

1. Set Your Global Git Configuration
2. Fork ACCRE Repository
3. Clone Fork to the ACCRE Cluster (or Your Personal Machine)
4. Submit Pull Request
5. Need Help?

Contributing code to ACCRE's Github page is a great way to foster collaboration in HPC research at Vanderbilt. Below are instructions for how to go about contributing code to any of ACCRE's Github repositories . Before getting started, you will need to create your own Github account (if you don't have one already). While you're at it, follow ACCRE's Github account so you can see when new content has been added. Also feel free to star any ACCRE repositories you find to be especially helpful!

1 Set Your Global Git Configuration

Once you have logged into the cluster, you will want to use the most recent version of git available through `setpkgs` :

```
[bob@vmeps13 ~]$ setpkgs -a git
[bob@vmeps13 ~]$ git --version
git version 1.8.3-rc3
```

You may want to stick the `setpkgs` command in your `.bashrc` so that you always pick up this version of git and not the system version (which is an older version). Next, set up your git global config with your Github user name and the email address linked to your Github account. For example:

```
[bob@vmeps13 ~]$ git config --global --add user.name "bob"
[bob@vmeps13 ~]$ git config --global --add user.email "bob@vanderbilt.edu"
[bob@vmeps13 ~]$ git config --global --list
user.name=bob
user.email=bob@vanderbilt.edu
```

This information will be stored in a file in your home directory (`~/.gitconfig`). Now your Github account will be listed as a contributor if you contribute code to an ACCRE git repository!

2 Fork ACCRE Repository

The next step is to download a copy of the appropriate ACCRE repository to your own Github account. This is called *forking* a repository. As an example, let's say you want to contribute a Python script to ACCRE's Python repository. To do this, first make sure you are signed into your Github account, then

navigate to the Python repository at ACCRE's Github page. Note that on ACCRE's homepage you will not see all repositories. You must click **Repositories** at the top of the page to get a full list.

At the top right of the page you will see a Fork button; simply click this button, and the repository will be forked to your Github profile. Navigate to your Github profile to convince yourself that you've successfully forked the repository.

3 Clone Fork to the ACCRE Cluster (or Your Personal Machine)

Now you need to download a copy of the repository to your ACCRE home directory or personal machine so you can edit your local copy of the repository. This is called **cloning**. When you click on your forked version of the repository in Github, you should see a box with **HTTPS clone URL** listed above the box. This box contains the URL of the repository, which you should copy and paste into your command line after typing `git clone`. For example:

```
[bob@vmeps13 ~]$ git clone https://github.com/bob/Python.git
Cloning into 'Python'...
remote: Counting objects: 21, done.
remote: Total 21 (delta 0), reused 0 (delta 0), pack-reused 21
Unpacking objects: 100% (21/21), done.
[bob@vmeps13 ~]$ ls -p | grep Python
Python/
```

Notice that user bob now has a new directory called **Python**, which is his local copy of the repository that he can edit without affecting the remote version of the repository stored at ACCRE's Github page.

Now you are ready to make changes to the repository. We suggest creating a new subdirectory in the top-level directory of the repository, with the name of the subdirectory indicating the type of code/example you are contributing. For example, if user bob has an example of doing clustering analysis within Python, he might name the subdirectory **clustering**. In general, user bob would then add files (Python script(s), SLURM script(s), any other subdirectories or miscellaneous files) inside of this subdirectory. You may also choose to create a **README** file in the new directory you've created, which will be displayed in Github when a user clicks on the example. We ask that you limit the total size of each example to < 100 MB.

Once you have added all your files and have thoroughly tested them to ensure their proper function (and left helpful documentation in the source code!), you are ready to commit these changes to your local repository. Below is an example of how to commit changes. This sequence of commands should work in most cases (note you must be inside the repository in order to execute these commands):

```

[ bob@vmpps13 Python]$ ls -p
clustering/ matplotlib/ README.md vectorization/
[ bob@vmpps13 Python]$ ls -p clustering/
clustering.py clustering.slurm README
[ bob@vmpps13 Python]$ git status
# On branch master
# Untracked files:
#   (use "git add ..." to include in what will be committed)
#
#   clustering/
nothing added to commit but untracked files present (use "git add" to track)
[ bob@vmpps13 Python]$ git add -A
[ bob@vmpps13 Python]$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD ..." to unstage)
#
#   new file:   clustering/README
#   new file:   clustering/clustering.py
#   new file:   clustering/clustering.slurm
#
[ bob@vmpps13 Python]$ git commit -m "Adding clustering example"
[ master a5249f0] Adding clustering example
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 clustering/README
create mode 100644 clustering/clustering.py
create mode 100644 clustering/clustering.slurm
[ bob@vmpps13 Python]$ git status
# On branch master
# Your branch is ahead of 'origin/master' by 1 commit.
#   (use "git push" to publish your local commits)
#
nothing to commit, working directory clean

```

Three important git commands are used here. `git status` gives you details about files that have and have not been committed to your local repository, `git add -A` will add files and any edits you've made to the local repository, and `git commit` will actually apply these changes to your local repository to make them permanent (the `-m` option allows you to add a message to your commit, which should be a brief description of the changes you've made to the repository).

Once you have committed all your changes, the last step is to push these changes to Github using `git push` :

```

[ bob@vmpps13 Python]$ git push
Username for 'https://github.com': bob
Password for 'https://bob@github.com':

```

```
Counting objects: 7, done.  
Delta compression using up to 16 threads.  
Compressing objects: 100% (4/4), done.  
Writing objects: 100% (4/4), 454 bytes | 0 bytes/s, done.  
Total 4 (delta 2), reused 0 (delta 0)  
To https://github.com/bob/Python.git  
    e5045fe..3fe0fa4  master -> master
```

Note that you will be prompted for your username and password. This is your Github username and password, NOT your ACCRE username and password. After you have completed this last step, you should now see these changes reflected when you view your fork (but not ACCRE's version of the repository) of the repository in Github.

4 Submit Pull Request

From Github, navigate to your fork of the repository, and click on **Pull Request**. You will be taken to a few screens where you can leave comments about the code you are contributing. Feel free to leave comments and confirm the pull request until it's clear that it has been submitted to the ACCRE admins for review. At this point, ACCRE admins will review the code you have contributed and have the option to accept or reject the changes. If all looks well, we will accept the changes and they will be included in the repository for other users to see and/or borrow!

5 Need Help?

Open a helpdesk ticket. We're happy to help!