

Making the Green Electronics Repo Your Own

Updated 10.10.16

This tutorial will show how to clone the green electronics repository hosted on Github and copy it into a private repository that you can manage and share with others. Following this tutorial will also allow you to pull patches in the future from the original green electronics Github repo.

READ THIS FIRST!!

One member of the team will have to create a private repository on Bitbucket using the steps below. All the other team members should also create Bitbucket accounts and skip to the end of the tutorial to remap their Git remotes to the private team repository after it has been created.

Clone the Repo

The first step is to clone the git repo from Github.

```
git clone https://github.com/ndanyliw/green-electronics.git
```

(If you have already cloned the repo, make sure your changes are committed and pull the latest code).

You now will have a local copy of the git repository on the machine which means you can commit code locally and pull from the upstream source. However you do not have push privileges to the Github repository so you cannot push changes to a remote target. If you just wanted to use git locally you could stop here. But, you most likely will want to host a private repository so you can collaborate with teammates on your code. To do this we recommend using [Bitbucket](#).

Creating a Private Repository

For the purpose of this tutorial, we will assume you will use Bitbucket to host your private repo. If this is not the case, you will need to modify some of the commands used. Only one teammate should do this step and everyone else should clone their private repository.

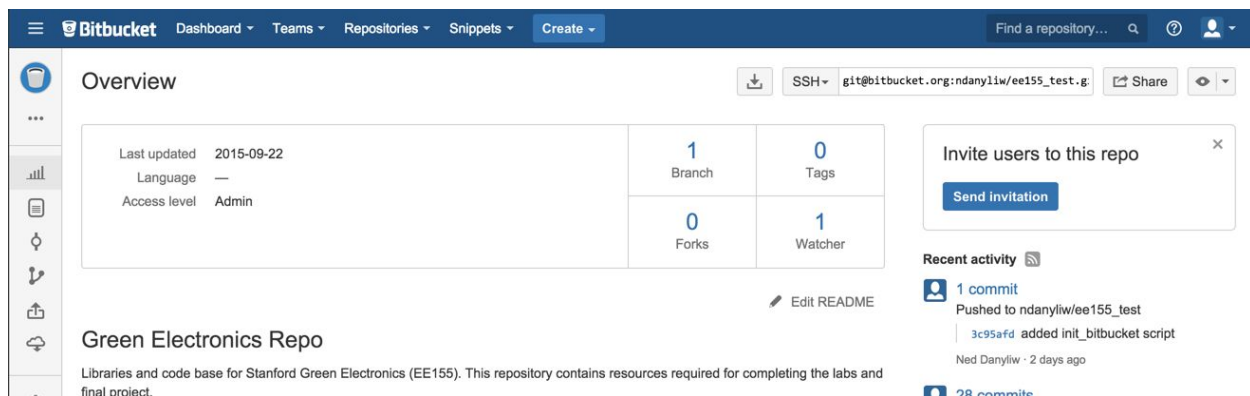
First make sure you have created a Bitbucket account. After doing so, just run the "init_bitbucket" script which will prompt you for your username and password and then work its git magic.

```
./init_bitbucket.sh
```

After running the script, you will have created a private repository on Bitbucket hosting your code and mapped that as your “origin” (where your default git push and pull go). To get the latest code from the class Github use `git pull upstream master`. It is recommended to do this at minimum before starting each lab as bugfixes and starter code will be added throughout the quarter.

Adding Collaborators

The last bit after creating the private repository is adding collaborators to your project who have access to it. To do this navigate to your repository on Bitbucket.

