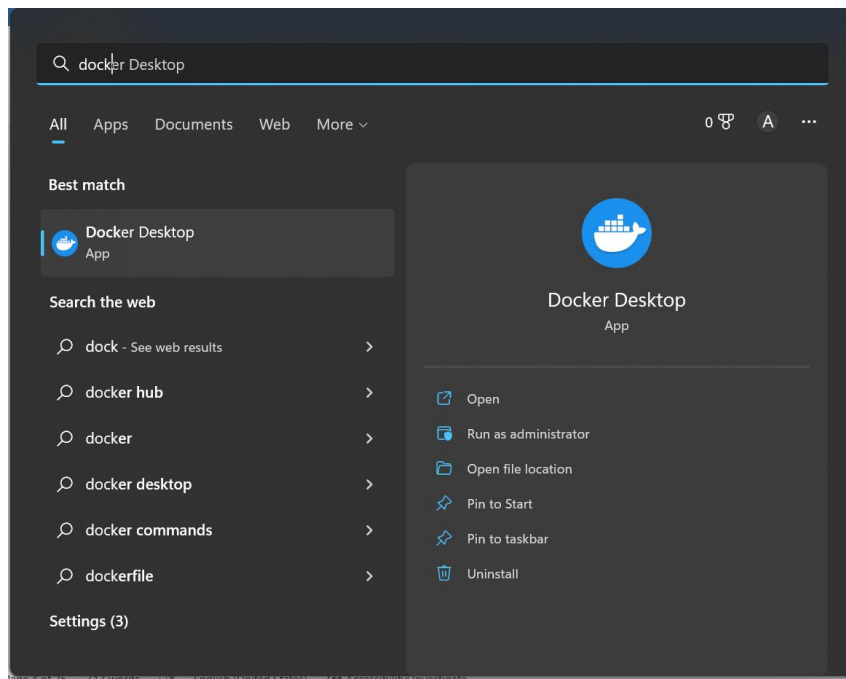
Downloading Docker Desktop from following url;

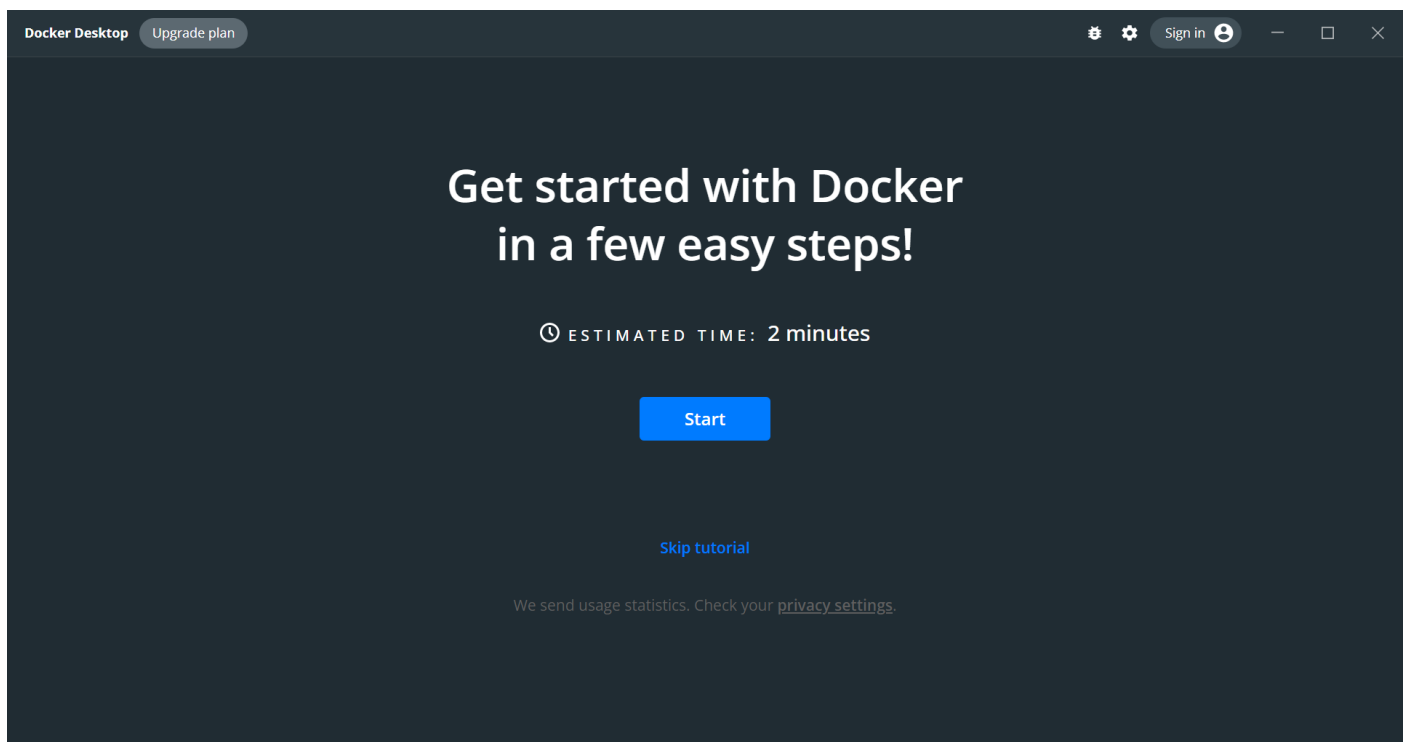
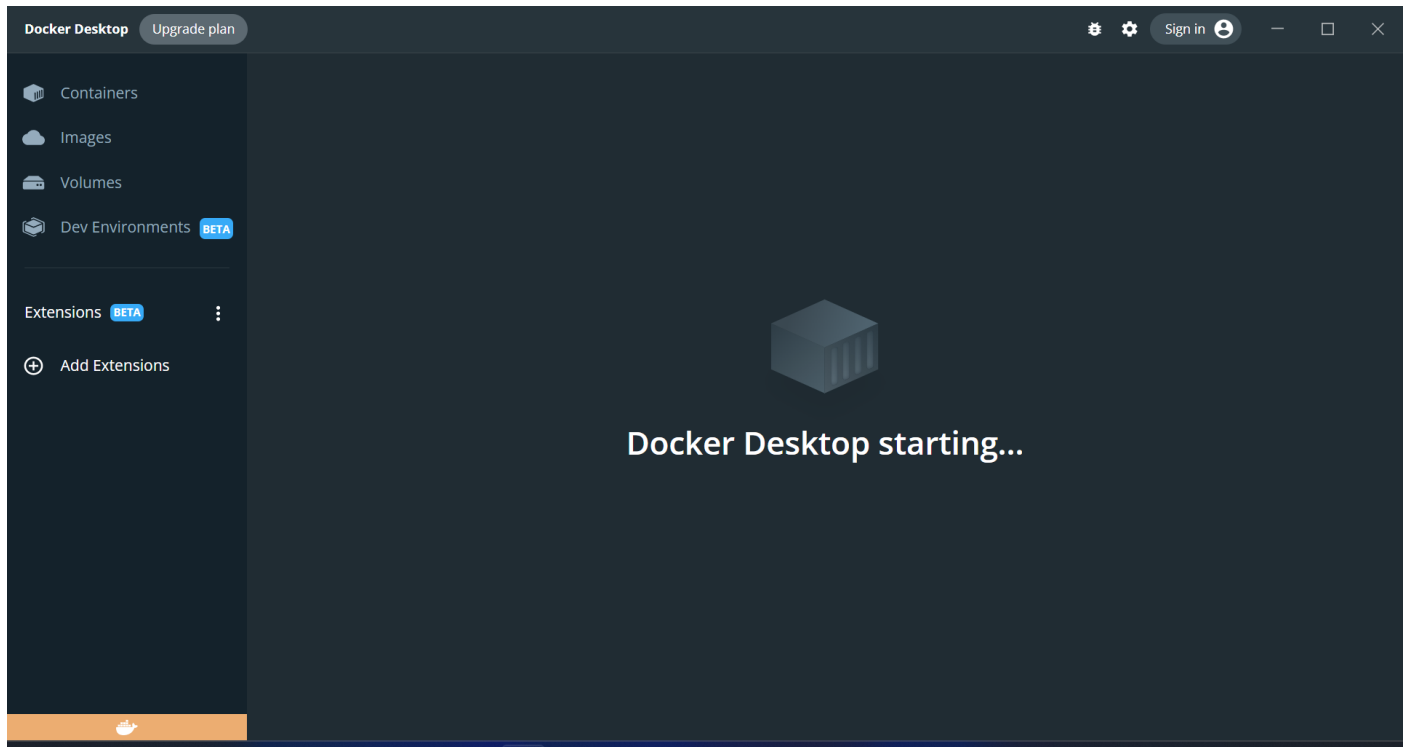
<https://www.docker.com/products/docker-desktop/>

