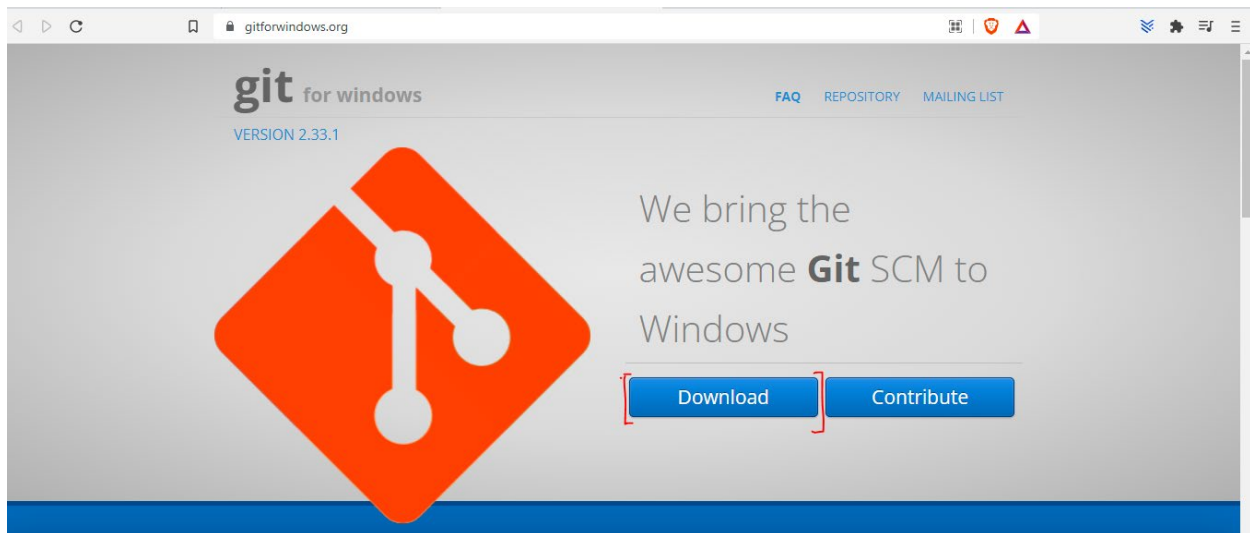



## Git for Windows

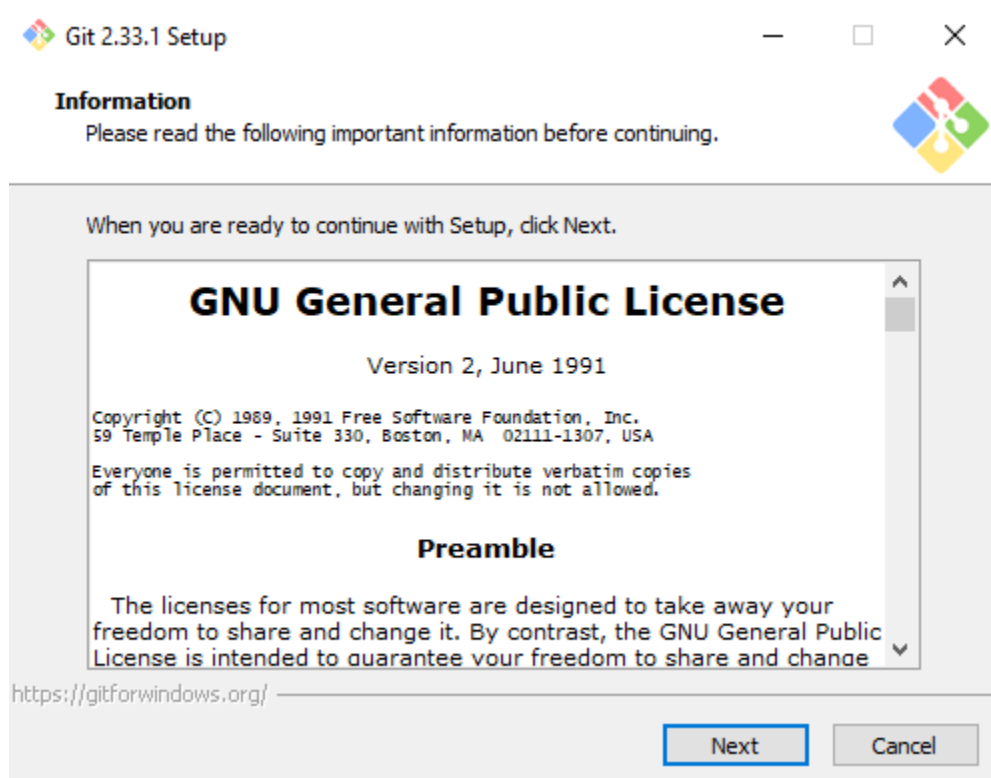
1. Download git <https://gitforwindows.org/>



