**JPOP Learnings:**

**9th June Session recording:**

<https://epam-my.sharepoint.com/:v:/p/sumit_trivedi/Ea7r0oSAzL5BidIWEB9DueoB90JWb4kpYh_0JenITYf_Lg>

git clone -b start https:/repo-url

Compile is deprecated in gradle 6 so use implementaiton only

<https://tomgregory.com/gradle-implementation-vs-compile-dependencies/>

Swagger:

<https://www.javainuse.com/spring/boot_swagger_annotations>

<http://localhost:8080/swagger-ui.html#/>

<http://localhost:8080/v2/api-docs>

<https://www.vojtechruzicka.com/documenting-spring-boot-rest-api-swagger-springfox/> \*\*\*

OPEN API:

<https://springdoc.org/>

PGADMIN:

Master password: test

Spring DATA JPA:

@SpringBootApplication(exclude = {DataSourceAutoConfiguration.class }) << Added to avoid mandatory db config.

**enable jpa repositories vs enable jpa auditing:**

<https://www.baeldung.com/database-auditing-jpa>

<https://dzone.com/articles/spring-data-jpa-auditing-automatically-the-good-stuff>

ResponseEntity??????

@Id @Type(type = "pg-uuid")private UUID id;

@Type specifies what type is used on the database. You need to specify since PostgreSQL has its own type for UUIDs: pg-uuid

**javax.validation.constraints.NotNull;**

Add this dependency:

implementation 'org.springframework.boot:spring-boot-starter-validation'

Persist enum value in database

<https://www.baeldung.com/jpa-persisting-enums-in-jpa>

@Enumerated(EnumType.STRING)

**private** Type type;

GIT

git init

git remote add origin <https://github.com/rupeshpatil761/book-service.git>

**git config** --global **user.name**

**git config** --global **user.email**

**Profiling:**

spring.profiles.active=dev << application.properties

application-dev.properties

**User Service:**

server.port=9091

<http://localhost:9091/swagger-ui.html>

@Column(name="email",unique = true)

private String email;

Library service:

**Composite Primary Keys**

<https://www.baeldung.com/jpa-composite-primary-keys>

\*\*Must have default constructor in LibraryId class

<https://stackoverflow.com/questions/56868587/spring-data-jpa-while-saving-duplicate-composite-key-doesnt-get-any-error/56871112>

**Spring Cloud:**

--Must have @GetMapping or other annotations at method level in proxy class.

--Must have @EnableFeignClients(basePackages = "com.library.service") annotation at main class level to enable creation of proxy beans.

--@PathVariable(“id”) << When using feign we need to explicitly mention variable name in quotes.

--"trace": "feign.FeignException$NotFound: [404] during [GET] to [http://localhost:8081/10] << Make sure we added the uri value in proxy interface endpoints e.g.

@GetMapping("/users/{id}") <<< Correct

@GetMapping("/{id}") << Wrong

--@LoadBalancerClient() <<Either name or value must be provided

--"trace": "feign.FeignException$ServiceUnavailable: [503] during [GET] to [http://user-service/users] [UserServiceProxy#getAllUsers()]: [Load balancer does not contain an instance for the service user-service]

GenericExceptionHandler

--https://medium.com/@aamine/customized-exception-handling-in-spring-boot-e4546e180a2d

<https://stackoverflow.com/questions/41401009/load-balancer-does-not-have-available-server-for-client>

How we can have multi line config: Like below

say-hello:

ribbon:

eureka:

enabled: false

listOfServers: localhost:8090,localhost:9092,localhost:9999

ServerListRefreshInterval: 15000

**versions-spring-boot-spring-cloud-ribbon-not-working**

<https://spring.io/guides/gs/spring-cloud-loadbalancer/>

<https://stackoverflow.com/questions/65469735/versions-spring-boot-spring-cloud-ribbon-not-working>

<https://github.com/spring-cloud-samples/spring-cloud-intro-demo>

<https://dzone.com/articles/spring-cloud-netflix-load-balancer-with-ribbonfeig>

**crudrepository vs jparepository spring boot**

[JpaRepository](http://static.springsource.org/spring-data/data-jpa/docs/current/api/org/springframework/data/jpa/repository/JpaRepository.html) extends [PagingAndSortingRepository](http://static.springsource.org/spring-data/data-commons/docs/current/api/org/springframework/data/repository/PagingAndSortingRepository.html) which in turn extends [CrudRepository](http://static.springsource.org/spring-data/data-commons/docs/current/api/org/springframework/data/repository/CrudRepository.html).

