## Prepare the Start Code for CI and CD

How to use our start code: <https://github.com/dat3startcode/rest-jpa-devops-startcode>

*This assumes you have created a droplet as explained in our setup instructions*

## Initial Setup (Do not open the project in your IDE for anything below)

1. Before you start verify that your local docker-environment is started and startcode\_test is available
2. In a terminal (git bash on Windows) clone the project. CD into the project folder and delete the .git folder and Do "Your own" git init
3. In the project folder run this: **mvn clean test** (make sure your docker environment is up)
4. When everything above is fine, create your OWN repository for the project on github. Commit and Push your code to this repo.
5. Go to ***Travis-ci.com*** and Sign up with GitHub
6. Accept the Authorization of Travis CI. You’ll be redirected to GitHub
7. Click the green Activate button, and select the new repository to be used with Travis CI
8. Now you are ready for the next steps :-)

## Deploy the project (manually) via Maven

Before you start make sure you have all these details ready (make a small document with the values FIRST)

**The user you have created on your droplet MySQL server, with GRANTS to all databases:**

User: \_\_\_\_\_\_\_\_\_

Password \_\_\_\_\_\_\_\_\_

**The user you have created to allow deployment on Tomcat**

(On your droplet look in tomcat/tomcat-users.xml if you have forgot):

User: \_\_\_\_\_\_\_\_\_

Password \_\_\_\_\_\_\_\_\_

**The name to your server** https://xxxxxx.dk (make sure you can access /manager/html)

1. Open your project in your IDE, since we have to make a few changes to the code and pom-file
2. Open the pom-file, and locate the properties-section at the start of the file. Change the value for **remote.server** to a URL pointing to YOUR droplet
3. On your droplet, either using workbench locally or via the SQL-client on the droplet, create a new database called **startcode\_demo**.
4. ssh into your droplet, and navigate into the root of the cloned docker project
5. Stop your docker-servers: docker-compose down
6. With nano, open **docker-compose.yml**

Under the web: section find the lines given below and change USER, and PW to your values, and change CONNECTION\_STR to point **startcode\_demo** (If you have an *existing Java-project*, using a database on your droplet, follow the instructions given in DigitalOcean Setup instructions "*Create another database and set environment variables to host additional Java Projects*")

**CONNECTION\_STR: "jdbc:mysql://db:3306/startcode"**

**USER: "dev"**

**PW: "ax2"**

DEPLOYED: "PROD"

1. Save the file, rebuild and run using these commands:

**docker-compose down**

**docker-compose build**

**docker-compose up -d**

1. Back in a LOCAL terminal (git bash on Windows), in the root of the Java-project, type:  
   **mvn clean test -Dremote.user=USER -Dremote.password=PW tomcat7:deploy**

The values for user and password above, are those YOU have added in tomcat-users.xml during the initial setup of your droplet.

*HOPEFULLY, those values are NOT* ***admin*** *and* ***admin123****, if they are, do the following:*

**docker-compose down**

With nano, open **tomcat/tomcat-users.xml** Change username and password and save.

run **docker-compose build --no-cache**

Start the servers again: **docker-compose up -d**

1. If everything was fine the project was deployed to your droplet, ready to use with the remote database. Test like this:

**URL\_OR\_DomainName/devops-starter/api/xxx/count**

## 

## Let Travis deploy your project - if it builds and all tests are green

*In this step we will let travis deploy our project if it builds and all tests are green :-)*

1. Login to Travis using Github, and select your project on the dashboard
2. Click "More options" and select "settings"
3. Create two Environment Variables with names and values as sketched below (two steps)

REMOTE\_USER : XXXXXX

REMOTE\_PW : XXXXXXX

*These are the values you added in step 8, when we deployed via maven by yourself. In this part, since you did it yourself, secrets were not an issue. On Travis it is. So this ensures that no one else can see your credentials*

1. Now make a small change to index.html (one that is easy to see after deploy)
2. In a terminal, in the root of the project, type: **mvn clean test** (always build and test before you commit)
3. If everything was fine, commit and push your changes
4. On travis-ci.org verify that your commit has been detected and a "build cycle" has started
5. If everything was fine (might take a few minutes) verify that Travis has deployed your new war file to your droplet