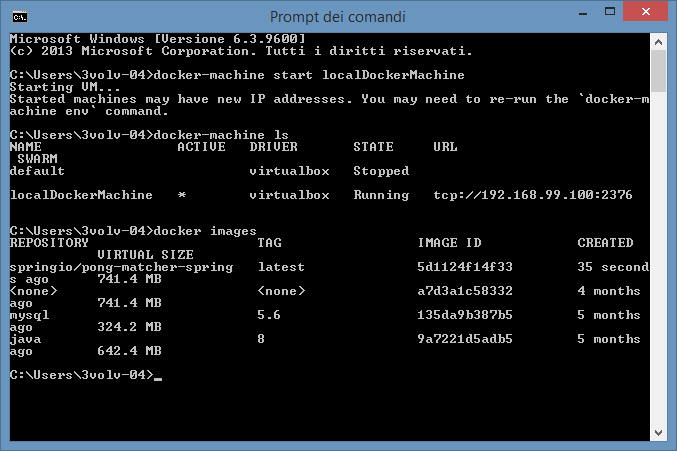
DOCKER LOCAL – DEMO HOW TO CREATE A CONTAINER

1. Brief introduction to docker
   1. Images
   2. Container
2. Run docker machine

**docker-machine start localDockerMachine**



**docker-machine ls ->** List of docker machine with URL infos

**docker images -> list of images**

1. Run maven build machine
   1. Workspace -> C:\development\workspace\01\_spring-boot-rest-backing-services
   2. 02-demo-mysql build project\_loc:pong-matcher-spring} **package docker:build**

[INFO] --- docker-maven-plugin:0.2.3:build (default-cli) @ pong-matcher-spring ---

[INFO] Copying C:\development\02-demo-mysql\target\pong-matcher-spring-1.0.0.BUILD-SNAPSHOT.jar -> C:\development\02-demo-mysql\target\docker\pong-matcher-spring-1.0.0.BUILD-SNAPSHOT.jar

[INFO] Copying src\main\docker\Dockerfile -> C:\development\02-demo-mysql\target\docker\Dockerfile

[INFO] Copying src\main\docker\dockerfileTemplate -> C:\development\02-demo-mysql\target\docker\dockerfileTemplate

[INFO] Building image springio/pong-matcher-spring

Step 0 : FROM java:8

---> 9a7221d5adb5

Step 1 : VOLUME /tmp

---> Using cache

---> 52e9b30dca40

Step 2 : ADD /pong-matcher-spring-1.0.0.BUILD-SNAPSHOT.jar app.jar

---> 5fe0d7c51477

Removing intermediate container cac525ab5725

Step 3 : RUN bash -c 'touch /app.jar'

---> Running in 59734b6a4cf7

---> a9475820ed71

Removing intermediate container 59734b6a4cf7

Step 4 : ENTRYPOINT java -Djava.security.egd=file:/dev/./urandom -jar /app.jar

---> Running in a7641b9dae96

---> 5d1124f14f33

Removing intermediate container a7641b9dae96

Successfully built **5d1124f14f33**

**docker images -> list of images check the new one**

**pom -> implementation**

<plugin>

<groupId>com.spotify</groupId>

<artifactId>docker-maven-plugin</artifactId>

<version>0.2.3</version>

<configuration>

<imageName>springio/pong-matcher-spring</imageName>

<dockerDirectory>src/main/docker</dockerDirectory>

<resources>

<resource>

<directory>C:\development\02-demo-mysql\target</directory>

<include>pong-matcher-spring-1.0.0.BUILD-SNAPSHOT.jar</include>

</resource>

</resources>

</configuration>

</plugin>

**Dockerfile**

FROM java:8

VOLUME /tmp

ADD /pong-matcher-spring-1.0.0.BUILD-SNAPSHOT.jar app.jar

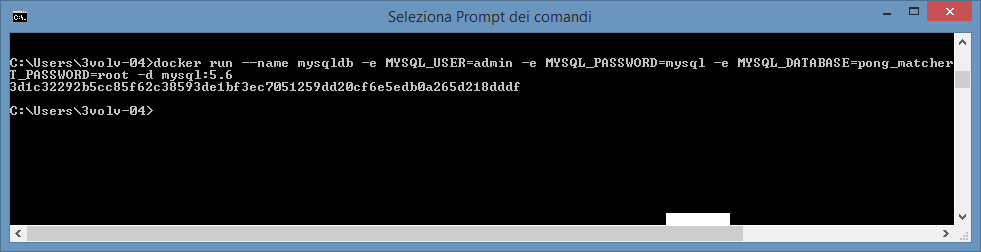
RUN bash -c 'touch /app.jar'

ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/app.jar"]

**RUN THE CONTAINERS (APP + DATABASE)**

**DATABASE FROM SCRATCH**

docker run --name mysqldb -e MYSQL\_USER=admin -e MYSQL\_PASSWORD=mysql -e MYSQL\_DATABASE=pong\_matcher\_spring\_development -e MYSQL\_ROOT\_PASSWORD=root -d mysql:5.6

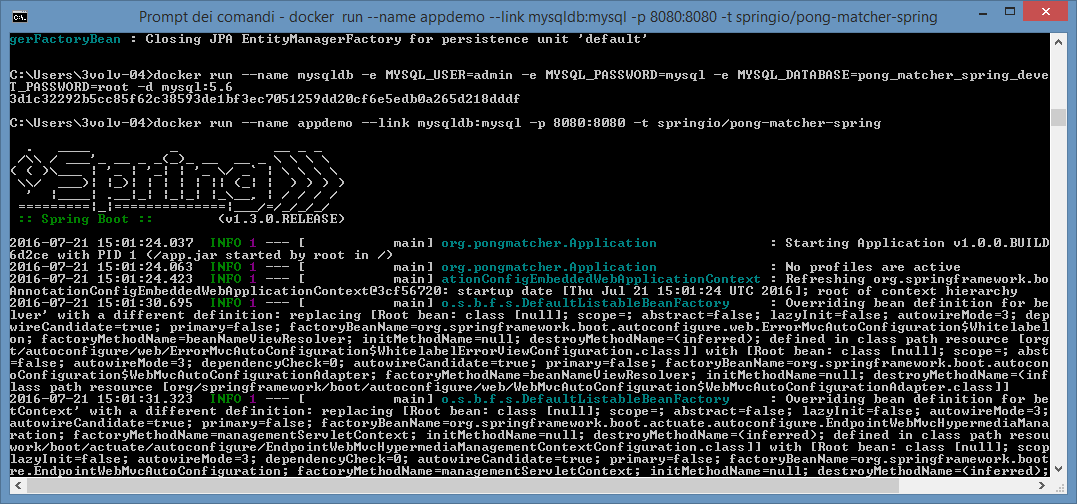


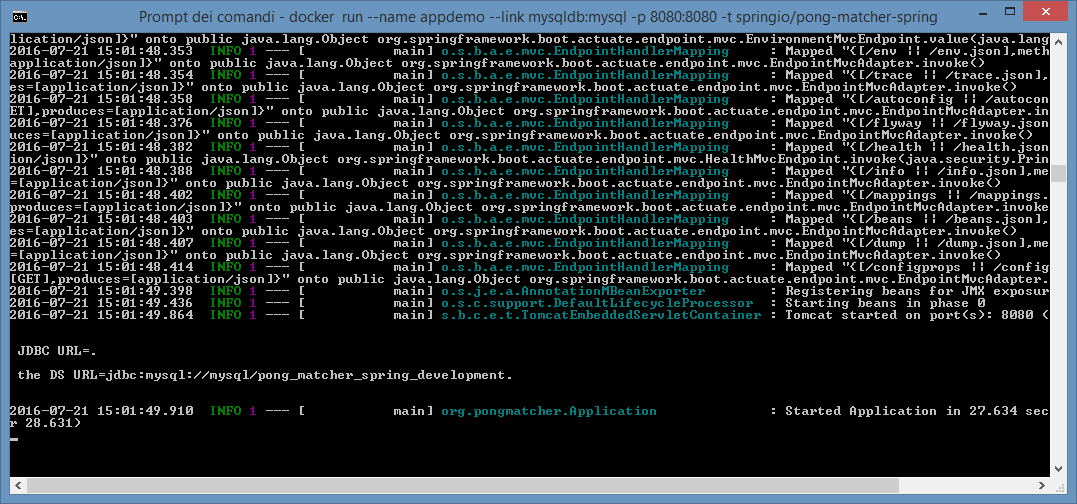
**APPLICATION CONTAINER (no output)**

docker run --name appdemo --link mysqldb:mysql -p 8080:8080 -d springio/pong-matcher-spring