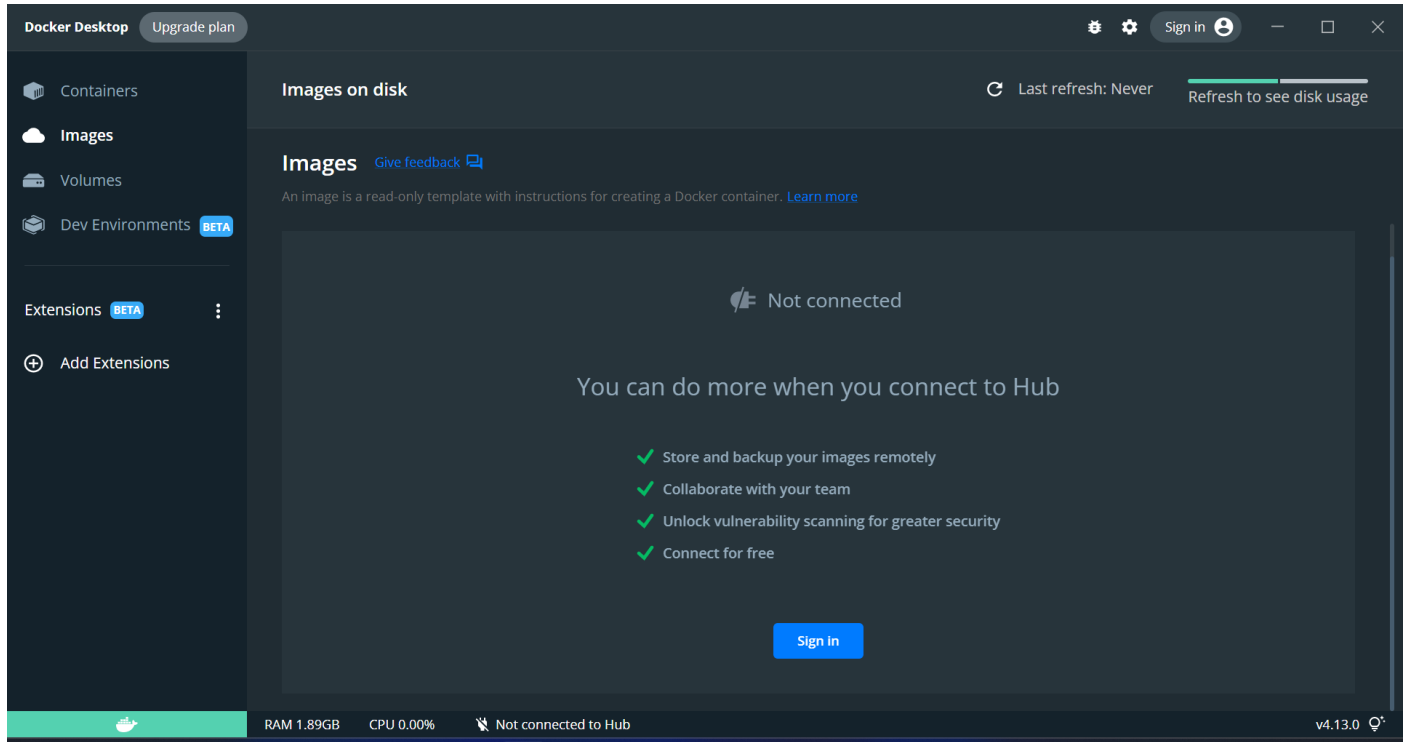




Open the Docker Desktop app in windows.







Pull MongoDB Image

Open the CMD and type

```
>docker pull mongo
```



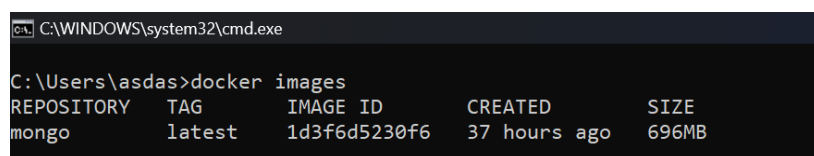
```

C:\Users\asdas>docker pull mongo
Using default tag: latest
latest: Pulling from library/mongo
eaead16dc43b: Pull complete
8a00eb9f68a0: Pull complete
f683956749c5: Pull complete
b33b2f05ea20: Pull complete
3a342bea915a: Pull complete
fa956ab1c2f0: Pull complete
138a8542a624: Pull complete
acab179a7f07: Pull complete
f88335710e84: Pull complete
Digest: sha256:3b9bfc35335710340afe1e98c870491b2a969fd93b62505b4617eab73d97cec6
Status: Downloaded newer image for mongo:latest
docker.io/library/mongo:latest

C:\Users\asdas>

```

```
>docker images
```



```

C:\Users\asdas>docker images
REPOSITORY    TAG       IMAGE ID      CREATED        SIZE
mongo         latest    1d3f6d5230f6  37 hours ago   696MB

```

Running Container with port mapping

```
>docker run -d -p 27017:27017 --name mongocontainer mongo:latest
```

```

C:\Users\asdas>docker run -d -p 27017:27017 --name mongocontainer mongo:latest
799eeb91b04e9e96c8eb81fad75c68ecb40381a88cbd5bc791d83e81a5073774

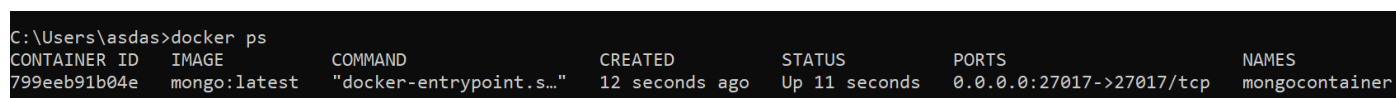
```

-d means detach mode also it is not compulsory

-p means port

Use **docker ps** command to show running containers

```
>docker ps
```



```

C:\Users\asdas>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
799eeb91b04e   mongo:latest "docker-entrypoint.s..." 12 seconds ago Up 11 seconds 0.0.0.0:27017->27017/tcp          mongocontainer

```


To show both running and stopped containers.

