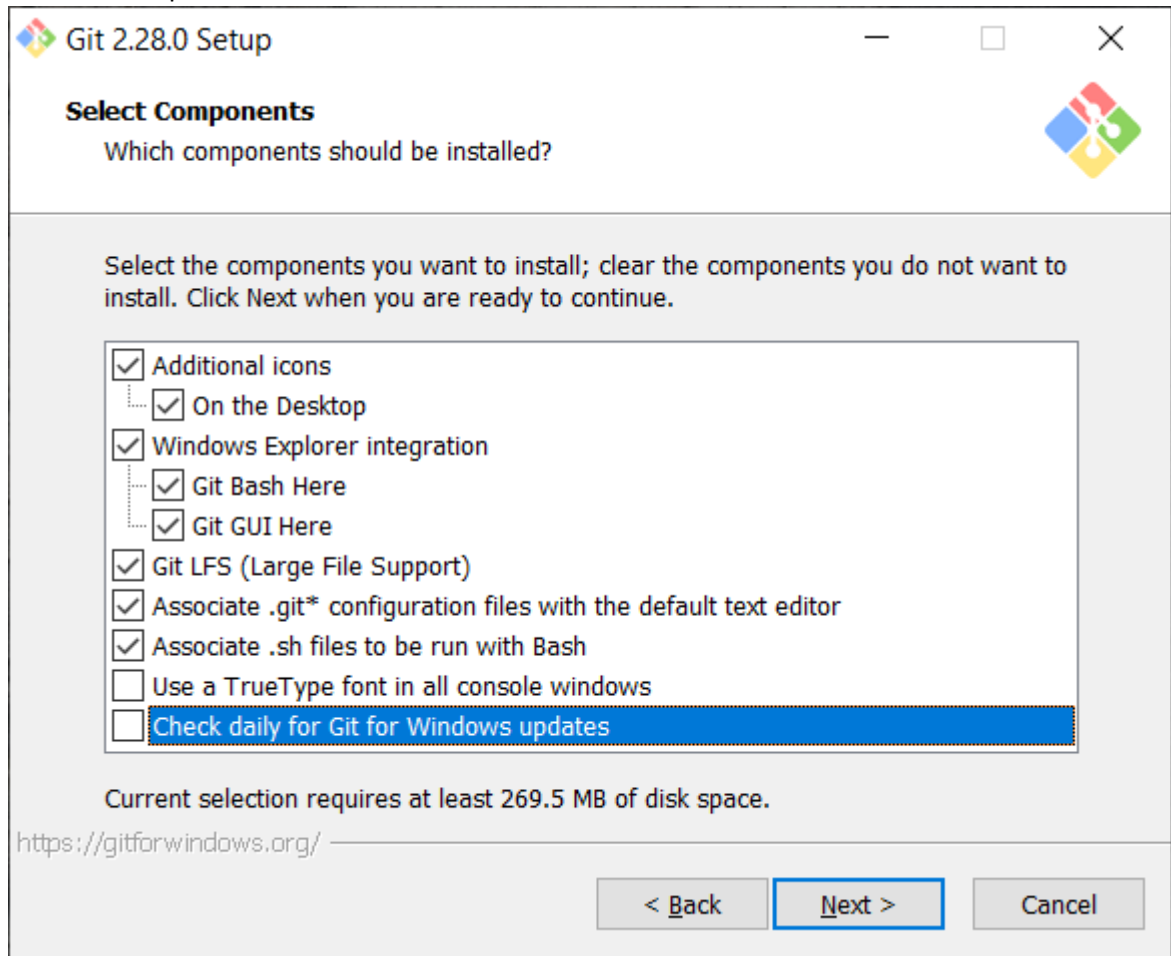## Windows installation:

