TP 2 : Introduction à Docker Dans un Contexte DevOps

DevOps

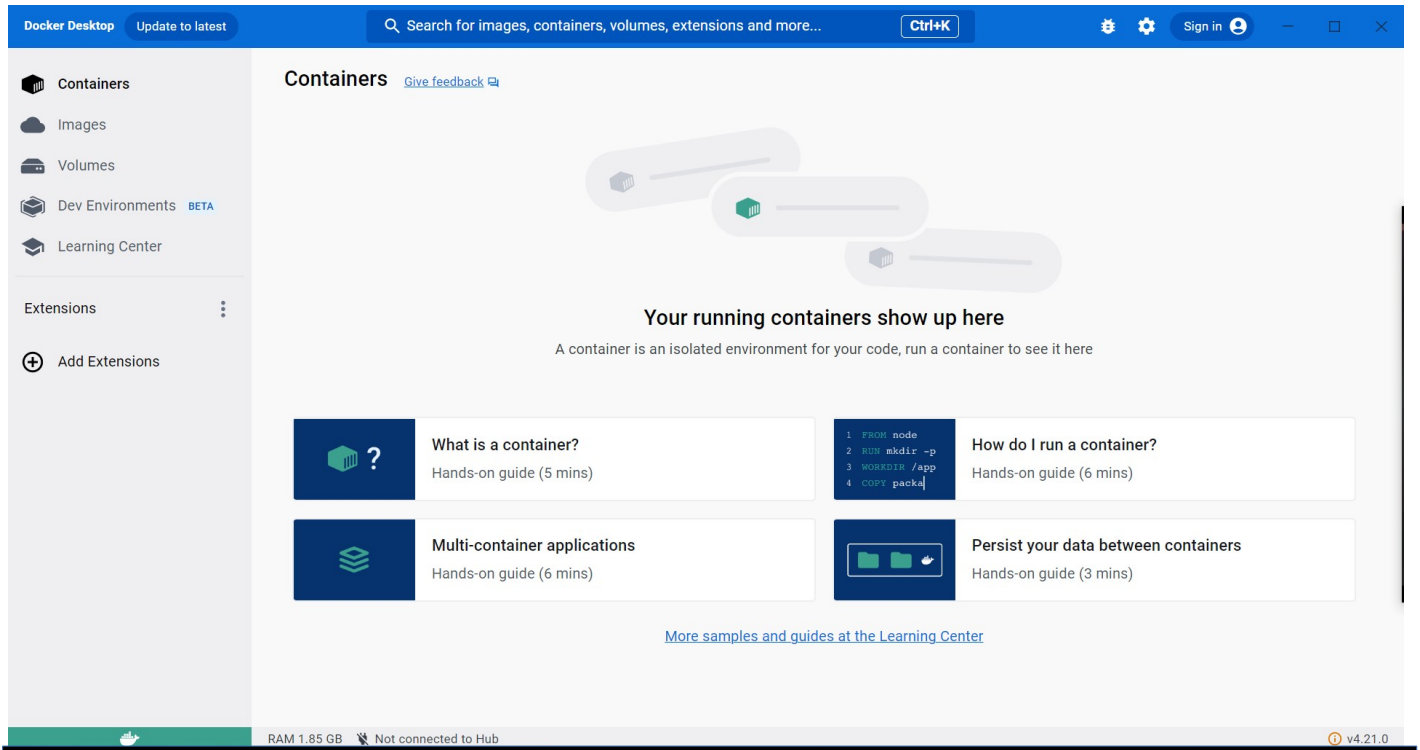
RT5_groupe2

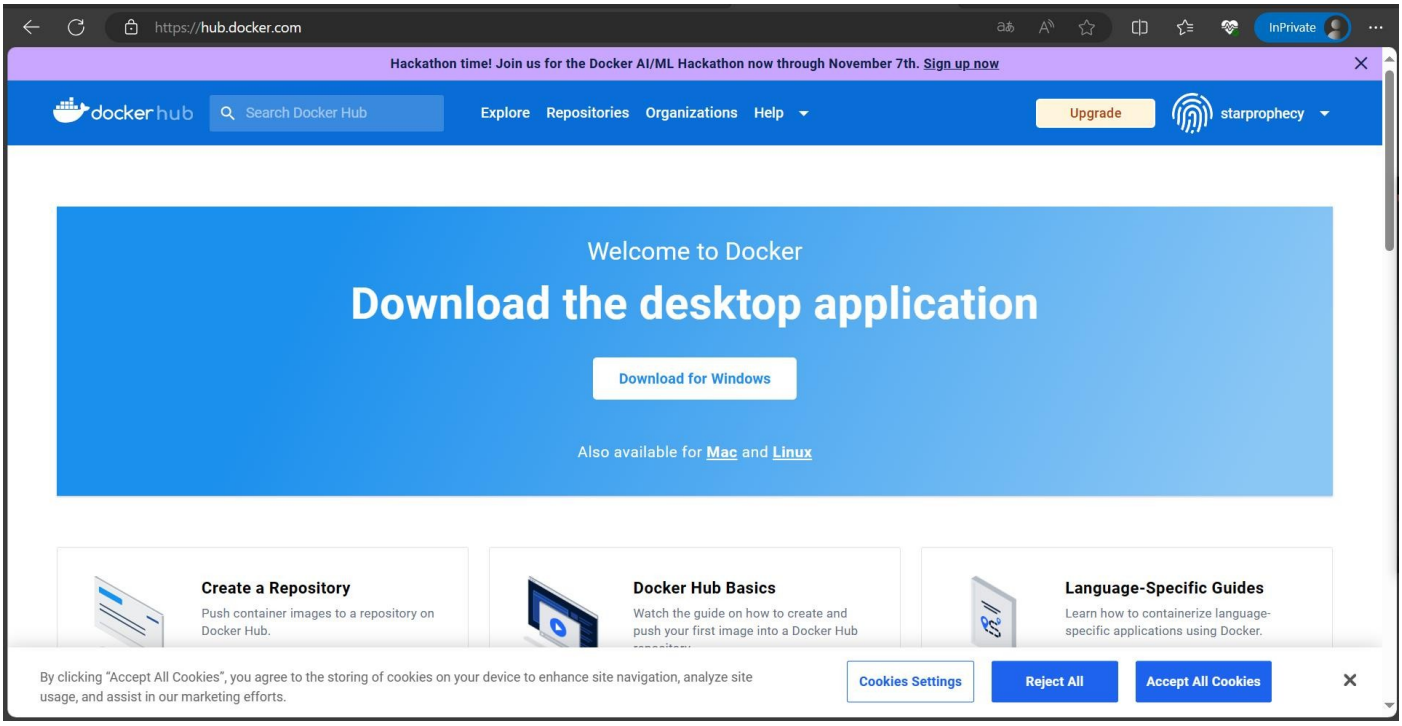
- Réalisé par:

- Gharsallah Ahmed
- Bouchhiwa Hassen
- Ben Jemaa Mouhib

Année universitaire: 2023-2024

1- Introduction à docker



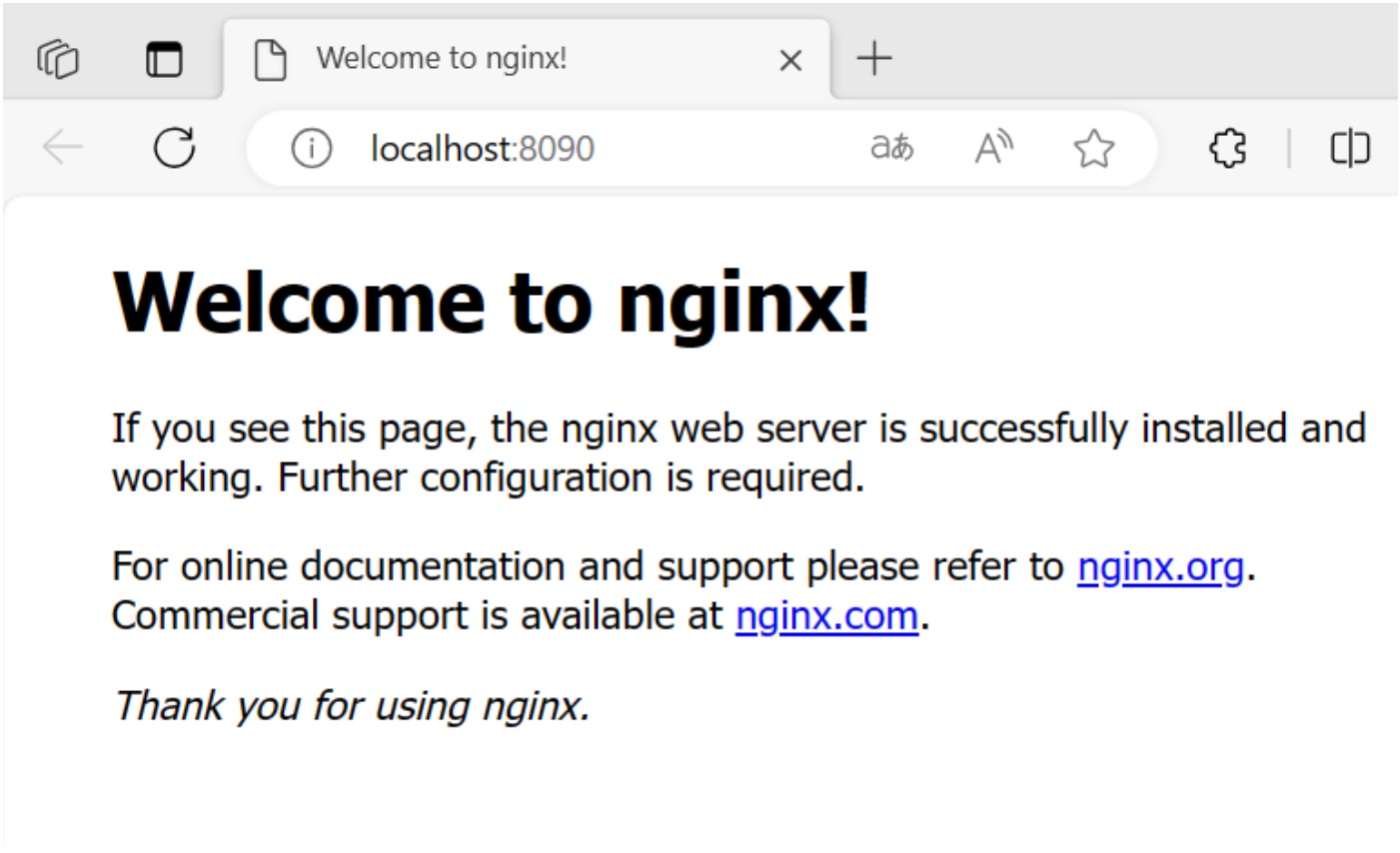


2- Utilisation de docker

- Création d'un conteneur Docker à partir d'une image existante : nginx port 8090

```
Sélection C:\Windows\System32\cmd.exe
Microsoft Windows [version 10.0.19045.3448]
(c) Microsoft Corporation. Tous droits réservés.

[redacted]>docker run -d -p 8090:80 --name nginx8090 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
a378f10b3218: Pull complete
4dfff0708538: Pull complete
2135e49ace4b: Pull complete
c843f6b280ce: Pull complete
6f35ab6f1400: Pull complete
6c538b49fa4a: Pull complete
d57731fb9008: Pull complete
Digest: sha256:b4af4f8b6470febf45dc10f564551af682a802eda1743055a7dfc8332dffa595
Status: Downloaded newer image for nginx:latest
74911f6f05967bb4fcdd5d9cf1f00f8058426b754b03bec399cfd56f8c08e76f
```

-Test et validation du fonctionnement du conteneur nginx



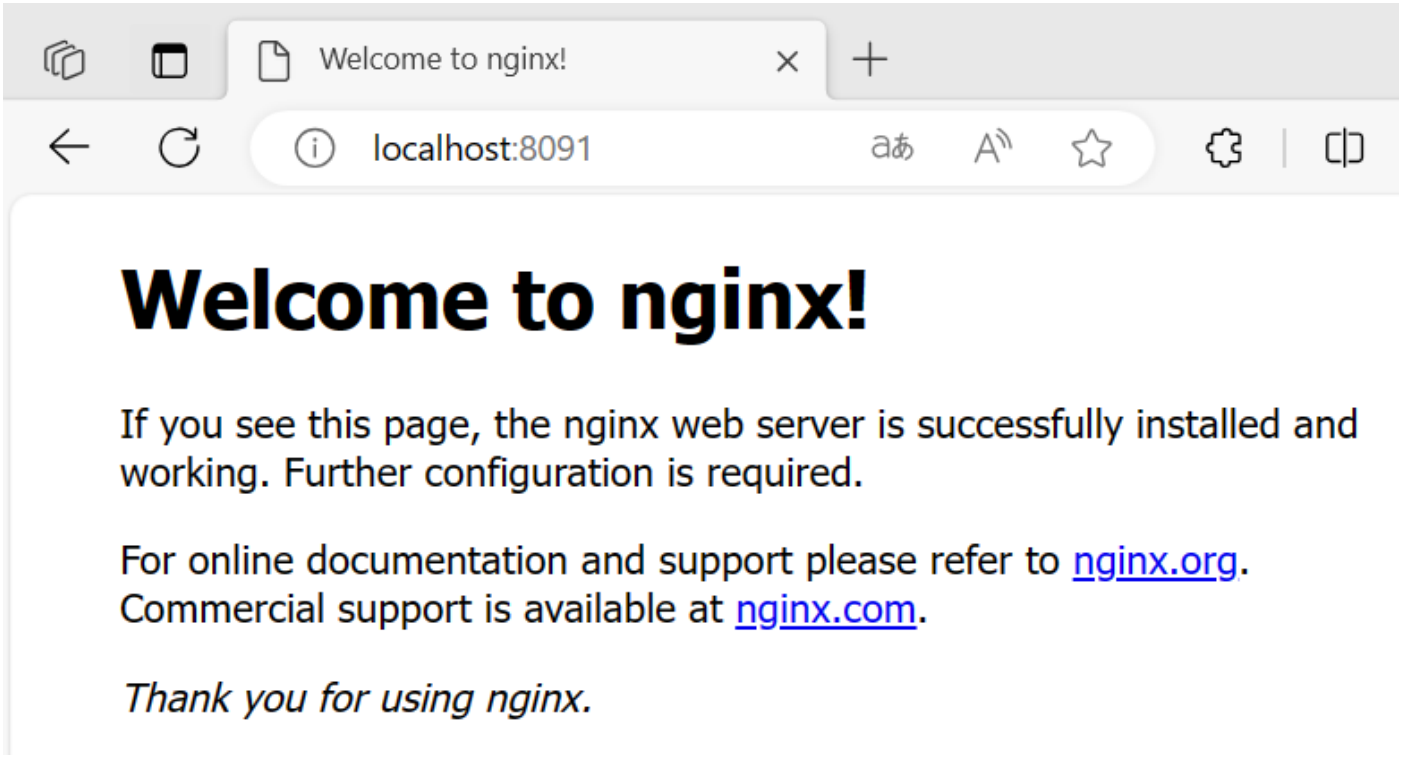
- Exploration du contenu du Conteneur :

```
[redacted]>
[redacted]>docker exec -it nginx8090 bash
root@74911f6f0596:/#
```

```
C:\Windows\System32\cmd.exe - docker  exec -it nginx8090 bash
root@74911f6f0596:/home# cd ..
root@74911f6f0596:/# ls
bin   dev                 docker-entrypoint.sh  home  lib32  libx32  mnt  proc  run   srv  tmp  var
boot  docker-entrypoint.d  etc                   lib   lib64  media   opt  root  sbin  sys  usr
root@74911f6f0596:/# cd home
root@74911f6f0596:/home#
```

- Création d'un deuxième Conteneur nginx sur le port 8091

```
C:\Windows\System32\cmd.exe
root@74911f6f0596:/home# cd ..
root@74911f6f0596:/# ls
bin   dev                 docker-entrypoint.sh  home  lib32  libx32  mnt  proc  run   srv  tmp  var
boot  docker-entrypoint.d  etc                   lib   lib64  media   opt  root  sbin  sys  usr
root@74911f6f0596:/# cd home
root@74911f6f0596:/home# exit
exit

[redacted]>docker run -d -p 8091:80 --name nginx8091 nginx
c8c54dcff4d227ba03668dc99df1038c04449052b43bc13f960f90fd8a14634f

[redacted]>
```



- Gestion des conteneurs Docker : liste,démarrage, arrêt, suppression.

*- Liste des conteneurs Docker en cours d'exécution :

```
[redacted]>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                    NAMES
c8c54dcff4d2   nginx     "/docker-entrypoint.…"   8 minutes ago    Up 8 minutes    0.0.0.0:8091->80/tcp     nginx8091
74911f6f0596   nginx     "/docker-entrypoint.…"   14 minutes ago   Up 14 minutes   0.0.0.0:8090->80/tcp     nginx8090
```

*- Arrêter un conteneur Docker :

```
>docker stop 74911f6f0596
74911f6f0596

>docker stop c8c54dcff4d2
c8c54dcff4d2

>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

>
```

*- Démarrer un conteneur Docker :

```
>docker start c8c54dcff4d2
c8c54dcff4d2

>docker start 74911f6f0596
74911f6f0596

>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS         PORTS                   NAMES
c8c54dcff4d2   nginx     "/docker-entrypoint.…"   13 minutes ago   Up 7 seconds   0.0.0.0:8091->80/tcp    nginx8091
74911f6f0596   nginx     "/docker-entrypoint.…"   19 minutes ago   Up 1 second    0.0.0.0:8090->80/tcp    nginx8090

>
```

