Guide to setting up Bitbucket and Heroku for the project

Vittorio Zaccaria

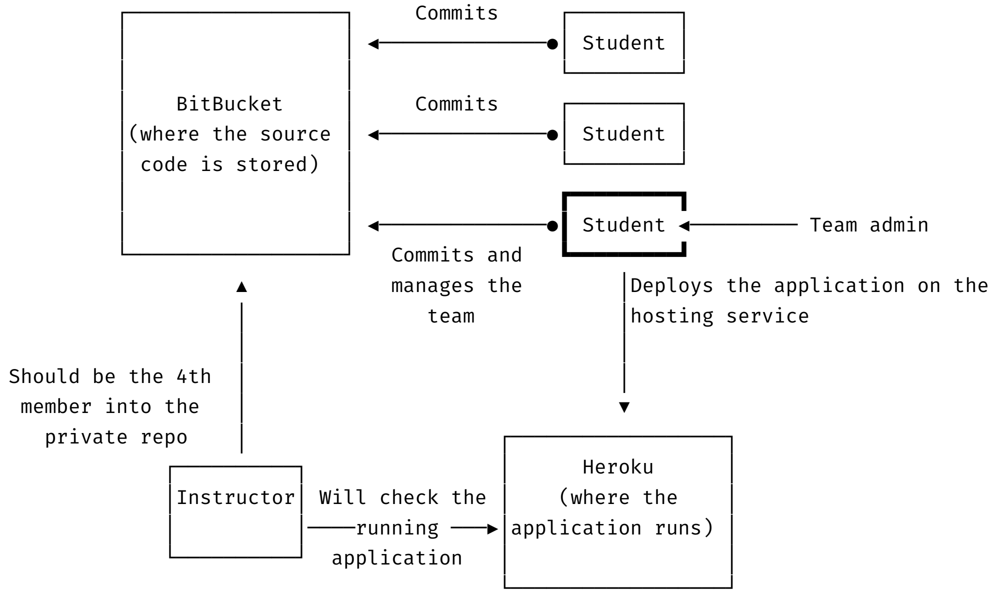
February 25, 2019

Contents

1 *Introduction*

Every project should host its code on a ***private*** Bitbucket Git repository and provide a running application on the Heroku hosting platform. As far as this guide is concerned, we will distinguish the following actors:

- Instructor (V. Zaccaria/M. Gelsomini)
- Team (composed of max. 3 students, where one of them will be elected as *team administrator*)



All the team members are required to commit and push their work to the common team repository under their individual name (read on to see how to create an account on Bitbucket), so that instructors will be able to discern the actual contribution ratio for each of them. The team manager (one of the three students, perhaps the most tinkerer) is responsible for

- management of the Bitbucket repository (e.g., tagging the final release, team member management and so on..), and
- pushing the application into production on the hosting service (Heroku).

Before beginning with the practical steps of this guide (a part from creating an account on BitBucket), we strongly urge you to ***take acquaintance with git*** by reading thoroughly the following links:

- Learn Git with Bitbucket Cloud
- Beginner / What is version control
- Beginner / What is git
- Collaborating / Syncing. Learn the basics of git ***remote***, ***fetch***, ***pull*** and ***push***.
- Collaborating / Centralized workflow. Among the available ones, we suggest this workflow for your team as it is very simple.

1.1 Using Git on your machine

The suggested centralized workflow dictates every developer to have his own local copy of the entire project. To work locally on the project you could use either the command line `git` (every OS has its own way of installing it), or a graphical client. For example, Bitbucket's company has its own free application for Mac and Windows (`https://www.sourcetreeapp.com/`), feel free to check it out!.

1.2 IMPORTANT What should I commit over to git?

When working with a Node.js based project, a common mistake done by everyone is to include in the repository the `node_modules` directory. This directory ***should not be stored in git***! This post should clarify how to avoid this mistake.

2 Setting up Bitbucket

2.1 Introduction

In the following sections, we will guide you to create your own account on Bitbucket and, if you are the lucky (!) team administrator we will instruct you on how to setup the team's git repository containing the project. Before submitting the repository to the instructors, you will have to tag its state as FINAL and add one of us (instructors) as an additional team member so that we can inspect your code and the contributions of each member of the team.

2.1.1 What goes into a repository

The repository will contain the code both for the backend and the frontend. In fact, when deployed to Heroku, the project should run as it is without further intervention. You will have some boilerplate code as an initial starting point when initializing the repository. Keep reading on.

