Overview of Version Control Systems and Git

Imagine you are a member of a team that is developing an app and working from different locations within India, the Americas, and Europe. You have been tasked with adding a new feature to the app, while another developer in Europe needs to fix a bug that is not related to the new feature you are working on. Yet another developer in the US could be working on something entirely different. The source code that the three of you need to work on, however, is the same. How do you ensure that all developers work on the code simultaneously and that the source code gets updated with all the changes made by each developer? How do you get access to all the versions with changes clearly identified across versions, along with the reasons why the changes were made? In short, how do you ensure you do not end up creating folders called 'Code-working copy', 'Code-latest copy', and 'Code-final final with bug fixes'?

The simple and short answer is: by leveraging version control software.

Objectives:

After completing this module, you'll be able to:

- Explain why using a version control system is essential to any development endeavor, especially in a team environment.
- Identify the advantages of a good version control system.
- Describe Git and GitHub as version control tools of choice for full-stack development.

Software Development and Version Control

As you might have rightly guessed, your efficiency as a developer will depend a lot on your ability to use version control software. More importantly, it will also depend on what kind of version control software you use. Since most of us today work in a distributed team environment, you'll probably be using a distributed version control system, such as Git and Mercurial, rather than a centralized one, such as CVS or Perforce.

Benefits of Using a Version Control System

1. **Complete change history of all files**: A good version control software can track changes made by multiple users over the years. The history can include the user who made the changes, the date of the change, and the written notes about why a change was made. Tracking complete history helps go back to previous versions for analysis of causes of bugs, especially in older versions of the software.
2. **Branching and merging**: There's much more than multiple users working concurrently that a good version control software can provide: individual users can work on their own

files known as ‘branches’, which can then be merged seamlessly into the distributed source code.

3. **Annotated history for better traceability**: A good version control system will also provide the ability to annotate every change with a message that summarizes the purpose of the change. This allows for tracing all changes and connecting them to project management and bug-tracking software. An annotated history also helps make harmonious changes that are aligned with the overall vision and design of a project.

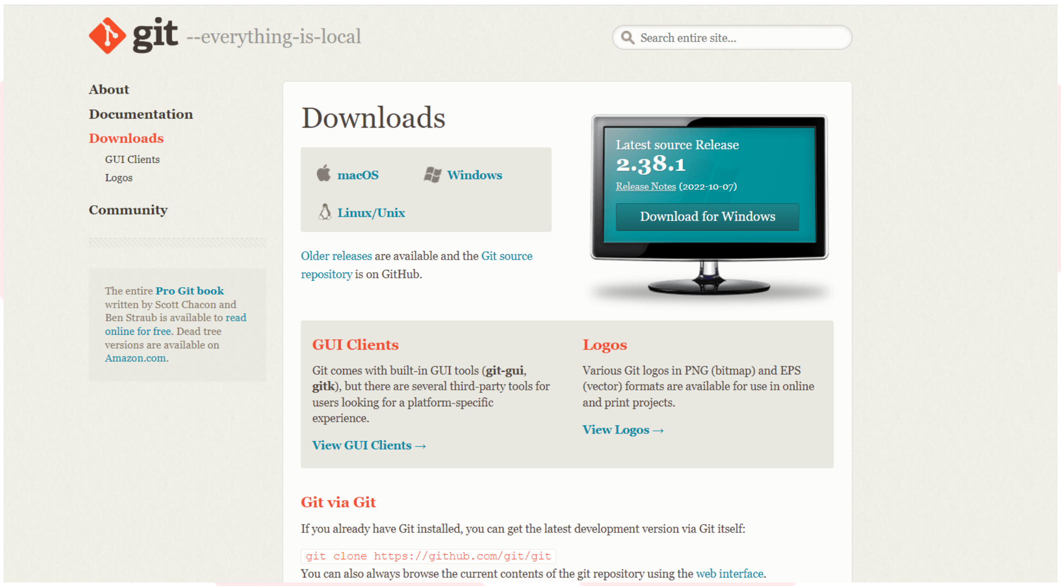
About Git and GitHub

Git is the version control system you’ll be using extensively during this program and at the workplace. It is a free and open-source, distributed version control system under the GNU General Public License version 2. Because it offers most of the sought-after features that make development in distributed teams a breeze, the ability to leverage Git is one of the most sought-after skills for a full-stack developer.

GitHub is a cloud-based service that allows managing Git repositories easily. What it means is that if you use Git as a version control system, then using GitHub will help you manage your Git repositories easily. Note that Git and GitHub are completely independent of each other, and it’s optional to use them together. However, using both for your development projects helps manage your development effort better.