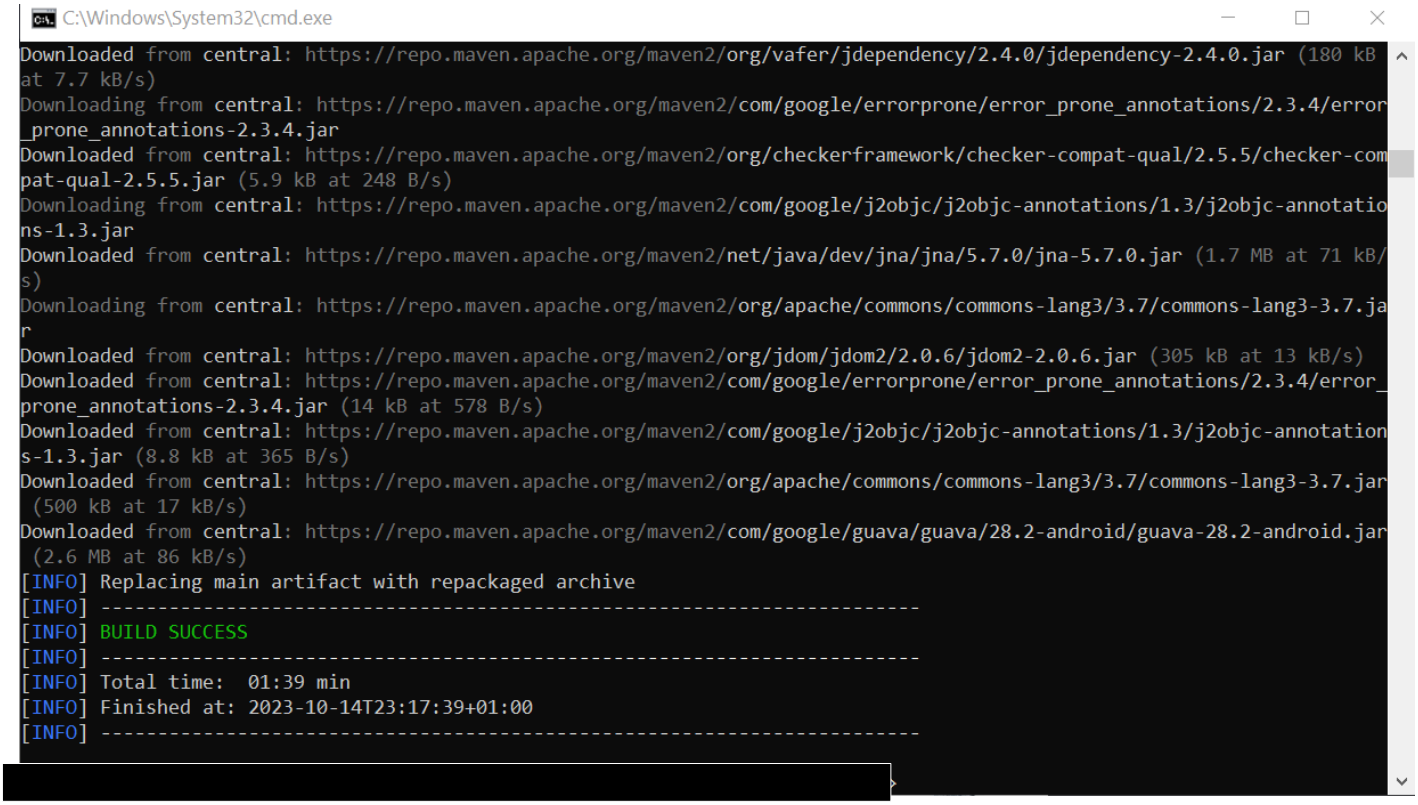
*- suppression d'un conteneur docker :

```
>docker stop c8c54dcff4d2
c8c54dcff4d2

>docker stop 74911f6f0596
74911f6f0596

>docker rm c8c54dcff4d2
c8c54dcff4d2

>docker rm 74911f6f0596
74911f6f0596

>
```

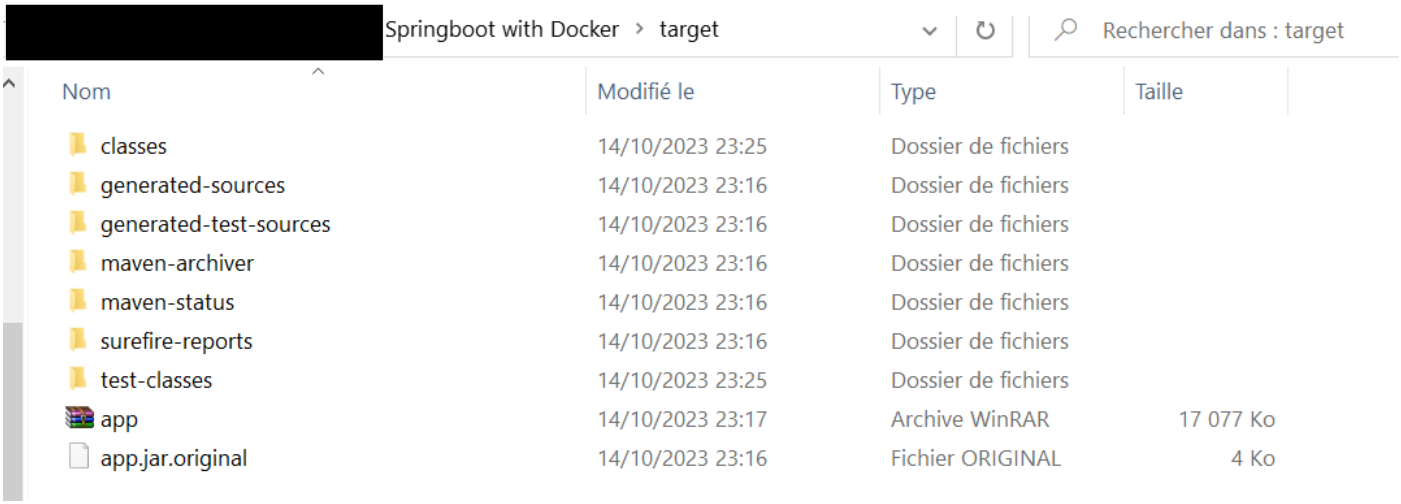
4. Exécution d'une application SpringBoot dans un conteneur Docker.

- Créer une Application SpringBoot avec un seule Contrôleur qui affiche le message «Bonjour »

Il faut faire « mvn clean package » pour faire le « build » de l'application.



- Ecrire le Dockerfile pour créer une image de cette application

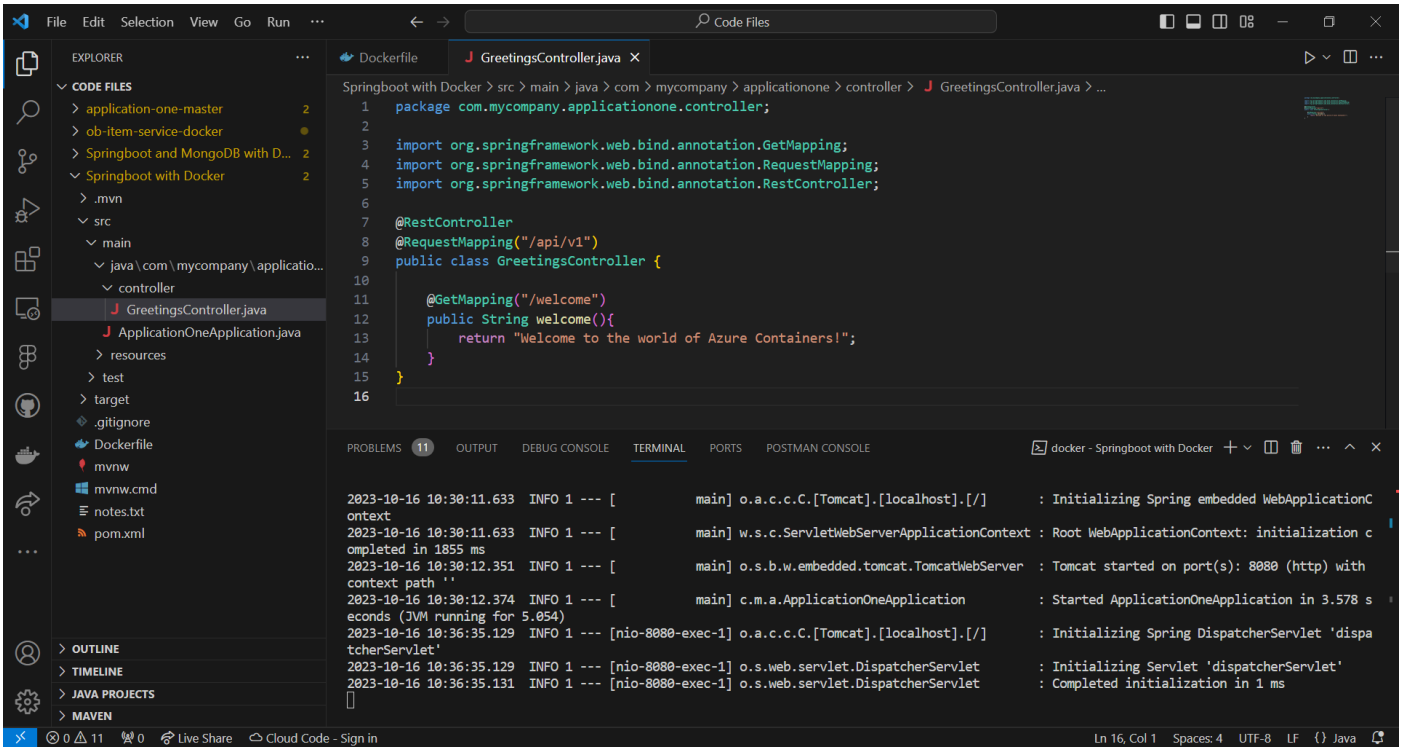


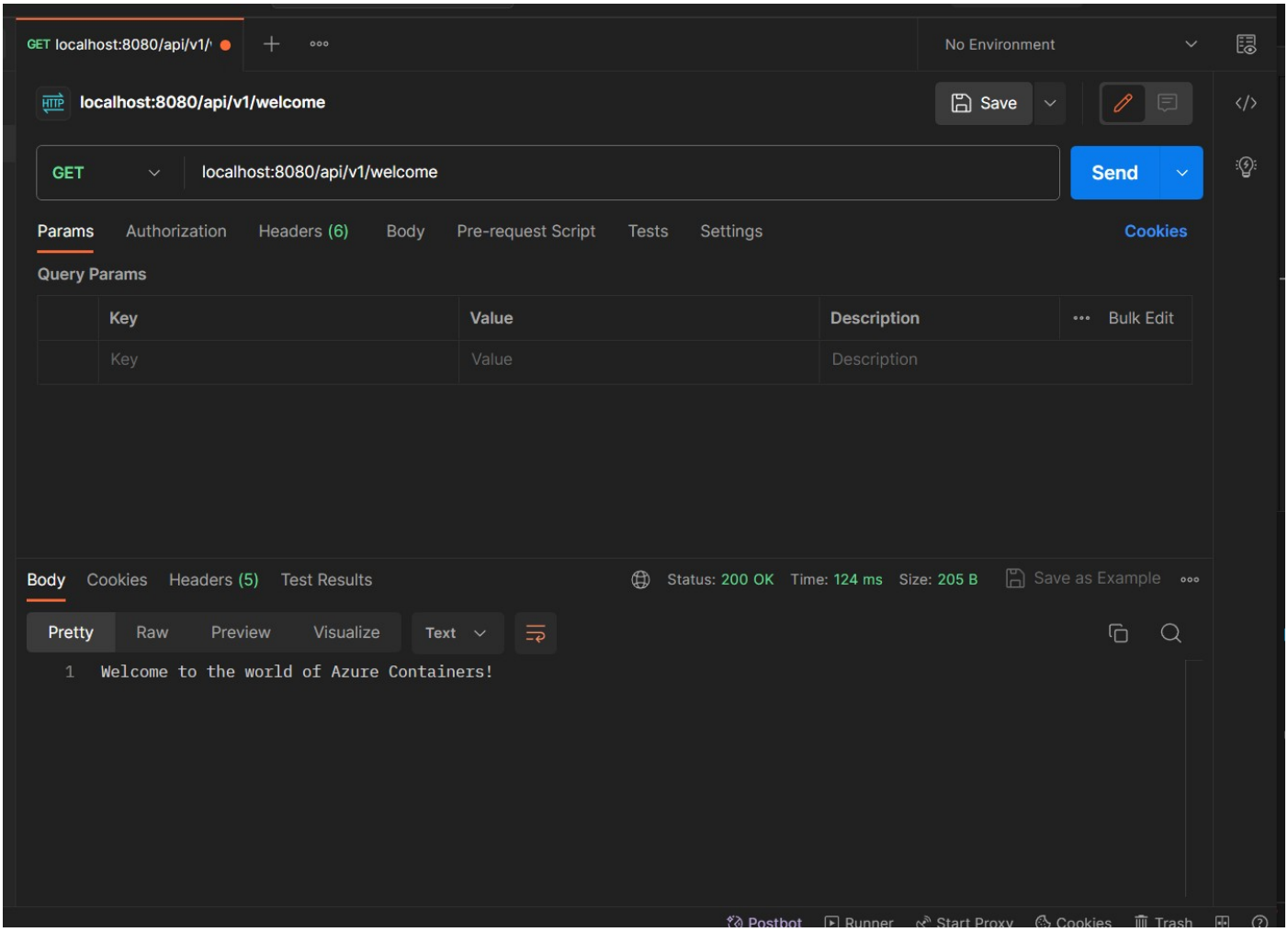
```
[redacted] docker build -t app .
[+] Building 0.7s (2/2) FINISHED                                                                    docker:default
 => [internal] load build definition from Dockerfile                                                         0.3s
 => => transferring dockerfile: 2B                                                                           0.1s
 => [internal] load .dockerignore                                                                            0.4s
 => => transferring context: 2B                                                                              0.0s
ERROR: failed to solve: failed to read dockerfile: open /var/lib/docker/tmp/buildkit-mount3139265740/Dockerfile: no such file or dir
ectory
[redacted] cd '.\Springboot with Docker\'
[redacted]r> docker build -t app .
2023/10/16 11:27:30 http2: server: error reading preface from client //./pipe/docker_engine: file has already been closed
[+] Building 6.6s (7/7) FINISHED                                                                    docker:default
 => [internal] load build definition from Dockerfile                                                         0.1s
 => => transferring dockerfile: 130B                                                                         0.0s
 => [internal] load .dockerignore                                                                            0.1s
 => => transferring context: 2B                                                                              0.0s
 => [internal] load metadata for docker.io/library/openjdk:11                                                5.9s
 => [internal] load build context                                                                            0.1s
 => => transferring context: 64B                                                                             0.0s
 => [1/2] FROM docker.io/library/openjdk:11@sha256:99bac5bf83633e3c7399aed725c8415e7b569b54e03e4599e580fc9cdb7c21ab  0.0s
 => CACHED [2/2] ADD target/app.jar app.jar                                                                  0.0s
 => exporting to image                                                                                       0.1s
 => => exporting layers                                                                                      0.0s
 => => writing image sha256:803cb4f31de28a1f41f3942866aab426245ffde29ef98d263784b3a5984e465d                 0.0s
 => => naming to docker.io/library/app                                                                       0.0s

What's Next?
```

- Exécuter cette image dans un Conteneur

```
PROBLEMS 11  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  POSTMAN CONSOLE        docker - Springboot with Docker

[redacted] docker run -p 8080:8080 app

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.5.5)

2023-10-16 10:30:09.653  INFO 1 --- [           main] c.m.a.ApplicationOneApplication          : Starting ApplicationOneApplication v0.0.1-SN
APSHOT using Java 11.0.16 on 014d043b30bf with PID 1 (/app.jar started by root in /)
2023-10-16 10:30:09.659  INFO 1 --- [           main] c.m.a.ApplicationOneApplication          : No active profile set, falling back to defau
lt profiles: default
2023-10-16 10:30:11.351  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
2023-10-16 10:30:11.379  INFO 1 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2023-10-16 10:30:11.380  INFO 1 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.
53]
2023-10-16 10:30:11.633  INFO 1 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationC
ontext
2023-10-16 10:30:11.633  INFO 1 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization c
ompleted in 1855 ms
2023-10-16 10:30:12.351  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with
context path ''
2023-10-16 10:30:12.374  INFO 1 --- [           main] c.m.a.ApplicationOneApplication          : Started ApplicationOneApplication in 3.578 s
econds (JVM running for 5.054)

Sign in                                                  Ln 15, Col 2  Spaces: 4  UTF-8  LF  {} Java
```

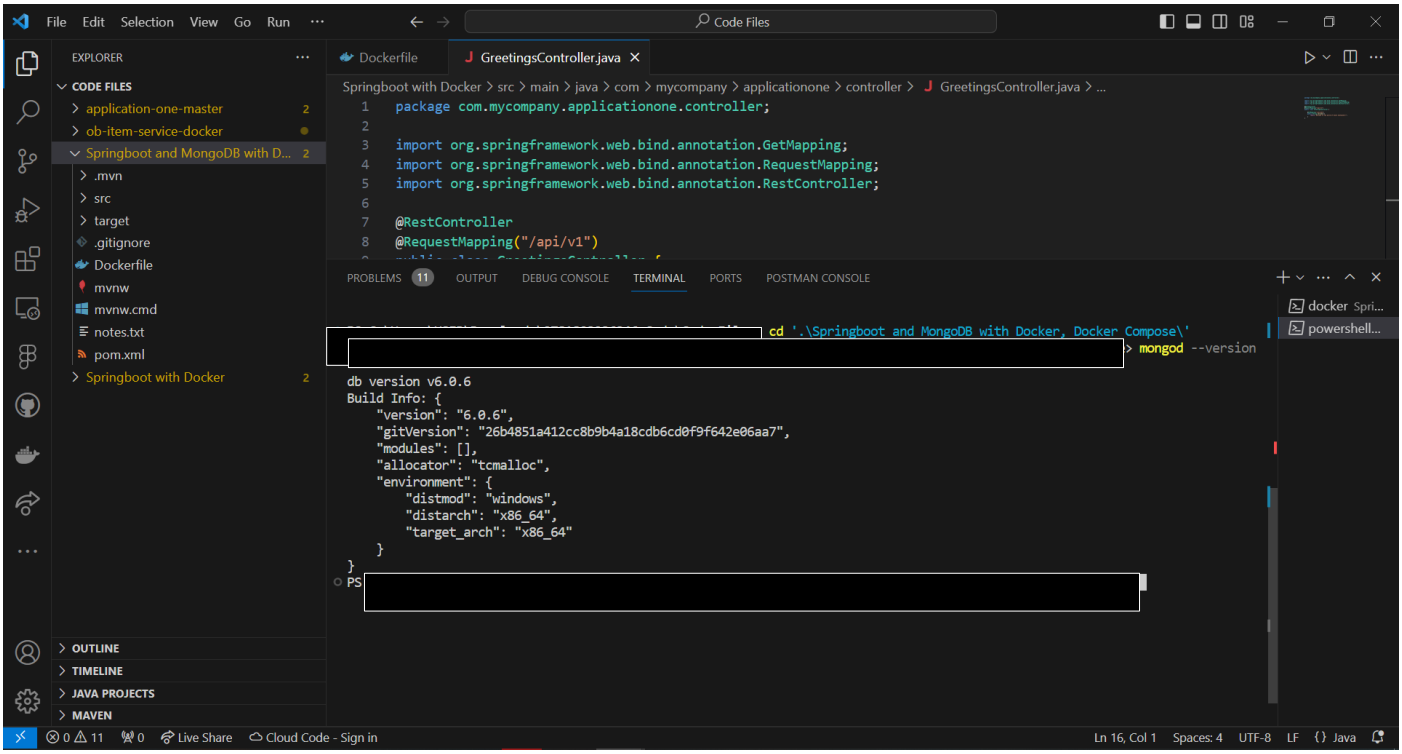
- Compréhension des couches d'image et de la réutilisabilité des images.