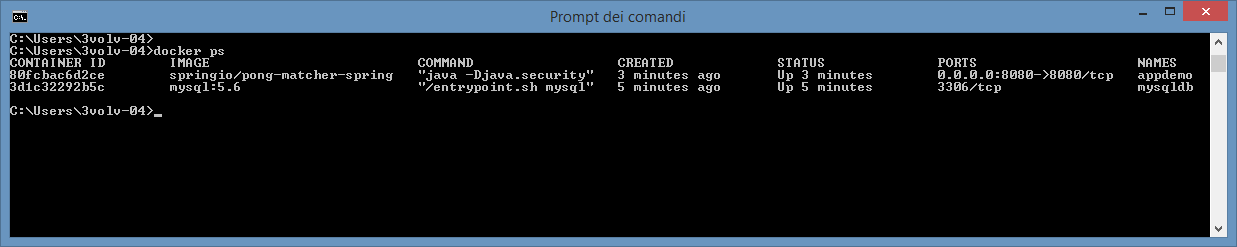
**APPLICATION CONTAINER**

docker run --name appdemo --link mysqldb:mysql -p 8080:8080 -t springio/pong-matcher-spring



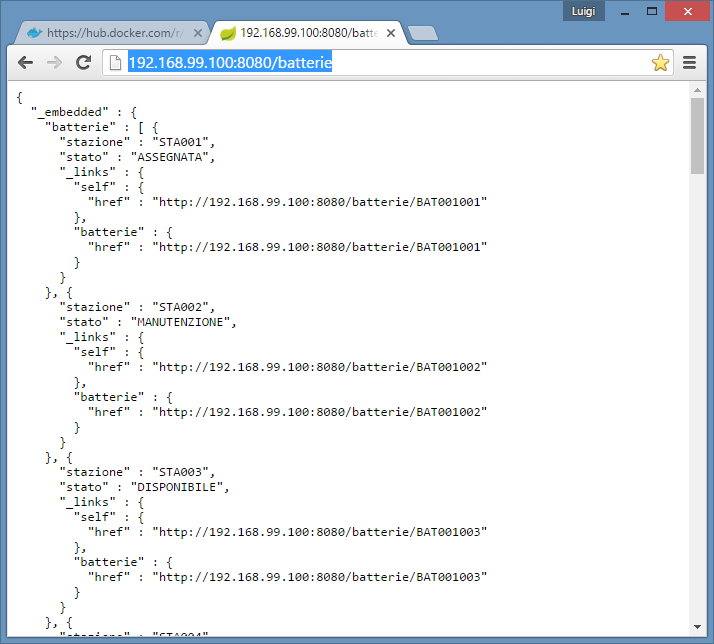


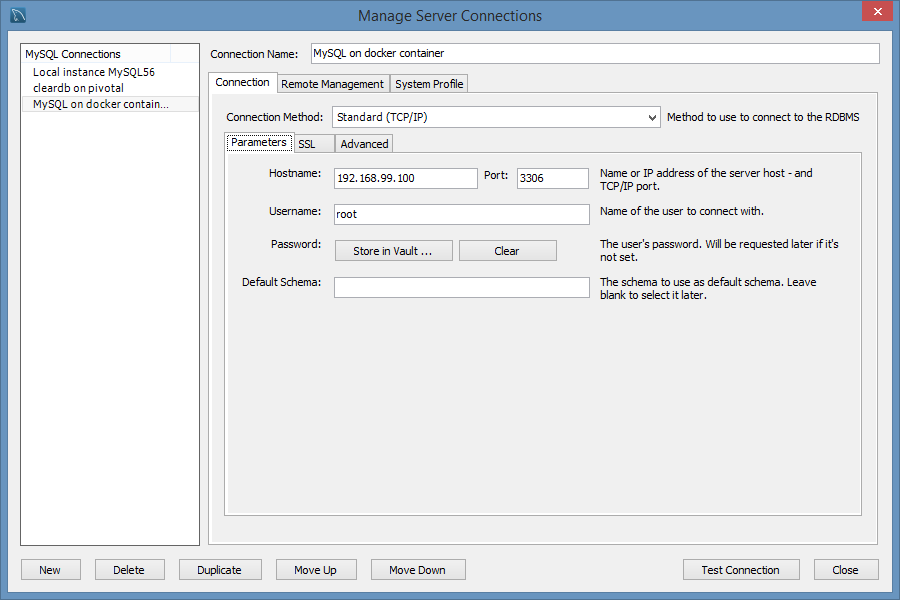
**CONTAINER IN EXECUTION**



**APPLICATION ACCESS**

http://192.168.99.100:8080/batterie



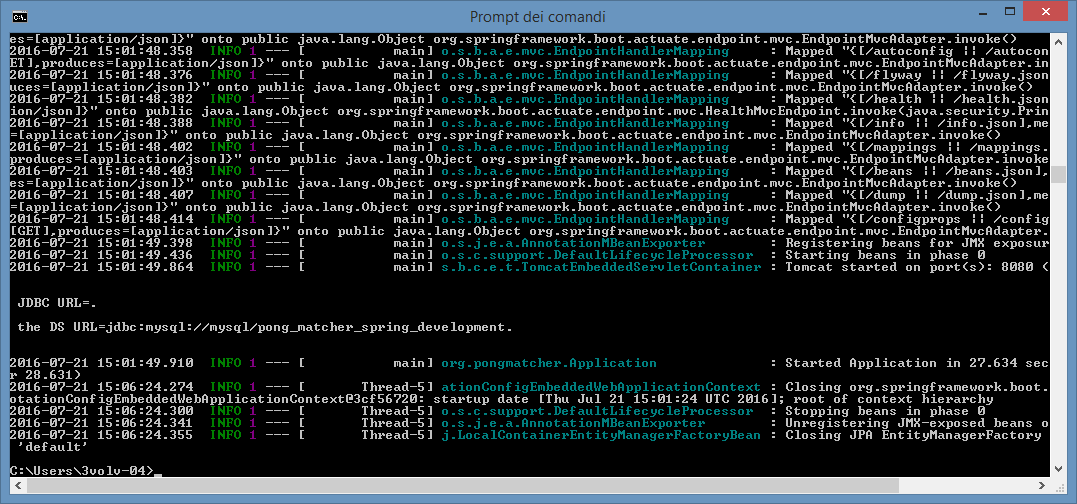