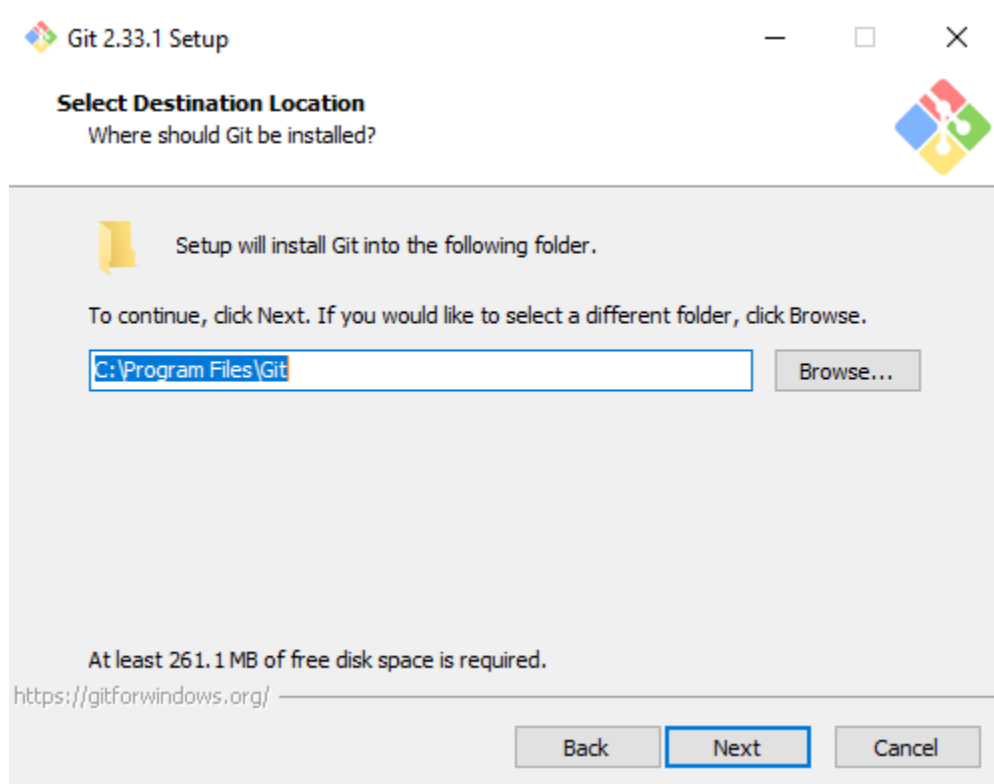
2. Open Download (double click on it)

 Git-2.33.1-64-bit.exe	10/28/2021 2:35 PM	Application	49,010 KB
---	--------------------	-------------	-----------

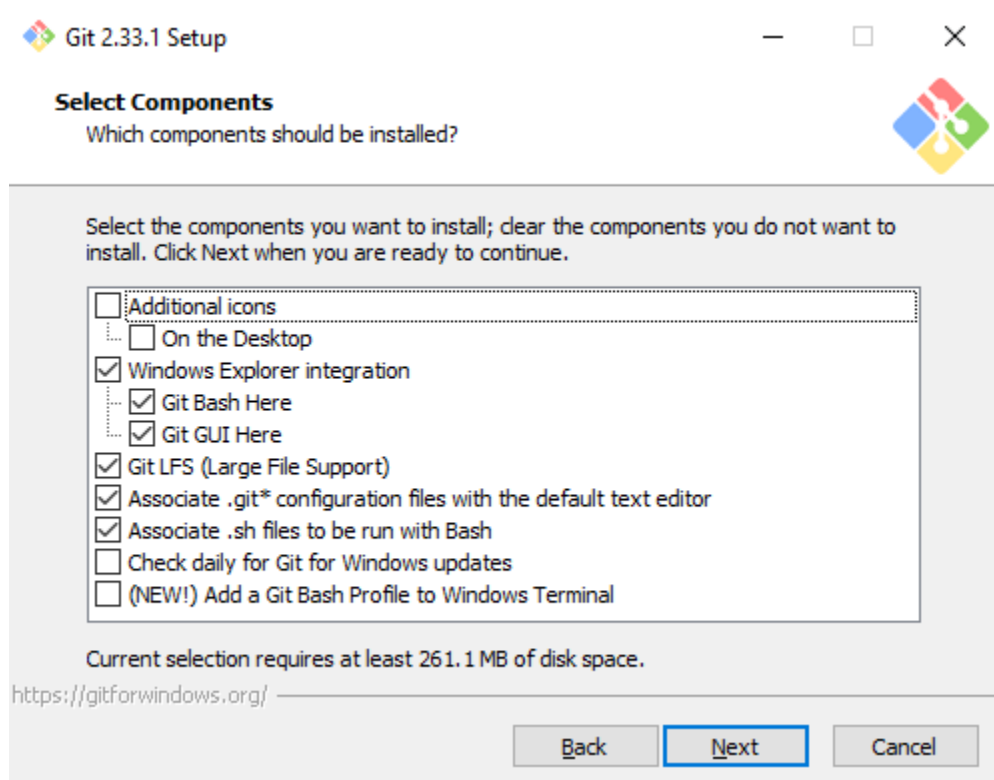
3. A window requesting permission to allow app to make changes will appear, click YES
4. Click Next