5. Compréhension de l'Application SpringBoot

- Installer MongoDB et Créer les Bases et Collections Nécessaires



- Tester l'Application SpringBoot Fournie et vérifier qu'elle accède à la Base Correctement en utilisant JUnit et Postman.

```
C:\Windows\System32\cmd.exe - mvn  spring-boot:run
[INFO] Attaching agents: []

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.4.9)

2023-10-16 11:45:11.543  INFO 18240 --- [           main] c.m.o.ObItemServiceApplication           : Starting ObItemServiceApplication using Java 15.0.1 on DESKTOP-JOKLOT4 with PID 18240 (C
:\Users\USER\Downloads\9781803236346_Code\Code Files\Springboot and MongoDB with Docker, Docker Compose\target\classes started by Lenovoo in C:\Users\USER\Downloads\9781803236346_Code\Code
Files\Springboot and MongoDB with Docker, Docker Compose)
2023-10-16 11:45:11.546  INFO 18240 --- [           main] c.m.o.ObItemServiceApplication           : The following profiles are active: local
2023-10-16 11:45:12.513  INFO 18240 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data MongoDB repositories in DEFAULT mode.
2023-10-16 11:45:12.597  INFO 18240 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 75 ms. Found 2 MongoDB repository interfaces
.
2023-10-16 11:45:14.066  INFO 18240 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8081 (http)
2023-10-16 11:45:14.125  INFO 18240 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2023-10-16 11:45:14.127  INFO 18240 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.50]
2023-10-16 11:45:14.409  INFO 18240 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2023-10-16 11:45:14.410  INFO 18240 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2764 ms
2023-10-16 11:45:14.916  INFO 18240 --- [           main] org.mongodb.driver.cluster               : Cluster created with settings {hosts=[host.docker.internal:28000], mode=SINGLE, required
ClusterType=UNKNOWN, serverSelectionTimeout='30000 ms'}
2023-10-16 11:45:16.339  INFO 18240 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8081 (http) with context path ''
2023-10-16 11:45:16.360  INFO 18240 --- [           main] c.m.o.ObItemServiceApplication           : Started ObItemServiceApplication in 6.046 seconds (JVM running for 6.796)
2023-10-16 11:45:27.871  INFO 18240 --- [.internal:28000] org.mongodb.driver.cluster               : Exception in monitor thread while connecting to server host.docker.internal:28000

com.mongodb.MongoSocketOpenException: Exception opening socket
        at com.mongodb.internal.connection.SocketStream.open(SocketStream.java:70) ~[mongodb-driver-core-4.1.2.jar:na]
        at com.mongodb.internal.connection.InternalStreamConnection.open(InternalStreamConnection.java:143) ~[mongodb-driver-core-4.1.2.jar:na]
        at com.mongodb.internal.connection.DefaultServerMonitor$ServerMonitorRunnable.lookupServerDescription(DefaultServerMonitor.java:188) ~[mongodb-driver-core-4.1.2.jar:na]
        at com.mongodb.internal.connection.DefaultServerMonitor$ServerMonitorRunnable.run(DefaultServerMonitor.java:144) ~[mongodb-driver-core-4.1.2.jar:na]
        at java.base/java.lang.Thread.run(Thread.java:832) ~[na:na]
Caused by: java.net.SocketTimeoutException: Connect timed out
        at java.base/sun.nio.ch.NioSocketImpl.timedFinishConnect(NioSocketImpl.java:546) ~[na:na]
        at java.base/sun.nio.ch.NioSocketImpl.connect(NioSocketImpl.java:597) ~[na:na]
        at java.base/java.net.SocksSocketImpl.connect(SocksSocketImpl.java:333) ~[na:na]
        at java.base/java.net.Socket.connect(Socket.java:648) ~[na:na]
        at com.mongodb.internal.connection.SocketStreamHelper.initialize(SocketStreamHelper.java:78) ~[mongodb-driver-core-4.1.2.jar:na]
        at com.mongodb.internal.connection.SocketStream.initializeSocket(SocketStream.java:79) ~[mongodb-driver-core-4.1.2.jar:na]
        at com.mongodb.internal.connection.SocketStream.open(SocketStream.java:65) ~[mongodb-driver-core-4.1.2.jar:na]
        ... 4 common frames omitted
```

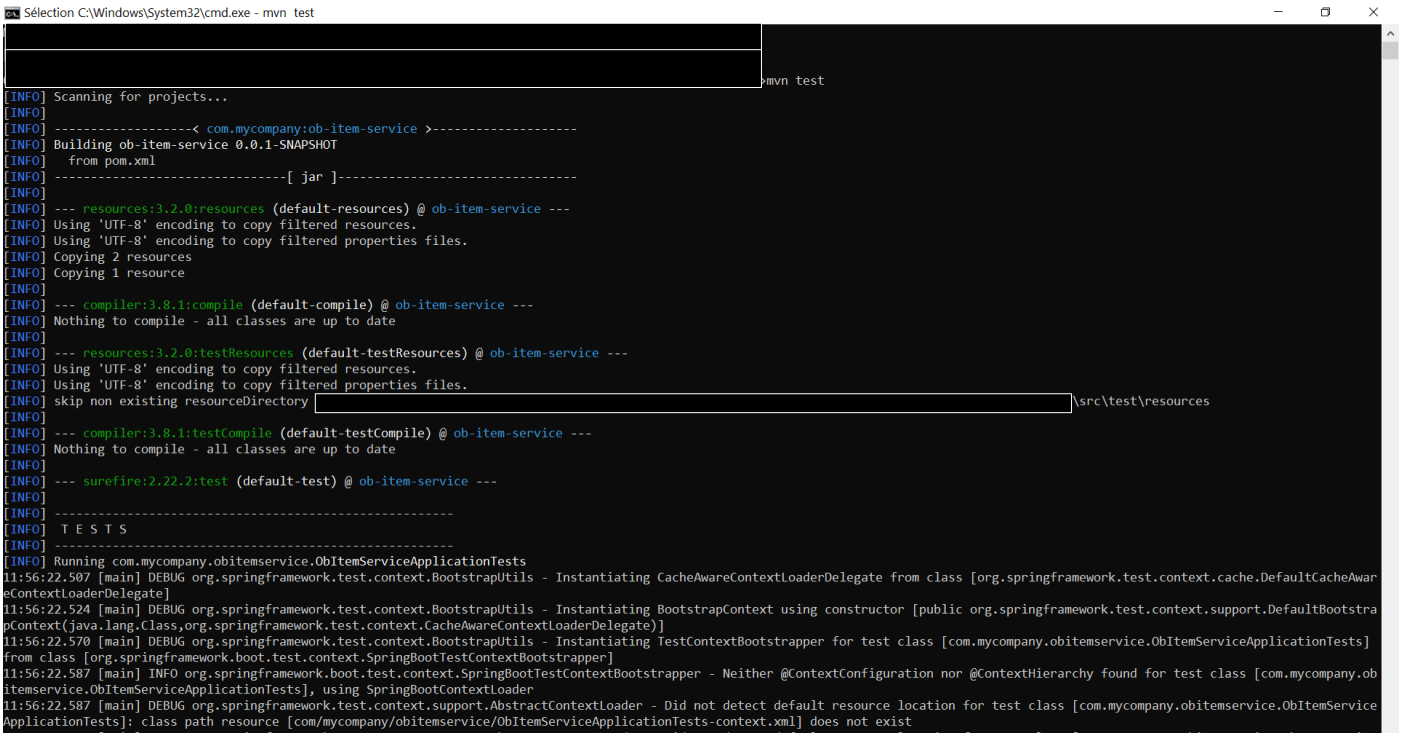
En utilisant JUnit :

Ceci est la classe JUnit :

Dockerfile | GreetingsController.java | *ObItemServiceApplicationTests.java* ×

Springboot and MongoDB with Docker, Docker Compose > src > test > java > com > mycompany > obitemservice > ObItemServiceApplicationTests.java > ...

```java
package com.mycompany.obitemservice;

import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest
class ObItemServiceApplicationTests {

    @Test
    void contextLoads() {
    }

}
```



7. Docker Compose

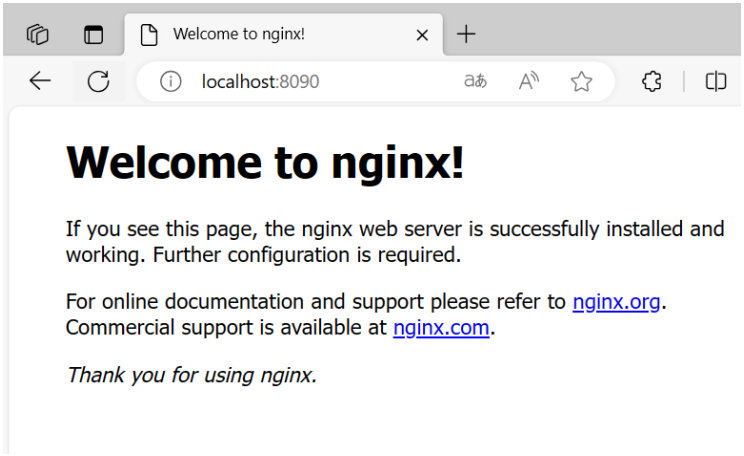
Running the docker compose file :

```yaml
version: "3"
services:
  mongodb-container-one:
    image: mongo:latest
    container_name: "mongodb-container-one"
    ports:
      - 28000:27017
  springboot-with-mongodb-container-one:
    image: ranjan715/springboot-with-mongodb:v11
    container_name: springboot-with-mongodb-container-one
    ports:
      - 8090:8081
    links:
      - mongodb-container-one
```

```
C:\Windows\System32\cmd.exe - docker-compose  up -d
[redacted]>docker-compose up -d
[+] Running 4/19
 - springboot-with-mongodb-container-one 8 layers [⠿⠿⠿⠿⠿⠿⠿⠿] 97.11MB/312.6MB Pulling                                    20.6s
   - 955615a668ce Extracting [=====>                                            ]  6.128MB/54.93MB                          17.9s
   ⠿ 2756ef5f69a5 Download complete                                                                                          2.7s
   ⠿ 911ea9f2bd51 Download complete                                                                                         11.0s
   - 27b0a22ee906 Downloading [==========================================>       ]  49.49MB/54.57MB                          17.9s
   ⠿ 785dffb36c6c Download complete                                                                                         11.0s
   ⠿ 3fccb14f0369 Download complete                                                                                         11.0s
   - 4a0c30fedd9c Downloading [==========>                                       ]  41.49MB/203.1MB                          17.9s
   - d73d22fb6202 Waiting                                                                                                   17.9s
 - mongodb-container-one 9 layers [⠿⠿⠿⠿⠿⠿⠿⠿⠿]      0B/0B      Pulling                                                        20.6s
   - 43f89b94cd7d Waiting                                                                                                   17.7s
   - 54a7480baa9d Waiting                                                                                                   17.7s
   - 7f9301fbd7df Waiting                                                                                                   17.7s
   - 5e4470f2e90f Waiting                                                                                                   17.7s
   - 40d046ff8fd3 Waiting                                                                                                   17.7s
   - e062d62b861e Waiting                                                                                                   17.7s
   - 72919e34fde8 Waiting                                                                                                   17.7s
   - ab22810dfc64 Waiting                                                                                                   17.7s
   - fb05c29fbdf5 Waiting                                                                                                   17.7s
```

The container has already started :

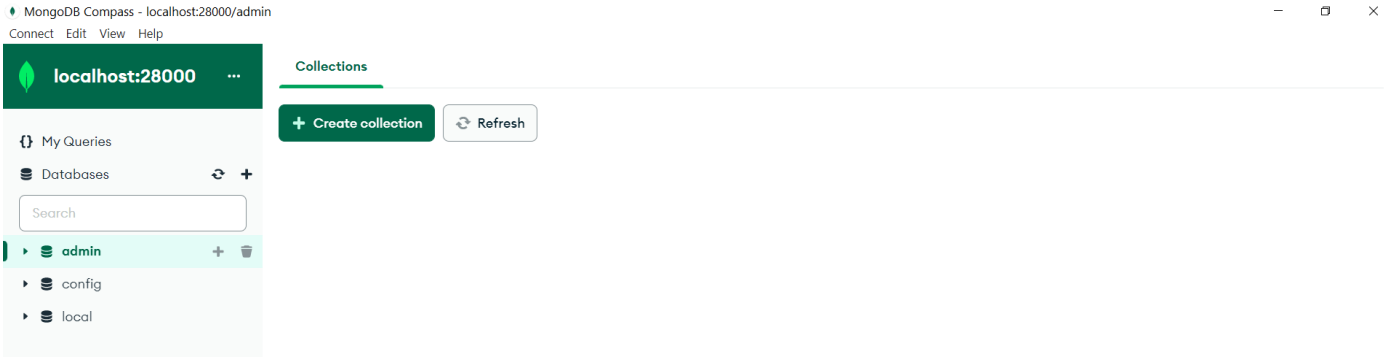
```
C:\Windows\System32\cmd.exe
[redacted]>docker-compose up -d
[+] Running 19/19
 ⠿ springboot-with-mongodb-container-one 8 layers [⠿⠿⠿⠿⠿⠿⠿⠿]      0B/0B      Pulled                                          60.9s
   ⠿ 955615a668ce Pull complete                                                                                              21.2s
   ⠿ 2756ef5f69a5 Pull complete                                                                                              21.7s
   ⠿ 911ea9f2bd51 Pull complete                                                                                              22.4s
   ⠿ 27b0a22ee906 Pull complete                                                                                              27.9s
   ⠿ 785dffb36c6c Pull complete                                                                                              28.5s
   ⠿ 3fccb14f0369 Pull complete                                                                                              28.6s
   ⠿ 4a0c30fedd9c Pull complete                                                                                              57.7s
   ⠿ d73d22fb6202 Pull complete                                                                                              58.1s
 ⠿ mongodb-container-one 9 layers [⠿⠿⠿⠿⠿⠿⠿⠿⠿]      0B/0B      Pulled                                                          84.4s
   ⠿ 43f89b94cd7d Pull complete                                                                                              31.4s
   ⠿ 54a7480baa9d Pull complete                                                                                              31.5s
   ⠿ 7f9301fbd7df Pull complete                                                                                              32.2s
   ⠿ 5e4470f2e90f Pull complete                                                                                              32.3s
   ⠿ 40d046ff8fd3 Pull complete                                                                                              32.4s
   ⠿ e062d62b861e Pull complete                                                                                              32.5s
   ⠿ 72919e34fde8 Pull complete                                                                                              32.6s
   ⠿ ab22810dfc64 Pull complete                                                                                              81.3s
   ⠿ fb05c29fbdf5 Pull complete                                                                                              81.4s
[+] Running 3/3
 ⠿ Network resources_default                     Created                                                                     0.3s
 ⠿ Container mongodb-container-one               Started                                                                     4.5s
 ⠿ Container springboot-with-mongodb-container-one  Started                                                                 5.0s
[redacted]>
```



⇨ The application works

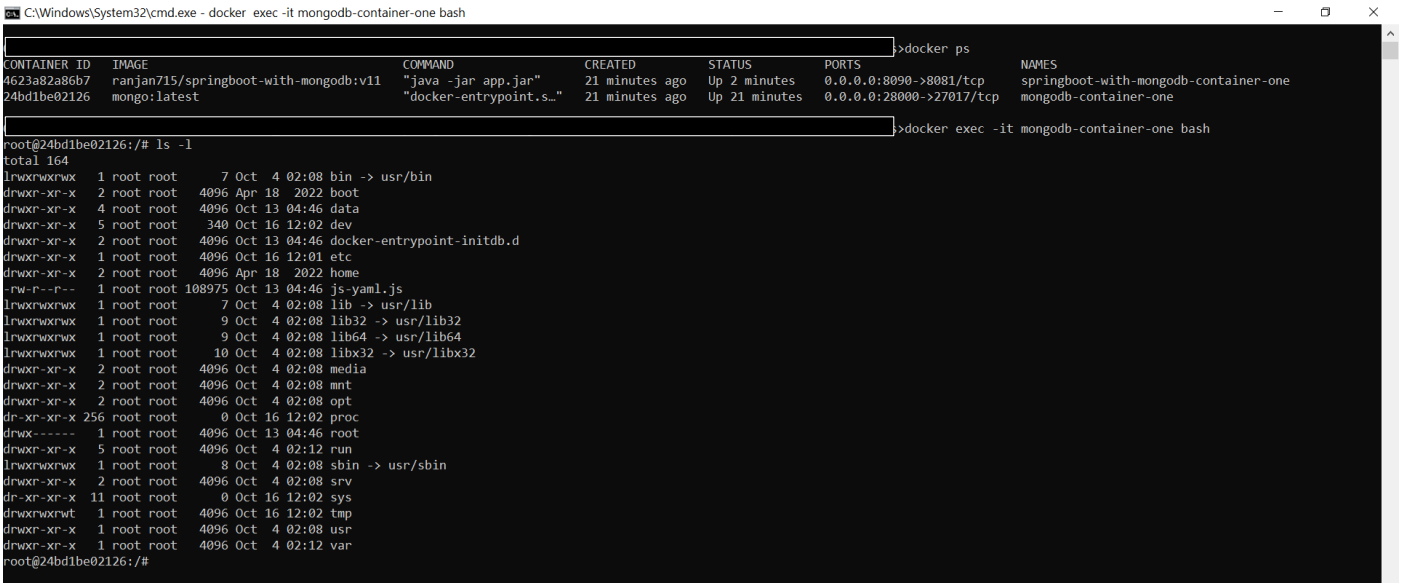
```
es>docker ps
CONTAINER ID   IMAGE                                  COMMAND                  CREATED          STATUS          PORTS                     NAMES
574eaf631307   ranjan715/springboot-with-mongodb:v11  "java -jar app.jar"      3 minutes ago    Up 3 minutes    0.0.0.0:8090->8081/tcp    springboot-with-mongodb-container-one
c78f4b127ed9   mongo:latest                           "docker-entrypoint.s…"   3 minutes ago    Up 3 minutes    0.0.0.0:28000->27017/tcp  mongodb-container-one
014d043b30bf   app                                    "java -jar app.jar"      43 minutes ago   Up 43 minutes   0.0.0.0:8080->8080/tcp    clever_dewdney
es>
```