Git Installation Guide

1. Create an account on https://github.com/ and log in.
2. Select the compatible downloading file for your Operating System. Visit https://git-scm.com/downloads and follow the instructions given below.



git --everything-is-local

Search entire site...

About
Documentation
Downloads
GUI Clients
Logos
Community

The entire Pro Git book written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Downloads

macOS Windows
Linux/Unix

Older releases are available and the Git source repository is on GitHub.

Latest source Release
2.38.1
Release Notes (2022-10-07)
Download for Windows

GUI Clients

Git comes with built-in GUI tools (git-gui, gitk), but there are several third-party tools for users looking for a platform-specific experience.

View GUI Clients →

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

View Logos →

Git via Git

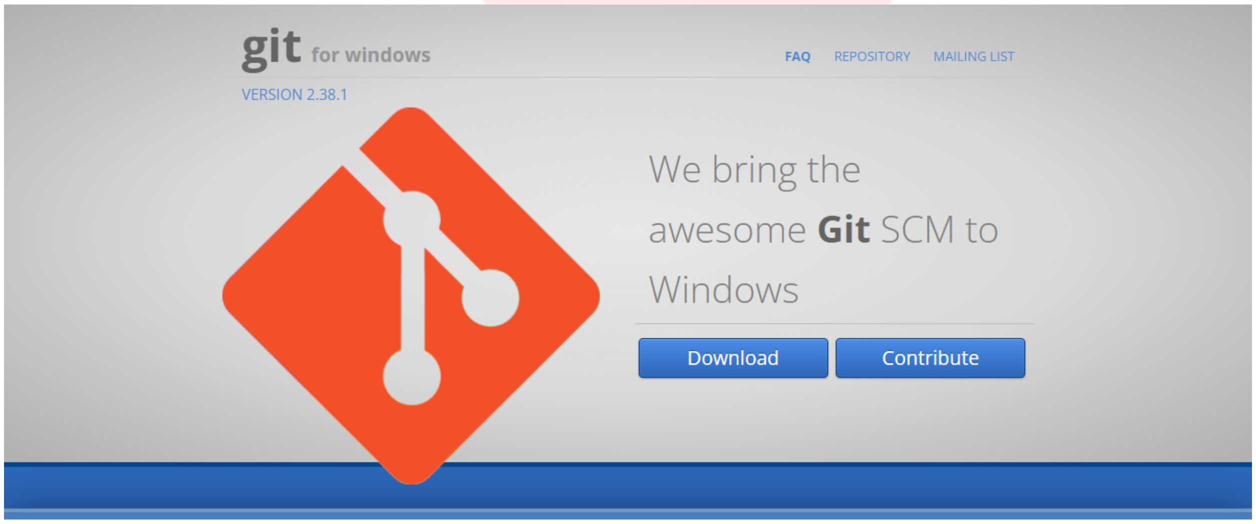
If you already have Git installed, you can get the latest development version via Git itself:

```
git clone https://github.com/git/git
```

You can also browse the current contents of the git repository using the web interface.

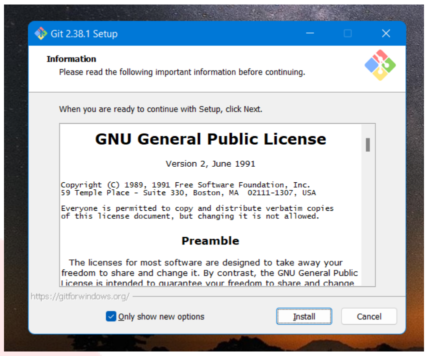
To Install Git for Windows

1. Go to https://gitforwindows.org/ and download the latest Git for Windows installer. For a detailed guide please refer to the links given in the Resource section below.



2. Run the installer to open the Git Setup wizard screen. Follow the Next and Finish prompts to complete the installation.

   a. Select the default installation options, and proceed further with the installation.



Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change

3. Open Command Prompt.
4. Run the following commands to configure your Git username and email. These details will be associated with any commits that you create:
   a. $ git config --global user.name "<username>"
   b. $ git config --global user.email "<email>"

Add SSH Key into the GitHub Account

An SSH key is an access credential for the SSH (secure shell) network protocol. It’s an authenticated and encrypted secure network protocol. SSH is used for remote file transfer, network management, and remote operating system access. SSH uses a pair of keys (public and private) to initiate secure sharing between remote users. Follow the below steps to connect to GitHub with SSH Key from Windows:

1. Open PowerShell
2. Run the command to create SSH keys: $ ssh-keygen

To start, store a public SSH key on GitHub. This is validated against a locally stored private key that Git uses to validate and establish a connection. GitHub SSH keys are created with the ssh-keygen tool that comes prepackaged with updated versions of Windows.