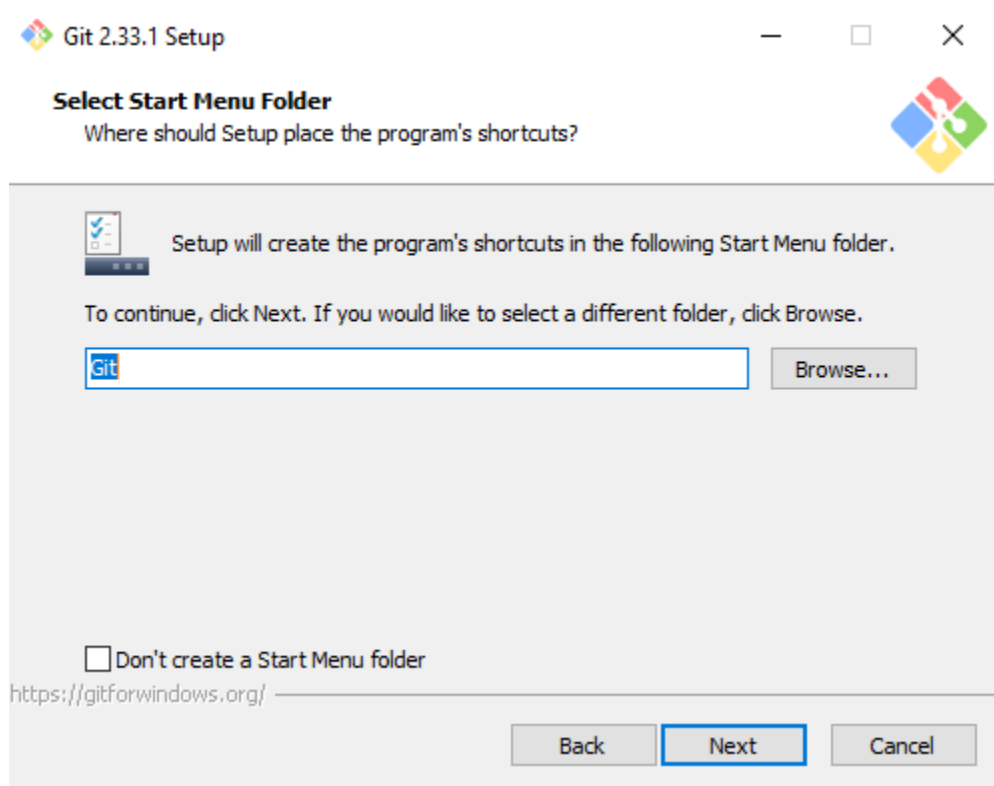
5. Click Next



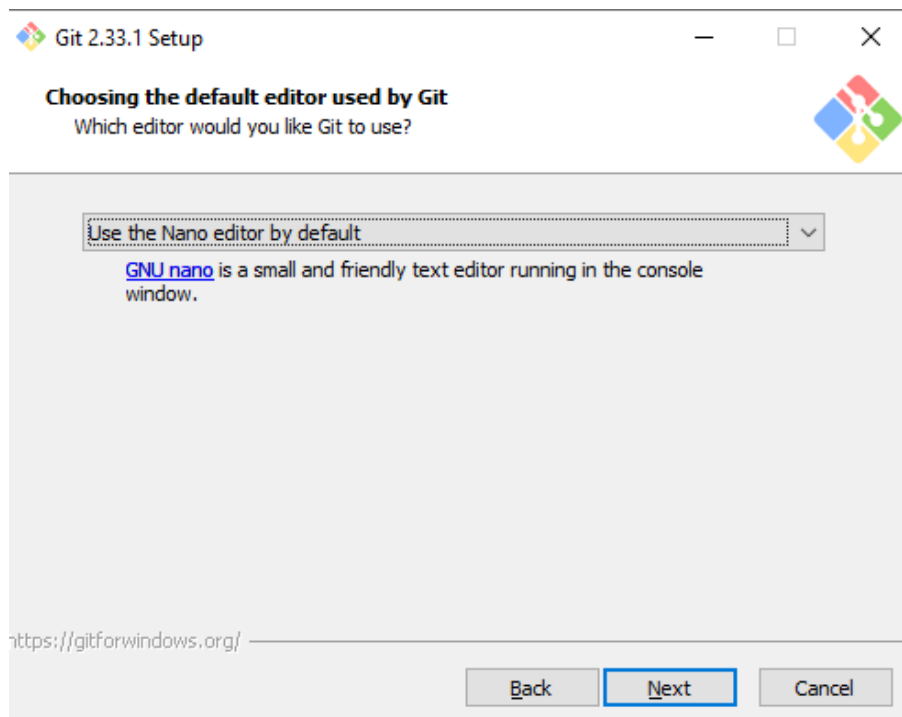
6. Click Next



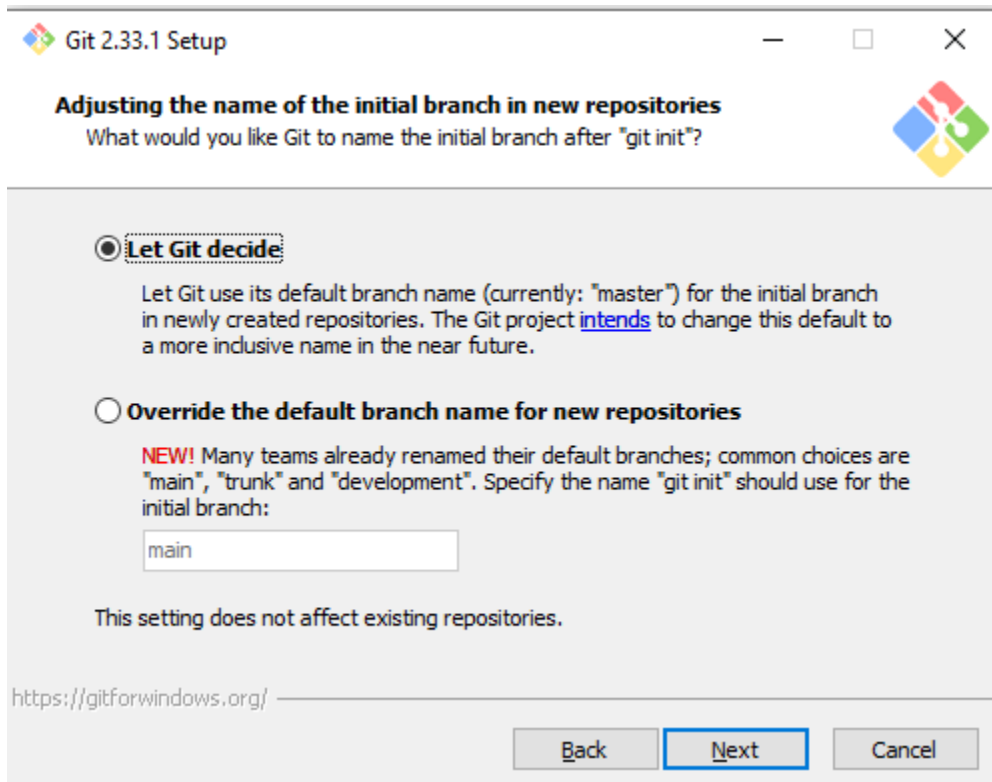
7. Click Next



8. Select "Use the Nano editor by default" and click Next



9. "Let Git Decide" and click Next



The screenshot shows the 'Git 2.33.1 Setup' window with the title 'Adjusting the name of the initial branch in new repositories'. The subtitle asks 'What would you like Git to name the initial branch after "git init"?'. There are two radio button options. The first option, 'Let Git decide', is selected. Below it, a paragraph explains that Git will use its default branch name 'master' and that the project intends to change this to a more inclusive name. The second option, 'Override the default branch name for new repositories', is unselected. Below it, a paragraph explains that many teams have renamed their default branches to 'main' or 'development', and a text box contains the word 'main'. A note states that this setting does not affect existing repositories. At the bottom, there is a URL 'https://gitforwindows.org/' and three buttons: 'Back', 'Next' (which is highlighted with a blue border), and 'Cancel'.

Git 2.33.1 Setup

**Adjusting the name of the initial branch in new repositories**  
What would you like Git to name the initial branch after "git init"?

☒ **Let Git decide**

Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories. The Git project [intends](#) to change this default to a more inclusive name in the near future.