Connected on localhost :28000 with mongodb



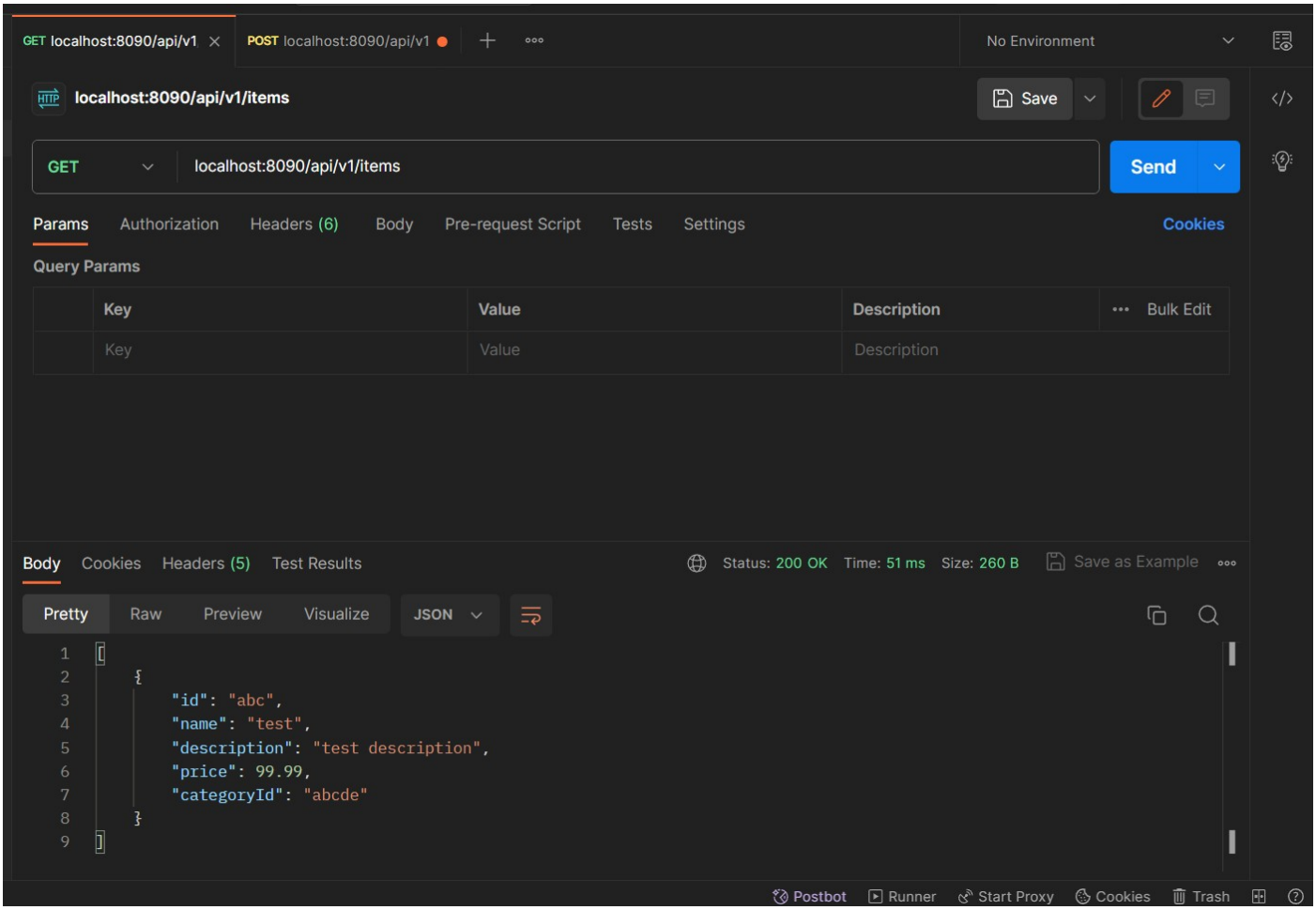
⇨ It communicates very well

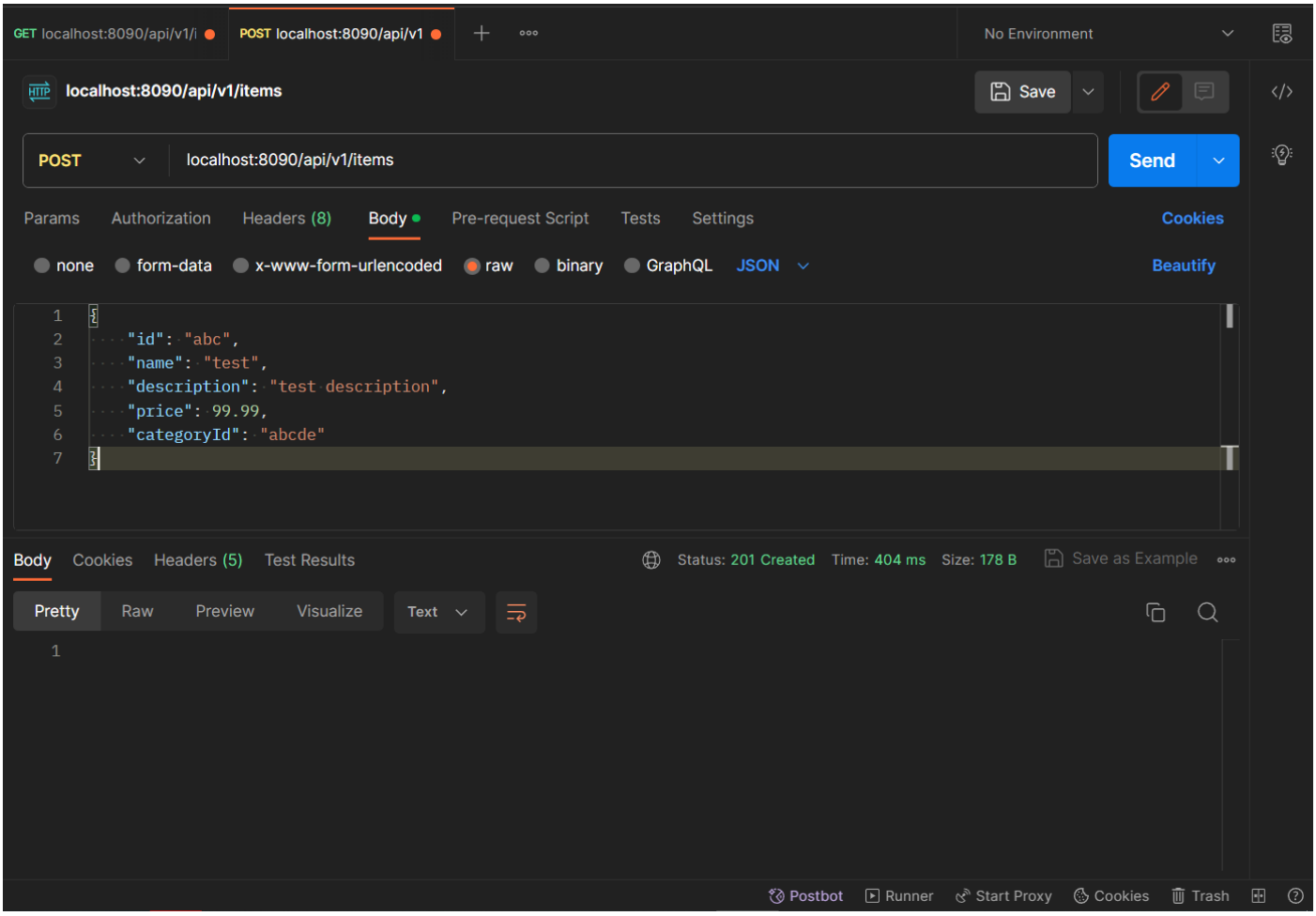
- Explorer le contenu des deux containers pour vérifier l'absence d'exceptions



```
C:\Users\USER\Downloads\9781803236346_Code\Code Files\Springboot and MongoDB with Docker, Docker Compose\src\main\resources>docker exec -it springboot-with-mongodb-container-one bash
root@4623a82a86b7:/# ls -l
total 22344
-rwxr-xr-x   1 root root 22809867 Sep 24  2021 app.jar
drwxr-xr-x   1 root root     4096 Sep  3  2021 bin
drwxr-xr-x   2 root root     4096 Apr 10  2021 boot
drwxr-xr-x   5 root root      340 Oct 16 12:21 dev
drwxr-xr-x   1 root root     4096 Oct 16 12:01 etc
drwxr-xr-x   2 root root     4096 Apr 10  2021 home
drwxr-xr-x   1 root root     4096 Sep  2  2021 lib
drwxr-xr-x   2 root root     4096 Sep  2  2021 lib64
drwxr-xr-x   2 root root     4096 Sep  2  2021 media
drwxr-xr-x   2 root root     4096 Sep  2  2021 mnt
drwxr-xr-x   2 root root     4096 Sep  2  2021 opt
dr-xr-xr-x 257 root root        0 Oct 16 12:21 proc
drwx------   1 root root     4096 Sep  3  2021 root
drwxr-xr-x   3 root root     4096 Sep  2  2021 run
drwxr-xr-x   1 root root     4096 Sep  3  2021 sbin
drwxr-xr-x   2 root root     4096 Sep  2  2021 srv
dr-xr-xr-x  11 root root        0 Oct 16 12:21 sys
drwxrwxrwt   1 root root     4096 Oct 16 12:21 tmp
drwxr-xr-x   1 root root     4096 Sep  2  2021 usr
drwxr-xr-x   1 root root     4096 Sep  2  2021 var
root@4623a82a86b7:/#
```

Faites communiquer les deux containers et vérifier le bon fonctionnement de l'application avec Postman





- Quelle est l'utilité de DockerCompose ?

L'utilité de DockerCompose est qu'il est un fichier de configuration YAML utilisé pour définir les services d'une application avec une architecture organisée de façon qu'elle peut être construite en utilisant une seule commande pour la création et le lancement des services.

8. Intégration de Docker dans un environnement DevOps

- Etudier et Expliquer rôle de Docker dans un flux de développement et de déploiement continu.

Docker permet de « partager » des applications tout développées sous formes d'images qui peuvent être construites en des containers, en exécutant ces containers, on est sur que l'environnement nécessaire pour que l'application fonctionne correctement existe.

9. Création de premiers Pipeline avec Jenkins

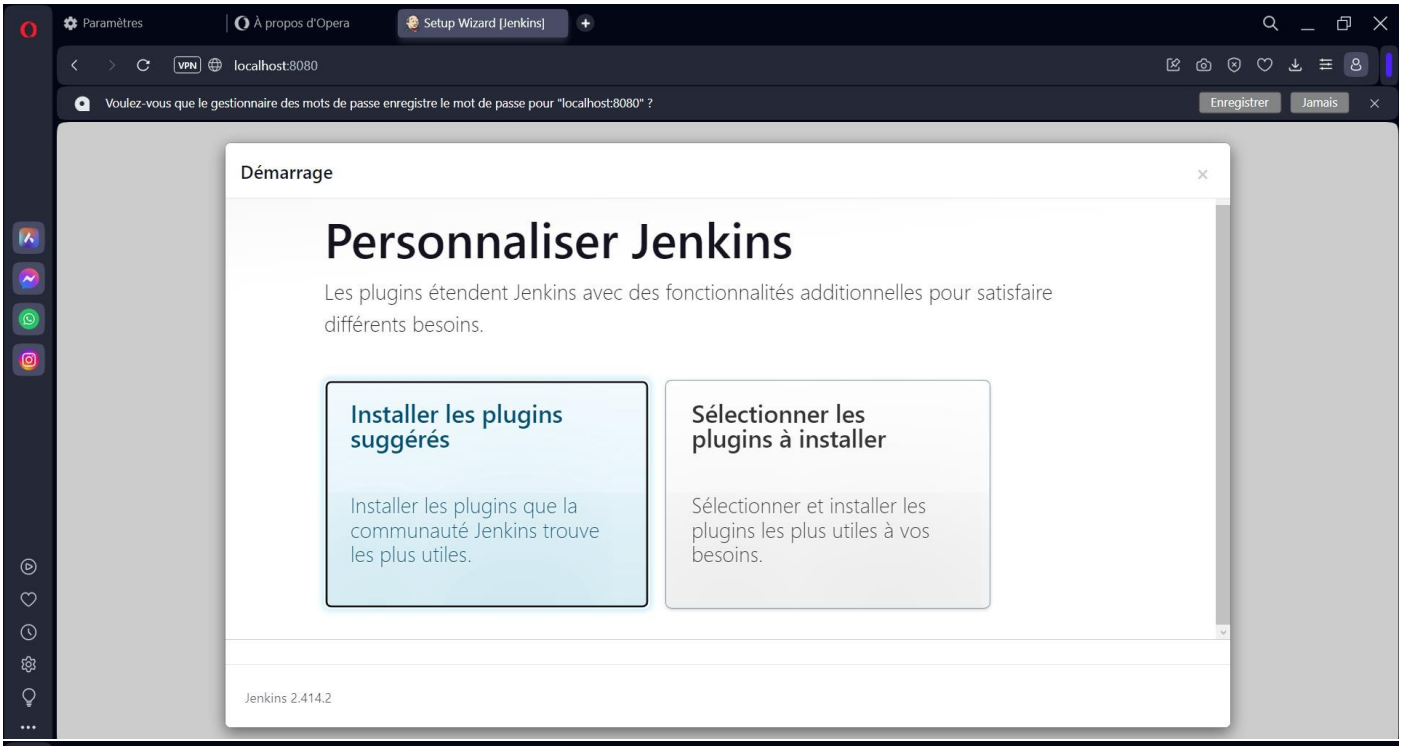
- Installation de Jenkins

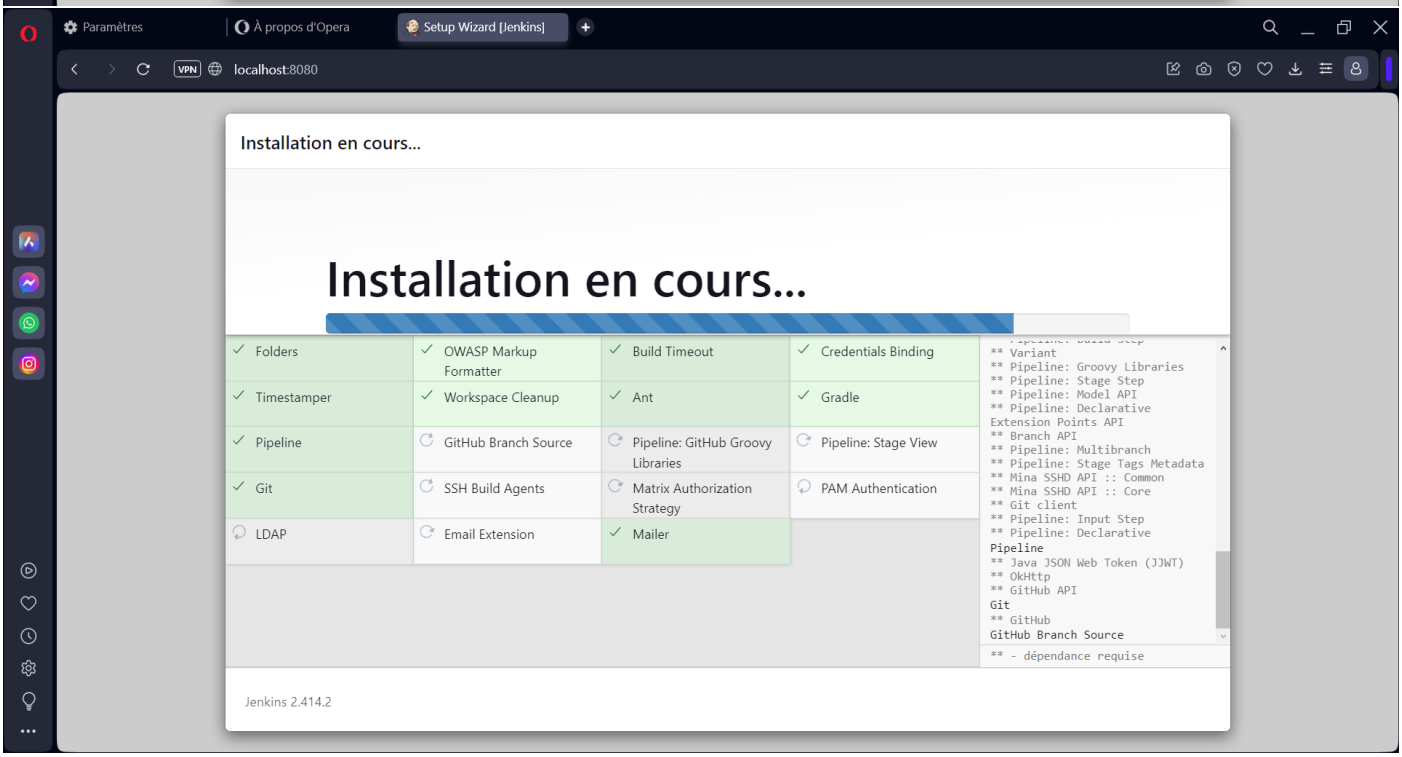
C:\Windows\System32\cmd.exe - docker build -t jenkins .