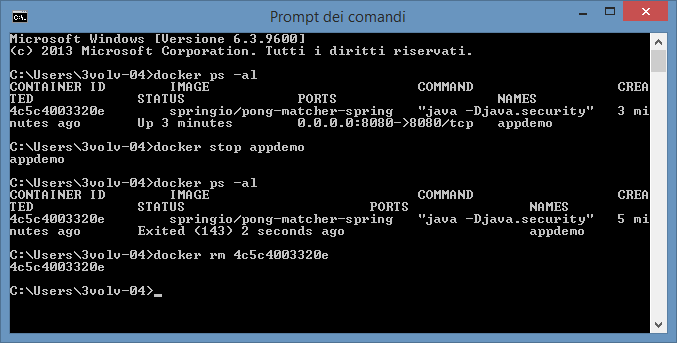
**STOP AND REMOVE CONTAINER**

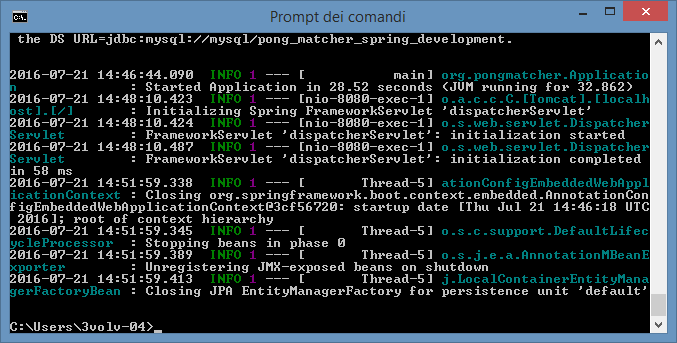
docker ps –al

docker stop appdemo

docker rm appdemo







## REMOTE DOCKER HUB

PROJECT :-> 01\_spring\_boot\_rest\_backing\_services SU GITHUB

DOCKERFILE TEMPLATE AFTER BUILD -> DOCKERFILE ON GIT dockerAutomationTest/mvn\_repo