☐ **Override the default branch name for new repositories**

**NEW!** Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:

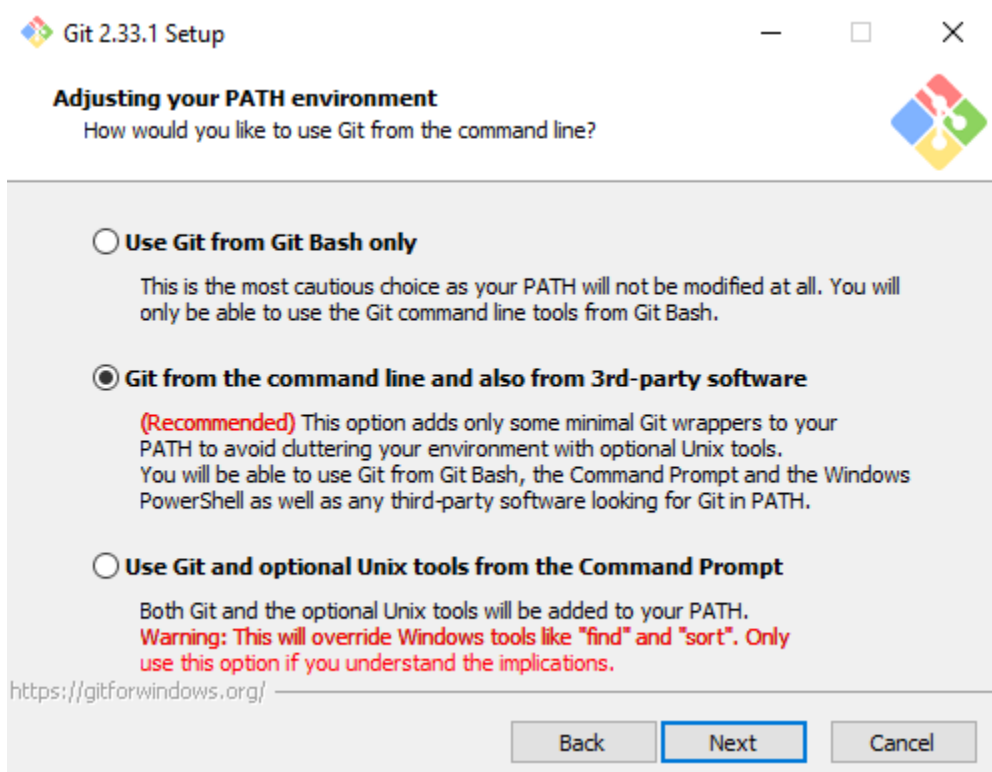
main

This setting does not affect existing repositories.

<https://gitforwindows.org/>

Back Next Cancel

10. "Git from the command line and also from 3<sup>rd</sup>-party software" click Next



The screenshot shows the 'Git 2.33.1 Setup' window with the title 'Adjusting your PATH environment'. The subtitle asks 'How would you like to use Git from the command line?'. There are three radio button options. The first option, 'Use Git from Git Bash only', is unselected. Below it, a paragraph explains that this is the most cautious choice as the PATH will not be modified. The second option, 'Git from the command line and also from 3rd-party software', is selected. Below it, a paragraph explains that this option adds minimal Git wrappers to the PATH to avoid cluttering the environment with optional Unix tools. The third option, 'Use Git and optional Unix tools from the Command Prompt', is unselected. Below it, a paragraph explains that both Git and optional Unix tools will be added to the PATH, with a warning that this will override Windows tools like 'find' and 'sort'. At the bottom, there is a URL 'https://gitforwindows.org/' and three buttons: 'Back', 'Next' (which is highlighted with a blue border), and 'Cancel'.