Their main functions are:

* [CrudRepository](http://static.springsource.org/spring-data/data-commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) mainly provides CRUD functions.
* [PagingAndSortingRepository](http://static.springsource.org/spring-data/data-commons/docs/current/api/org/springframework/data/repository/PagingAndSortingRepository.html) provides methods to do pagination and sorting records.
* [JpaRepository](http://static.springsource.org/spring-data/data-jpa/docs/current/api/org/springframework/data/jpa/repository/JpaRepository.html) provides some JPA-related methods such as flushing the persistence context and deleting records in a batch.

Because of the inheritance mentioned above, JpaRepository will have all the functions of CrudRepository and PagingAndSortingRepository. So if you don't need the repository to have the functions provided by JpaRepository and PagingAndSortingRepository , use CrudRepository.

**"org.springframework.dao.InvalidDataAccessApiUsageException: No EntityManager with**

**actual transaction available for current thread - cannot reliably process 'remove' call; nested exception is javax.persistence.TransactionRequiredException: No EntityManager with actual transaction available for current thread - cannot reliably process 'remove' call**

>> <https://stackoverflow.com/questions/32269192/spring-no-entitymanager-with-actual-transaction-available-for-current-thread>

\*\*Use Transcational at serviceImpl level or public level method

**Open Questions:**

1. What is diff between Ribbon and LoadBalancer
2. What changes needs to be done in order to use Loadbalancer and Feign
3. What are diff implementations of cloud load balancer?
4. Remove Id from try swagger body
5. Save method how updates entry automatically : issueBook api
6. Having single repo and create separate branch for each service \*\*\*

<https://dzone.com/articles/spring-boot-and-swagger-documenting-restful-servic>

**Learn and implement:**

**spring-boot-app-with-flyway-and-postgres \*\*\*\*\*\*\*\*\*\*\***

<https://dzone.com/articles/build-a-spring-boot-app-with-flyway-and-postgres>

<https://spring.io/projects/spring-cloud-contract>

axon framework spring boot: <https://www.baeldung.com/axon-cqrs-event-sourcing>

<https://docs.axoniq.io/reference-guide/axon-framework/spring-boot-integration>

==================================

select \* from book;

select \* from library\_user;

select \* from library;

**Spring cloud dependency updates:**

**If you are using 2.4.0, you need to add this dependency to the pom.xml:**

implementation 'org.springframework.cloud:spring-cloud-starter-bootstrap' << Not mandatory

**Spring cloud config server:**

**Setup local git repo:**

1. Create new folder >> git init >> add new files >> git add . >> git commit (No need to push)

@EnableConfigServer on main class and configure git repo uri:

spring.cloud.config.server.git.uri=file:///C:/Users/Rupesh\_Patil/Downloads/JPOP Workspace/git-localconfig-repo

Access: <http://localhost:8888/user-service/default>

<http://localhost:8888/user-service/stage>

**Connecting microservice to config server:**

1. Rename application.properties to bootstrap.properties
2. Add below dependencies in user-service:

implementation 'org.springframework.cloud:spring-cloud-starter-bootstrap'

implementation 'org.springframework.cloud:spring-cloud-starter-config'

1. Add below dependencies in config-server:

implementation 'org.springframework.cloud:spring-cloud-config-server'

implementation 'org.springframework.cloud:spring-cloud-starter-config'

Disabling discovery from client.

eureka.client.enabled=false

<https://github.com/shabbirdwd53/Springboot-Microservice/tree/kubernetes>

https://stackoverflow.com/questions/35142105/how-to-selectively-disable-eureka-discovery-client-with-spring

**RefreshScope:**

1. Created a UserConfiguration class
2. Added below property in bootstrap.properties of user-service( running on port 9091)

management.endpoints.web.exposure.include=refresh

1. Added properties in user-service-dev.properties of git repo and committed changes
2. Called POST localhost:9091/actuator/refresh endpoint using POSTMAN

