
Course Setup and Projects

Objective

The aim of this exercise is to create a couple of git repositories of your own with projects in that we can use Jenkins to build later in the course. We will manually build the projects to ensure that everything is working as intended.

Overview

We will create a Bitbucket repository for the hello-scalatra dino web app.

Create an account on bitbucket and fork the project we will be building

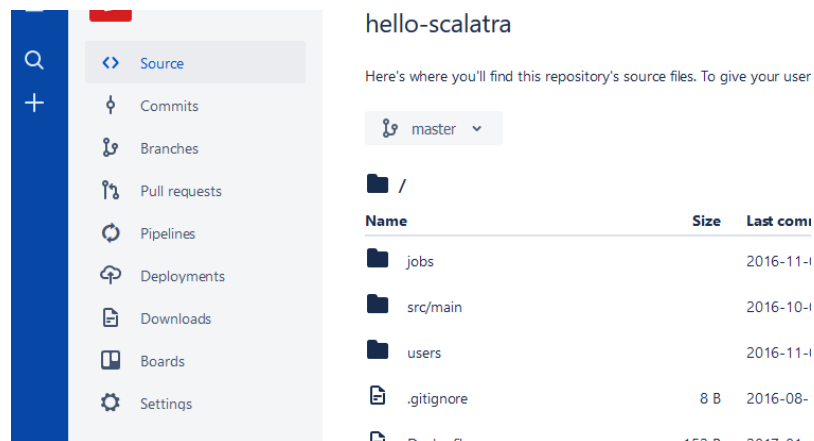
You will need a Bitbucket account for this course. Create one if you haven't done so yet.

When we link git and Jenkins together the instructions will assume that you are using a bitbucket account. There are alternate methods for connecting GitLab, which you can investigate if you have time!

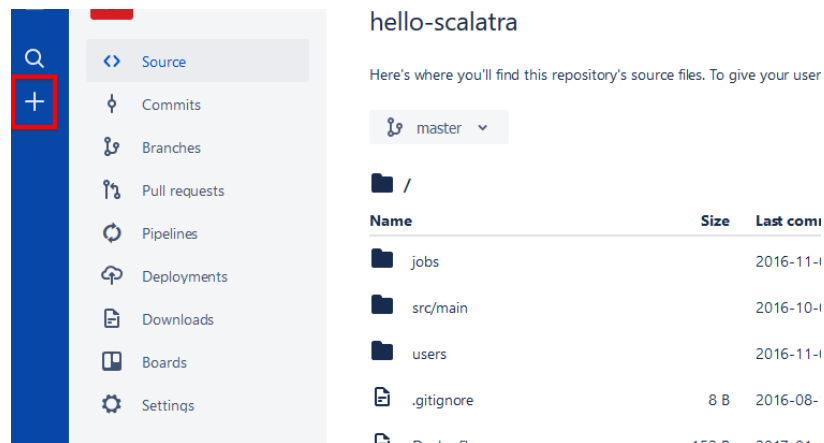
Once you have an account, sign into Bitbucket and navigate to the hello-scalatra project in a **new** browser tab.

<https://bitbucket.org/pmwtraining/hello-scalatra>

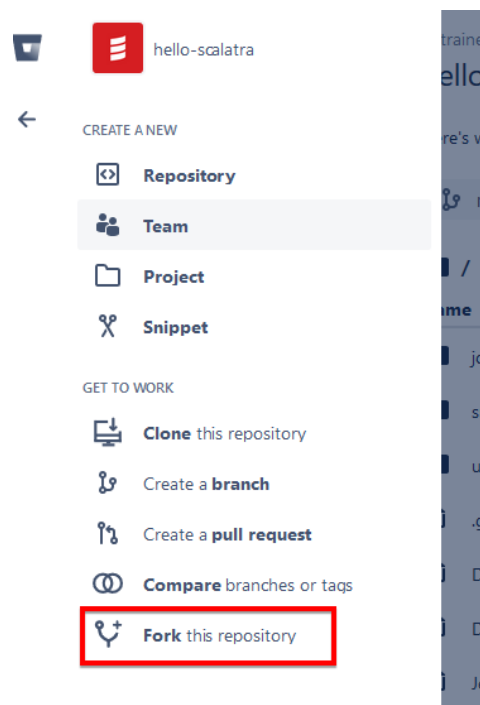
This is a simple hello-world project written in Scala.