FROM java:8

MAINTAINER l.bennardis@email.it

VOLUME /tmp

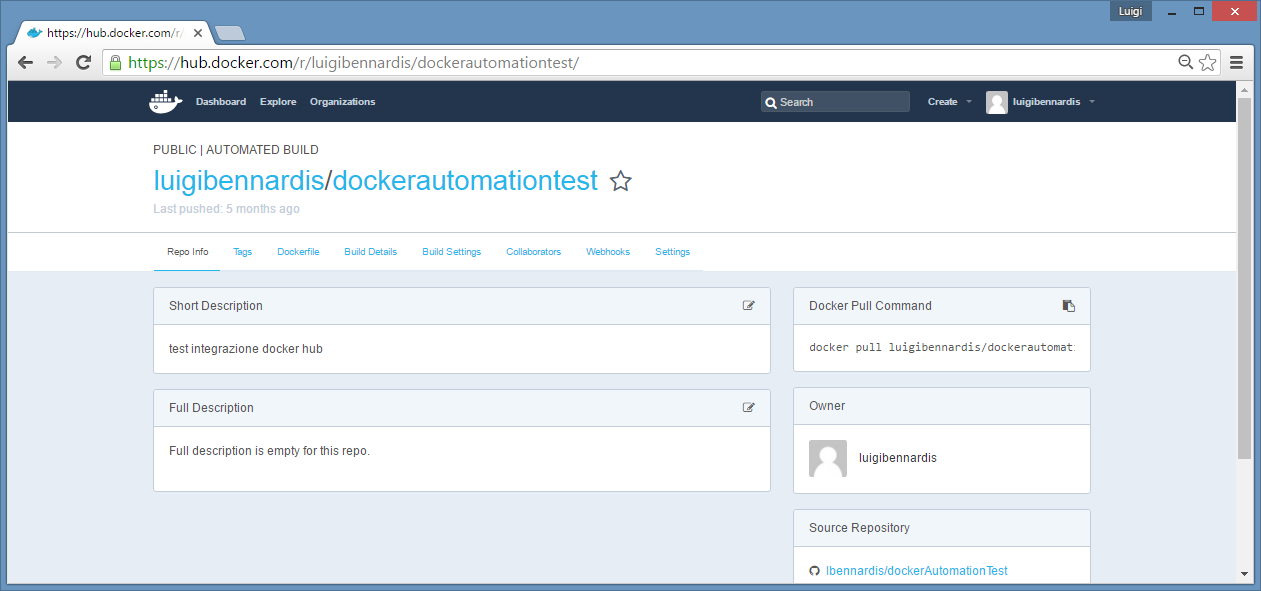
RUN mkdir /temp

RUN git clone -b mvn-repo https://github.com/lbennardis/dockerAutomationTest.git /temp

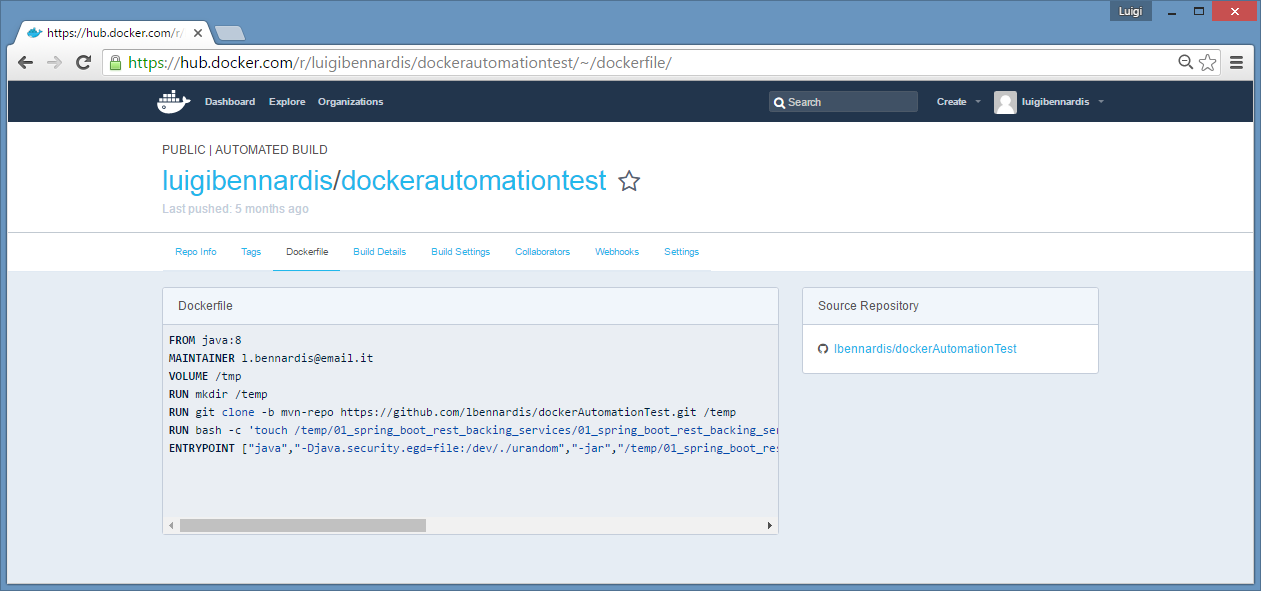
RUN bash -c 'touch /temp/01\_spring\_boot\_rest\_backing\_services/01\_spring\_boot\_rest\_backing\_services/@version@/@jar\_name@-@version@.jar'

ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/temp/01\_spring\_boot\_rest\_backing\_services/01\_spring\_boot\_rest\_backing\_services/@version@/@jar\_name@-@version@.jar"]