Git 2.33.1 Setup

**Adjusting your PATH environment**  
How would you like to use Git from the command line?

☐ **Use Git from Git Bash only**

This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

☒ **Git from the command line and also from 3rd-party software**

**(Recommended)** This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.

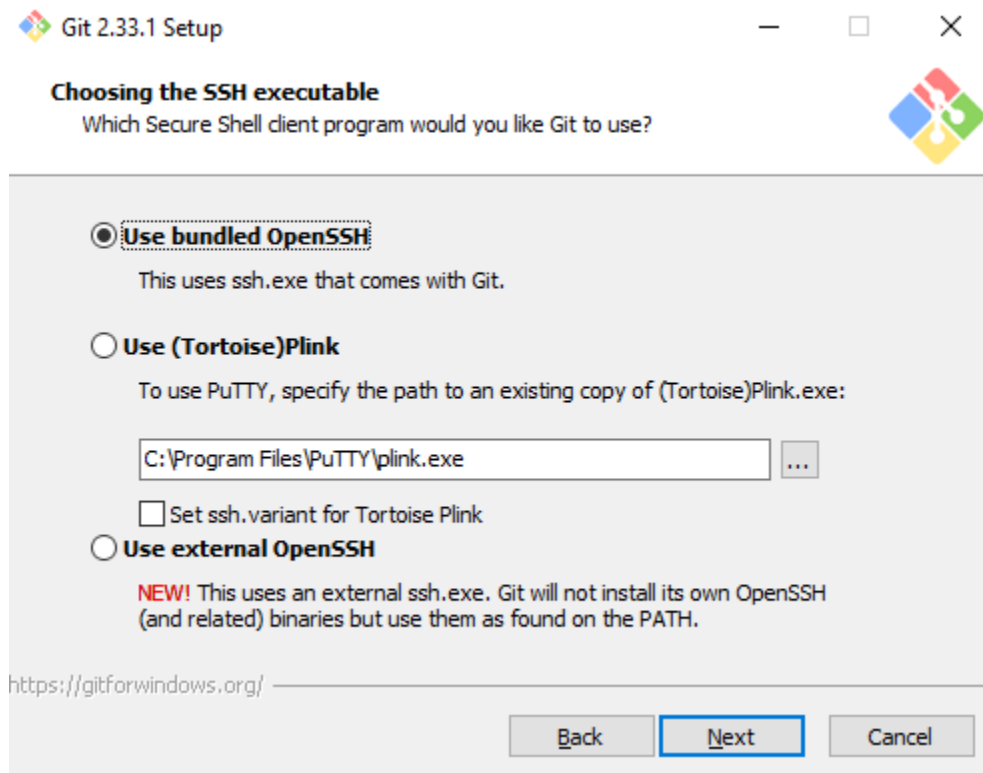
☐ **Use Git and optional Unix tools from the Command Prompt**

Both Git and the optional Unix tools will be added to your PATH.  
**Warning:** This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