Click on the + symbol.



Fork this repository to create a copy in your account.



Leave the defaults as they are.

Fork pmwtraining / hello-scalatra

Workspace

Peter McNaughton

▼

Project*

Untitled project

▼

Name*

hello-scalatra

Access level

☐ Private repository

> Advanced settings

Fork repository

Cancel

Clone the hello-scalatra project to your **Maven EC2 Machine and build it**

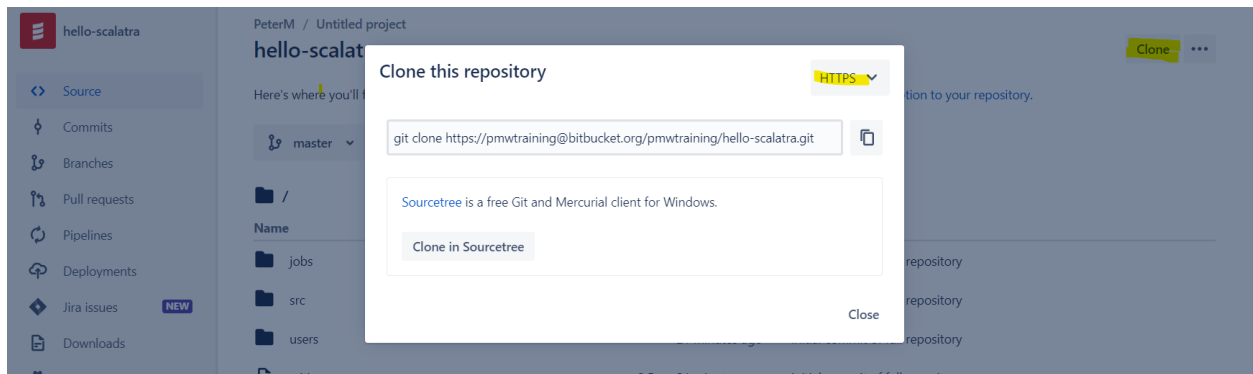
Clone your project with the following:

```
#install git - if not already installed
$ sudo yum install git -y
#clone the project
$ cd ~
$ git clone https://username@bitbucket.org/username/projectname
# You can find your repository URL on the bitbucket website on
the top right-hand side of the page. [Clone] It will be of the
form:
```

<https://username@bitbucket.org/username/projectname>

For example:

<https://pmwtraining@bitbucket.org/pmwtraining/hello-scalatra>



Now we have a local copy of the repository, we can build and run the project with the following:

```
$ cd hello-scalatra #this will be the directory created when you
                        #cloned your project
$ mvn package        #builds the project
$ mvn jetty:run       #runs the project using the jetty webserver
```

Check you see BUILD SUCCESS in the terminal window:

```
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-interpolation/1.15/plexus-interpolation-1.15.jar (60 KB at 1554.1 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/xpp3/xpp3_min/1.1.4c/xpp3_min-1.1.4c.jar (25 KB at 658.7 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-filtering/1.0-beta-2/maven-filtering-1.0-beta-2.jar (33 KB at 790.3 KB/sec)
Downloaded: https://repo.maven.apache.org/maven2/com/thoughtworks/xstream/xstream/1.3.1/xstream-1.3.1.jar (422 KB at 7659.9 KB/sec)
[INFO] Packaging webapp
[INFO] Assembling webapp [hello-scalatra-new] in [/home/ec2-user/hello-scalatra/target/hello-scalatra]
[INFO] Processing war project
[INFO] Copying webapp resources [/home/ec2-user/hello-scalatra/src/main/webapp]
[INFO] Webapp assembled in [174 msecs]
[INFO] Building war: /home/ec2-user/hello-scalatra/target/hello-scalatra.war
[INFO] WEB-INF/web.xml already added, skipping
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 04:14 min
[INFO] Finished at: 2018-05-03T23:39:03+00:00
[INFO] Final Memory: 15M/40M
[INFO] -----
[ec2-user@ip-10-0-1-51 hello-scalatra]$
```

To see if your project is running point a browser at:
http://[maven_ip]:8080/

You should see:



Hello, World!

Stop the scalatra project running by pressing Ctrl+C in the maven ec2 terminal