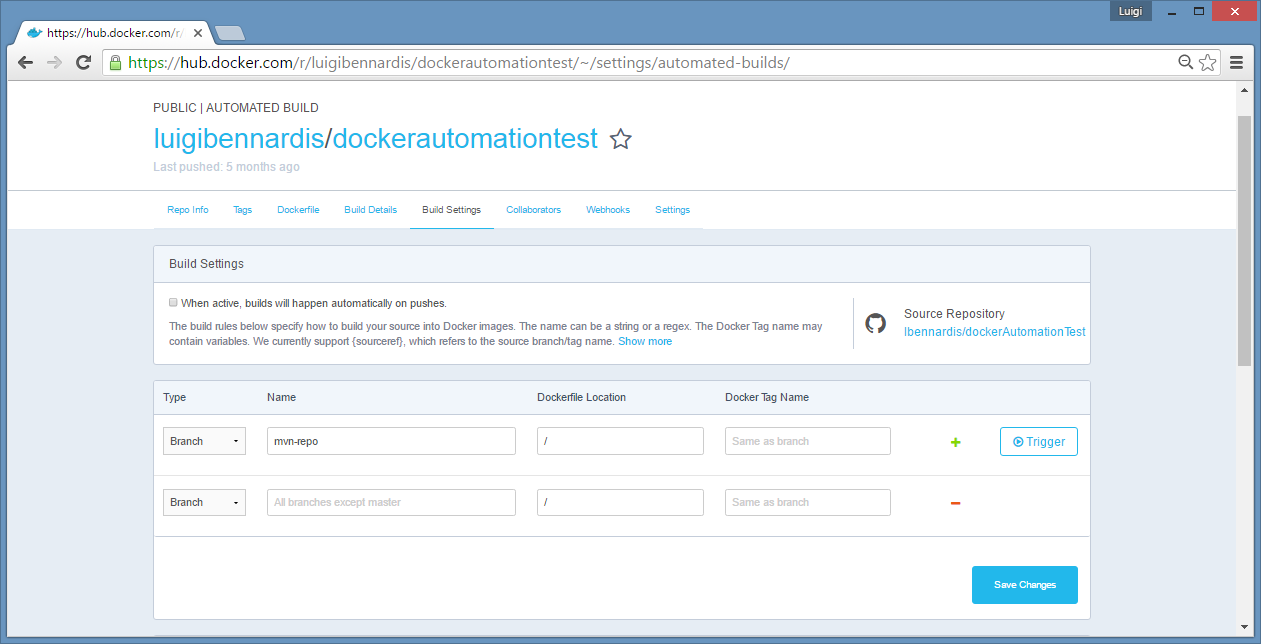
**Dockerhub**



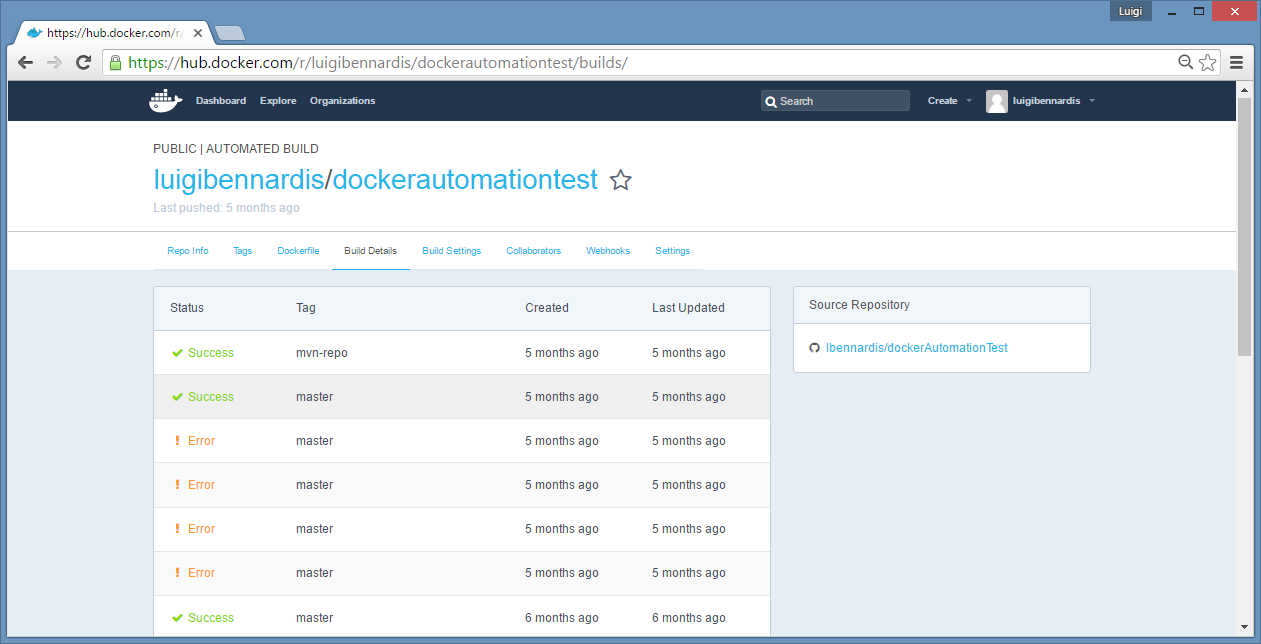
**Dockerfile**



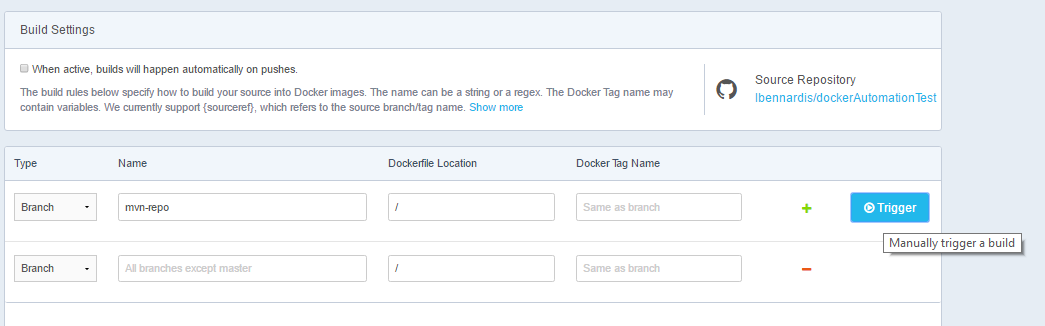
**Dockerfile settings**



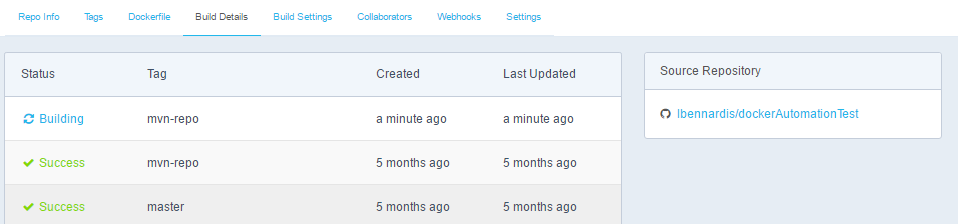