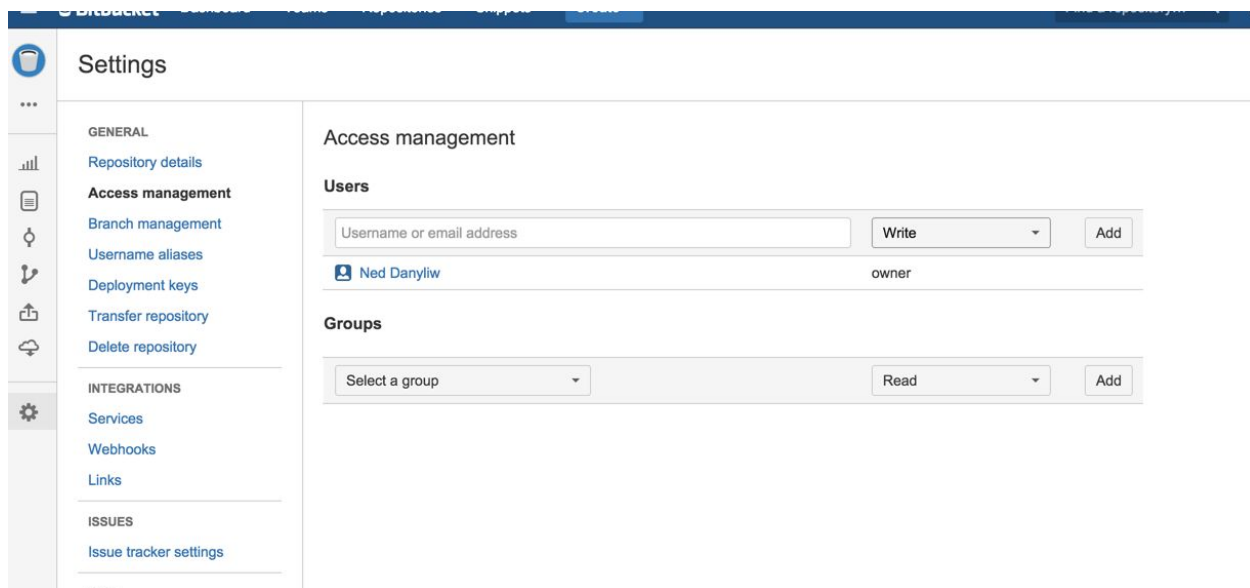


The screenshot shows the Bitbucket interface for a repository named "Green Electronics Repo". The top navigation bar includes "Dashboard", "Teams", "Repositories", "Snippets", and "Create". The repository overview displays the following information:

Last updated	2015-09-22	1	0
Language	—	Branch	Tags
Access level	Admin	0	1
		Forks	Watcher

Additional details include an "SSH" link with the URL `git@bitbucket.org:ndanyliw/ee155_test.g`, a "Share" button, and an "Invite users to this repo" dialog box with a "Send invitation" button. The "Recent activity" section shows 1 commit: "Pushed to ndanyliw/ee155_test" by user 3c95afd, adding an `init_bitbucket` script, 2 days ago.

Click the “Share” button on the upper left and then “Manage this repository” on the dialog box that appears. From this menu you can add your teammates and adjust their access privileges.



The screenshot shows the "Settings" page for the repository, with the "Access management" section selected. The left sidebar lists various settings categories: GENERAL (Repository details, Access management, Branch management, Username aliases, Deployment keys, Transfer repository, Delete repository), INTEGRATIONS (Services, Webhooks, Links), and ISSUES (Issue tracker settings).

The "Access management" section is titled "Access management" and contains two sub-sections:

- Users:** A form to add users with a "Username or email address" input field, a "Write" dropdown menu, and an "Add" button. Below this, a user named "Ned Danyliw" is listed with the role "owner".
- Groups:** A form to add groups with a "Select a group" dropdown menu, a "Read" dropdown menu, and an "Add" button.

Your teammates should now be able to clone this repository.

Remapping to Group Repository

All your teammates will now need to remap their Git remote repositories to point to the group repository by default and still track the shared class repository. Each teammate will need to run the following commands from the “green-electronics” repository:

```
git remote add upstream https://github.com/ndanyliw/green-electronics.git
```

```
git remote set-url origin <url of team bitbucket>
```

This will now allow everyone to pull and push from the team repository (assuming they have been added as collaborators) and pull from the class repository.

To pull updates from the class repository:

```
git pull upstream master
```

To pull updates from the team repository

```
git pull origin master
```

To push updates to the team repository

```
git push origin master
```