>docker ps -a

```
C:\Users\asdas>docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
799eeb91b04e   mongo:latest   "docker-entrypoint.s..." 5 hours ago    Exited (255)  About an hour ago    0.0.0.0:27017->27017/tcp    mongocontainer
376bbc3f87d8   57854908ef7f   "java -jar /spring-b..." 6 hours ago    Exited (143)  6 hours ago             reverent_groth
endieck
5702d952264a   97d30ade3c93   "docker-entrypoint.s..." 13 months ago  Exited (1)    31 hours ago             todoimage-cont
ainer
f13794fd492e   97d30ade3c93   "docker-entrypoint.s..." 13 months ago  Exited (1)    13 months ago             kind_lehmann
```

>docker exec -it mongocontainer bash

root@aa316a63cbd6:/# mongosh

```
mongosh mongodb://127.0.0.1:27017
C:\Users\asdas>docker exec -it mongocontainer bash
root@11343561eae:/# mongosh
Current Mongosh Log ID: 635d3a840f5728c9523e2926
Connecting to:  mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.6.0
Using MongoDB:  6.0.2
Using Mongosh:  1.6.0

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

-----
The server generated these startup warnings when booting
2022-10-29T14:25:33.733+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://docs.mongodb.org/core/prodnotes-filesystem
2022-10-29T14:25:34.702+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2022-10-29T14:25:34.703+00:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting it to 'never'
2022-10-29T14:25:34.703+00:00: vm.max_map_count is too low
-----

Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
-----

test> |
```

show dbs

```
mongosh mongodb://127.0.0.1:27017
test> show dbs
admin    40.00 KiB
config  108.00 KiB
local   40.00 KiB
test>
```

```
db.emp.insertMany(
  [
    {eid: 1001, name: "Ramesh", sal: 35000},
    {eid: 1002, name: "Somesh", sal: 45000},
    {eid: 1003, name: "Rupesh", sal: 55000},
    {eid: 1004, name: "Naresh", sal: 65000},
    {eid: 1005, name: "Kamesh", sal: 75000}
  ]
)
```

```
test> db.emp.insertMany(
...   [
...     {eid: 1001, name: "Ramesh", sal: 35000},
...     {eid: 1002, name: "Somesh", sal: 45000},
...     {eid: 1003, name: "Rupesh", sal: 55000},
...     {eid: 1004, name: "Naresh", sal: 65000},
...     {eid: 1005, name: "Kamesh", sal: 75000}
...   ]
... )
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("635d3ceec599ca940d06e55f"),
    '1': ObjectId("635d3ceec599ca940d06e560"),
    '2': ObjectId("635d3ceec599ca940d06e561"),
    '3': ObjectId("635d3ceec599ca940d06e562"),
    '4': ObjectId("635d3ceec599ca940d06e563")
  }
}
```

db.employee.find()

```
mongosh mongodb://127.0.0.1
}
test> db.emp.find()
[
  {
    _id: ObjectId("635d3ceec599ca940d06e55f"),
    eid: 1001,
    name: 'Ramesh',
    sal: 35000
  },
  {
    _id: ObjectId("635d3ceec599ca940d06e560"),
    eid: 1002,
    name: 'Somesh',
    sal: 45000
  },
  {
    _id: ObjectId("635d3ceec599ca940d06e561"),
    eid: 1003,
    name: 'Rupesh',
    sal: 55000
  },
  {
    _id: ObjectId("635d3ceec599ca940d06e562"),
    eid: 1004,
    name: 'Naresh',
    sal: 65000
  },
  {
    _id: ObjectId("635d3ceec599ca940d06e563"),
    eid: 1005,
    name: 'Kamesh',
    sal: 75000
  }
]
```

To exit

```
test> exit
root@11343561eae:/# exit
exit
C:\Users\asdas>
```

To stop container;

>docker stop mongocontainer

To remove container;

>docker ps

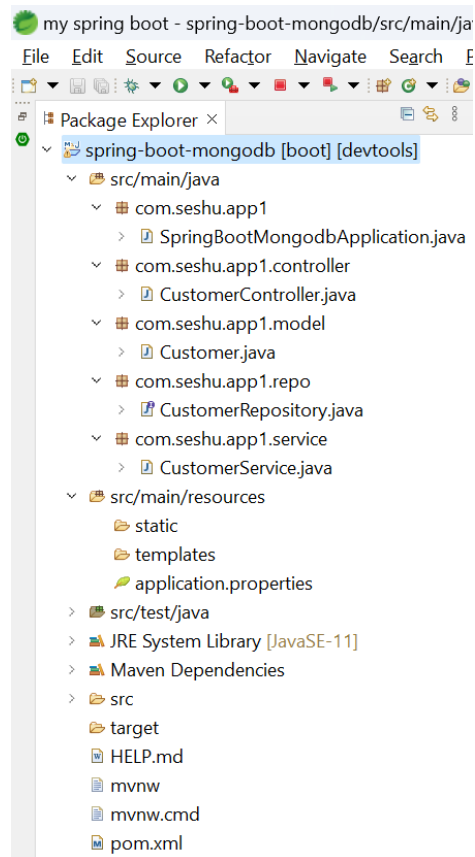
```
C:\Users\asdas>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                    NAMES
11343561eae    mongo:latest "docker-entrypoint.s..." 26 minutes ago Up 26 minutes 0.0.0.0:27017->27017/tcp  mongocontainer

C:\Users\asdas>
C:\Users\asdas>docker rm -f 11343561eae
11343561eae

C:\Users\asdas>
C:\Users\asdas>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                    NAMES
```

Spring Boot Rest and MongoDB Docker Container

Generate spring boot project with following dependencies;
Spring web, Spring Dev tools and Spring MongoDB



application.properties

```
spring.data.mongodb.database=customerdb
spring.data.mongodb.port=27017
spring.data.mongodb.host=mongocontainer
```

Customer.java

```
package com.seshu.appl.model;

import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;

@Document
public class Customer {
    @Id
    private int customerId;
    private String customerName;
    private long customerPhoneNo;
    private String customerEmail;
    public int getCustomerId() {
        return customerId;
    }
    public void setCustomerId(int customerId) {
        this.customerId = customerId;
    }
    public String getCustomerName() {
        return customerName;
    }
    public void setCustomerName(String customerName) {
        this.customerName = customerName;
    }
    public long getCustomerPhoneNo() {
        return customerPhoneNo;
    }
    public void setCustomerPhoneNo(long customerPhoneNo) {
        this.customerPhoneNo = customerPhoneNo;
    }
    public String getCustomerEmail() {
        return customerEmail;
    }
    public void setCustomerEmail(String customerEmail) {
        this.customerEmail = customerEmail;
    }
}
```

CustmerRepository.java

```
package com.seshu.appl.repo;

import org.springframework.data.mongodb.repository.MongoRepository;
import org.springframework.stereotype.Repository;

import com.seshu.appl.model.Customer;

@Repository
public interface CustomerRepository extends
MongoRepository<Customer,Integer> {
}
```