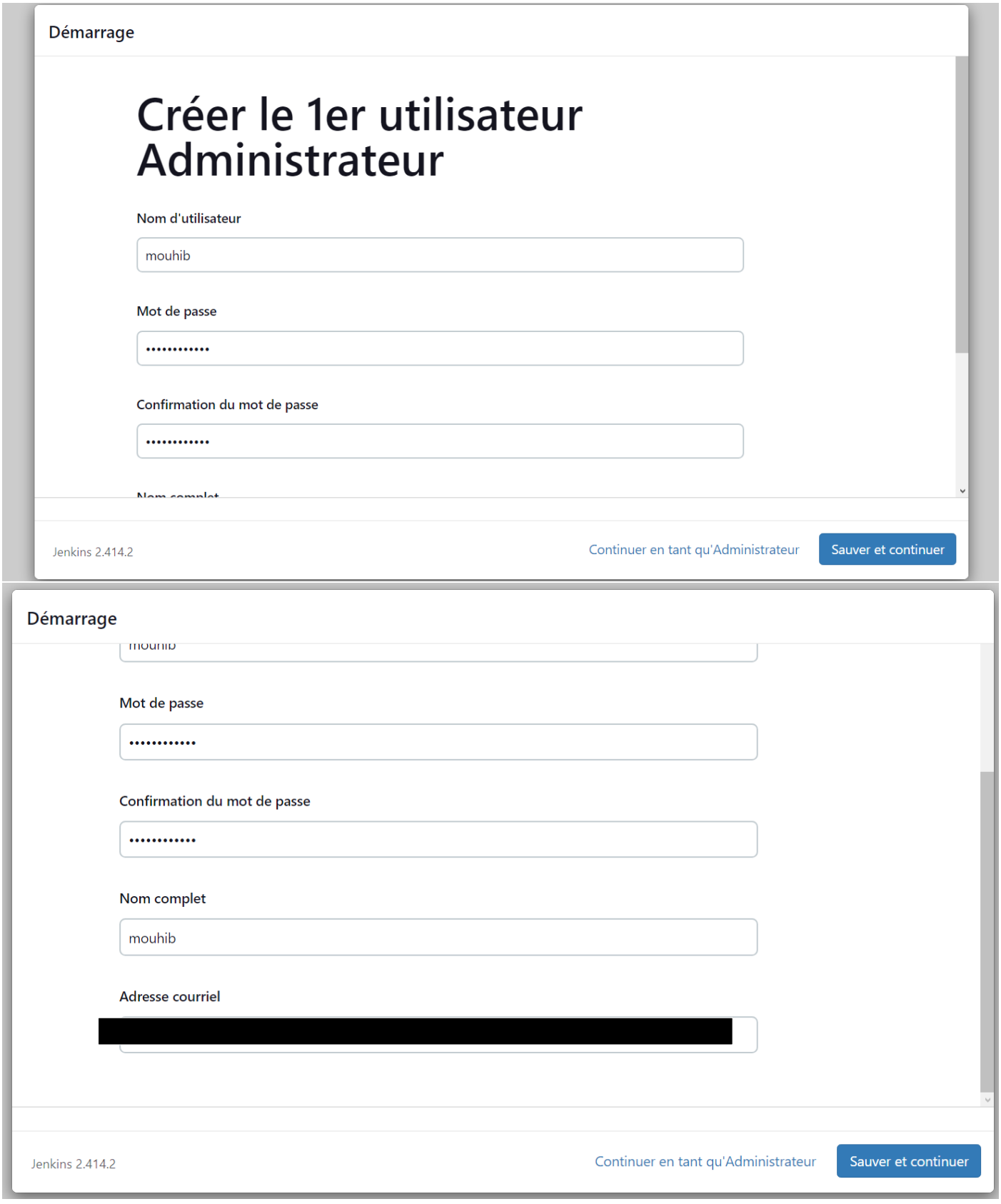
```
[+] Building 122.2s (4/9)                                                                                                          docker:default
 => [internal] load .dockerignore                                                                                                             0.3s
 => => transferring context: 2B                                                                                                               0.0s
 => [internal] load build definition from Dockerfile                                                                                          0.2s
 => => transferring dockerfile: 505B                                                                                                          0.0s
 => [internal] load metadata for docker.io/jenkins/jenkins:lts                                                                                2.4s
 => [1/6] FROM docker.io/jenkins/jenkins:lts@sha256:b705323eaf70a7da4c1eed8b816f33dff2d5c8c3671170a2c17cf77aa4f15432                        41.9s
 => => resolve docker.io/jenkins/jenkins:lts@sha256:b705323eaf70a7da4c1eed8b816f33dff2d5c8c3671170a2c17cf77aa4f15432                         0.1s
 => => sha256:2a15645303916751e9fd781fd403051199d94052d5d18be6e1b2398aab1f79c1 2.77kB / 2.77kB                                               0.0s
 => => sha256:3d0d48f61941fc7cb5253b07efd932d06ee6e335445f6f535ac3d1d49e385a7e 12.62kB / 12.62kB                                             0.0s
 => => sha256:b705323eaf70a7da4c1eed8b816f33dff2d5c8c3671170a2c17cf77aa4f15432 3.12kB / 3.12kB                                               0.0s
 => => sha256:5e0093fb8d5f36e86b93870a544e50bf21f3968e872f91ad0a2bc9abdc672472 8.68MB / 8.68MB                                               1.2s
 => => sha256:012c0b3e998c1a0c0bedcf712eaaafb188580529dd026a04aa1ce13fdb39e42b 49.56MB / 49.56MB                                            19.3s
 => => sha256:f6154b0007a4a7dcee550de8b63ddbdf177961a972a33c54b2c4390dc4a873bf 61.32MB / 61.32MB                                            23.4s
 => => sha256:34d3735955786e60f8bb8e7bd6fa2dfd7739dc047ba49c637c18a5d54b99d23c 1.23kB / 1.23kB                                               1.5s
 => => sha256:10cad8b7dbf876c2afb6402c97060b77d723bd6838daef8c2fadab362864f768 183B / 183B                                                   1.7s
 => => sha256:2471c7529ac37af126ea6c1afad134f08afa5c684f7e10816e314d286ce7dc38 89.33MB / 89.33MB                                            26.0s
 => => sha256:655eafc427231f280d11bcd10dfa0887a3165d2eb9345ab44ed75063bed17ae7 189B / 189B                                                  20.3s
 => => extracting sha256:012c0b3e998c1a0c0bedcf712eaaafb188580529dd026a04aa1ce13fdb39e42b                                                     3.8s
 => => sha256:3145851d1a67db7eb3dd0d7f6734a26340476b9d1f6db58668088d72b4833847 6.09MB / 6.09MB                                              23.5s
 => => sha256:614d8a8c6a95f7a9ab30388cf16a55f34eb54e0bcc52d4c3ef5033ac46200cd8 77.15MB / 77.15MB                                            38.9s
 => => sha256:95f5dcd4e08b1d468bac93b830b9aa4d9c0f302d5be66c478dc70fb48fb6bf62 1.93kB / 1.93kB                                              23.8s
 => => extracting sha256:f6154b0007a4a7dcee550de8b63ddbdf177961a972a33c54b2c4390dc4a873bf                                                     3.3s
 => => sha256:7c8f869ed080beb2a95df25c497c699a86746c448fbb3a6f29bfd42313ec2a49 1.16kB / 1.16kB                                              24.4s
 => => sha256:3a7be85a26d2887982180264f604ffb596884aa270ed5a80eed3b30711cfd44c 392B / 392B                                                  24.6s
 => => sha256:d754f693a7c2efe59a3b9340ff01604beb630ca1094f9024926d915957cd7fcd 265B / 265B                                                  24.8s
 => => extracting sha256:5e0093fb8d5f36e86b93870a544e50bf21f3968e872f91ad0a2bc9abdc672472                                                     0.1s
 => => extracting sha256:34d3735955786e60f8bb8e7bd6fa2dfd7739dc047ba49c637c18a5d54b99d23c                                                     0.0s
 => => extracting sha256:10cad8b7dbf876c2afb6402c97060b77d723bd6838daef8c2fadab362864f768                                                     0.0s
 => => extracting sha256:2471c7529ac37af126ea6c1afad134f08afa5c684f7e10816e314d286ce7dc38                                                     0.8s
 => => extracting sha256:655eafc427231f280d11bcd10dfa0887a3165d2eb9345ab44ed75063bed17ae7                                                     0.0s
 => => extracting sha256:3145851d1a67db7eb3dd0d7f6734a26340476b9d1f6db58668088d72b4833847                                                     0.1s
 => => extracting sha256:614d8a8c6a95f7a9ab30388cf16a55f34eb54e0bcc52d4c3ef5033ac46200cd8                                                     0.7s
 => => extracting sha256:95f5dcd4e08b1d468bac93b830b9aa4d9c0f302d5be66c478dc70fb48fb6bf62                                                     0.0s
 => => extracting sha256:7c8f869ed080beb2a95df25c497c699a86746c448fbb3a6f29bfd42313ec2a49                                                     0.0s
 => => extracting sha256:3a7be85a26d2887982180264f604ffb596884aa270ed5a80eed3b30711cfd44c                                                     0.0s
 => => extracting sha256:d754f693a7c2efe59a3b9340ff01604beb630ca1094f9024926d915957cd7fcd                                                     0.0s
 => [2/6] RUN apt-get update -qq     && apt-get install -qqy apt-transport-https ca-certificates curl gnupg2 software-properties-common     77.5s
 => => # Unpacking libxml2:amd64 (2.9.14+dfsg-1.3~deb12u1) ...
 => => # Selecting previously unselected package shared-mime-info.
 => => # Preparing to unpack .../08-shared-mime-info_2.2-1_amd64.deb ...
 => => # Unpacking shared-mime-info (2.2-1) ...
 => => # Preparing to unpack .../09-libcurl3-gnutls_7.88.1-10+deb12u4_amd64.deb ...
 => => # Unpacking libcurl3-gnutls:amd64 (7.88.1-10+deb12u4) over (7.88.1-10+deb12u1) ...
```

Sélection C:\Windows\System32\cmd.exe - docker run -it -p 8080:8080 -p 50000:50000 -v /var/run/docker.sock:/var/run/docker.sock -v jenkins_home:/var/jenkins_home jenkins

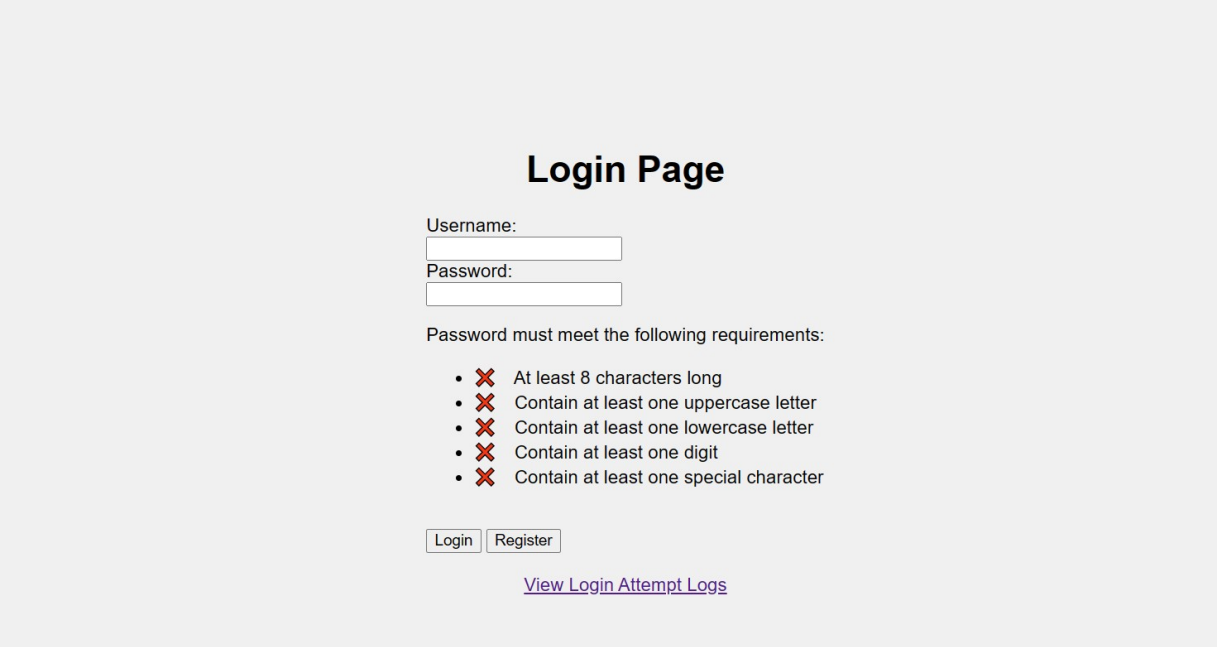
```
2023-10-16 12:54:20.402+0000 [id=1]     INFO    org.eclipse.jetty.server.Server#doStart: jetty-10.0.15; built: 2023-04-11T17:25:14.480Z; git: 68017dbd00236bb7e187330d7585a059610f661d; jvm 1
1.0.20.1+1
2023-10-16 12:54:20.680+0000 [id=1]     INFO    o.e.j.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /, did not find org.eclipse.jetty.jsp.JettyJspServlet
2023-10-16 12:54:20.735+0000 [id=1]     INFO    o.e.j.s.s.DefaultSessionIdManager#doStart: Session workerName=node0
2023-10-16 12:54:21.222+0000 [id=1]     INFO    hudson.WebAppMain#contextInitialized: Jenkins home directory: /var/jenkins_home found at: EnvVars.masterEnvVars.get("JENKINS_HOME")
2023-10-16 12:54:21.380+0000 [id=1]     INFO    o.e.j.s.handler.ContextHandler#doStart: Started w.@28cb9120{Jenkins v2.414.2,/,file:///var/jenkins_home/war/,AVAILABLE}{/var/jenkins_home/war
}
2023-10-16 12:54:21.419+0000 [id=1]     INFO    o.e.j.server.AbstractConnector#doStart: Started ServerConnector@37883b97{HTTP/1.1, (http/1.1)}{0.0.0.0:8080}
2023-10-16 12:54:21.439+0000 [id=1]     INFO    org.eclipse.jetty.server.Server#doStart: Started Server@53aad5d5{STARTING}[10.0.15,sto=0] @1888ms
2023-10-16 12:54:21.443+0000 [id=25]    INFO    winstone.Logger#logInternal: Winstone Servlet Engine running: controlPort=disabled
2023-10-16 12:54:21.662+0000 [id=32]    INFO    jenkins.InitReactorRunner$1#onAttained: Started initialization
2023-10-16 12:54:21.720+0000 [id=40]    INFO    jenkins.InitReactorRunner$1#onAttained: Listed all plugins
2023-10-16 12:54:22.462+0000 [id=40]    INFO    jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
2023-10-16 12:54:22.470+0000 [id=39]    INFO    jenkins.InitReactorRunner$1#onAttained: Started all plugins
2023-10-16 12:54:22.477+0000 [id=30]    INFO    jenkins.InitReactorRunner$1#onAttained: Augmented all extensions
2023-10-16 12:54:22.738+0000 [id=30]    INFO    jenkins.InitReactorRunner$1#onAttained: System config loaded
2023-10-16 12:54:22.739+0000 [id=33]    INFO    jenkins.InitReactorRunner$1#onAttained: System config adapted
2023-10-16 12:54:22.740+0000 [id=33]    INFO    jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2023-10-16 12:54:22.742+0000 [id=33]    INFO    jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
2023-10-16 12:54:22.808+0000 [id=33]    INFO    jenkins.install.SetupWizard#init:

***************************************************************
***************************************************************
***************************************************************

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

[redacted]

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

***************************************************************
***************************************************************
***************************************************************

WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.codehaus.groovy.vmplugin.v7.Java7$1 (file:/var/jenkins_home/war/WEB-INF/lib/groovy-all-2.4.21.jar) to constructor java.lang.invoke.MethodHandles$Lo
okup(java.lang.Class,int)
WARNING: Please consider reporting this to the maintainers of org.codehaus.groovy.vmplugin.v7.Java7$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
2023-10-16 12:55:00.615+0000 [id=30]    INFO    jenkins.InitReactorRunner$1#onAttained: Completed initialization
2023-10-16 12:55:00.815+0000 [id=24]    INFO    hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
```

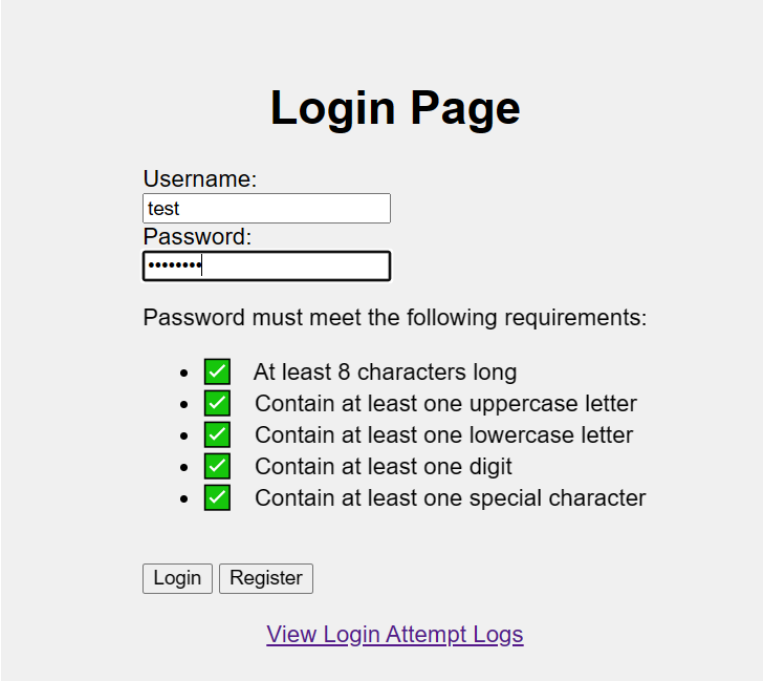

Paramètres
À propos d'Opera
Setup Wizard [Jenkins]
localhost:8080
Voulez-vous que le gestionnaire des mots de passe enregistre le mot de passe pour "localhost:8080" ?
Enregistrer
Jamais
Démarrage
Personnaliser Jenkins
Les plugins étendent Jenkins avec des fonctionnalités additionnelles pour satisfaire différents besoins.
Installer les plugins suggérés
Installer les plugins que la communauté Jenkins trouve les plus utiles.
Sélectionner les plugins à installer
Sélectionner et installer les plugins les plus utiles à vos besoins.
Jenkins 2.414.2


Paramètres
À propos d'Opera
Setup Wizard [Jenkins]
localhost:8080
Installation en cours...
Installation en cours...
Folders
OWASP Markup Formatter
Build Timeout
Credentials Binding
Timestamper
Workspace Cleanup
Ant
Gradle
Pipeline
GitHub Branch Source
Pipeline: GitHub Groovy Libraries
Pipeline: Stage View
Git
SSH Build Agents
Matrix Authorization Strategy
PAM Authentication
LDAP
Email Extension
Mailer
** Variant
** Pipeline: Groovy Libraries
** Pipeline: Stage Step
** Pipeline: Model API
** Pipeline: Declarative
Extension Points API
** Branch API
** Pipeline: Multibranch
** Pipeline: Stage Tags Metadata
** Mina SSHD API :: Common
** Mina SSHD API :: Core
** Git client
** Pipeline: Input Step
** Pipeline: Declarative
Pipeline
** Java JSON Web Token (JJWT)
** OkHttp
** GitHub API
Git
** GitHub
GitHub Branch Source
** - dépendance requise
Jenkins 2.414.2

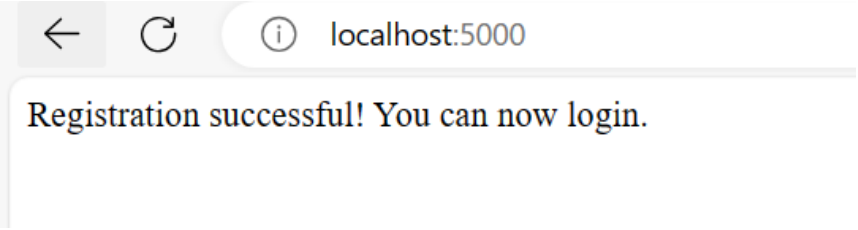


On va travailler avec une simple web app développée en python Flask, elle permet de faire un simple login avec des contraintes sur le mot de passe défini et de visualiser les logs des essais de logins faits (les logs sont sous forme de texte ou graphiquement en utilisant une bibliothèque de javascript chartjs) :

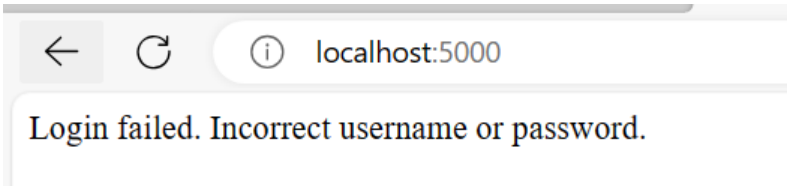




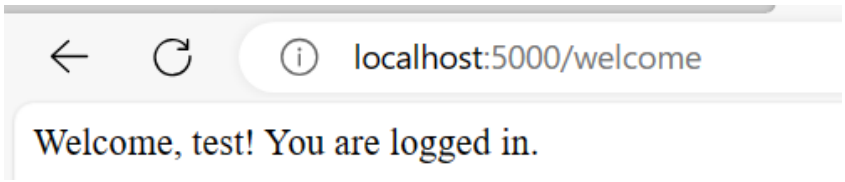
Exemple de « register » :



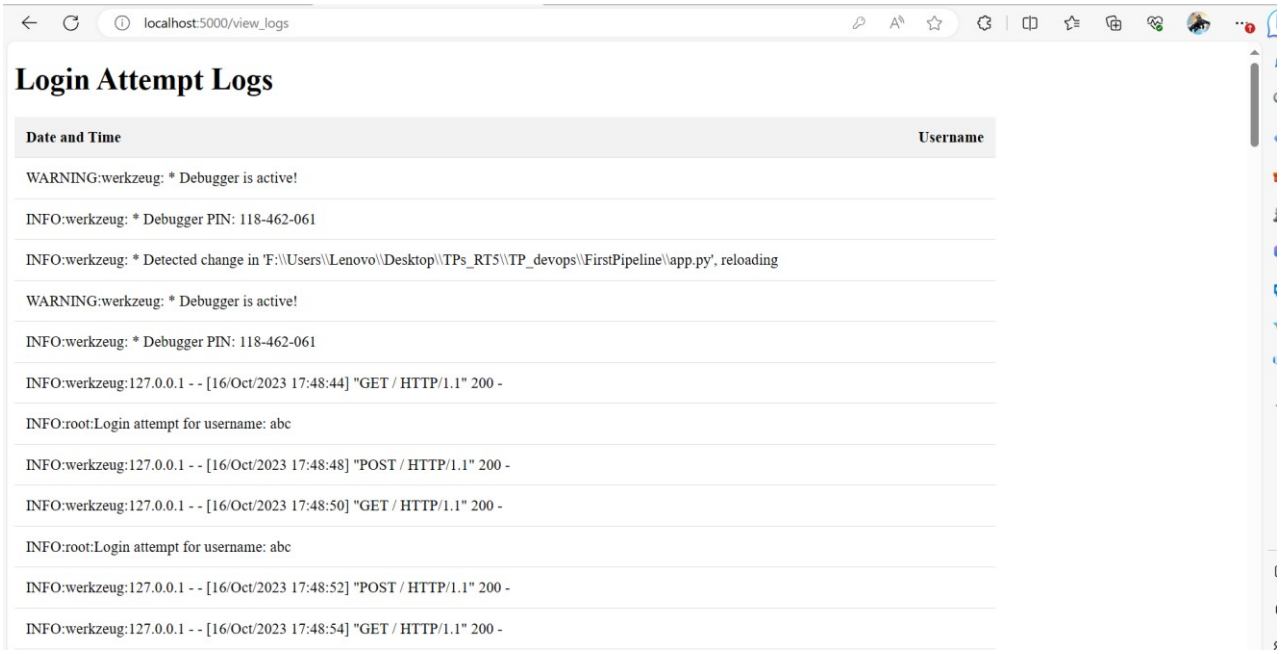
Exemple de « login » faux :