1) Download the installer file from https://git-scm.com/downloads
2) Open the installer and click next
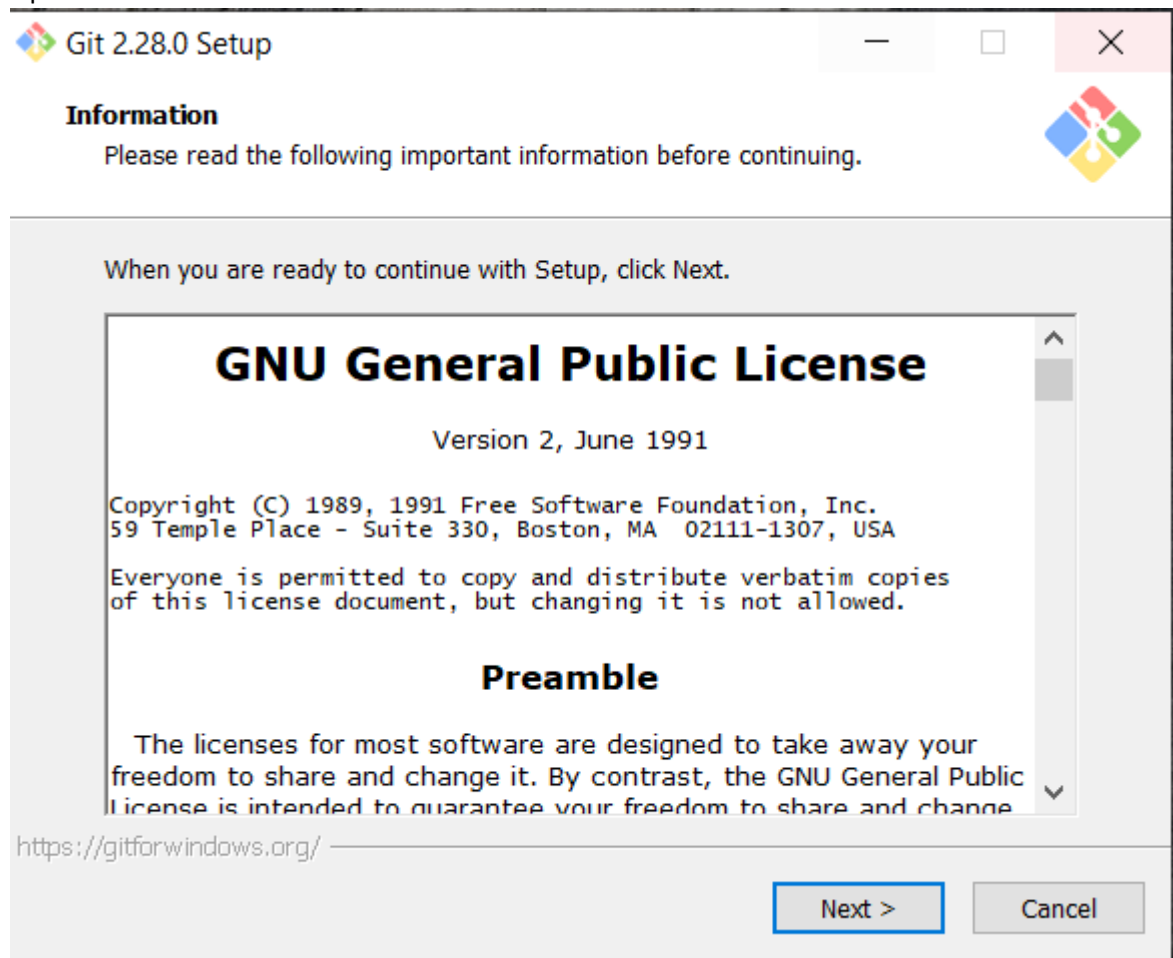
3) Choose the directory where you want to install git bash and click next

Git 2.28.0 Setup — □ ×

Select Destination Location
Where should Git be installed?

Setup will install Git into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Git   Browse...