CustomerService.java

```
package com.seshu.appl.service;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.seshu.appl.model.Customer;
import com.seshu.appl.repo.CustomerRepository;

import java.util.List;

@Service
public class CustomerService {
    @Autowired
    private CustomerRepository customerRepository;

    public Customer saveCustomerDetails(Customer customer) {
        return customerRepository.save(customer);
    }

    public List<Customer> getAllCustomerDetails() {
        return customerRepository.findAll();
    }
}
```

CustomerController.java

```
package com.seshu.appl.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.seshu.appl.model.Customer;
import com.seshu.appl.service.CustomerService;

@RestController
@RequestMapping("api/v1/customerservice/")
public class CustomerController {
    @Autowired
    private CustomerService customerService;

    @PostMapping("customer")
    public ResponseEntity<?> saveCustomer(@RequestBody Customer
customer) {
        return new
ResponseEntity<>(customerService.saveCustomerDetails(customer),
HttpStatus.INTERNAL_SERVER_ERROR);
    }

    @GetMapping("customer")
    public ResponseEntity<?> getAllCustomer() {
        return new
ResponseEntity<>(customerService.getAllCustomerDetails(),
HttpStatus.INTERNAL_SERVER_ERROR);
    }
}
```

SpringBootMongodbApplication.java

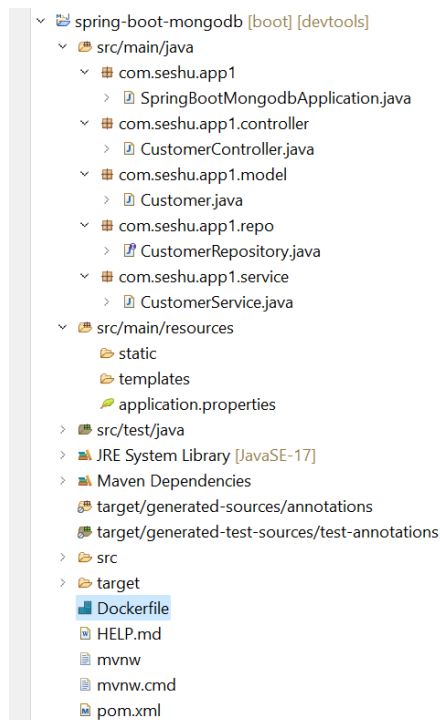
```
package com.seshu.app1;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class SpringBootMongodbApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringBootMongodbApplication.class,
args);
    }
}
```

Create one Dockerfile inside project



- Right click on Project => New => file => Enter file name as “Dockerfile” => finish.
- Naming convention wise Dockerfile name starts with D capital.

Note: Without any extension

- Write bellow details inside **Dockerfile**.

Dockerfile

```
FROM openjdk:17
EXPOSE 8080
ADD target/spring-boot-mongodb.jar spring-boot-mongodb.jar
ENTRYPOINT ["java", "-jar", "/spring-boot-mongodb.jar"]
```

Update the pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.7.0</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.seshu</groupId>
  <artifactId>spring-boot-mongodb</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>spring-boot-mongodb</name>
  <description>Demo project for Spring Boot</description>
  <properties>
    <java.version>17</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-
mongodb</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-devtools</artifactId>
      <scope>runtime</scope>
      <optional>true</optional>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
```

```

        </dependency>
    </dependencies>

    <build>
        <finalName>spring-boot-mongodb</finalName>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

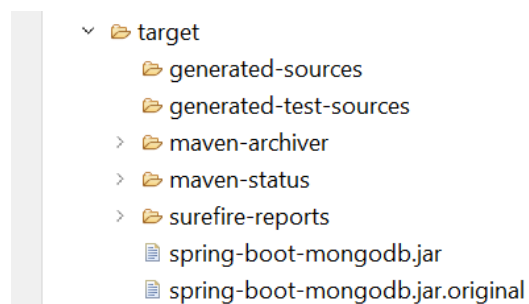
</project>

```

Note:

Whatever the name you are providing inside **<finalName>spring-boot-mongodb</finalName>** tag by that file name a new jar file (**spring-boot-mongodb.jar**) is created inside target folder.

If this tag is not provided, by default jar name will be **artifactId name** which is mentioning pom.xml.

**Generate Jar file**

Right click on Project -> Run As -> Maven clean

Right click on Project -> Run As -> Maven Install

Creating Image;

Go to the project location;

```
spring-boot-mongodb>docker build -t spring-boot-mongodb:1.0 .
```

Here dot(.) indicate current directory, must give one space

```
C:\Windows\System32\cmd.e  +  v

C:\workspace\my spring boot\spring-boot-mongodb>docker build -t spring-boot-mongodb:1.0 .
[+] Building 88.4s (7/7) FINISHED
=> [internal] load build definition from Dockerfile                                0.1s
=> => transferring dockerfile: 184B                                              0.1s
=> [internal] load .dockerignore                                                  0.1s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/openjdk:17                    4.5s
=> [1/2] FROM docker.io/library/openjdk:17@sha256:528707081fdb9562eb819128a9f85ae7fe000e2fbaeaf9f87662e7b3f38cb7d8 77.6s
=> => resolve docker.io/library/openjdk:17@sha256:528707081fdb9562eb819128a9f85ae7fe000e2fbaeaf9f87662e7b3f38cb7d8 0.1s
=> => sha256:38a980f2cc8accf69c23deae6743d42a87eb34a54f02396f3fcd7c2d06e2c5b 42.11MB / 42.11MB 25.2s
=> => sha256:528707081fdb9562eb819128a9f85ae7fe000e2fbaeaf9f87662e7b3f38cb7d8 1.04kB / 1.04kB 0.0s
=> => sha256:98f0304b3a3b7c12ce641177a99d1f3be56f532473a528fda38d53d519cafb13 954B / 954B 0.0s
=> => sha256:5e28ba2b4cdb3a7c3bd0ee2e635a5f6481682b77eabf8b51a17ea8bfe1c05697 4.45kB / 4.45kB 0.0s
=> => sha256:a7203ca35e75e068651c9907d659adc721dba823441b78639fde66fc988f042f 187.53MB / 187.53MB 66.5s
=> => sha256:de849f1cfbe60b1c06a1db83a3129ab0ea397c4852b98e3e4300b12ee57ba111 13.53MB / 13.53MB 11.9s
=> => extracting sha256:38a980f2cc8accf69c23deae6743d42a87eb34a54f02396f3fcd7c2d06e2c5b 10.2s
=> => extracting sha256:de849f1cfbe60b1c06a1db83a3129ab0ea397c4852b98e3e4300b12ee57ba111 2.9s
=> => extracting sha256:a7203ca35e75e068651c9907d659adc721dba823441b78639fde66fc988f042f 8.0s
=> [internal] load build context                                                3.9s
=> => transferring context: 23.69MB                                             3.7s
=> [2/2] ADD target/spring-boot-mongodb.jar spring-boot-mongodb.jar           5.2s
=> => exporting to image                                                        0.6s
=> => exporting layers                                                         0.5s
=> => writing image sha256:a4275f66a700cf2aa70d6a0fe1f2e81431afccaf10d514764a6b1ba83d6565a4 0.0s
=> => naming to docker.io/library/spring-boot-mongodb:1.0                   0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

