

Introduction to Git — Fall 2022

Lecture 0: Setup



Slides: <https://hackmd.io/@git-fall-2022/L0-setup#/>

Installing and setting up Git

We will use Git from the command line in this course. This is how you will use it on HPC2N's systems, and this way it will also be easier to understand what is going on while you are learning to use Git.

Graphical tools exists for Git, see below list for a few. All entries on the list are free and unless otherwise mentioned for Windows, macOS, and Linux:

- git-scm (<https://git-scm.com>) comes with a basic GUI
- Git Kraken (<https://www.gitkraken.com/>)
- Github Desktop (<https://desktop.github.com/>) Windows and macOS only
- Sourcetree (<https://www.sourcetreeapp.com/>) Windows and macOS only
- TortoiseGit (<https://tortoisegit.org/>) Windows only

Installing and setting up Git - continued

- Install Git, if you have not already
- Create a repository with `git init`
- Set your name and email with `git config` (local, global, system)
- Test by creating a file
- Then adding the file with `git add`
- Then committing the file with `git commit`

When this is done, you will clone the course materials.

Git install - Windows

- Go to the Git-scm website (<https://git-scm.com/downloads>) and click “Windows” to download the Windows version. It will automatically start download of the .exe file.
- The downloaded file can be installed by double-clicking and choosing “Run”.
- Click “Yes” to let it be installed and then “Next” to accept the GNU GPL.
- The default options you are presented with should work, and we recommend using those.
- NOTE: when it comes to choosing the default editor, we recommend using Vim, unless you have a preferred editor.

Git install - macOS

If you have installed XCode (or its Command Line Tools), Git may already be installed. To find out, open a terminal and enter `git --version`.

If Git is not installed, you have several installation options. Apple maintains their own fork of Git, but it is usually a few versions behind, so we do not recommend installing that.

- SourceForge: <https://sourceforge.net/projects/git-osx-installer/files/>
- [Git-scm.com](https://git-scm.com/downloads): <https://git-scm.com/downloads>
- If you have Homebrew: `brew install git`

Git install - Linux

Installing Git on Linux depends on which distro you are running.

- `sudo apt-get install git` (Ubuntu, Debian)
- `sudo dnf install git` (RHEL, CentOS)
- **<https://git-scm.com/download/linux>** (other)

Configure git

First check that you have git installed:

```
$ git --version
```

Now configure git with

- `git config (local, global, system)`

You should at least set your global name and email (just once):

```
$ git config --global user.name "Your Name"  
$ git config --global user.email "yourname@example.com"
```

Configure git - continued

Setting the editor (once) is also a good idea:

```
$ git config --global core.editor <editor>
```

Choices for editor could be:

- nano
- vim
- emacs

Test your Git installation

Create an example folder and change to that, then create a file test.txt. On Linux you would do this:

```
$ mkdir <mydir>
$ cd <mydir>
$ touch test.txt
```

Now initialize a repository and stage the new file:

```
$ git init
Initialized empty Git repository in /home/bbrydsoe/test-
git/.git/
$ git add test.txt
```

Now *commit* the change. The editor which you configured earlier should open. Add an example commit message:

```
$ git commit test.txt  
[master (root-commit) ff8b6f6] Test of git  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 test.txt
```

Test your Git installation - continued

Now let us look at the log:

```
$ git log
commit ff8b6f699d98c72d5cffc64d65a1c618b976b45a (HEAD ->
master)
Author: Birgitte Brydsö <bbrydsoe@cs.umu.se>
Date:   Thu Sep 17 13:53:59 2020 +0200

    Test of git
```

When you do `git log`, you should see something like the above, but with name, email, and commit message different. If that is the case, your Git should be configured correctly.

Download the course materials

For the individual hands-on part of the course, we have created some course materials which you will download from the course GitHub:

<https://github.com/hpc2n/course-intro-git> (normally you click the green “Code” button to get the link to clone or download)

- Please go to the terminal window where you have downloaded and set up Git.
- Change the directory to wherever you wish to have the course material.
- *Do one of:*
 - 1. Download the zipfile (directly with wget or elsewhere then transfer) and unzip
 - 2. `git clone https://github.com/hpc2n/course-intro-git.git`

Web based Git repositories

There are several web based Git repository. Some of the more popular ones are:

- GitHub (<https://github.com/>)
- GitLab (<https://www.gitlab.com>)
- Bitbucket (<https://bitbucket.org>)
- SourceForge (<https://sourceforge.net/>)

We are going to use GitHub for the part of the hands-on where you will be working together in groups.

Please go to

- <https://github.com/>

and sign up for an account.

Create a new SSH key for GitHub - Linux and macOS

This part will be done during the exercises on day 5, but you can create and add your SSH key to GitHub now if you want to.