**Build details**

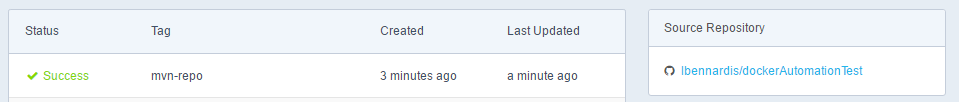


**Trigger a build**



**BUILD DETAILS**





**PULL IMAGE NOTE THE VERSION IS THE BRANCH NAME (MVN-REPO)**

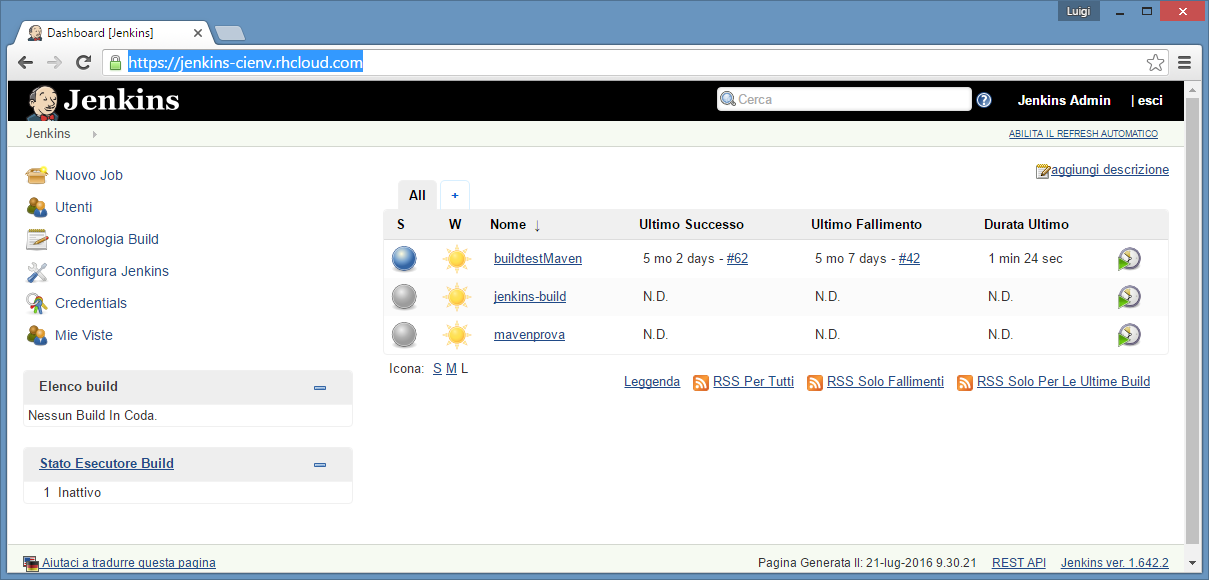
docker pull luigibennardis/dockerautomationtest:mvn-repo