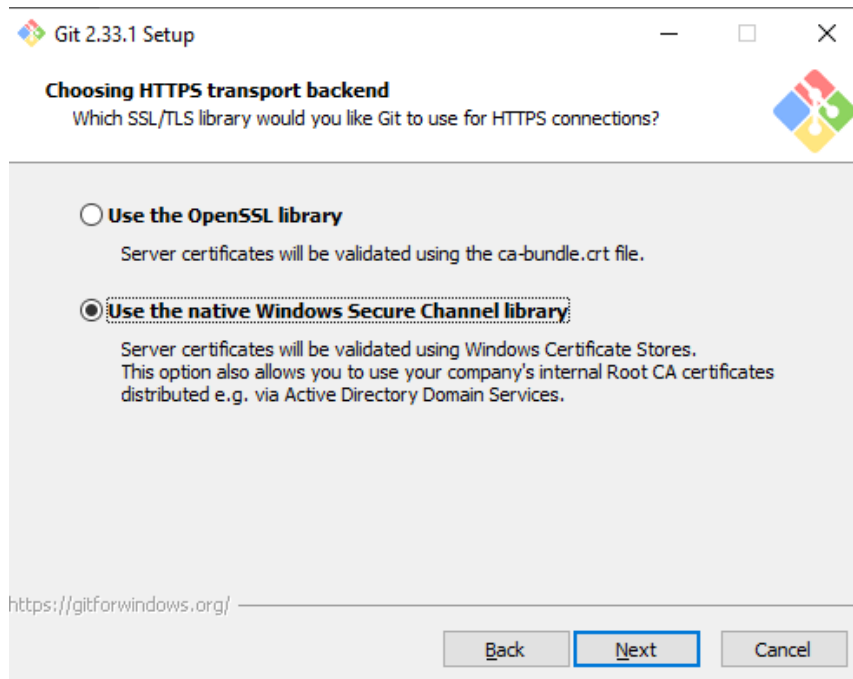
<https://gitforwindows.org/>

Back Next Cancel

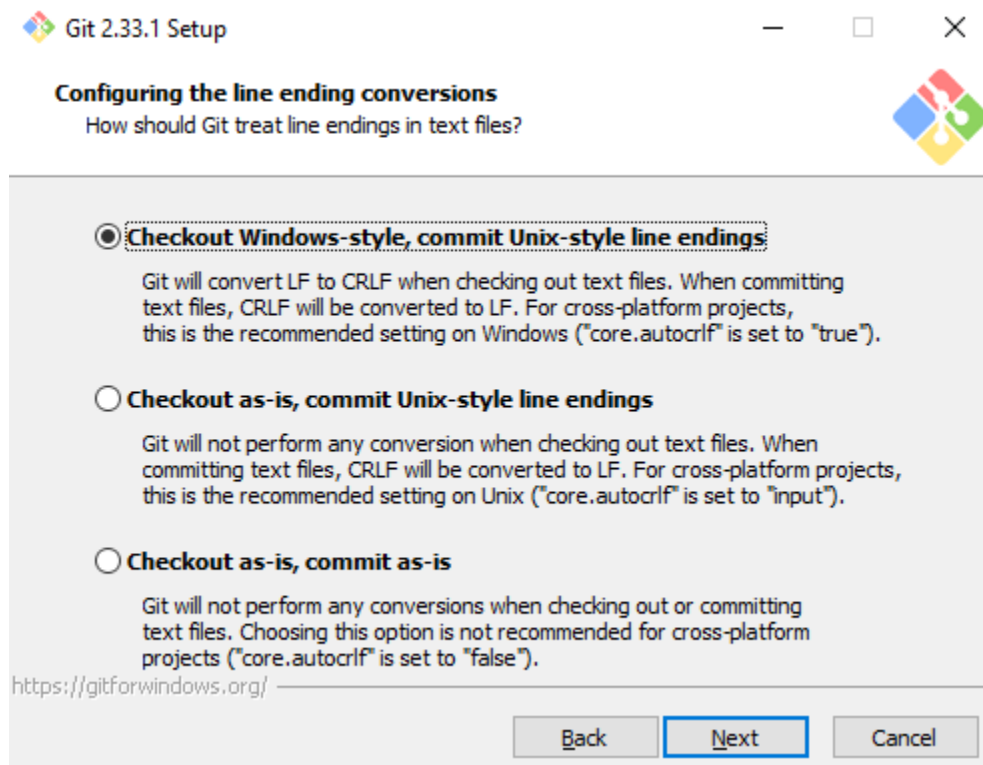
11. "Use bundled OpenSSH", click Next



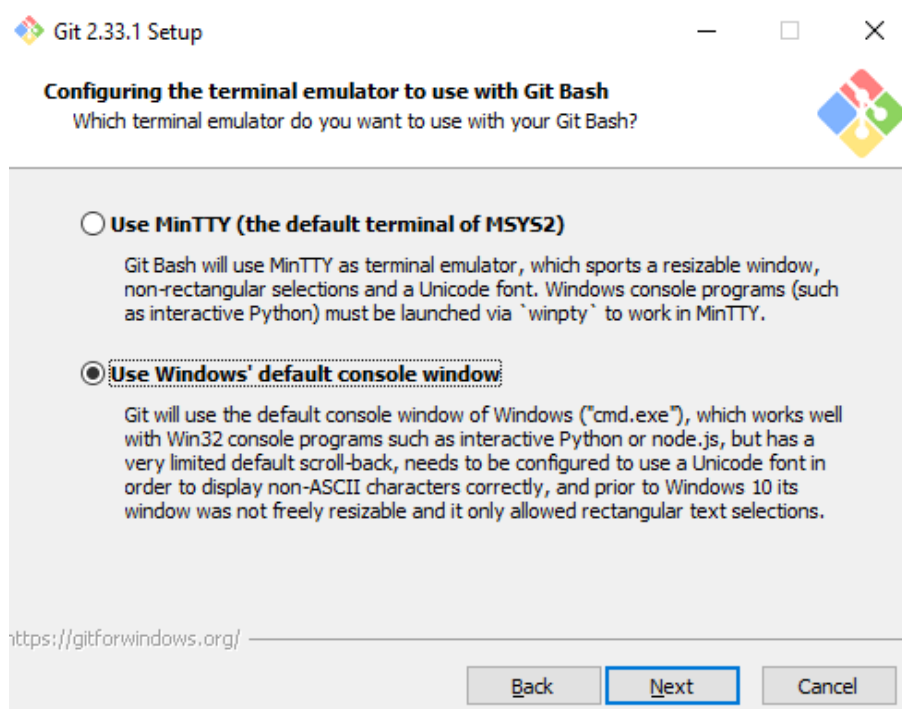
12. "Use the native Windows Secure Channel library", click Next



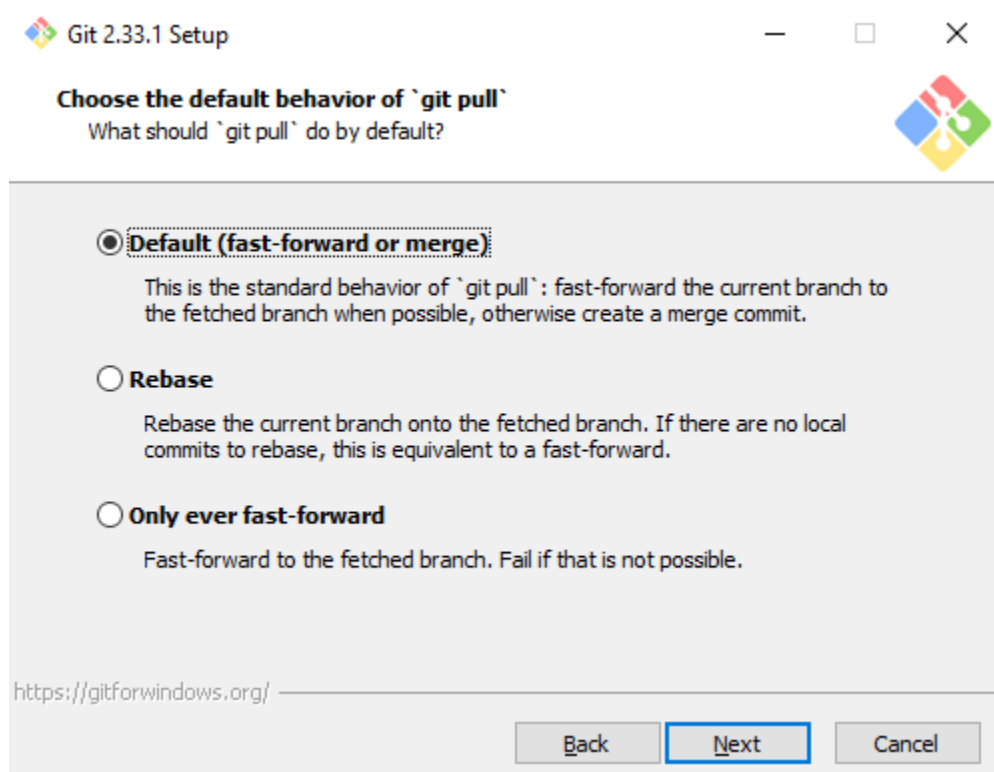
13. “Checkout Windows-style, commit Unix-style line endings”, click Next