1. Open a terminal. In the command below, “GitHub” is a label added to the key for clarity. You can add any you want:
 - a. Do this

```
$ ssh-keygen -t ed25519 -C "GitHub"
```

- b. If you have an older system, this may work better

```
$ ssh-keygen -t rsa -b 4096 -C "GitHub"
```

2. You will be asked for a file to save the key. Unless you have an existing SSH key, accept the default.
3. Enter a passphrase and repeat it.
4. Add the key to the ssh-agent. Here we assume the default name for the legacy systems - change to what your key was called:

```
$ eval "$(ssh-agent -s)"  
$ ssh-add ~/.ssh/id_rsa
```

5. Switch to the `.ssh` folder, open the file `id_rsa.pub` and copy it.
Do NOT add any newlines or whitespace!

Create a new SSH key for GitHub - Windows

This part will be done during the exercises on day 5, but you can create and add your SSH key to GitHub now if you want to.

1. Open Git Bash. In the command below, “GitHub” is a label added to the key for clarity. You can add any you want:
 - a. Do this

```
$ ssh-keygen -t ed25519 -C "GitHub"
```

- b. If you have an older system, this may work better

```
$ ssh-keygen -t rsa -b 4096 -C "GitHub"
```

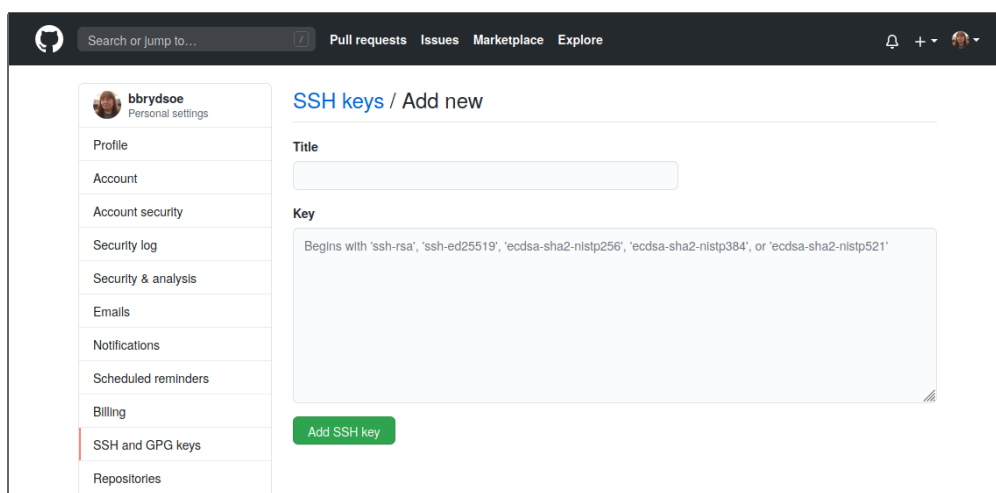
2. You will be asked for a file to save the key. Unless you have an existing SSH key, accept the default.
3. Enter a passphrase and repeat it.
4. Add the key to the ssh-agent. Here we assume the default name for the legacy systems - change to what your key was called:


```
$ eval "$(ssh-agent -s)"  
$ ssh-add ~/.ssh/id_rsa
```

5. Switch to the `.ssh` folder, open the file `id_rsa.pub` and copy it.
Do NOT add any newlines or whitespace!

Adding the SSH key to GitHub

1. On GitHub, click your avatar in the top right corner and pick “Settings”.
2. Choose “SSH and GPG keys”
3. Click “Add new SSH key”
4. Add a descriptive label for the key in the “Title” field. In the key field you paste the content of the key (`~/.ssh/id_rsa.pub`)



5. Click “Add SSH key”
6. Confirm your GitHub password if you are prompted for it.

Testing the SSH keys

1. Open a terminal / the Git bash
2. `$ ssh -T git@github.com`
3. It will look similar to this:

```
$ ssh -T git@github.com
The authenticity of host 'github.com (140.82.121.4)' can't
be established.
RSA key fingerprint is
SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,140.82.121.4' (RSA)
to the list of known hosts.
Enter passphrase for key '/home/bbrydsoe/.ssh/id_rsa':
Hi bbrydsoe! You've successfully authenticated, but GitHub
does not provide shell access.
```

4. Verify that the resulting message contains your username.

Editor, Linux

Vim

- You may need to install it first. (`sudo apt-get install vim`)
- Start with `vim <filename>` to open a file for editing. The file will be created if it does not exist before.
- Type `i` to enter 'insert' mode to be able to write in the editor.
- Use ESC to go to 'command' mode and then `:wq` to save and exit the editor.
- When you are in 'command' mode, typing `dd` will delete the whole line your cursor is on.

Nano

- Start with `nano <filename>` to open a file for editing. The file will be created if it does not exist before.
- Ctrl-x will exit the editor, asking first if you want to save the file.

CHANGED 6 DAYS AGO

OWNED THIS NOTE

43 views  