<https://spring.io/guides/gs/centralized-configuration/>

<https://medium.com/analytics-vidhya/spring-cloud-config-server-and-good-practice-of-refresh-scope-usage-ef65d0fee379>

<https://www.youtube.com/watch?v=yNnLICy2zk4>

<https://www.baeldung.com/configuration-properties-in-spring-boot>

**Flyway:**

There are several tools to handle database migrations, and one of the most popular is [Flyway](https://flywaydb.org/), which works flawlessly with Spring Boot. Briefly, Flyway looks for SQL scripts on your project’s resource path and runs all scripts not previously executed in a defined order. Flyway stores what files were executed into a particular table called SCHEMA\_VERSION

Dependency: implementation 'org.flywaydb:flyway-core'

Create v2\_\_ddl.sql and v3\_\_data.sql files under /resources/db/migration

Change spring.jpa.hibernate.ddl-auto=validate

spring.flyway.baseline-on-migrate = true << without this schema table will not get created

Caused by: org.flywaydb.core.api.FlywayException: Found more than one migration with version 1

>> ddl and data file must not have same version id

And Version must be always greater than the current version schema version table

Successfully baselined schema with version: 1 >>> So ddl version must be greater than 1 i.e. 2 and data version must be 3

Book Server Flyway integration with same version files: Got below error

Error 1) flywayexception validate failed migration checksum mismatch for migration version 2:

>>>> Tried changing files version in book service after that got below error

Error 2) /validate-failed-detected-applied-migration-not-resolved-locally-flyway

>>> Same flyway\_schema\_version table we cannot use for multiple tables. Added below property in conf file:

>>spring.flyway.table=flyway\_book\_version

select \* from library\_user;

select \* from book;

select \* from flyway\_book\_version;

select \* from flyway\_library\_user\_version;

<https://dzone.com/articles/build-a-spring-boot-app-with-flyway-and-postgres>

<https://www.thomasvitale.com/spring-cloud-config-basics/#commento-login-box-container>

**Naming Server:**

Error: Add a spring.config.import=configserver: property to your configuration.

Added bootstrap dependency in build.gradle

implementation 'org.springframework.cloud:spring-cloud-starter-config'  
implementation 'org.springframework.cloud:spring-cloud-starter-bootstrap'  
implementation 'org.springframework.cloud:spring-cloud-starter-netflix-eureka-server'

**Annotations:**

@EnableEurekaServer

Client Apps Side:

Dependency:

implementation 'org.springframework.cloud:spring-cloud-starter-bootstrap'  
implementation 'org.springframework.cloud:spring-cloud-starter-config'  
implementation 'org.springframework.cloud:spring-cloud-starter-netflix-eureka-client'

@EnableDiscoveryClient

Bootstrap.properties

eureka.client.serviceUrl.defaultZone=http://localhost:8761/eureka

**API Gateway:**

Dependencies:

implementation 'org.springframework.cloud:spring-cloud-starter-gateway'  
implementation 'org.springframework.cloud:spring-cloud-starter-netflix-eureka-client'

ERROR:

io.netty.resolver.dns.DnsResolveContext$SearchDomainUnknownHostException: Search domain query failed. spring cloud api gateway

<https://github.com/spring-cloud/spring-cloud-gateway/issues/2091>

<https://github.com/reactor/reactor-netty/issues/1431>

To enable api-gateway feature we need add property in app.properties file of api-gateway

<http://localhost:8765/USER-SERVICE/users> <<< Whitelable error without adding above property

<http://localhost:8765/books> <<< internally api gateway redirects request to book service by doing load balancing

[**http://localhost:8765/library/books/12**](http://localhost:8765/library/books/12)

**DynamoDB Impl:**

implementation 'software.amazon.awssdk:dynamodb:2.16.87'

implementation 'com.github.derjust:spring-data-dynamodb:5.1.0'

<https://github.com/aws/aws-sdk-java-v2/#using-the-sdk>

<https://www.baeldung.com/spring-data-dynamodb>

<https://github.com/michaellavelle/spring-data-dynamodb/tree/v5.1.0>

<https://github.com/awsdocs/aws-doc-sdk-examples/tree/master/java/example_code>

<https://stackoverflow.com/questions/60706042/spring-boot-application-the-bean-could-not-be-registered>

<https://tech.smartling.com/getting-started-with-amazon-dynamodb-and-java-universal-language-850fa1c8a902>

<https://www.youtube.com/watch?v=3ay92ZdCgwQ>

Que 1) How to take integer id as primary key and autogererate key in dyanmodb.