2.2 Creating an account

1. Go to Bitbucket and create a free account with your ***institutional mail address*** (the one finishing with `polimi.it`)
2. When asked for creating the username, use the following ***mandatory*** template (mind the dashes '-'):

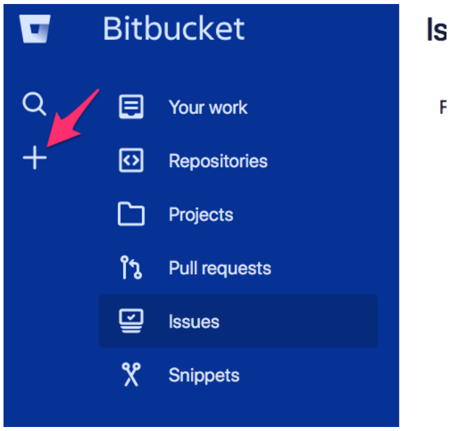
```
polimi-hyp-2018-<your codice persona>
```

Now, if you are not your team's administrator you can stop here and start to study some git fundamentals. You will be added to the team by the team administrator. However, if you are the team administrator, please read the following two sections (Creating a team and creating the project repository).

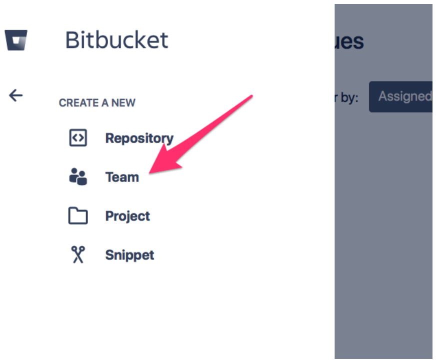
2.3 ADMIN ONLY *Creating a team*

This guide is meant for the team administrator. Here you will find instructions on how to setup a team and create the project repository with the correct naming conventions.

1. Once logged in click on the plus button:



2. Then create a team



by using the following ***mandatory*** template as 'Team name'. Mind the dashes!

```
polimi-hyp-2018-team-<your codice persona>
```

and add your other team members.

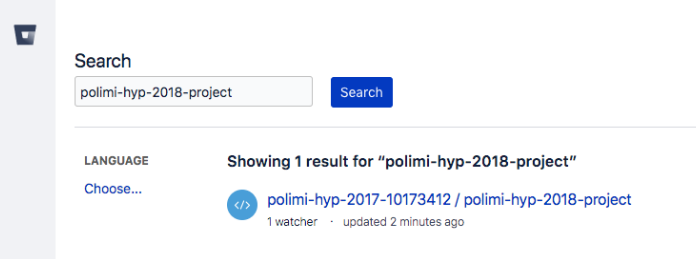
2.4 ADMIN ONLY *Creating the project repository*

This section is meant for the team administrator. Here you will find instructions on how to setup the repository of the project, by forking (copying) an existing one.

1. Go to the bitbucket repository search:

`https://bitbucket.org/repo/all`

search for the `polimi-hyp-2017-10173412 / polimi-hyp-2018-project` (***instructor's repository***)



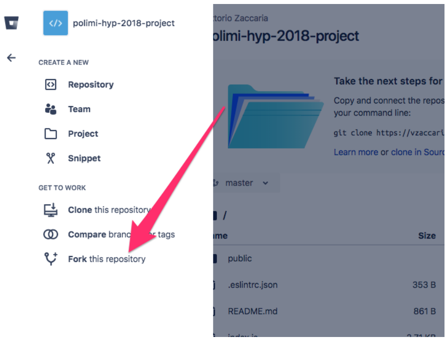
Showing 1 result for "polimi-hyp-2018-project"

polimi-hyp-2017-10173412 / polimi-hyp-2018-project

1 watcher · updated 2 minutes ago

and click on it.

2. Fork it (which is a fancy way of saying "copy it") by clicking the plus button and "Fork this repository" on the left sidebar



3. When asked, assign ***your team*** as owner of the fork; then assign the name and project name to your repo and ***check "this is a private repository"*** (no public repository will be accepted for the project). Finally, click fork this repository.

3 *ADMIN ONLY* *Setting up Heroku*

3.1 *What is Heroku*

Heroku is a *platform as a service* (PaaS) that will allow you to host your running application. Once started on Heroku, your application will be accessible globally with a unique URL. Heroku supports Node.js natively and you can deploy your app by simply connecting your Bitbucket repo (the one we created in the previous sections).