In Windows PowerShell, issue the following ssh-keygen command to create GitHub SSH keys: $ ssh-keygen -o -t rsa -C "windows-ssh@mcnz.com"

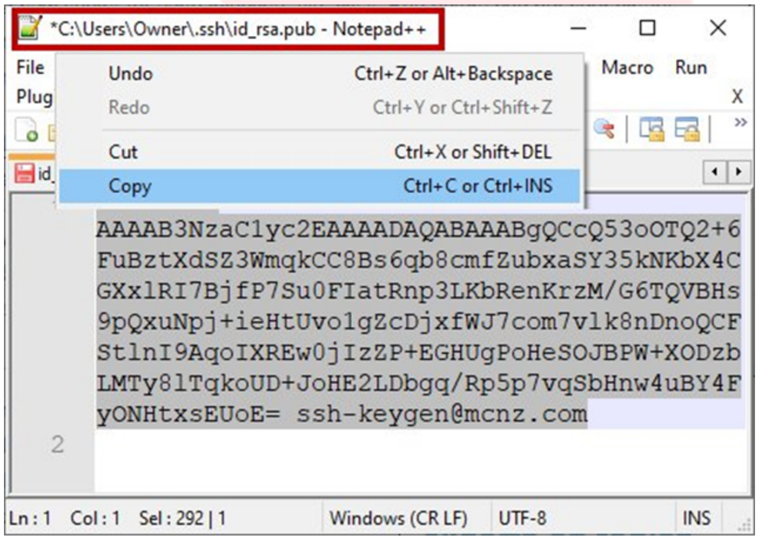
You will be asked for an optional passphrase. It’s permissible to click enter and leave this blank.

ssh-keygen flags	Purpose	Suggested
-C	Comments or metadata to add to the public key	Email address
-t	The type of GitHub SSH key to create	RSA
-o	Use the newest OpenSSH format	Leave blank

You will also be asked for a location to save the GitHub SSH keys on Windows. Again, just click enter to accept the default location, which is the .ssh folder under the user's home directory. The Windows GitHub SSH keys live in the .ssh folder under the current user's home directory.

GitHub SSH Configuration

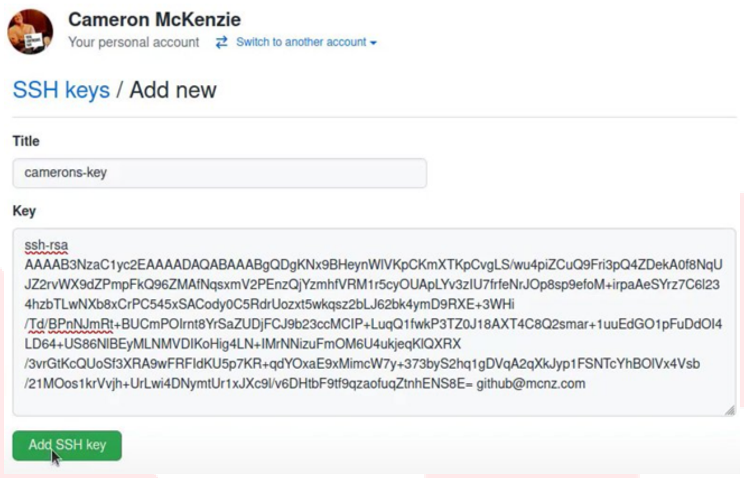
1. Copy the value of the SSH public key-
2. Open the SSH public key in a text editor such as Notepad++, perform a Select All, and copy the key.


*C:\Users\Owner\.ssh\id_rsa.pub - Notepad++
File
Plug
Macro
Run
Undo Ctrl+Z or Alt+Backspace
Redo Ctrl+Y or Ctrl+Shift+Z
Cut Ctrl+X or Shift+DEL
Copy Ctrl+C or Ctrl+INS
Ln : 1 Col : 1 Sel : 292 | 1
Windows (CR LF) UTF-8 INS

```
AAAAB3NzaC1yc2EAAAADAQABAAABgQCcQ53oOTQ2+6
FuBztXdSZ3WmqkCC8Bs6qb8cmfZubxaSY35kNKbX4C
GXxlRI7BjfP7Su0FIatRnp3LKbRenKrzM/G6TQVBHs
9pQxuNpj+ieHtUvolgZcDjxfWJ7com7vlk8nDnoQCF
StlnI9AqoIXREw0jIzZP+EGHUgPoHeSOJBPW+XODzb
LMTy8lTqkoUD+JoHE2LDbgq/Rp5p7vqSbHnw4uBY4F
yONHtxsEUoE= ssh-keygen@mcnz.com
```

3. Copy the public GitHub SSH key and store this value as a registered SSH key in your GitHub account. With the SSH key copied, log into GitHub, navigate to your account settings, and paste the public key as a new SSH key.