At least 269.6 MB of free disk space is required.

https://gitforwindows.org/

< Back   Next >   Cancel
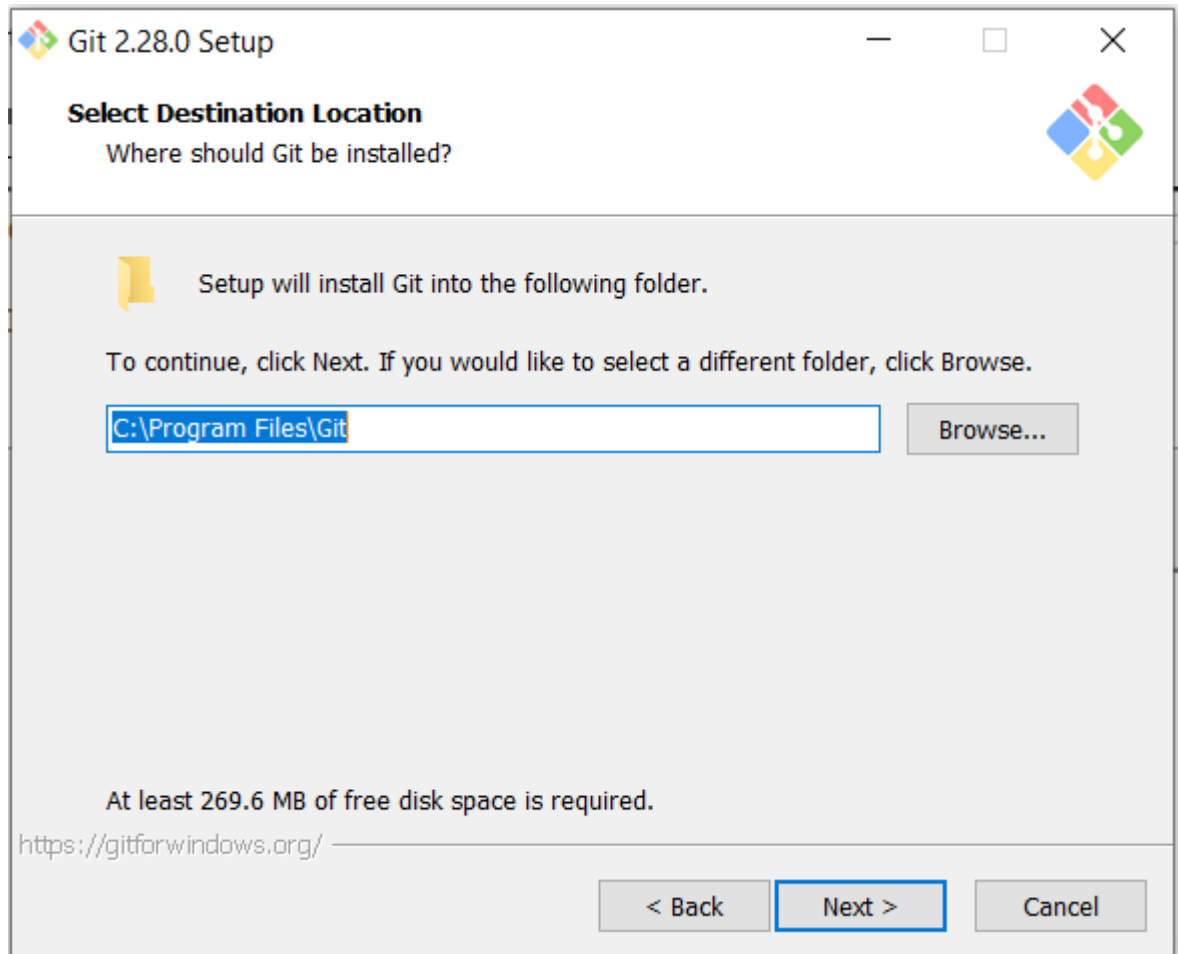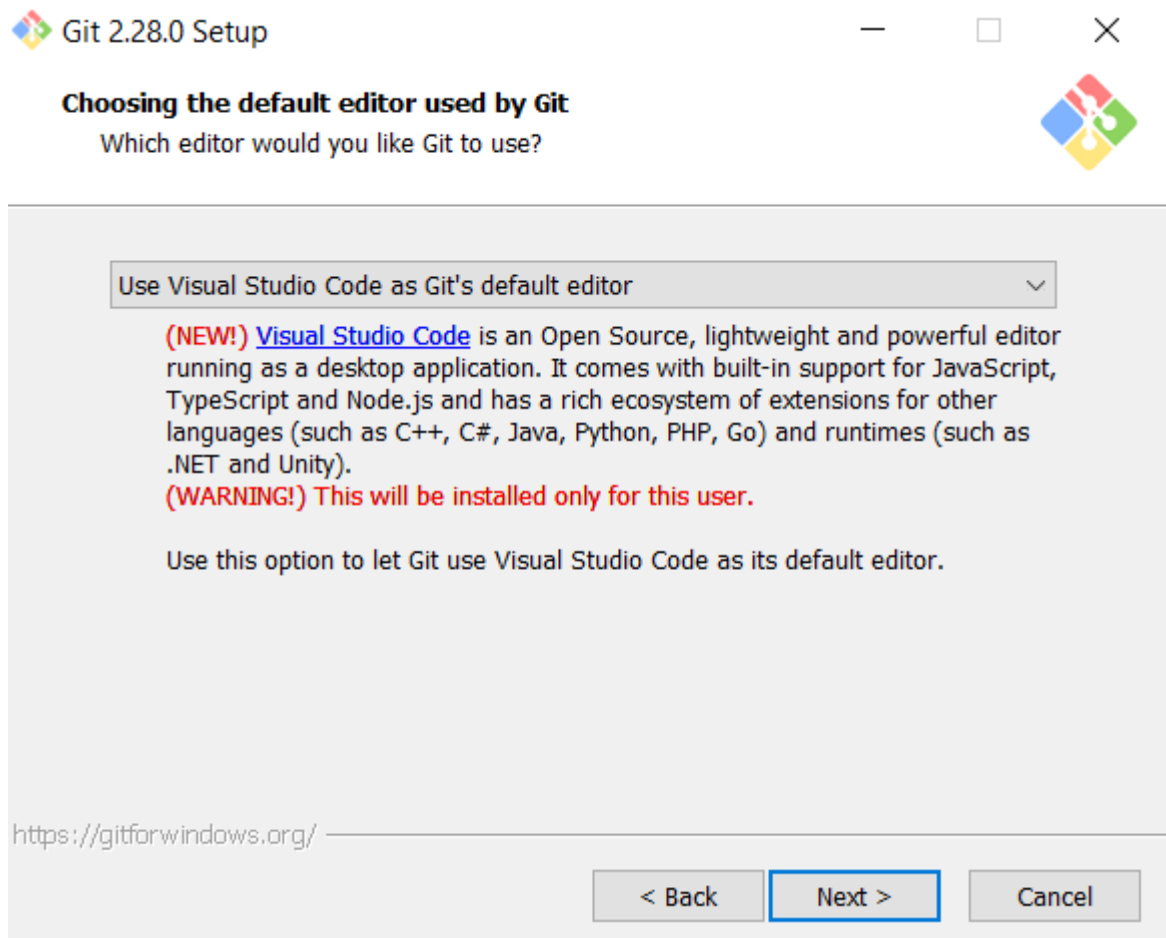