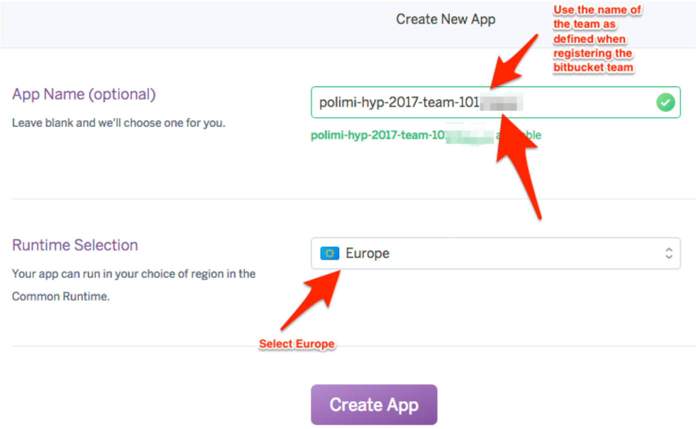
This guide is only for the team administrator. He will be in charge of creating an instance of the app on Heroku and pushing regularly updates from the current state of the repo.

3.2 *Register on Heroku and create an app (just once at the beginning)*

1. Register to Heroku and create an app (to register use your (team administrator) email).

2. Login and click create new app

3. Name your app by using the team name



Create New App

App Name (optional)

Leave blank and we'll choose one for you.

polimi-hyp-2017-team-101[redacted]

polimi-hyp-2017-team-10[redacted] a…ble

Runtime Selection

Your app can run in your choice of region in the Common Runtime.

Europe

Use the name of the team as defined when registering the bitbucket team

Select Europe

3.3 *Connecting the Bitbucket repository to Heroku*

1. Download the 'heroku' CLI program (major platforms supported) and login:

```
» heroku login
  Enter your Heroku credentials.
  Email: ***********
  Password (typing will be hidden):
  Logged in as ***********
```

2. Go to the directory where you store locally your Bitbucket repository

```
» cd <myprojectdirectory>
```

3. Connect your repo to heroku

```
» heroku git:remote -a polimi-hyp-2018-team-101****
```

3.4 *Adding a SQL database*

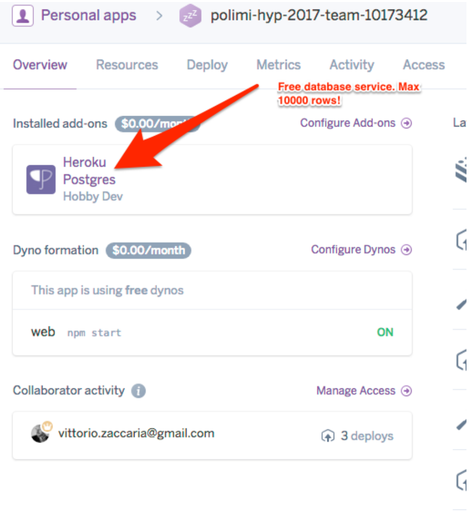
For our web application, we need to setup a SQL database on Heroku. We are going to tell Heroku to use Postgres directly from the command line:

```
heroku addons:create heroku-postgresql:hobby-dev
```

where `hobby-dev` means we are just using the free version of ***Postgres*** which is limited to 10,000 rows). You should see something like:

```
Database has been created and is available...
```

If you go to the Heroku panel, you will see the new add-on running; clicking on it will bring you to the page where you can administer your database (e.g. for deleting it, if you need).



Free database service. Max 10000 rows!

Installed add-ons $0.00/mo

Heroku Postgres
Hobby Dev

Dyno formation $0.00/month

This app is using free dynos

web `npm start` ON

Collaborator activity

vittorio.zaccaria@gmail.com 3 deploys

3.5 *Deploy your app!*

Every time the administrator repo contains the latest commits and is ready to be deployed, the administrator can push it by typing:

```
» cd <myprojectdirectory>
» git push heroku master
```

The app will restart automatically; you will see messages in the terminal telling you the address of your app:

```
...
https://polimi-hyp-2018-team-****.herokuapp.com/ deployed to Heroku
...
```

Click to that link and check out your shiny new app!

3.6 *Can I use Postgres locally?*

If you want to use ***Postgres*** also locally, you definitely can; however you should install on your machine the whole ***Postgres*** package which I definitely not recommend since it implies the installation of a service running in background. Just use SQLite locally and ***Postgres*** on Heroku since, as you will see, our application will allow you to switch between the two without changing much of the code.