Save the Public key in your GitHub Account Settings


Cameron McKenzie
Your personal account ⇄ Switch to another account ▾

SSH keys / Add new

Title
camerons-key

Key
```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDgKNx9BHeynWlVKpCKmXTKpCvgLS/wu4piZCuQ9Fri3pQ4ZDekA0f8NqU
JZ2rvWX9dZPmpFkQ96ZMAfNqsxmV2PEnzQjYzmhfVRM1r5cyOUApLYv3zIU7frfeNrJOp8sp9efoM+irpaAeSYrz7C6l23
4hzbTLwNXb8xCrPC545xSACody0C5RdrUozxt5wkqsz2bLJ62bk4ymD9RXE+3WHi
/Td/BPnNJmRt+BUCmPOlrnt8YrSaZUDjFCJ9b23ccMCIP+LuqQ1fwkP3TZ0J18AXT4C8Q2smar+1uuEdGO1pFuDdOl4
LD64+US86NIBEyMLNMVDIKoHig4LN+IMrNNizuFmOM6U4ukjeqKlQXRX
/3vrGtKcQUoSf3XRA9wFRFldKU5p7KR+qdYOxaE9xMimcW7y+373byS2hq1gDVqA2qXkJyp1FSNTcYhBOlVx4Vsb
/21MOos1krVvjh+UrLwi4DNymtUr1xJXc9l/v6DHtbF9tf9qzaofuqZtnhENS8E= github@mcnz.com
```

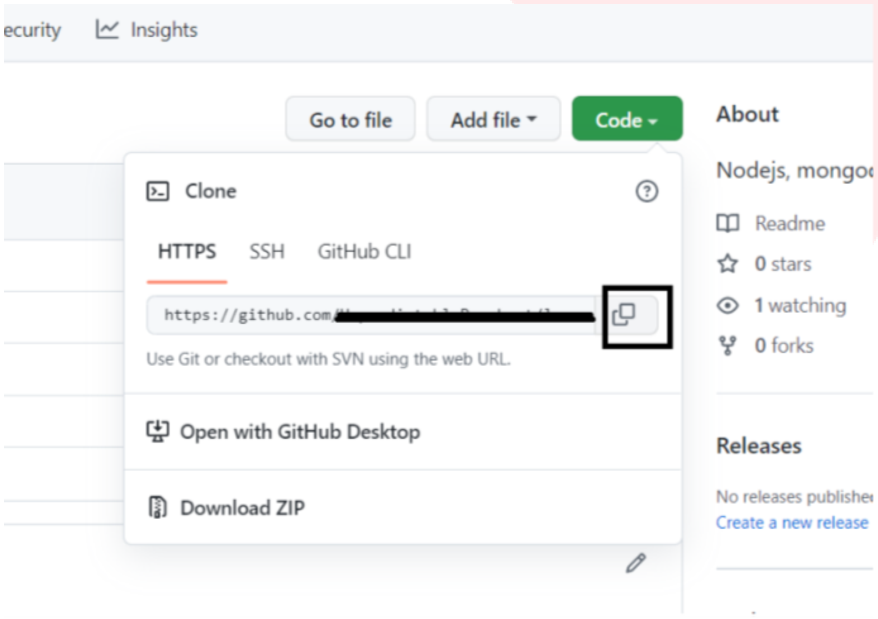
To obtain a secure, GitHub SSH Windows connection, you must register the public key in your online account settings.

Perform a Git Clone Operation using Your Repo's SSH URL

With the SSH keys generated, and the public key registered in your GitHub account, you can now use Git to connect to GitHub over SSH on Windows.

Create a Repository on GitHub: Link the Repository with Your Local Project Folder

Copy the SSH URL from the GitHub page of the repository you wish to clone and then provide that SSH link to the Git clone command: $ git clone <repository path>



https://github.com/[redacted]

Use Git or checkout with SVN using the web URL.

Resources

- Guide for Git installation:

  https://github.com/git-guides/install-git

  https://www.atlassian.com/git/tutorials/install-git

- Guide for Git installation on Windows:

  https://phoenixnap.com/kb/how-to-install-git-windows

- Guide for Git installation on Mac:

  https://phoenixnap.com/kb/install-git-on-mac

- Guide for Git installation on Linux:

  https://www.makeuseof.com/install-configure-git-on-linux/

- Adding SSH Key into GitHub:

  https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/GitHub-SSH-Windows-Example

- Git commands:

  https://www.atlassian.com/git/glossary

- Git Documentation:

  https://git-scm.com/doc