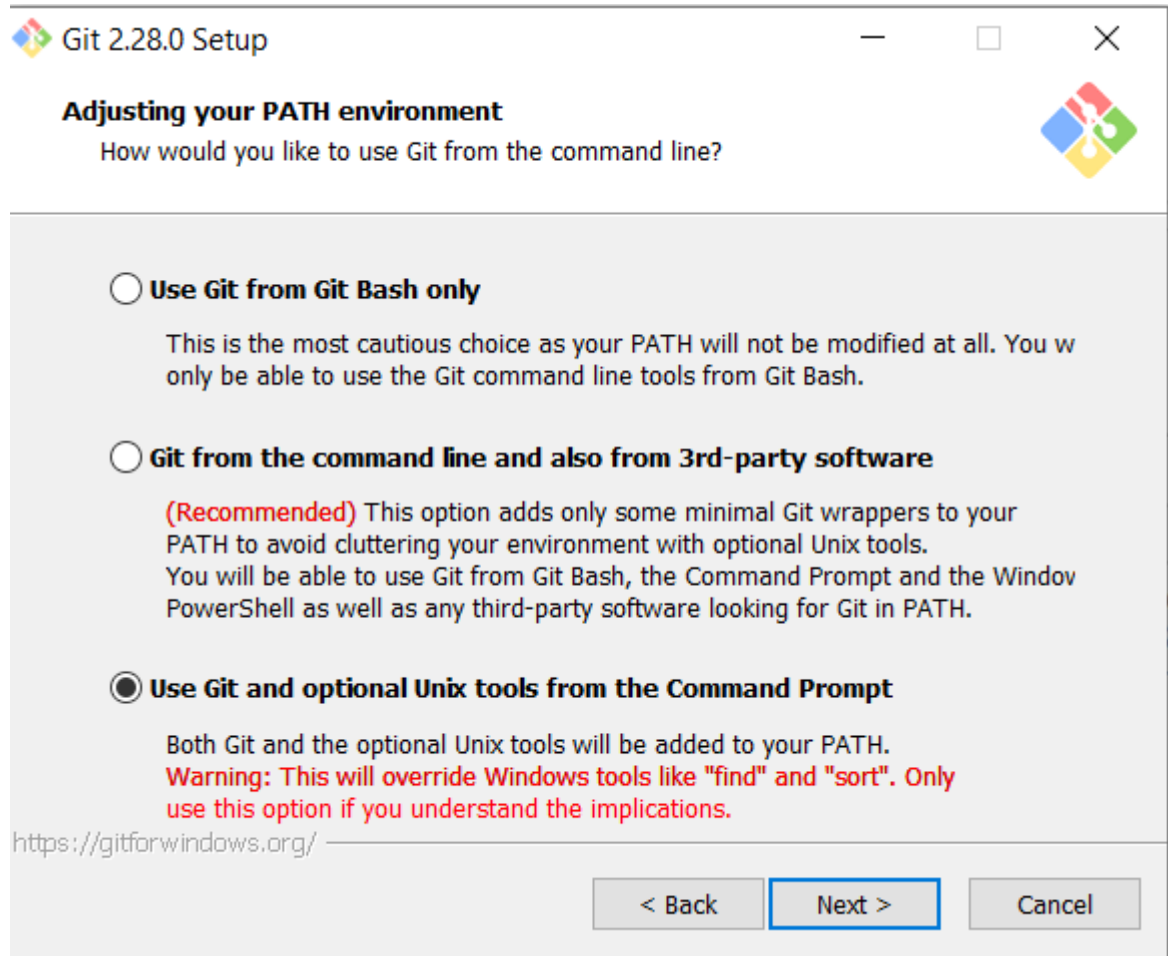
4) Choose this options and click next