//BUILD TASK INSTRUMENTED BY JENKINS@OPENSHIFT

<https://jenkins-cienv.rhcloud.com/>

USER: admin

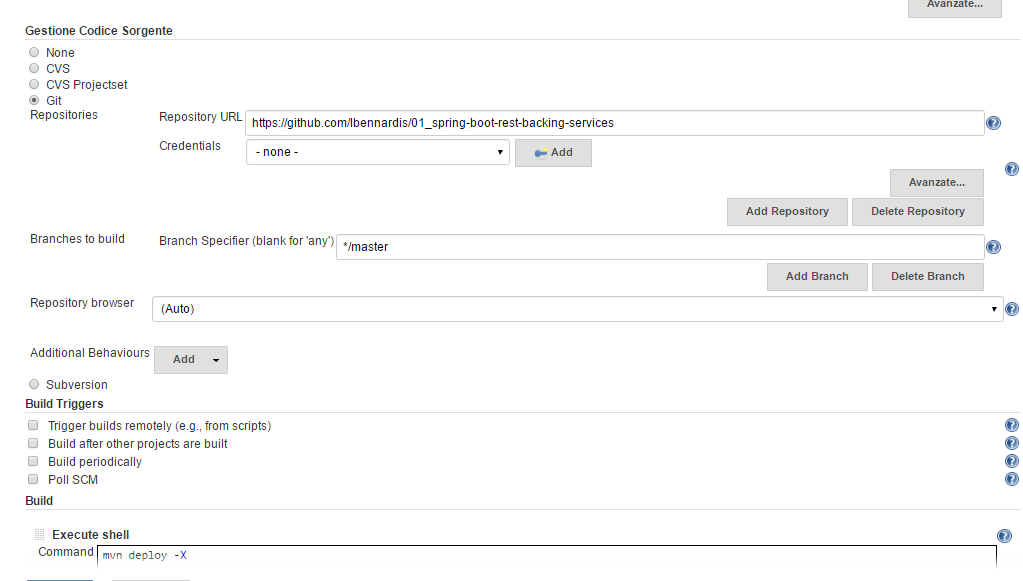
PASS: musical



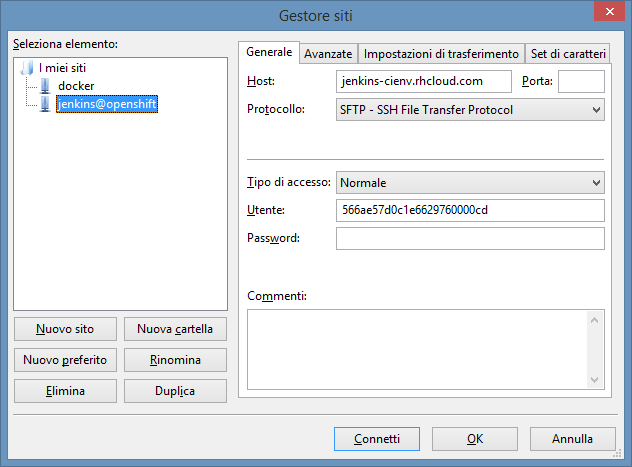
repoURL: <https://github.com/lbennardis/01_spring-boot-rest-backing-services>

command: mvn deploy –X

IMPLEMENTAZIONE TASK



FILEZILLA TO JENKINS



/var/lib/openshift/566ae57d0c1e6629760000cd/app-root/data/buildjenkins/mvn-repo