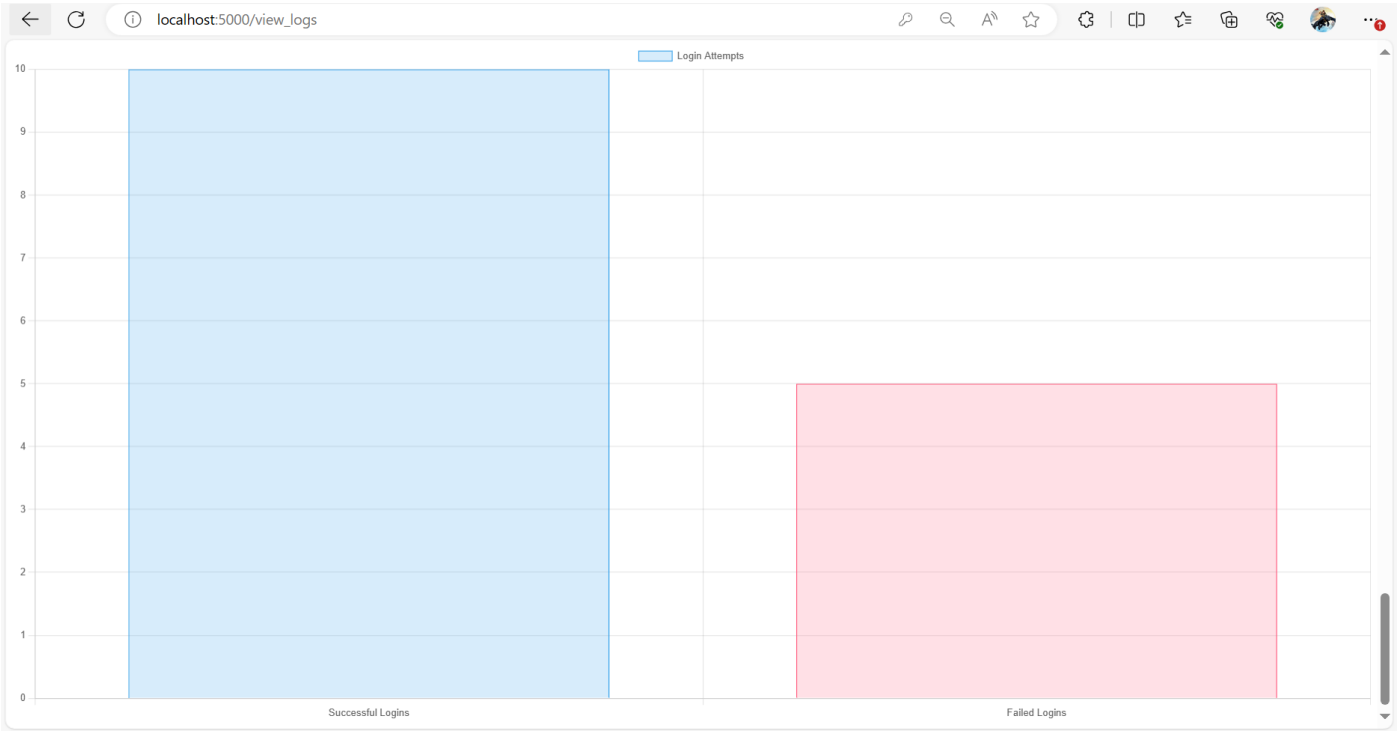
Exemple de « login » correcte :

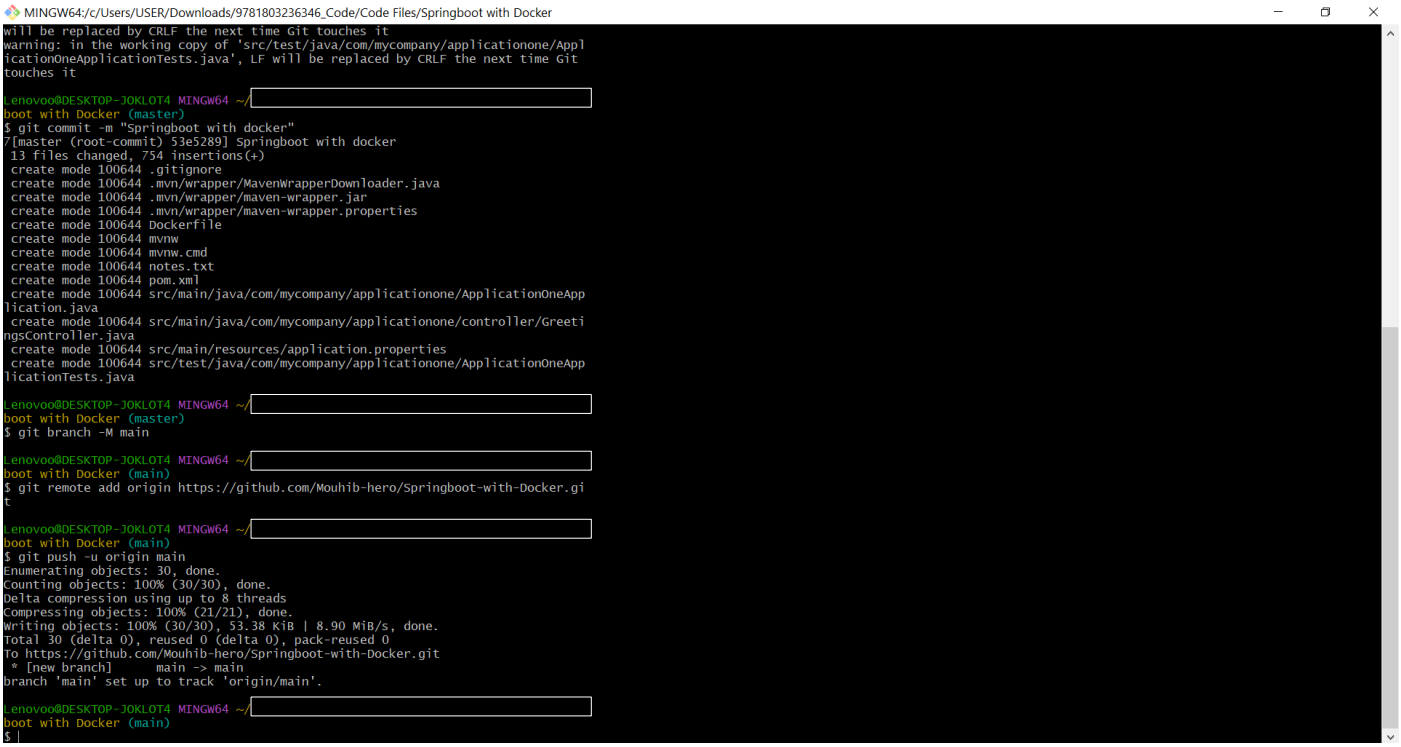


Visualisation des logs d'essais de login (login attempts) en text :

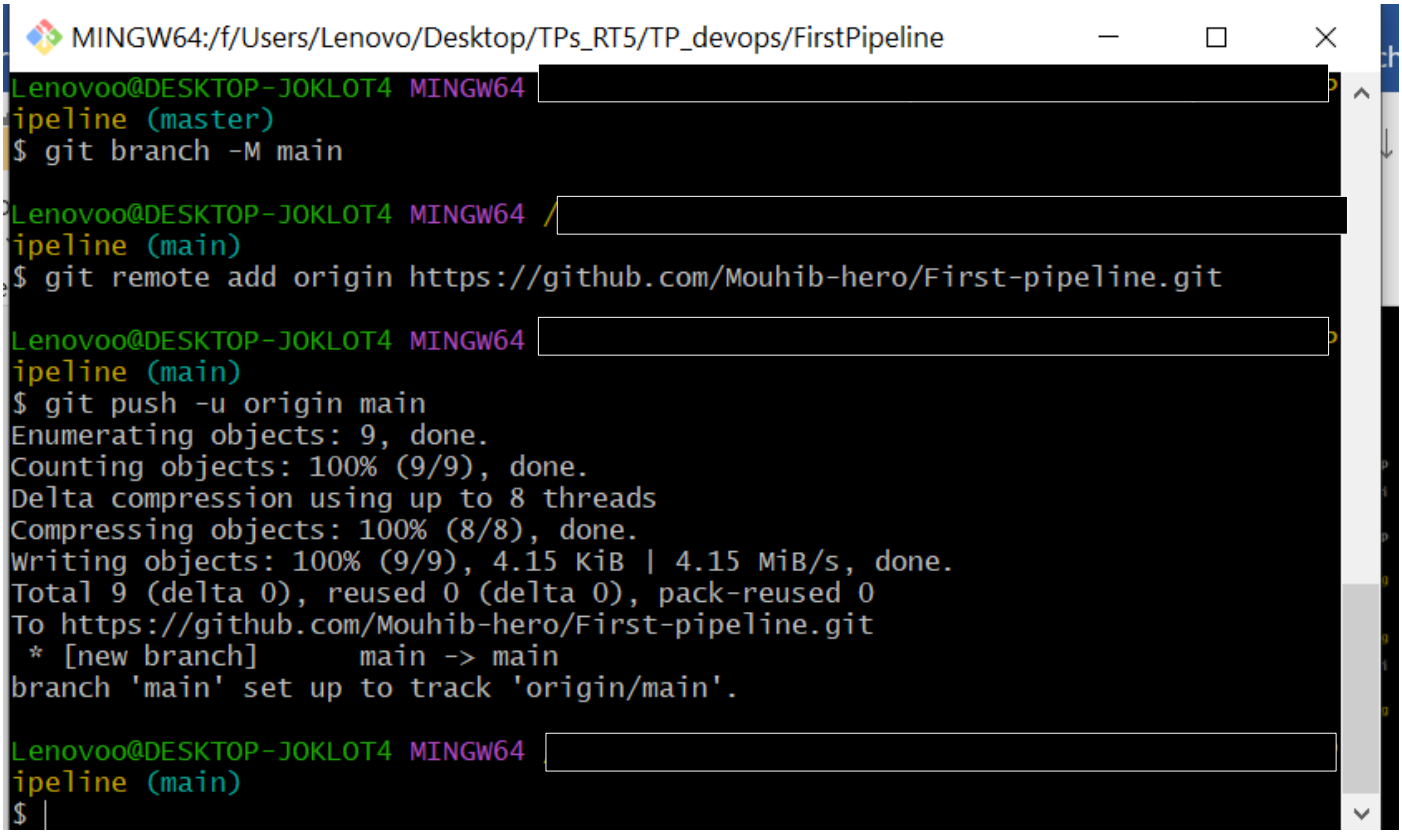


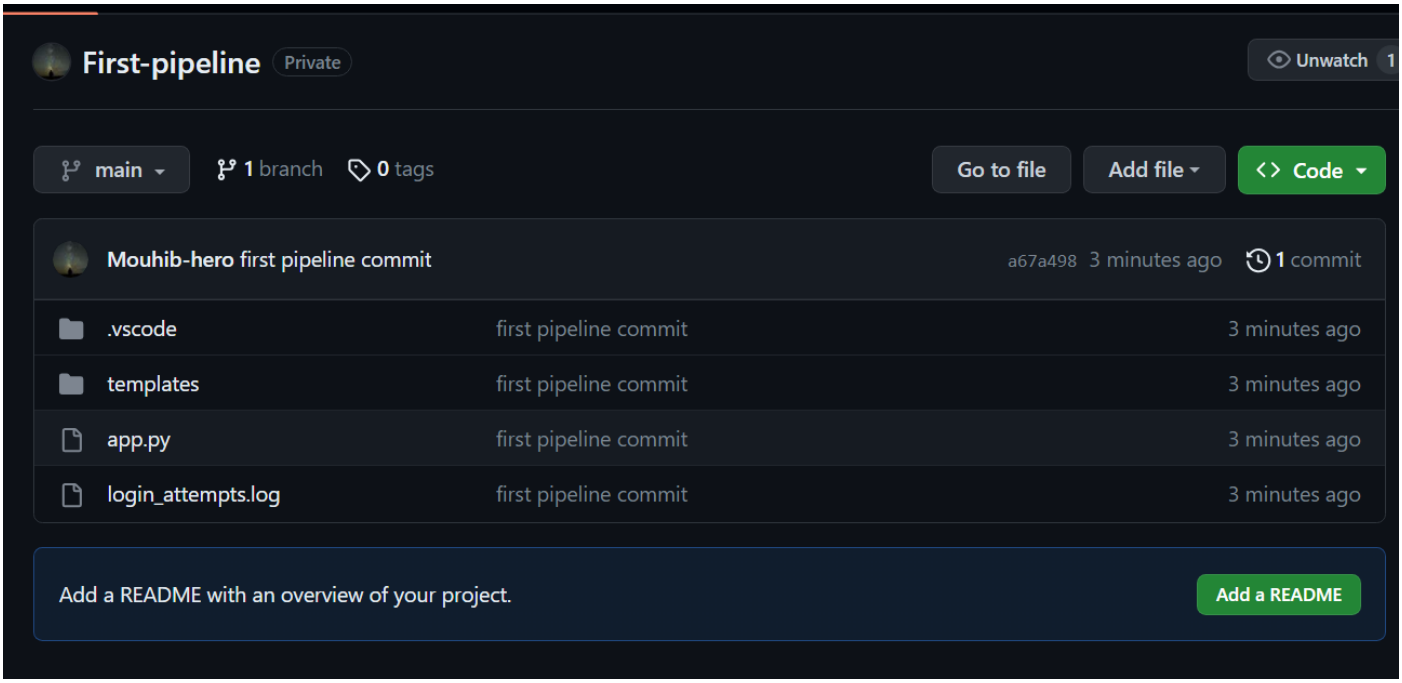
Visualisation des logs d'essais de login (login attempts) avec chartjs :



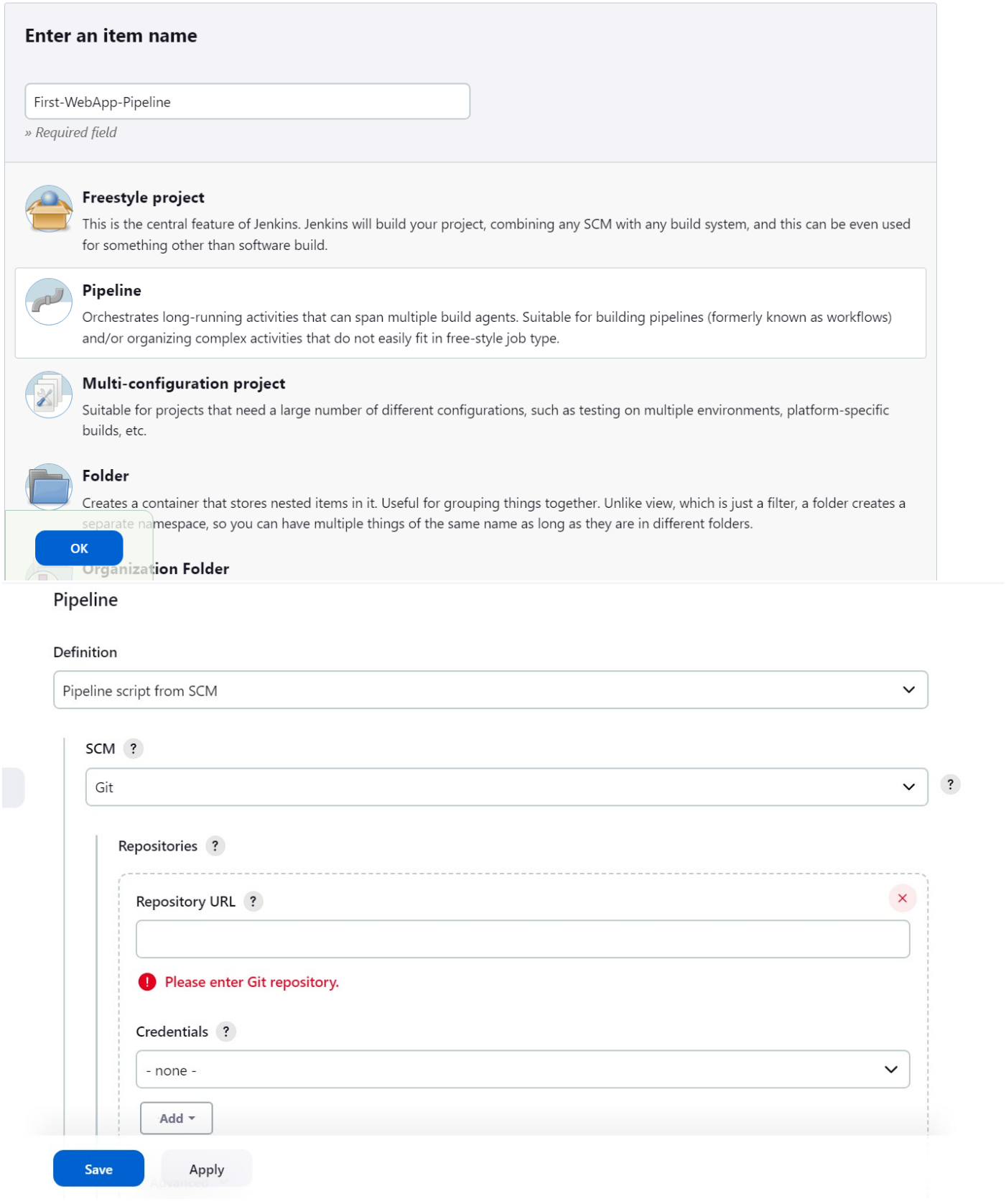


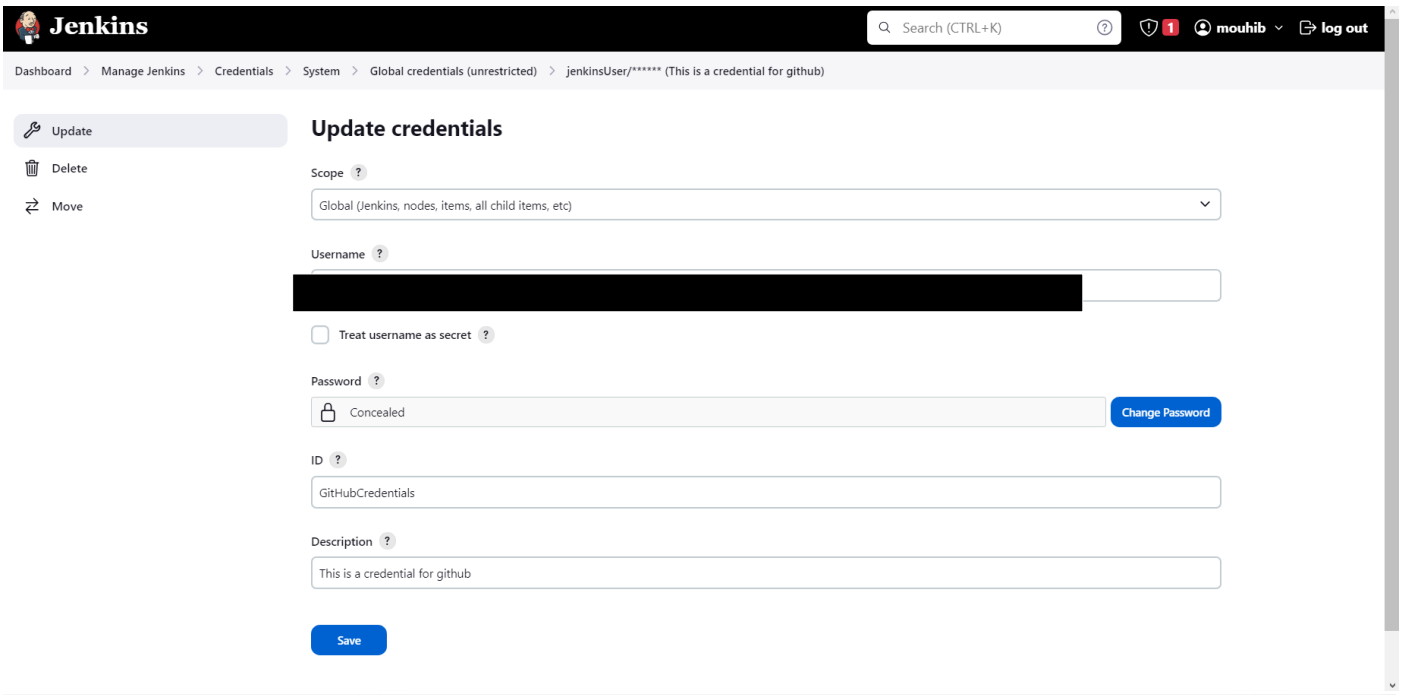
On fait push pour notre simple web application

