Time Tracking Deploying App and DB Necessary Commands

This document has been created to give the necessary commands to do the basics related to the deploy of the application and data base.

To deploy in Test environment a new war:

After making some changes in the tool, if we want to update this new release to Test environment we should create the war file in our workspace, push it to the Nexus Repository and finally from the OpenShift platform create the application and create a route to be accessed from an URL.

The following commands shows how to do all the necessary steps after making the changes in local to deploy it in Test environment.

0) **Change the necessary properties and settings to deploy it in the Test/Pro Environment.** (This is shown in the 'Time Tracking Local and Pro Code Differences' document)

1) Build war file:

```
$ mvn clean install -DskipTests
```

2) Build docker image:

```
$ docker build -f Dockerfile -t <image name> .
```

```
$ docker build -f Dockerfile -t timetrackingapptest .
```

3) Tag the docker image:

```
$ docker tag <image name> <nexus repository url>/<image name>:<optional version number tag>
```

```
$ docker tag timetrackingapptest gt00-gttt-docker-hosted.docker.nexus.zurich.com/timetrackingapptest:tst01
```

4) Login to Nexus Repository

```
$ docker login gt00-gttt-docker-hosted.docker.nexus.zurich.com
```

*If want to pull image from Nexus

```
$ docker pull gt00-gttt-docker-hosted.docker.nexus.zurich.com/timetrackingapptest:tst01
```

5) Push the image to the Nexus repository:

```
$ docker push <nexus repository url>/<image name>:<optional version number tag>
```

```
$ docker push gt00-gttt-docker-hosted.docker.nexus.zurich.com/timetrackingapptest
```

6) Login into OpenShift

```
$ oc login https://ocp.zurich.com
```

7) Remove if needed previous app from:

- Application -> Routes
- Application -> Services
- Application -> Pods
- Application -> Deployments

8) Create the application on OpenShift. The database connection details are added as environment variables:

```
$ oc new-app --name=<app name> -e spring_datasource_url=jdbc:mysql://<mysql pod service name>:3306/time_tracking_db?useSSL=false -e spring_datasource_username=<database username> -e spring_datasource_password=<database password> <nexus repository url>/<image name>
```

```
$ oc new-app --name=timetrackingapptest -e
spring_datasource_url=jdbc:mysql://timetrackingdb-
v1:3306/time_tracking_db?useSSL=false"&"allowPublicKeyRetrieval=true -e
spring_datasource_username=pperezcivit -e
spring_datasource_password=tTracking2019 gt00-gttt-docker-
hosted.docker.nexus.zurich.com/timetrackingapptest:tst01
```

9) Once the application is created, an OpenShift Route is required to access the application outside of OpenShift:

```
$ oc create route edge <name for route> --hostname=<name of
application>.ocpapps.zurich.com --service=<application service name> --
insecure-policy=Redirect
```

```
$ oc create route edge timetrackingapptest --
hostname=timetrackingtest.ocpapps.zurich.com --service=timetrackingapptest --
insecure-policy=Redirect
```

10) Change the port where the app is exposed:

Applications -> Routes -> select app -> Actions -> Edit -> 8080

11) The application is now available at the URL given to the --hostname parameter:

https://timetrackingtest.ocpapps.zurich.com/

Deploying a Data Base in OpenShift

Unlike the application that has several changes, the database once is created, it doesn't suffer so many modifications.

Anyway, here is explained how to create a new one

From the Service Catalog page from https://ocp.zurich.com/ select MySQL from the catalog and **fill in with the following parameter:**

Memory Limit: 512Mi
Namespace: openshift
Database Service Name: timetrackingdb
Username: pperezcivit
Password: tTracking2019
Root password: tTracking2019
DataBase Name: time_tracking_db
Volume Capacity: 5Gi
MySQL version: 5.7

Giving as a result:

-> The following service(s) have been created in your project: timetrackingdb.
-> Username: pperezcivit
-> Password: tTracking2019
-> Database Name: time_tracking_db
-> Connection URL: mysql://timetrackingdb:3306/

It will create an entry into:

- Application -> Deployments
- Application -> Services
- Resources -> Secrets
- Storage