

# Workspace Configuration

## TABLE OF CONTENTS

- 1 [Git name and email](#)
- 2 [SSH keys](#)
- 3 [Repositories](#)

After you connect to the Workspace with SSH, you should set up Git for your repositories.

## Git name and email

Run these commands to set up your Name and Email that will be used for your Git commits. Make sure to replace "Your Name" and "your\_email@berkeley.edu" with your name and email.

```
git config --global user.name "Your Name"
git config --global user.email "your_email@berkeley.edu"
```

Additionally, configure Git to merge when pulling from `staff`:

```
git config pull.rebase false
```

*Note: Please make sure that you are inside the cloned repo to run the command above. If you want to run the command above from the root, and make it affect all of the repos in the machine please use*

```
git config --global pull.rebase false
```

## SSH keys

You will need to setup your SSH keys in order to authenticate with GitHub from your Workspace.

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

will generate an SSH key. Make sure to replace `your_email@example.com` with your own email.

Add the output of

```
cat ~/.ssh/id_ed25519.pub
```

to your Github account by following the instructions on [this page](#).

## Repositories

You will have access to two private repositories in this course: a personal repository for homework, and a group repository for projects. We will publish skeleton code for homeworks in [Berkeley-CS162/student0](#) and we will publish skeleton code for group projects in [Berkeley-CS162/group0](#). These two skeleton code repositories are already checked out in the home folder of your Workspace, inside `~/code/personal` and `~/code/group`.

You will use the “Remotes” feature of Git to pull code from our skeleton repos (when we release new skeleton code) and push code to your personal and group repos (when you submit code). Your working files will be stored within the Workspace. Back them up by pushing to your GitHub repo. Save your work early and often. Breaking up the implementation of a large feature into several, small, clear commits is good practice, and will make it easier to determine where in your codebases’ history a bug was introduced. Communication with course staff will often involve looking at the code and commits in your repo.

The Git Remotes feature allows you to link GitHub repositories to your local Git repository. We have already set up a remote called “staff” that points to our skeleton code repos on GitHub, for both your personal and group repo. You will now add your own remote that points to your private repo so that you can submit code.

You should have received the link to your personal private GitHub repo when you registered with the autograder earlier. Add a new remote by doing the following steps in your Workspace:

- 1 First, switch into your personal repository.

```
cd ~/code/personal
```

- 2 Then visit your personal repo on GitHub and find the SSH clone URL. It should have the form “git@github.com:Berkeley-CS162/...”
- 3 Now add the remote

```
git remote add personal YOUR_GITHUB_CLONE_URL
```

4 You can get information about the remote you just added

```
git remote -v  
git remote show personal
```

5 Pull the skeleton, make a test commit and push to `personal main`

```
git pull staff main  
touch test_file  
git add test_file  
git commit -m "Added a test file."  
git push -u personal main
```

*Note: You might encounter an issue when trying to run the `git push` command that states `Unprotected Private Key File`. If this is the case, you will have to update the permissions on the workspace. You can do so by running the commands below.*

```
chmod 0700 ~  
chmod 0700 ~/.ssh  
chmod 0600 ~/.ssh/authorized_keys
```

In this course, “main” is the default Git branch that you will use to push code to the autograder. You can create and use other branches, but only the code on your “main” branch will be graded. You should do this test commit as soon as possible – we want to know that everyone has this basic infrastructure in place.

- 1 Within 30 minutes, you should receive an email from the autograder. (If not, please notify the instructors via Ed). Check [cs162.org/autograder](https://cs162.org/autograder) for more information.