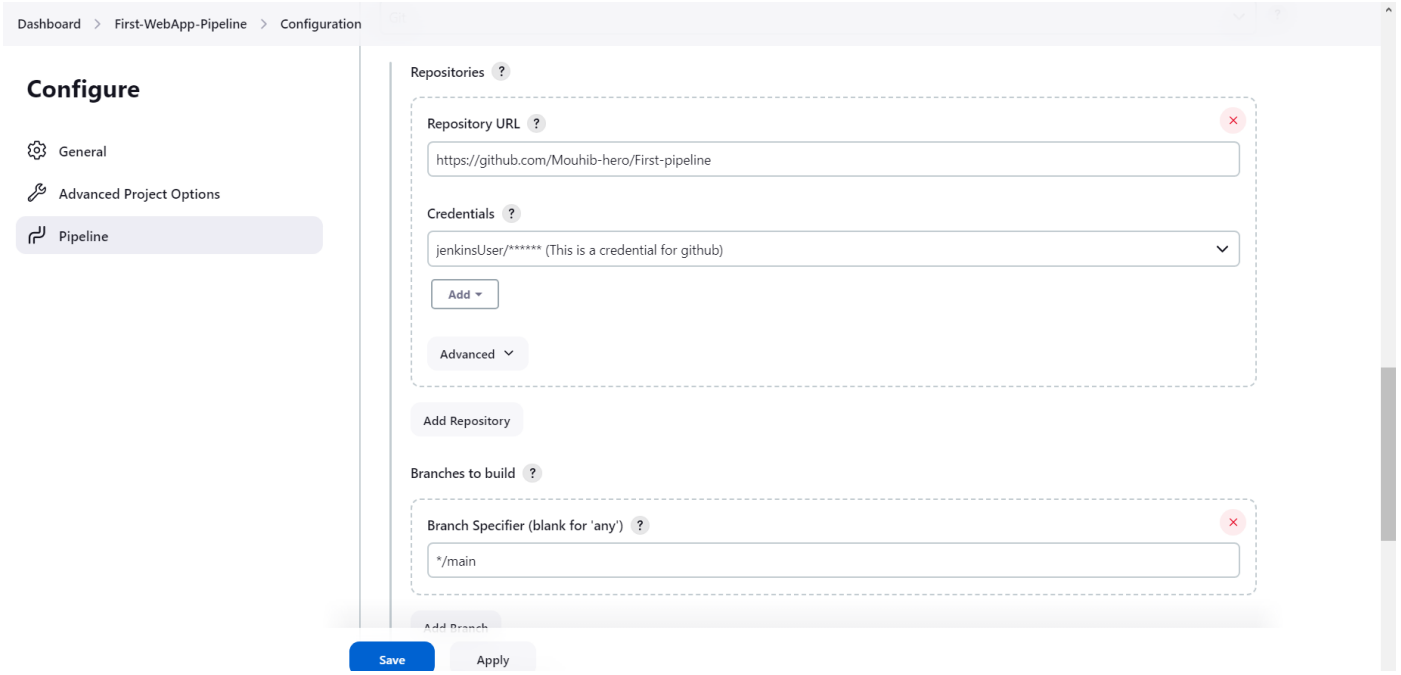


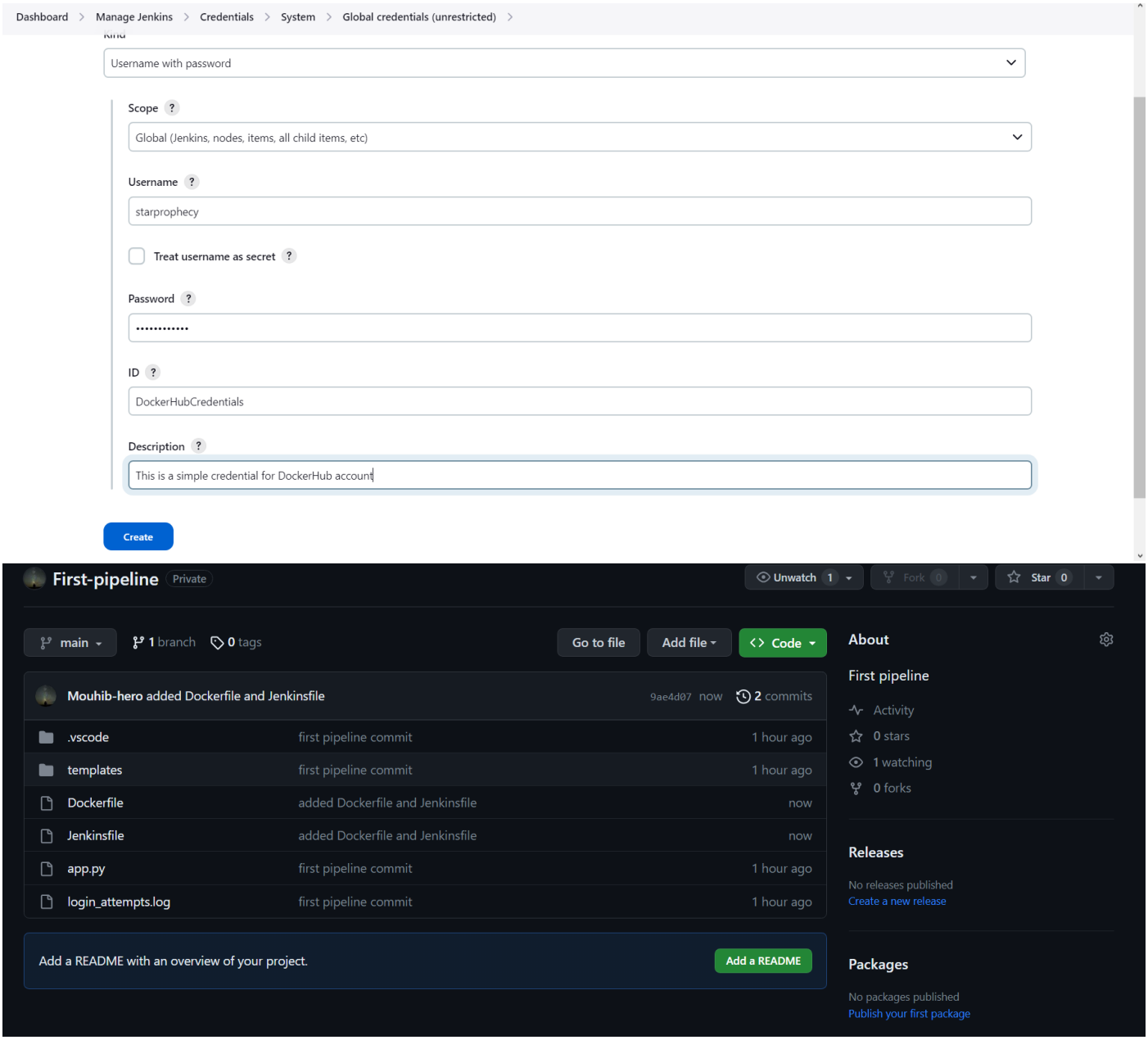


On va manipuler jenkins pour la création du pipeline :


Enter an item name
First-WebApp-Pipeline
» Required field
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
OK
Pipeline
Definition
Pipeline script from SCM
SCM
Git
Repositories
Repository URL
Please enter Git repository.
Credentials
- none -
Add
Save
Apply

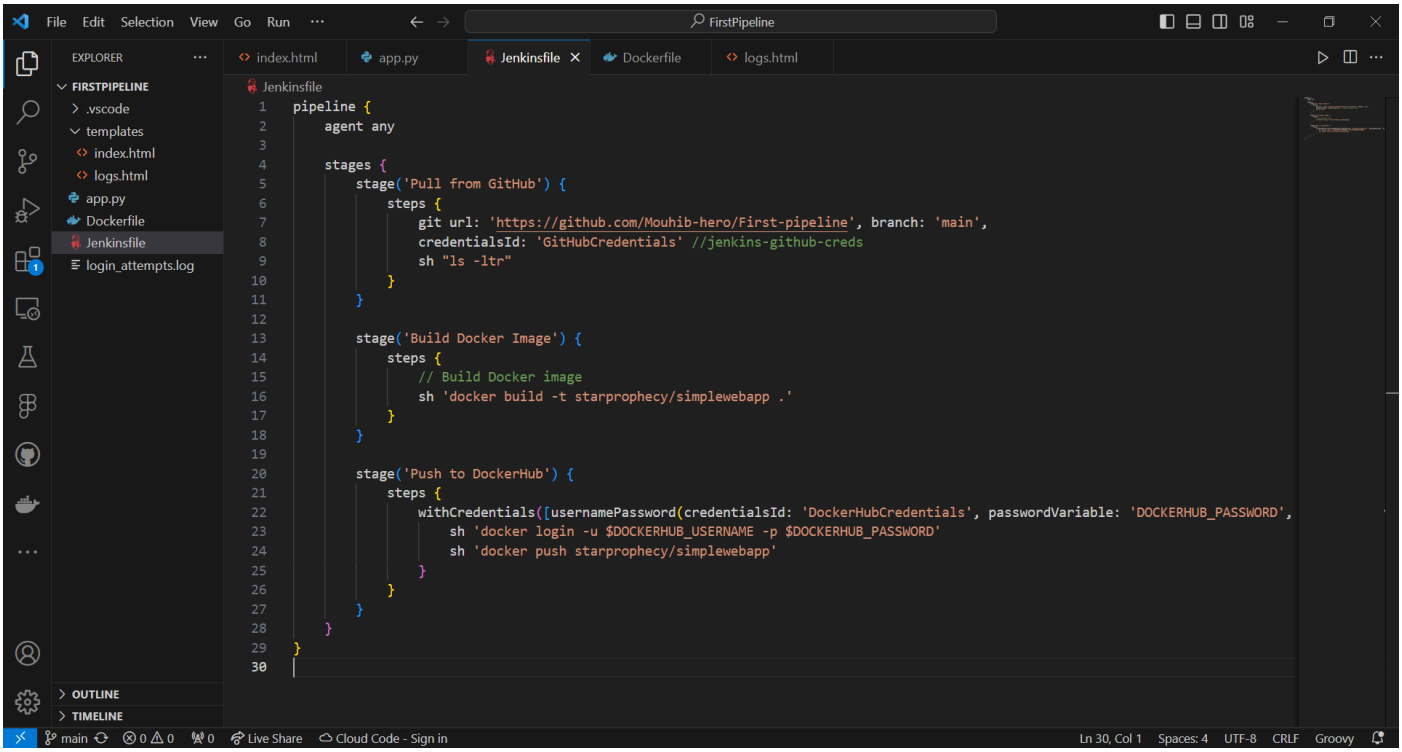

Jenkins
Search (CTRL+K)
mouhib
log out
Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) > jenkinsUser/****** (This is a credential for github)
Update
Delete
Move
Update credentials
Scope
Global (Jenkins, nodes, items, all child items, etc)
Username
[redacted]
Treat username as secret
Password
Concealed
Change Password
ID
GitHubCredentials
Description
This is a credential for github
Save

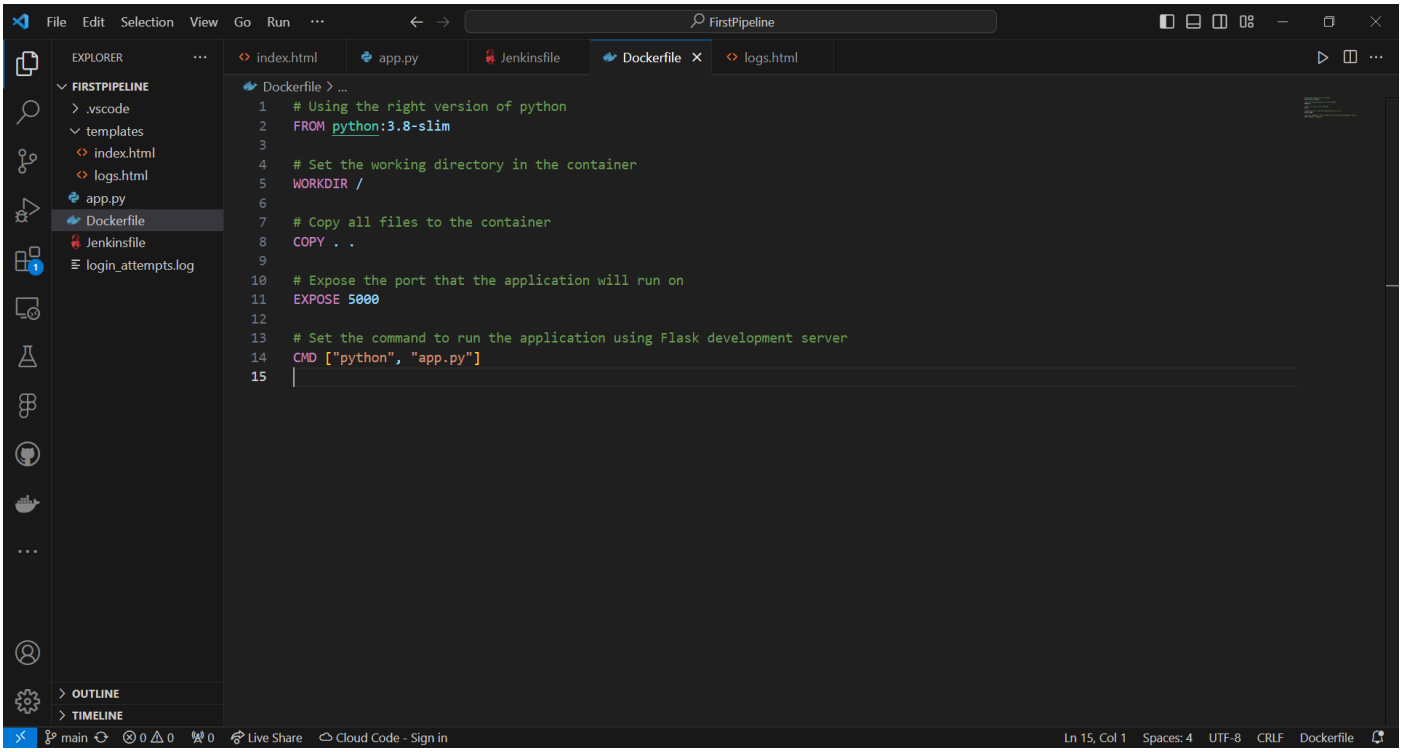

Dashboard > First-WebApp-Pipeline > Configuration
Configure
General
Advanced Project Options
Pipeline
Repositories
Repository URL
https://github.com/Mouhib-hero/First-pipeline
Credentials
jenkinsUser/****** (This is a credential for github)
Add
Advanced
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/main
Save
Apply



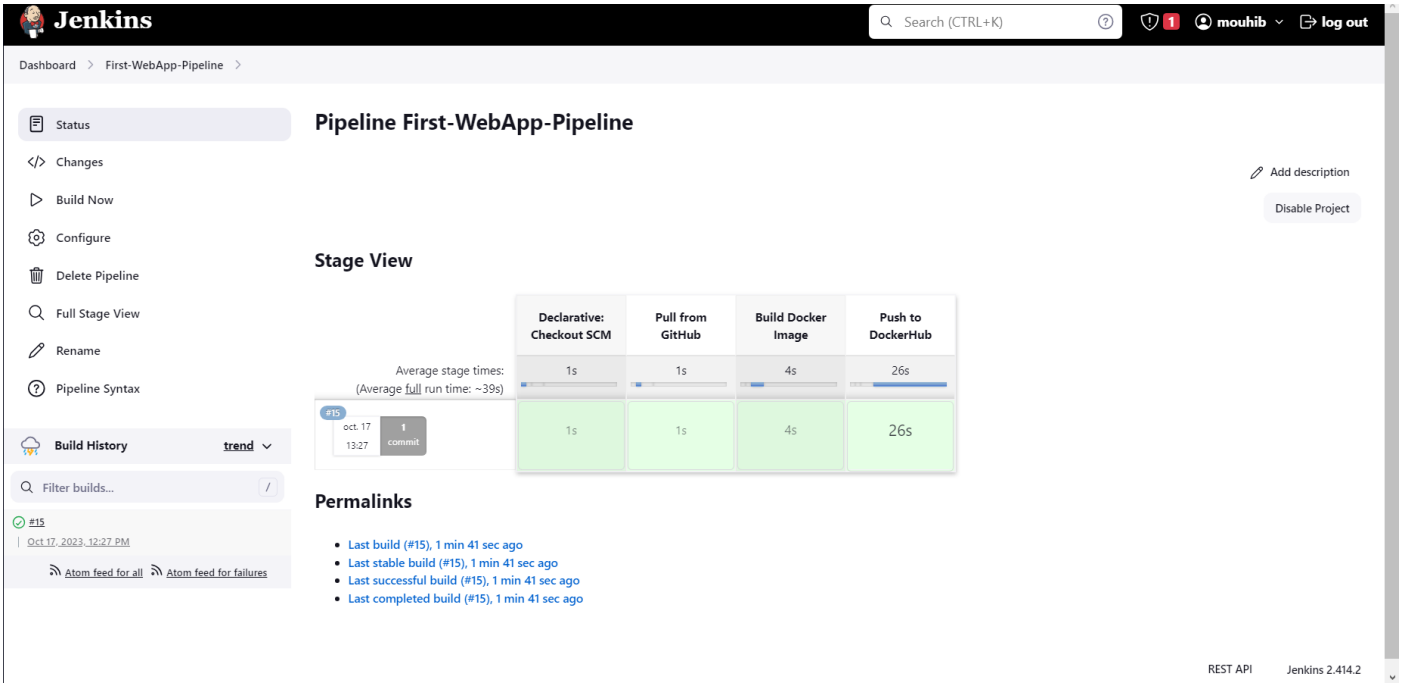
On clique sur build now ➔ erreur car la methode d'authentification n'est plus supportée.