C:\workspace\my spring boot\spring-boot-mongodb>
```

```
C:\workspace\my spring boot\spring-boot-mongodb>docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
spring-boot-mongodb	1.0	a4275f66a700	About a minute ago	495MB
mongo	latest	b70536aeb250	3 days ago	695MB

Run mongocontainer

```
>docker run -d -p 27017:27017 --name mongocontainer mongo:latest
```

Run spring-boot-mongodb

```
>docker run -p 8080:8080 --name spring-boot-mongodb --link mongocontainer:mongo -d spring-boot-mongodb:1.0
```

```
C:\Windows\System32\cmd.e
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
af2139cca62d   mongo:latest   "docker-entrypoint.s..." 26 minutes ago Up 26 minutes 0.0.0.0:27017->27017/tcp  mongocontainer

C:\workspace\my spring boot\spring-boot-mongodb>
C:\workspace\my spring boot\spring-boot-mongodb>docker rm -f af2139cca62d
af2139cca62d

C:\workspace\my spring boot\spring-boot-mongodb>docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
af2139cca62d   mongo:latest   "docker-entrypoint.s..." 26 minutes ago Up 26 minutes 0.0.0.0:27017->27017/tcp  mongocontainer

C:\workspace\my spring boot\spring-boot-mongodb>docker images
REPOSITORY    TAG          IMAGE ID      CREATED        SIZE
spring-boot-mongodb  1.0         a4275f66a700  11 minutes ago  495MB
mongo          latest      b70536aeb250  3 days ago    695MB

C:\workspace\my spring boot\spring-boot-mongodb>
C:\workspace\my spring boot\spring-boot-mongodb>docker run -d -p 27017:27017 --name mongocontainer mongo:latest
33e778d69f03c87a3f2e3aba95be9295739b770d95cf0778a1fd0e93ec743363

C:\workspace\my spring boot\spring-boot-mongodb>
C:\workspace\my spring boot\spring-boot-mongodb>docker run -p 8080:8080 --name spring-boot-mongodb --link mongocontainer:mongo -d spring-boot-mongodb:1.0
90ab8ffff4df8c798cd935576b4b2ad43738e309617ec5edf68339dfab2d92364

C:\workspace\my spring boot\spring-boot-mongodb>docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
90ab8ffff4df8  spring-boot-mongodb:1.0  "java -jar /spring-b..." 8 seconds ago  Up 7 seconds  0.0.0.0:8080->8080/tcp    spring-boot-mongodb
33e778d69f03   mongo:latest   "docker-entrypoint.s..." 29 seconds ago Up 28 seconds  0.0.0.0:27017->27017/tcp  mongocontainer
```

```
>docker logs spring-boot-mongodb
```

```
C:\Users\asdas>docker logs spring-boot-mongodb

:: Spring Boot :: (v2.7.0)

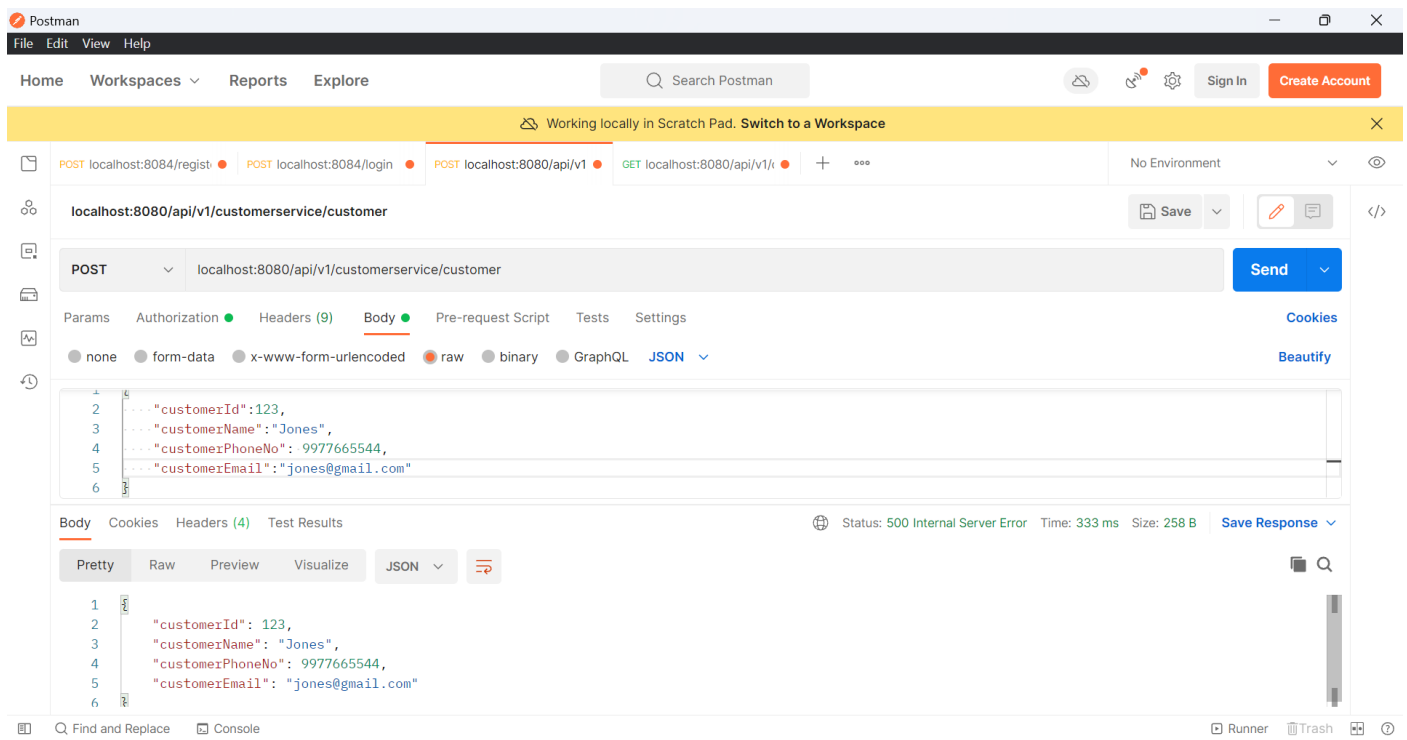
2022-06-08 19:23:40.264 INFO 1 --- [main] c.s.appl.SpringBootMongodbApplication : Starting SpringBootMongodbApplication v0.0.1-SNAPSHOT using Java 18.0.1.1 o
n 73e51603b0a2 with PID 1 (/spring-boot-mongodb.jar started by root in /)
2022-06-08 19:23:40.268 INFO 1 --- [main] c.s.appl.SpringBootMongodbApplication : No active profile set, falling back to 1 default profile: "default"
2022-06-08 19:23:41.945 INFO 1 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data MongoDB repositories in DEFAULT mode.
2022-06-08 19:23:42.247 INFO 1 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 295 ms. Found 1 MongoDB reposi
tory interfaces.
2022-06-08 19:23:43.629 INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2022-06-08 19:23:43.659 INFO 1 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-06-08 19:23:43.660 INFO 1 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.63]
2022-06-08 19:23:43.810 INFO 1 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2022-06-08 19:23:43.810 INFO 1 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 3447 ms
2022-06-08 19:23:44.095 INFO 1 --- [main] org.mongodb.driver.client : MongoClient with metadata {"driver": {"name": "mongo-java-driver|sync|sprin
g-boot", "version": "4.6.0"}, "os": {"type": "Linux", "name": "Linux", "architecture": "amd64"}, "version": "5.4.72-microsoft-standard-WSL2"}, "platform": "Java/Oracle Corpo
ration|18.0.1.1+2-6"} created with settings MongoClientSettings(readPreference=primary, writeConcern=WriteConcern(w=null, wTimeout=null ms, journal=null), retryWrites=true,
retryReads=true, readConcern=ReadConcern(level=null), credential=null, streamFactory=null, commandListeners=[], codecRegistry=ProvidersCodecRegistry(codecProviders=
[ValueCodecProvider(), BsonValueCodecProvider(), DBRefCodecProvider(), DBObjectCodecProvider(), DocumentCodecProvider(), IterableCodecProvider(), MapCodecProvider(), GeoIso
nCodecProvider(), GridFSFileCodecProvider(), Jsrs310CodecProvider(), JsonObjectCodecProvider(), BsonCodecProvider(), EnumCodecProvider(), com.mongodb.Jep395RecordCodecProvid
er@5dcd8c7a]), clusterSettings={hosts=[mongocontainer:27017], srvServiceName=mongodb, mode=SINGLE, requiredClusterType=UNKNOWN, requiredReplicaSetName='null', serverSelecto
r='null', clusterListeners=[]}, serverSelectionTimeout='30000 ms', localThreshold='30000 ms'}, socketSettings=SocketSettings(connectTimeoutMS=10000, readTimeoutMS=0, recei
veBufferSize=0, sendBufferSize=0), heartbeatSocketSettings=SocketSettings(connectTimeoutMS=10000, readTimeoutMS=10000, receiveBufferSize=0, sendBufferSize=0), connectionPoo
lSettings=ConnectionPoolSettings(maxSize=100, minSize=0, maxWaitTimeMS=120000, maxConnectionLifeTimeMS=0, maxConnectionIdleTimeMS=0, maintenanceInitialDelayMS=0, maintenanc
eFrequencyMS=60000, connectionPoolListeners=[], maxConnecting=2), serverSettings=ServerSettings(heartbeatFrequencyMS=10000, minHeartbeatFrequencyMS=500, serverListeners=[]
), serverMonitorListeners=[]), sslSettings=SslSettings(enabled=false, invalidHostNameAllowed=false, context=null), applicationName='null', compressorList=[], uuidRepresent
ation=JAVA_LEGACY, serverApi=null, autoEncryptionSettings=null, contextProvider=null)
2022-06-08 19:23:44.165 INFO 1 --- [container:27017] org.mongodb.driver.connection : Opened connection [connectionId{localValue:1, serverValue:1}] to mongoconta
iner:27017
2022-06-08 19:23:44.166 INFO 1 --- [container:27017] org.mongodb.driver.cluster : Monitor thread successfully connected to server with description ServerDesc
ription{address=mongocontainer:27017, type=STANDALONE, state=CONNECTED, ok=true, minWireVersion=0, maxWireVersion=13, maxDocumentSize=16777216, logicalSessionTimeoutMinutes
=30, roundTripTimeNanos=56446200}
2022-06-08 19:23:44.171 INFO 1 --- [container:27017] org.mongodb.driver.connection : Opened connection [connectionId{localValue:2, serverValue:2}] to mongoconta
iner:27017
2022-06-08 19:23:45.351 INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path '/'
2022-06-08 19:23:45.365 INFO 1 --- [main] c.s.appl.SpringBootMongodbApplication : Started SpringBootMongodbApplication in 5.84 seconds (JVM running for 6.719
s)
```