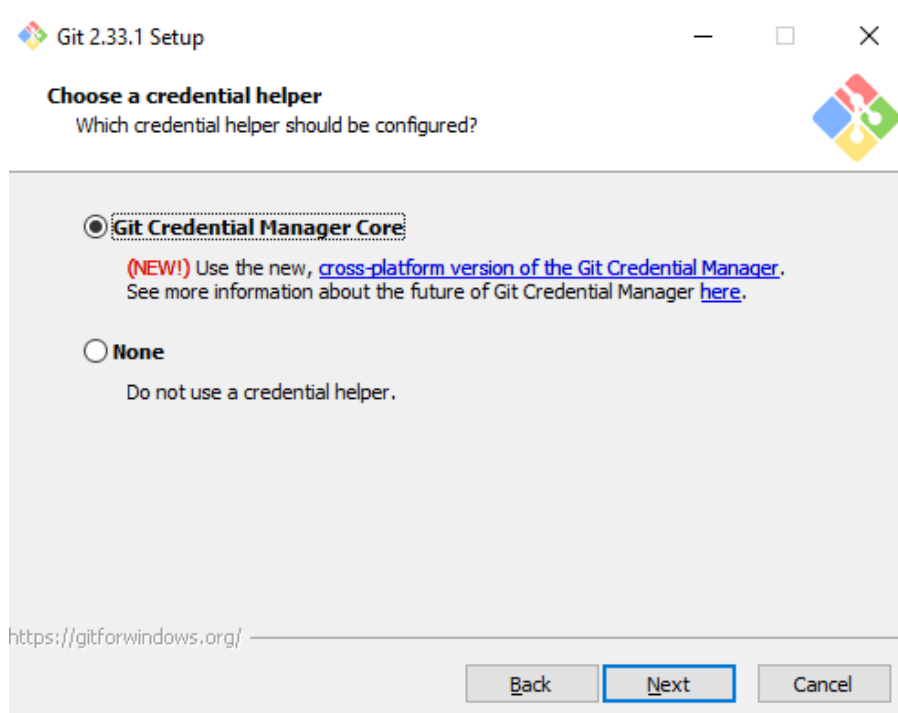
14. “Use Windows’ default console window”, click Next



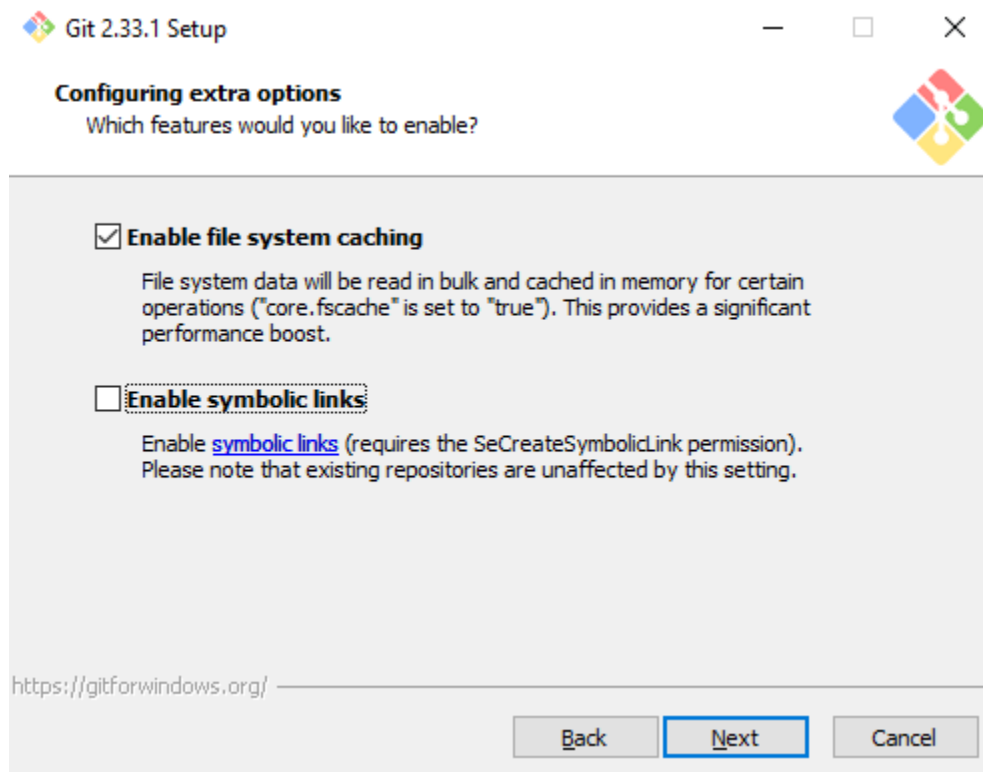
15. "Default (fast-forward or merge)"