**Errors:**

com.amazonaws.services.dynamodbv2.datamodeling.DynamoDBMappingException: could not instantiate class com.amazonaws.services

>>> Resolved by adding n-arg constructor

could not instantiate class com.amazonaws.services.dynamodbv2.datamodeling.DynamoDBGeneratedUuid$Generator

>>Resolved by making id field type as String

**Circuit Breaker:**

<https://www.exoscale.com/syslog/migrate-from-hystrix-to-resilience4j/>

Add below dependency:

implementation 'org.springframework.cloud:spring-cloud-starter-circuitbreaker-resilience4j'

Ref Links:

<https://www.exoscale.com/syslog/migrate-from-hystrix-to-resilience4j/>

<https://yurynino.medium.com/my-first-approach-moving-from-hystrix-to-resilience4j-21b590cc601>

<https://www.baeldung.com/resilience4j>

<https://docs.spring.io/spring-cloud-circuitbreaker/docs/current/reference/html/>

<https://www.youtube.com/watch?v=LaZGedpfAyM> <<

<https://www.baeldung.com/spring-cloud-circuit-breaker>

<https://github.com/spring-cloud-samples/spring-cloud-circuitbreaker-demo>

<https://reflectoring.io/circuitbreaker-with-resilience4j/>

<https://resilience4j.readme.io/docs/examples>

<https://www.youtube.com/watch?v=fkQHtw35alY>

<https://www.youtube.com/watch?v=IqDDw6ZhhgY>

<https://www.appsdeveloperblog.com/feign-error-handling-with-errordecoder/>

<https://cloud.spring.io/spring-cloud-netflix/multi/multi_spring-cloud-feign.html#spring-cloud-feign-hystrix-fallback>

<https://stackoverflow.com/questions/46662265/feign-client-error-handling> <\*\*\*\*\*\*\*\*\*

<https://stackoverflow.com/questions/55020389/spring-feign-client-exception-handling>

**June 15:**

* Configure**Service#1** enable **DynamoDB**
* Configure **Service#2** enable **AWS S3 Object Storage**
* Spring Cloud Contracts
* API Gateway

**Spring Cloud Contracts:**

Alternate libraries: <https://docs.pact.io/>

<https://www.baeldung.com/pact-junit-consumer-driven-contracts>

<https://spring.io/projects/spring-cloud-contract>

<https://github.com/spring-cloud-samples/spring-cloud-contract-samples>

<https://cloud.spring.io/spring-cloud-contract/2.0.x/multi/multi__spring_cloud_contract_verifier_setup.html>

<https://cloud.spring.io/spring-cloud-contract/reference/html/gradle-project.html>

<https://www.youtube.com/watch?v=GqN8OoODMOI>

mvn update ~= ./gradlew build --refresh-dependencies

mvn clean install ~= ./gradlew clean build

In real time projects… Individual team is responsible for producer and consumer contracts.

**Producer Side:**

1. Add maven / gradle support
2. Create contract in groovy src/test/resources/contracts
3. Create base setup
4. Build artifacts

You will find the generated test here:

\book-service\build\generated-test-sources\contractTest\java\org\springframework\cloud\contract\verifier\tests\ContractVerifierTest.java

<https://www.yawintutor.com/java-lang-exception-no-runnable-methods/>

**Consumer Side:**

1. Add maven / gradle support
2. Create contract test case (Mock MVC)
3. Run Stub
4. Run contract test case

\*\*\*\* Make sure you use the same version of Junit either junit or junit Jupiter in auto generated contract test and base contract test and same version at consumer side.