Testing Containerized Spring Boot MongoDB app;

Post Request;

localhost:8080/api/v1/customerservice/customer

```
{
  "customerId":123,
  "customerName":"Jones",
  "customerPhoneNo": 9977665544,
  "customerEmail":"jones@gmail.com"
}
```



Get Request;

localhost:8080/api/v1/customerservice/customer

Postman interface showing a GET request to `localhost:8080/api/v1/customerservice/customer`. The response is a JSON object with the following details:

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body (JSON):

```

1 {
2   "customerId": 123,
3   "customerName": "Jones",
4   "customerPhoneNo": 9977665544,
5   "customerEmail": "jones@gmail.com"
6 }
7
8

```

Status: 500 Internal Server Error. Time: 54 ms. Size: 260 B. Save Response

Check in Containerized MongoDB;

```
C:\Users\asdas>docker exec -it mongocontainer bash
```

```
root@87520b7f0b09:/# mongosh
```

```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.6.0
running for 9.015)

C:\workspace\my spring boot\spring-boot-mongodb>docker exec -it mongocontainer bash
root@833e778d69f03:/# mongosh
Current Mongosh Log ID: 635d49651aeae9c64059443a
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.6.0
Using MongoDB:      6.0.2
Using Mongosh:      1.6.0

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

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
The server generated these startup warnings when booting
2022-10-29T15:34:43.943+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/pr
odnotes-filesystem
2022-10-29T15:34:44.850+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2022-10-29T15:34:44.851+00:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting it to 'never'
2022-10-29T15:34:44.851+00:00: vm.max_map_count is too low
-----

-----
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
-----

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

test> use customerdb
switched to db customerdb
customerdb>

customerdb> show collections
customer
customerdb>

customerdb> db.customer.find()
[
  {
    _id: 101,
    customerName: 'Wills',
    customerPhoneNo: Long("8877665544"),
    customerEmail: 'wills@gmail.com',
    _class: 'com.seshu.appl.model.Customer'
  }
]
customerdb> |

```

```
show dbs
```

```
use customerdb
```

```
show collections
```

```
db.customer.find()
```


Removing the containers;

```
C:\workspace\my spring boot\spring-boot-mongodb>docker ps -a
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS                NAMES
90ab8fff4df8   spring-boot-mongodb:1.0  "java -jar /spring-b..."  11 minutes ago  Up 11 minutes  0.0.0.0:8080->8080/tcp  spring-boot-mongodb
33e778d69f03   mongo:latest          "docker-entrypoint.s..."  11 minutes ago  Up 11 minutes  0.0.0.0:27017->27017/tcp  mongocontainer

C:\workspace\my spring boot\spring-boot-mongodb>
C:\workspace\my spring boot\spring-boot-mongodb>docker rm -f 90ab8fff4df8
90ab8fff4df8

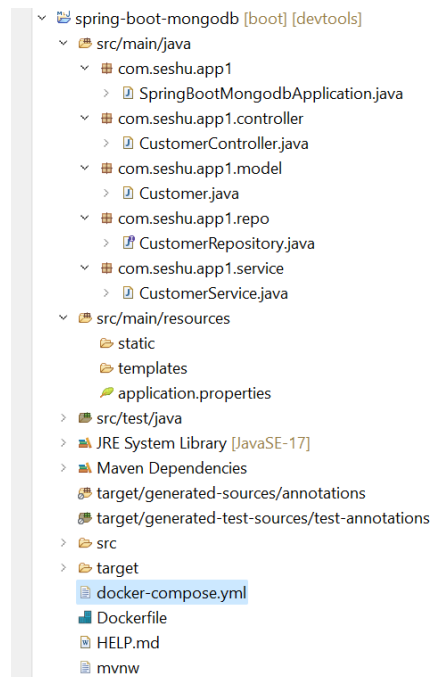
C:\workspace\my spring boot\spring-boot-mongodb>docker rm -f 33e778d69f03
33e778d69f03

C:\workspace\my spring boot\spring-boot-mongodb>docker ps -a
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS                NAMES
```

Executing Container using *docker-compose.yml*

Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your application's services. Then, with a single command, you create and start all the services from your configuration.

