Instructions for Installing Git

Dr. Julianna Oliveira

November 8, 2024

For Ubuntu/Debian-based systems

1. Update the package index:

   ```
   sudo apt update
   ```

2. Install Git:

   ```
   sudo apt install git
   ```

3. Verify the installation:

   ```
   git --version
   ```

   This command should return the installed version of Git.

4. **For Conda environment installation**: If you want to install Git within a specific conda environment, follow these steps:

   (a) Activate your conda environment:

   ```
   conda activate your_env_name
   ```

   (b) Install Git using conda:

   ```
   conda install git
   ```

   (c) Verify the installation:

   ```
   git --version
   ```

Installing Git on Windows

To install Git on Windows, follow these steps:

1. Download the Git installer for Windows from the official website:

   ```
   https://git-scm.com/download/win
   ```

2. Run the installer and follow the installation wizard.
3. During the installation, you will be prompted with several configuration options:
   - **Select Components**: You can leave the default options checked.
   - **Choosing the editor used by Git**: Select your preferred text editor (e.g., Vim, Nano, Notepad++).
   - **Adjusting your PATH environment**: Select the default option to enable Git from the command line.
   - **Choosing HTTPS transport backend**: Select ”Use the OpenSSL library.”
   - **Configuring the line ending conversions**: Select ”Checkout Windows-style, commit Unix-style line endings.”
4. After installation, open Git Bash from the Start menu.
5. Verify the installation by running:

   ```
   git --version
   ```

   This should return the installed version of Git.

Installing Git on macOS

To install Git on macOS, follow these steps:

1. Check if Git is already installed by running:

   ```
   git --version
   ```

   If Git is already installed, you should see the version number.

2. If Git is not installed, you can either install it using *homebrew* or by downloading the macOS Git installer from the official website:

```
https://git-scm.com/download/mac
```

After downloading, run the installer and follow the steps to complete the installation.

3. Verify the installation by running:

```
git --version
```

This should return the installed version of Git.

Configuration

Once Git is installed, you should configure your user name and email as follows:

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

Using Git to Update Your Code

Once Git is installed, let's go through the process of creating a GitHub repository and uploading your code.

Step 1: Creating an Online Repository on GitHub

1. Go to `https://github.com` and log in (or create a new account if you don't have one). 2. Once logged in, look for the **New** button. 3. Enter a name for your repository, like `my_project`, and add an optional description. 4. Select whether you want the repository to be **Public** or **Private**. 5. Click on **Create repository**.

Your GitHub repository is now created and ready to store your code!

Step 2: Setting Up Your Local Repository

Now, we'll set up a local repository on your computer and connect it to the GitHub repository.

A. Create a Local Project Folder

1. Open your terminal. 2. Navigate to the folder where you want to create your project (guidelines below assuming Unix):

```
cd path/to/your/directory
```

3. Create a new folder for your project:

```
mkdir my_project
cd my_project
```

B. Initialise Git in Your Local Folder

1. Run the following command to turn this folder into a Git repository:

```
git init
```

2. This command initialises an empty Git repository in your folder, allowing Git to start tracking changes.

Step 3: Connecting Your Local Repository to GitHub

1. In your GitHub repository page (from Step 2), you'll see an HTTPS URL in the format `https://github.com/username/my_project.git`. 2. Copy this URL, then, in your terminal, type:

```
git remote add origin https://github.com/username/my_project.git
```

This command links your local repository to the online GitHub repository.

Step 4: Adding and Committing Your Code Locally

1. If you've added any files to your project folder, it's time to tell Git to track these files. Run:

```
git add .
```

This command stages all files in the folder to be committed. If you only want to add a specific file, replace '.' with the file name.

2. Now, save a snapshot of your changes with a descriptive message:

```
git commit -m "Initial commit - adding project files"
```

Step 5: Pushing Your Code to GitHub

Now that your changes are committed locally, let's upload them to GitHub.

1. Use this command to push your code to the GitHub repository:

```
git push -u origin main
```

If it's the first time you're pushing to GitHub, it may ask you for your GitHub username and password.

2. Check your GitHub repository online. You should now see your files there!

Step 6: Making and Pushing Further Changes

After making changes to your code, you can repeat these steps to upload the updates:

1. Check which files were modified:

```
git status
```

2. Stage your changes:

```
git add .
```

3. Commit your changes:

```
git commit -m "Description of the changes made"
```

4. Push your changes to GitHub:

```
git push
```