./gradle clean build publishToMavenLocal

<https://stackoverflow.com/questions/59774232/how-to-run-local-contracts-with-spring-cloud-contract-stub-runner>

**Spring Cloud Sleuth, Zipkin and ELK:**

**traceId and span id**

**trace id will be same but span ids will be different**

<https://spring.io/projects/spring-cloud-sleuth>

<https://spring.io/projects/spring-cloud-consul>

<https://stackoverflow.com/questions/50907463/adding-custom-trace-id-in-sleuth>

<https://hub.docker.com/r/openzipkin/zipkin/>

<https://github.com/in28minutes/spring-microservices>

<https://www.rabbitmq.com/download.html>

<https://dzone.com/articles/spring-cloud-sleuth-rabbitmq-zipkin-elasticsearch>

<https://livebook.manning.com/book/spring-microservices-in-action-second-edition/chapter-11/v-7/158> <<<<<\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<https://medium.com/swlh/distributed-tracing-in-micoservices-using-spring-zipkin-sleuth-and-elk-stack-5665c5fbecf> <<<\*\*\*\*\*\*\*\*\*\*

docker run -it --rm --name rabbitmq -p 5672:5672 -p 15672:15672 rabbitmq:3-management

<http://localhost:5672/rabbitmq/>

docker run -d -p 9411:9411 openzipkin/zipkin

http://localhost:9411/zipkin/

Docker HUB:

rupeshpatil761 / rdpdockerhub461

Dependencies:

implementation 'org.springframework.boot:spring-boot-starter-amqp'  
implementation 'org.springframework.cloud:spring-cloud-sleuth-zipkin'  
implementation 'org.springframework.cloud:spring-cloud-starter-sleuth'  
testImplementation 'org.springframework.amqp:spring-rabbit-test'

curl -sSL <https://zipkin.io/quickstart.sh> | bash -s  
java -jar zipkin.jar

**Docker:**

Docker Image: Image is like a Class. Its version static version.

Docker Container: Container is like a Object. Running version of image is container

Docker HuB >> Repositories >> in28Minutes/test-app >> Tag 1.0.0.RELEASE

--docker run in28Minutes/test-app:1.0.0.RELEASE

--docker run **-p 5000:5000** in28Minutes/test-app:1.0.0.RELEASE

HostPort:ContainerPort

--docker run -p 5000:5000 **-d** in28Minutes/test-app:1.0.0.RELEASE << Detached mode

--docker logs <container-id>

-- docker logs **-f** <container-id> << tailing logs

--docker container ls << running containers

-- docker run -p **5001**:5000 **-d** in28Minutes/test-app:1.0.0.RELEASE

--docker images << shows locally pulled images

--docker container ls -a <<< all containers

--docker container stop 3894

------------------------------------------------------------------------------------------

**Docker Client**: It’s the interface where we execute commands.

**Docker Daemon**: It manages containers, local images, image registry.

Docker Desktop has both Client and Daemon.

**Containerization:**

Cloud Infra >> Host OS >> Docker Engine >> Containers

**Virtualization** >> Hardware >> Host OS >> Hypervisor >> [{GuestOS1, Software 1, Application1} , {GuestOS2, Software 2, Application2} ]

------------------------------------------------------------------------------------------

**Docker Image Commands:**

--docker images

--**docker tag** in28Minutes/test-app:1.0.0.RELEASE in28Minutes/test-app**:latest**

--**docker pull** mysql

--**docker search** mysql

--**docker image history** <image-id>

--**docker image inspect**<image-id>

--**docker image remove** <image-id> << Image will be removed from local

------------------------------------------------------------------------------------------

**Docker Container Commands:**

--docker container **run -p {hostport:container-port} -d**

--docker container **logs** <container-id>

--docker container **pause** <container-id>

--docker container **unpause** <container-id>

--docker container **inspect** <container-id>

--docker container **prune** << Will remove all stopped containers

--docker container **stop** <container-id> << Graceful shutdown (SIGTERM)

--docker container **kill** <container-id> << Immediately terminates the process (SIGKILL)

--docker run -p 5001:5000 -d **--restart=always** in28Minutes/test-app:1.0.0.RELEASE

------------------------------------------------------------------------------------------

--docker **events** << To see what events are happening on docker env.

--docker **top** <id> << what is the top process running in that container.

--docker **stats** <<< all the stats for running containers (memory and cpu utilization)

-- docker run -p 5001:5000 **-m 512m –cpu-quota 5000** -d in28Minutes/test-app:1.0.0.RELEASE

100000 << 1--% so we are assigning 5% i.e. 5000

--docker **system df** << Shows details of containers, images, volumes in tabular format.