Compose works in all environments: production, staging, development, testing, as well as CI workflows. It also has commands for managing the whole lifecycle of your application



Add the following file inside project;

docker-compose.yml

```
version: "3"
services:
  mongocontainer:
    image: mongo:latest
    container_name: "mongocontainer"
    ports:
      - 27017:27017
  spring-boot-mongodb:
    image: spring-boot-mongodb:1.0
    container_name: spring-boot-mongodb
    ports:
      - 8080:8080
    links:
      - mongocontainer
```

Open the CMD from docker-compose.yml file and run as fallows

```
C:\workspace\my spring boot\spring-boot-mongodb>docker-compose up
```

```
C:\Windows\System32\cmd.exe x + v
C:\workspace>my spring boot\spring-boot-mongodb>docker-compose up
[+] Running 3/3
 - Network spring-boot-mongodb_default Created 0.3s
 - Container mongocontainer Created 0.5s
 - Container spring-boot-mongodb Created 0.4s
Attaching to mongocontainer, spring-boot-mongodb
mongocontainer | {"k":{"$date":"2022-10-29T16:07:17.019+00:00"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"-", "msg":"Initialized wire specification", "attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6,"maxWireVersion":17,"isInternalClient":true}}}}
mongocontainer | {"k":{"$date":"2022-10-29T16:07:17.114+00:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"main", "msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
mongocontainer | {"k":{"$date":"2022-10-29T16:07:17.143+00:00"},"s":"I", "c":"NETWORK", "id":4648601, "ctx":"main", "msg":"Implicit TCP FastOpen unavailable. If TCP FastOpen is required, set tcpFastOpenServer, tcpFastOpenClient, and tcpFastOpenQueueSize."}
mongocontainer | {"k":{"$date":"2022-10-29T16:07:17.185+00:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"main", "msg":"Successfully registered PrimaryOnlyService", "attr":{"service":"TenantMigrationDonorService", "namespace":"config.tenantMigrationDonors"}}
mongocontainer | {"k":{"$date":"2022-10-29T16:07:17.186+00:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"main", "msg":"Successfully registered PrimaryOnlyService", "attr":{"service":"TenantMigrationRecipientService", "namespace":"config.tenantMigrationRecipients"}}
mongocontainer | {"k":{"$date":"2022-10-29T16:07:17.187+00:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"main", "msg":"Successfully registered PrimaryOnlyService", "attr":{"service":"ShardSplitDonorService", "namespace":"config.tenantSplitDonors"}}
mongocontainer | {"k":{"$date":"2022-10-29T16:07:17.190+00:00"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"main", "msg":"Multi threading initialize done"}
mongocontainer | {"k":{"$date":"2022-10-29T16:07:17.200+00:00"},"s":"I", "c":"CONTROL", "id":4615611, "ctx":"initandlisten", "msg":"MongoDB starting", "attr":{"pid":1,"port":27017,"dbPath":"/data/db","architecture":"64-bit","host":"c3ddc573532b2"}}
mongocontainer | {"k":{"$date":"2022-10-29T16:07:17.203+00:00"},"s":"I", "c":"CONTROL", "id":23403, "ctx":"initandlisten", "msg":"Build Info", "attr":{"buildInfo":{"version":"6.0.2", "gitVersion":"94f67dfc8b974f15343e7ea394d0d9deedba50e", "opensslVersion":"OpenSSL 1.1.1f 31 Mar 2020", "modules":[], "allocator":"tcmalloc", "environment":{"distmod":"Ubuntu2004","distarch":"x86_64", "target_arch":"x86_64"}}}}
mongocontainer | {"k":{"$date":"2022-10-29T16:07:17.203+00:00"},"s":"I", "c":"CONTROL", "id":51765, "ctx":"initandlisten", "msg":"Operating System", "attr":{"os":{"name":"Ubuntu", "version":"20.04"}}}
mongocontainer | {"k":{"$date":"2022-10-29T16:07:17.203+00:00"},"s":"I", "c":"CONTROL", "id":21951, "ctx":"initandlisten", "msg":"Options set by command line", "attr":{"options":{"net":{"bindIp":"*"}}}}
mongocontainer | {"k":{"$date":"2022-10-29T16:07:17.222+00:00"},"s":"I", "c":"STORAGE", "id":22297, "ctx":"initandlisten", "msg":"Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem", "tags":["startupWarnings"]}
mongocontainer | {"k":{"$date":"2022-10-29T16:07:17.227+00:00"},"s":"I", "c":"STORAGE", "id":22315, "ctx":"initandlisten", "msg":"Opening WiredTiger", "attr":{"config":"Create, cache_size=1395M, session_max=33000, eviction=(threads_min=4, threads_max=4), config_base=false, statistics=(fast), log=(enabled=true, remove=true, path=journal, compression=snappy), builtin_extension_config=(zstd:(compression_level=6)), file_manager=(close_idle_time=600, close_scan_interval=10, close_handle_minimum=2000), statistics_log=(wait=0), json_output=(error, message), verbose=[recovery_progress:1, checkpoint_progress:1, compact_progress:1, backup:0, checkpoint:0, compact:0, evict:0, history_store:0, recovery:0, rts:0, salvage:0, tiered:0, timestamp:0, transaction:0, verify:0, log:0], list"}}
mongocontainer | {"k":{"$date":"2022-10-29T16:07:23.131+00:00"},"s":"I", "c":"STORAGE", "id":4795906, "ctx":"initandlisten", "msg":"WiredTiger opened
```

```
C:\Windows\System32\cmd.exe + v
}, "storage": {}, "protocol": "op_msg", "durationMillis": 247}}
spring-boot-mongodb
spring-boot-mongodb
spring-boot-mongodb
spring-boot-mongodb
spring-boot-mongodb
spring-boot-mongodb
spring-boot-mongodb
:: Spring Boot :: (v2.7.5)
spring-boot-mongodb
spring-boot-mongodb | 2022-10-29 16:07:29.575 INFO 1 --- [main] c.s.app1.SpringBootMongodbApplication : Starting SpringBootMongodbApplication v0.0.1-SNAPSHOT using Java 17.0.2 on 59f-c88b92d18 with PID 1 (/spring-boot-mongodb.jar started by root in /)
spring-boot-mongodb | 2022-10-29 16:07:29.590 INFO 1 --- [main] c.s.app1.SpringBootMongodbApplication : No active profile set, falling back to 1 default profile: "default"
spring-boot-mongodb | 2022-10-29 16:07:32.531 INFO 1 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data MongoDB repositories in DEFAULT mode.
spring-boot-mongodb | 2022-10-29 16:07:32.755 INFO 1 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 197 ms. Found 1 MongoDB repository interfaces.
spring-boot-mongodb | 2022-10-29 16:07:34.400 INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
spring-boot-mongodb | 2022-10-29 16:07:34.437 INFO 1 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
spring-boot-mongodb | 2022-10-29 16:07:34.438 INFO 1 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.68]
spring-boot-mongodb | 2022-10-29 16:07:34.703 INFO 1 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
spring-boot-mongodb | 2022-10-29 16:07:34.703 INFO 1 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 4894 ms
spring-boot-mongodb | 2022-10-29 16:07:36.026 INFO 1 --- [main] org.mongodb.driver.client : MongoClient with metadata {'driver': {'name': 'mongo-java-driver|sync|spring-boot', 'version': '4.6.1'}, 'os': {'type': 'Linux', 'name': 'Linux', 'architecture': 'amd64', 'version': '5.4.72-microsoft-standard-WSL2'}, 'platform': 'Java|Oracle Corporation|17.0.2+8-86'} created with settings MongoClientSettings{readPreference=primary, writeConcern=WriteConcern{level=null, wtimeout=null ms, journal=null}, retryWrites=true, retryReads=true, readConcern=ReadConcern{level=null}, credential=null, streamFactoryFactory=null, commandListeners=[], codecRegistry=ProvidersCodecRegistry{codecsProviders=[ValueCodecProvider{}, BSONValueCodecProvider{}, DBRefCodecProvider{}, DBObjectCodecProvider{}, DocumentCodecProvider{}, IterableCodecProvider{}, MapCodecProvider{}, GeoJsonCodecProvider{}, GridFSFileCodecProvider{}, Jsr310CodecProvider{}, JsonObjectCodecProvider{}, BSONCodecProvider{}, EnumCodecProvider{}], com.mongodb.Jep395RecordCodecProvider@2a448449}, clusterSettings={hosts=[mongocontainer:27017], srvServiceName=mongodb, mode=SINGLE, requiredClusterType=UNKNOWN, requiredReplicaSetName='null', serverSelector='null', clusterListeners=[], serverSelectionTimeout='30000 ms', localThreshold='30000 ms'}, socketSettings=SocketSettings{connectTimeoutMS=10000, readTimeoutMS=0, receiveBufferSize=0, sendBufferSize=0, heartbeatSocketSettings=SocketSettings{connectTimeoutMS=10000, readTimeoutMS=10000, receiveBufferSize=0, sendBufferSize=0}, connectionPoolSettings=ConnectionPoolSettings{maxSize=100, minSize=0, maxWaitTimeMS=120000, maxConnectionLifeTimeMS=0, maxConnectionIdleTimeMS=0, maintenanceInterval=0}}
```