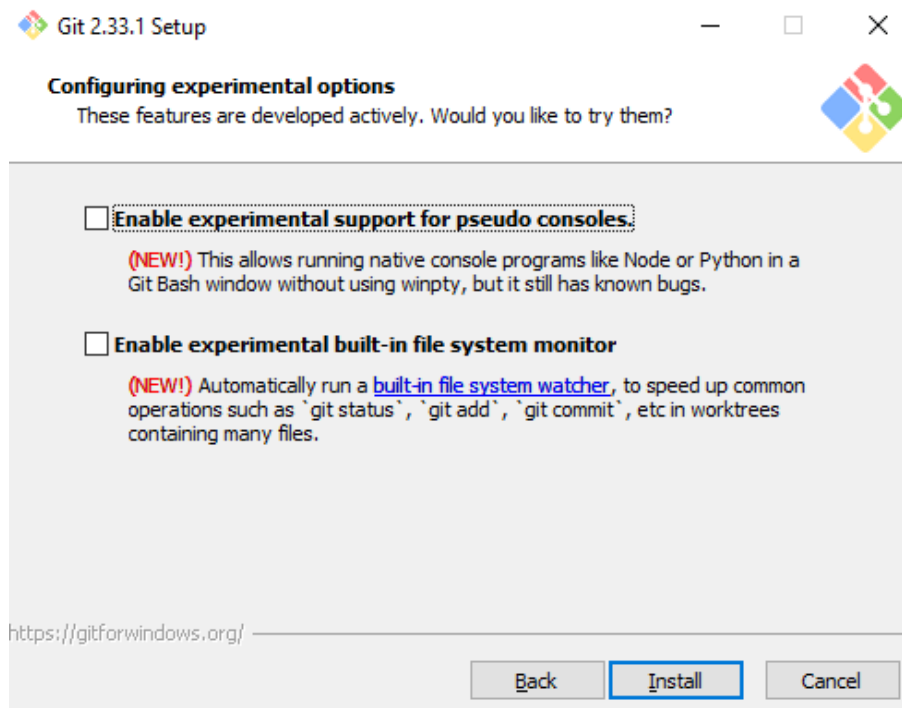
16. "Git Credential Manager Core", click Next



17. "Enable file system caching", click Next

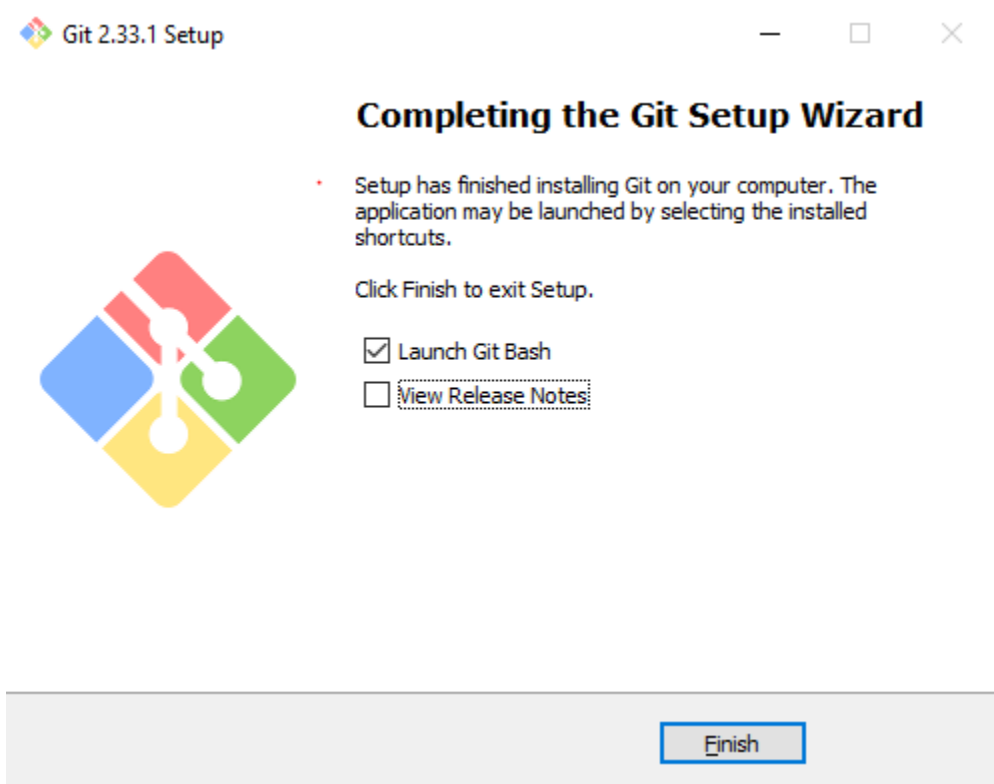


18. Click Install





19. "Launch Git Bash", click Finish



20. Git Bash will open up. To verify everything works

- Type the command "git --version"

```
CA: MINGW64:/c/Users/user  
user@UofM-Student MINGW64 ~ (master)  
$ git --version
```

- Hit Enter (on your keyboard), and you should get a similar output.

```
git version 2.33.1.windows.1
```