------------------------------------------------------------------------------------------

**Launching zipkin container using docker:**

docker pull openzipkin/zipkin

docker run -d -p 9411:9411 openzipkin/zipkin

localhost:9411/zipkin

<https://github.com/in28minutes/spring-microservices-v2/tree/main/04.docker>

Installing ELK Stack:

**beats-input.conf**

input {

tcp {

port => 5044

ssl => false

}

}

====================================================

**logstash-output.conf**

filter {

json {

source => "message"

}

}

output {

elasticsearch {

hosts => ["localhost:9200"]

manage\_template => false

index => "logstash-local"

}

}

====================================================

**Dockerfile (save as All types)**

FROM sebp/elk

# overwrite existing file

RUN rm /etc/logstash/conf.d/logstash-output.conf

COPY logstash-output.conf /etc/logstash/conf.d/logstash-output.conf

RUN rm /etc/logstash/conf.d/beats-input.conf

COPY beats-input.conf /etc/logstash/conf.d/beats-input.conf

====================================================

https://stackoverflow.com/questions/64985913/failed-to-solve-with-frontend-dockerfile

Go to The Files created directory and gitbash:

docker build . --tag local-elk

docker run -p 5601:5601 -p 9200:9200 -p 5044:5044 -it --name elk local-elk

the input device is not a TTY. If you are using mintty, try prefixing the command with 'winpty'

<https://stackoverflow.com/questions/48623005/docker-error-the-input-device-is-not-a-tty-if-you-are-using-mintty-try-prefi>

winpty docker run -p 5601:5601 -p 9200:9200 -p 5044:5044 -it --name elk local-elk

<https://www.youtube.com/watch?v=H_1BQdvdlEg> \*\*

<https://www.youtube.com/watch?v=I2ZS2Wlk1No>

<https://www.youtube.com/watch?v=-v1L3ym52I4> \*\* for windows 10

**Errors encountered:**

1. elasticsearch\_1 exited with code 78
2. ERROR: [2] bootstrap checks failed elasticsearch\_1 |

[1]: max virtual memory areas vm.max\_map\_count [65530] is too low, increase to at least [262144] elasticsearch\_1 |

<https://github.com/docker/for-win/issues/5202>

**Solution**:

Open powershell

wsl -d docker-desktop

echo "vm.max\_map\_count = 262144" > /etc/sysctl.d/99-docker-desktop.conf

[2]: the default discovery settings are unsuitable for production use; at least one of [discovery.seed\_hosts, discovery.seed\_providers, cluster.initial\_master\_nodes] must be configured

**Solution**: Add below settings under elasticsearch resource in docker-compose file.

enviornment:

- discovery.type=single-mode

Docker-compose up -d

<https://github.com/sohangp/SleuthAndZipkin/blob/master/customer-service/pom.xml>

<http://localhost:8761/> : Eureka

<http://localhost:5601/app/dev_tools#/console> : Kibana

<http://localhost:9411/zipkin/?lookback=15m&endTs=1627620378197&limit=10> : Zipkin

<https://medium.com/swlh/distributed-tracing-in-micoservices-using-spring-zipkin-sleuth-and-elk-stack-5665c5fbecf>

./logstash-pipeline/ports.conf:

-- Unknown seeting ssl => true

<https://www.youtube.com/watch?v=83Z1fOz0bOw> \* \*\*\*\*

<https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-delete-index.html>

<http://localhost:5601/app/dev_tools#/console?load_from=https:/www.elastic.co/guide/en/elasticsearch/reference/current/snippets/2045.console>

PUT /library-mgmt-system-docker

<https://www.youtube.com/watch?v=4rju8N60T7Q>

<https://github.com/deviantony/docker-elk/blob/main/docker-compose.yml>

<https://www.youtube.com/watch?v=6bXSfjwQVIc>

<https://github.com/reshmik/zipkin>

<https://stackoverflow.com/questions/50042514/what-is-the-dockerfile-extension>

**Dockerization:**

Create DockerFile in each microservice root directory

Create Docker-Compose file and define all microservices with required configuration.

\*\*Make sure the snapshot jars are available /build/libs folder

\*\*Need to create the docker images first

So, run below command for it

docker build -f book-service/Dockerfile -t jpop/book-service:0.0.1-SNAPSHOT -t jpop/book-service:latest .

Here, we will have new version with 0.0.2-snapshot and latest version will also get replaced.

And finally run docker-compose up