Test service in postman

Postman interface showing a POST request to `localhost:8080/api/v1/customerservice/customer`. The request body is a JSON object with the following data:

```

1 {
2   "customerId": 123,
3   "customerName": "Jones",
4   "customerPhoneNo": 9977665544,
5   "customerEmail": "jones@gmail.com"
6 }

```

The response status is `500 Internal Server Error` with a time of `277 ms` and size of `258 B`.

Postman interface showing a GET request to `localhost:8080/api/v1/customerservice/customer`. The response status is `500 Internal Server Error` with a time of `42 ms` and size of `260 B`.

To stop;

Press ctrl+c

```
C:\Windows\System32\cmd.e  X  +  v
mongocontainer | {"t":{"$date":"2022-10-29T16:07:36.210+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn2","msg":"client metadata","attr":{"
"remote":"172.18.0.3:54084","client":"conn2","doc":{"driver":{"name":"mongo-java-driver|sync|spring-boot","version":"4.6.1"},"os":{"type":"Linux","name":"Li
nux","architecture":"amd64","version":"5.4.72-microsoft-standard-WSL2"},"platform":"Java/Oracle Corporation/17.0.2+8-86"}}}
spring-boot-mongodb | 2022-10-29 16:07:36.301 INFO 1 --- [container:27017] org.mongodb.driver.connection : Opened connection [connectionId{loca
lValue:2, serverValue:2}] to mongocontainer:27017
spring-boot-mongodb | 2022-10-29 16:07:36.302 INFO 1 --- [container:27017] org.mongodb.driver.cluster : Monitor thread successfully connecte
d to server with description ServerDescription{address=mongocontainer:27017, type=STANDALONE, state=CONNECTED, ok=true, minWireVersion=0, maxWireVersion=17,
maxDocumentSize=16777216, logicalSessionTimeoutMinutes=30, roundTripTimeNanos=113215200}
spring-boot-mongodb | 2022-10-29 16:07:36.309 INFO 1 --- [container:27017] org.mongodb.driver.connection : Opened connection [connectionId{loca
lValue:1, serverValue:1}] to mongocontainer:27017
spring-boot-mongodb | 2022-10-29 16:07:38.647 INFO 1 --- [ main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (htt
p) with context path ''
spring-boot-mongodb | 2022-10-29 16:07:38.674 INFO 1 --- [ main] c.s.appl.SpringBootMongodbApplication : Started SpringBootMongodbApplication
in 13.013 seconds (JVM running for 21.665)
spring-boot-mongodb | 2022-10-29 16:07:51.713 INFO 1 --- [nio-8080-exec-2] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServle
t 'dispatcherServlet'
spring-boot-mongodb | 2022-10-29 16:07:51.713 INFO 1 --- [nio-8080-exec-2] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServ
let'
spring-boot-mongodb | 2022-10-29 16:07:51.715 INFO 1 --- [nio-8080-exec-2] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms
mongocontainer | {"t":{"$date":"2022-10-29T16:07:53.135+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","
attr":{"remote":"172.18.0.3:54096","uid":"429d4483-831c-47b1-ab06-47f76e5f96d2","connectionId":3,"connectionCount":3}}
mongocontainer | {"t":{"$date":"2022-10-29T16:07:53.137+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn3","msg":"client metadata","attr":{"
"remote":"172.18.0.3:54096","client":"conn3","doc":{"driver":{"name":"mongo-java-driver|sync|spring-boot","version":"4.6.1"},"os":{"type":"Linux","name":"Li
nux","architecture":"amd64","version":"5.4.72-microsoft-standard-WSL2"},"platform":"Java/Oracle Corporation/17.0.2+8-86"}}}
spring-boot-mongodb | 2022-10-29 16:07:53.139 INFO 1 --- [nio-8080-exec-2] org.mongodb.driver.connection : Opened connection [connectionId{loca
lValue:3, serverValue:3}] to mongocontainer:27017
mongocontainer | {"t":{"$date":"2022-10-29T16:07:59.686+00:00"},"s":"I", "c":"STORAGE", "id":20320, "ctx":"conn3","msg":"createCollection","attr":
{"namespace":"customerdb.customer","uuidDisposition":"generated","uuid":{"$uuid":{"$uuid":"f2080052-9c94-4733-b398-952434ee4e97"},"options":{}}}}
mongocontainer | {"t":{"$date":"2022-10-29T16:07:59.767+00:00"},"s":"I", "c":"INDEX", "id":20345, "ctx":"conn3","msg":"Index build: done buildin
g","attr":{"buildUUID":null,"collectionUUID":{"$uuid":{"$uuid":"f2080052-9c94-4733-b398-952434ee4e97"},"namespace":"customerdb.customer","index":"_id","ide
nt":"index-8-7065004119212930095","collectionIdent":"collection-7-7065004119212930095","commitTimestamp":null}}
Gracefully stopping... (press Ctrl+C again to force)
[+] Running 2/2
 - Container spring-boot-mongodb Stopped 0.9s
 - Container mongocontainer Stopped 0.9s
canceled
C:\workspace\my spring boot\spring-boot-mongodb>
```