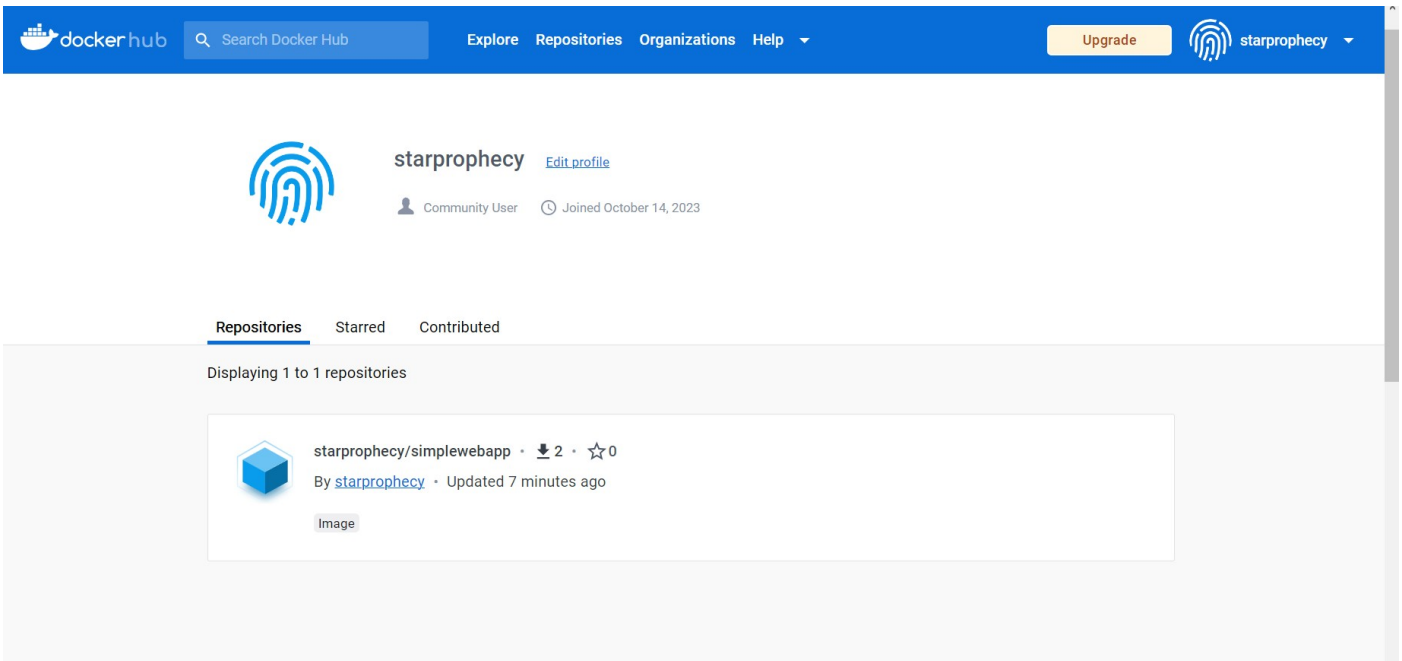
Donc on utilisera cette methode d'authentification : github adresse email avec le token github en tant que mot de passe





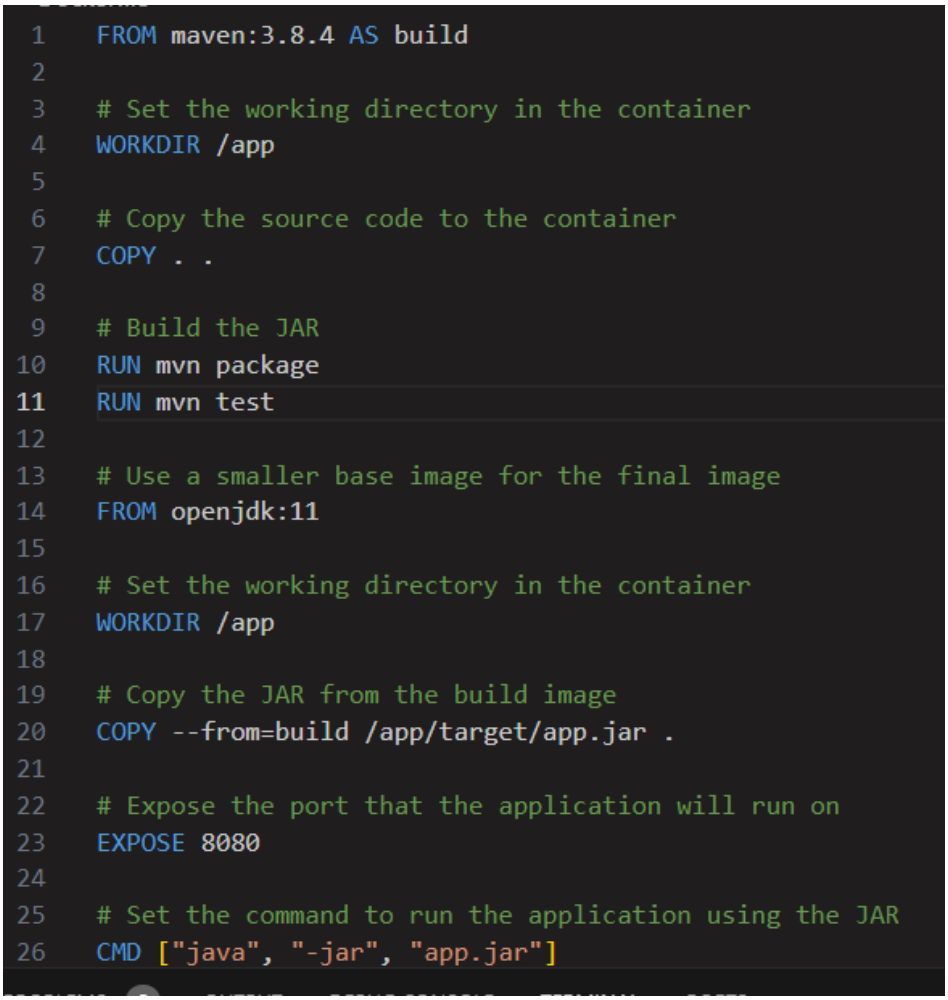
On fait des builds sur notre pipeline et on remarque que ça a marché sans aucun problème


Jenkins
Search (CTRL+K)
mouhib
log out
Dashboard
First-WebApp-Pipeline
Status
Changes
Build Now
Configure
Delete Pipeline
Full Stage View
Rename
Pipeline Syntax
Build History
trend
Filter builds...
#15
Oct 17, 2023, 12:27 PM
Atom feed for all
Atom feed for failures
Pipeline First-WebApp-Pipeline
Add description
Disable Project
Stage View
Declarative: Checkout SCM
Pull from GitHub
Build Docker Image
Push to DockerHub
Average stage times:
(Average full run time: ~39s)
1s
1s
4s
26s
#15
oct. 17
13:27
1 commit
Permalinks
Last build (#15), 1 min 41 sec ago
Last stable build (#15), 1 min 41 sec ago
Last successful build (#15), 1 min 41 sec ago
Last completed build (#15), 1 min 41 sec ago
REST API
Jenkins 2.414.2


docker hub
Search Docker Hub
Explore
Repositories
Organizations
Help
Upgrade
starprophecy
starprophecy
Edit profile
Community User
Joined October 14, 2023
Repositories
Starred
Contributed
Displaying 1 to 1 repositories
starprophecy/simplewebapp
2
0
By starprophecy
Updated 7 minutes ago
Image

Création d'un deuxième pipeline en utilisant l'application Springboot avec Docker :

Le pipeline qu'on va créer dans cette partie fait la même chose que le pipeline qui précède, la différence est dans les applications utilisées. Le code de l'application fournie « springboot with docker » sera utilisé pour cette partie. On doit clarifier que le dossier « target » dans lequel on trouve généralement le fichier .jar de l'application ne doit pas être « pushed » sur github puisqu'on doit le générer de nouveau à chaque fois dans notre pipeline. Pour cela, on ajoute la commande qui permet de créer ce type de fichier .jar « mvn clean package » dans le fichier dockerfile donc il correspondra à :



```
1   FROM maven:3.8.4 AS build
2
3   # Set the working directory in the container
4   WORKDIR /app
5
6   # Copy the source code to the container
7   COPY . .
8
9   # Build the JAR
10  RUN mvn package
11  RUN mvn test
12
13  # Use a smaller base image for the final image
14  FROM openjdk:11
15
16  # Set the working directory in the container
17  WORKDIR /app
18
19  # Copy the JAR from the build image
20  COPY --from=build /app/target/app.jar .
21
22  # Expose the port that the application will run on
23  EXPOSE 8080
24
25  # Set the command to run the application using the JAR
26  CMD ["java", "-jar", "app.jar"]
```

Files

main

Go to file t

- .mvn
- src
- target
- .gitignore
- Dockerfile
- Jenkinsfile
- mvnw
- mvnw.cmd
- notes.txt
- pom.xml

Documentation • Share feedback

Code | Blame | 29 lines (27 loc) · 942 Bytes | Code 55% faster with GitHub Copilot | Raw

```
pipeline {
    agent any

    stages {
        stage('Pull from GitHub') {
            steps {
                git url: 'https://github.com/Lquit/tpdevops', branch: 'main',
                credentialsId: 'GitHubCredentials' //jenkins-github-creds
                sh "ls -ltr"
            }
        }

        stage('Build Docker Image') {
            steps {
                // Build Docker image
                sh 'docker build -t lquit/simplewebapp .'
                //sh './mvnw clean compile'
            }
        }
        stage('Push to DockerHub') {
            steps {
                withCredentials([usernamePassword(credentialsId: 'DockerHubCredentials', passwordVariable: 'DOCKERHUB_PASSWORD', us
                    sh 'docker login -u $DOCKERHUB_USERNAME -p $DOCKERHUB_PASSWORD'
                    sh 'docker push lquit/simplewebapp'
                }
            }
        }
    }
}
```

Lancement du pipeline :

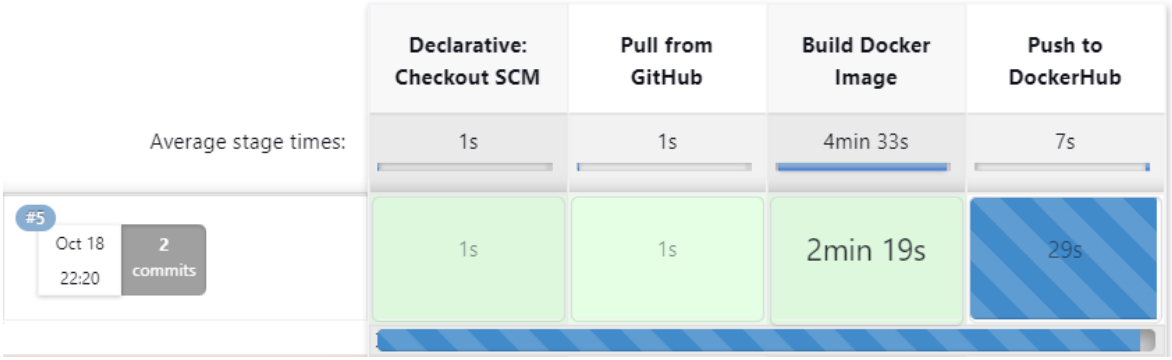
habla #13 Console [Jenkins]

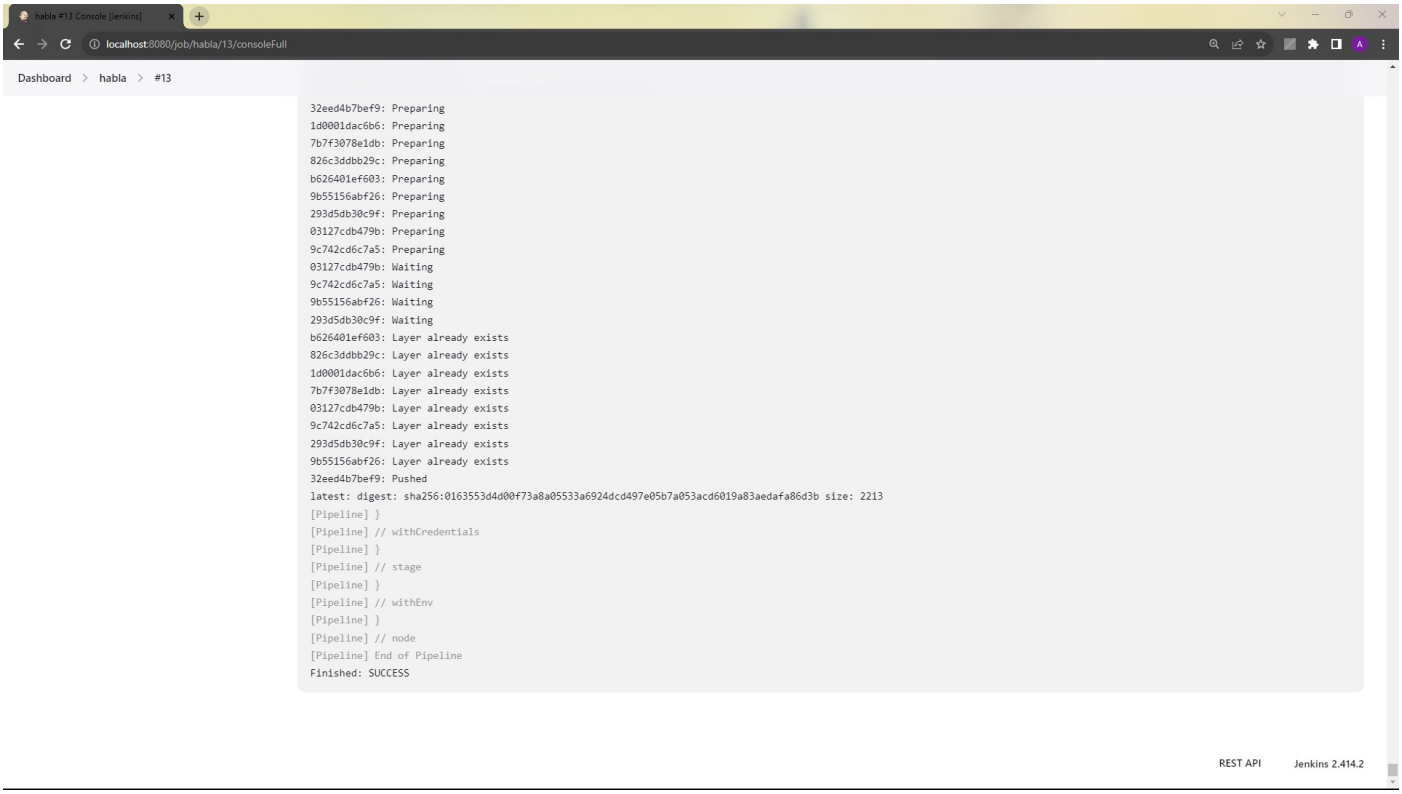
localhost:8080/job/habla/13/consoleFull

Dashboard > habla > #13

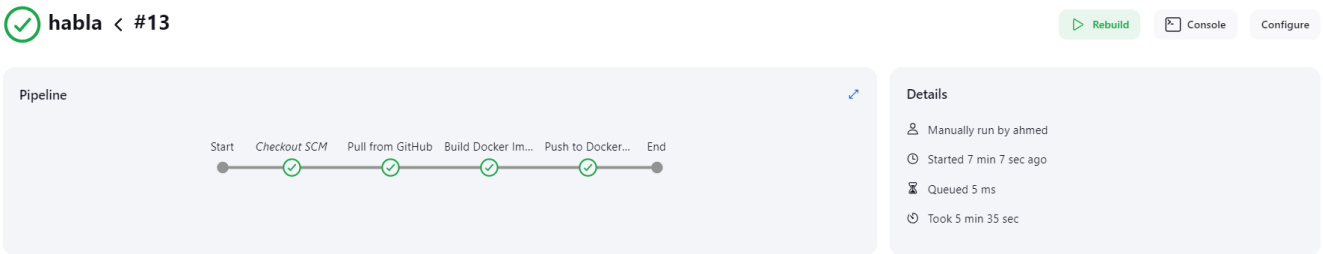
```
'org.springframework.boot.test.context.SpringBootContextLoader', parent = [null]], attributes =
map['org.springframework.test.context.web.ServletTestExecutionListener.activateListener' -> true]], class annotated with @DirtiesContext [false] with mode [null].
#11 4.726 23:44:44.081 [main] DEBUG org.springframework.test.context.support.TestPropertySourceUtils - Adding inlined properties to environment: {spring.jmx.enabled=false,
org.springframework.boot.test.context.SpringBootTestContextBootstrapper=true}
#11 5.002
#11 5.002   .   ____          _            __ _ _
#11 5.002  /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
#11 5.002 ( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
#11 5.002  \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
#11 5.002   '  |____| .__|_| |_|_| |_\__, | / / / /
#11 5.002  =========|_|==============|___/=/_/_/_/
#11 5.003  :: Spring Boot ::                (v2.5.5)
#11 5.004
#11 5.054 2023-10-18 23:44:44.408  INFO 56 --- [           main] c.m.a.ApplicationOneApplicationTests     : Starting ApplicationOneApplicationTests using Java 17.0.2 on
buildkitsandbox with PID 56 (started by root in /app)
#11 5.057 2023-10-18 23:44:44.412  INFO 56 --- [           main] c.m.a.ApplicationOneApplicationTests     : No active profile set, falling back to default profiles: default
#11 6.831 2023-10-18 23:44:46.185  INFO 56 --- [           main] c.m.a.ApplicationOneApplicationTests     : Started ApplicationOneApplicationTests in 2.095 seconds (JVM running
for 3.211)
#11 7.299 [INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 3.075 s - in com.mycompany.applicationone.ApplicationOneApplicationTests
#11 7.636 [INFO]
#11 7.636 [INFO] Results:
#11 7.636 [INFO]
#11 7.636 [INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
#11 7.636 [INFO]
#11 7.641 [INFO] ------------------------------------------------------------------------
#11 7.641 [INFO] BUILD SUCCESS
#11 7.641 [INFO] ------------------------------------------------------------------------
#11 7.643 [INFO] Total time:  5.653 s
#11 7.644 [INFO] Finished at: 2023-10-18T23:44:46Z
#11 7.644 [INFO] ------------------------------------------------------------------------
#11 DONE 7.7s

#12 [stage-1 2/3] WORKDIR /app
#12 CACHED

#13 [stage-1 3/3] COPY --from=build /app/target/app.jar .
#13 DONE 0.1s

#14 exporting to image
```

Stage View





On génère le schéma du pipeline à l'aide d'un plugin qu'on a installé dans jenkins (« pipeline graph view ») :



⇨ Ce schéma est identique pour les deux pipelines puisqu'on a utilisé la même structure du pipeline mais avec des applications différentes (l'une avec Flask et l'autre avec Springboot).

Conclusion :

Dans ce TP nous avons appris à manipuler Docker en utilisant des images nginx dans le début du TP, de manipuler une application springboot avec docker, de voir les communications entre des containers (mongodb et une application springboot) et d'utiliser docker compose pour la mise en place de deux ressources en même temps tout en lançant des tests avec JUnit dans les méthodes de tests définies dans le code.

Dans la dernière partie du TP, nous avons travaillé sur la création de deux pipelines en utilisant l'outil Jenkins. Ces deux pipelines permettent de récupérer du code d'une application web en Flask depuis github ou bien une application springboot depuis github, Construire une image docker à partir d'un docker file qui existe dans la même repository que l'application web (si c'est le cas de l'application springboot, on ajoute aussi qu'il faut créer un fichier .jar à travers la commande « mvn clean package » qui se fait dans le dockerfile) faire un build de l'image de l'application et puis faire un push de cette image dans dockerhub repository de nos propres comptes.