5) Click next
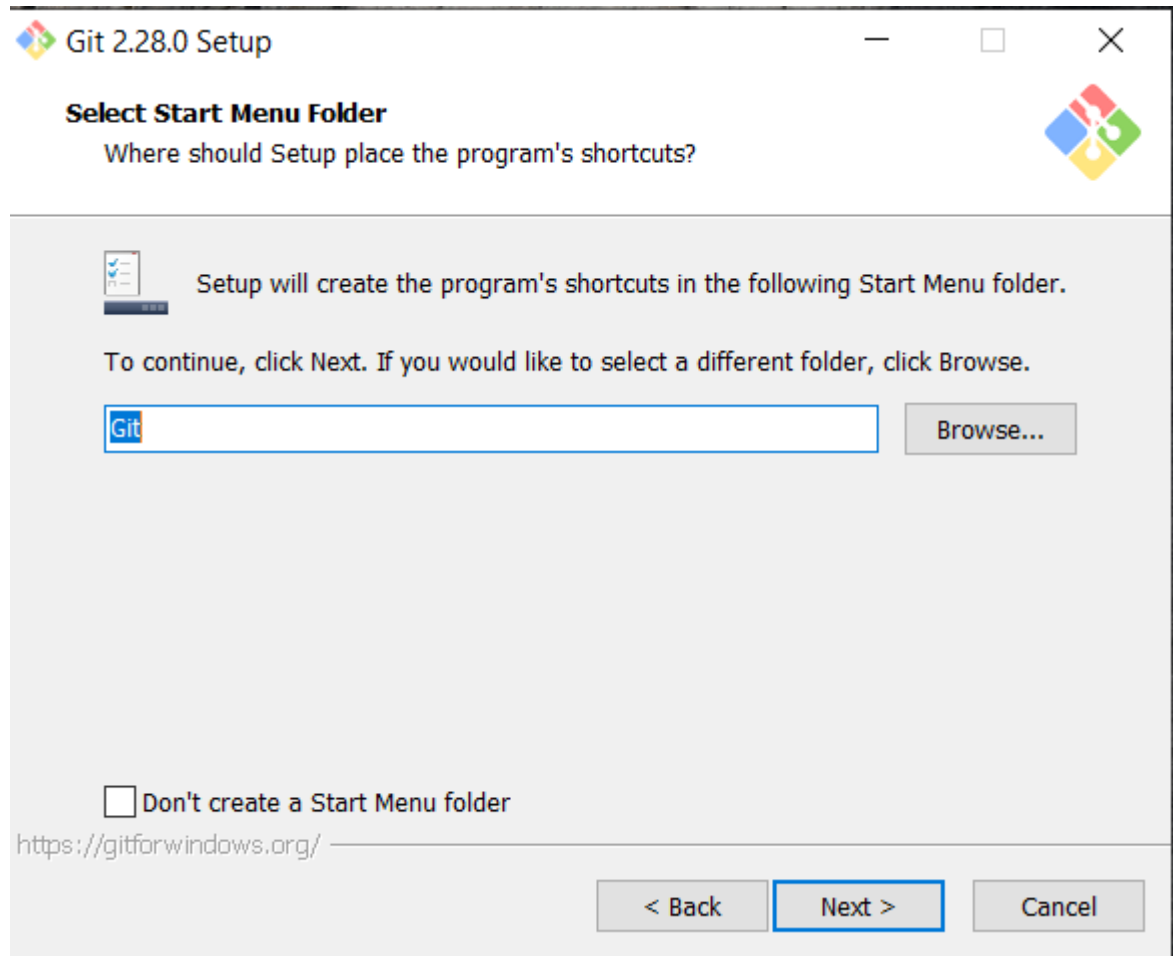
6) Choose the text editor which you are using and click next

Git 2.28.0 Setup — □ ✕

Choosing the default editor used by Git
Which editor would you like Git to use?

Use Visual Studio Code as Git's default editor ⌄

(NEW!) Visual Studio Code is an Open Source, lightweight and powerful editor running as a desktop application. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).
(WARNING!) This will be installed only for this user.

Use this option to let Git use Visual Studio Code as its default editor.

https://gitforwindows.org/

< Back   Next >   Cancel

7) Choose either 2nd or 3rd option and click next

Git 2.28.0 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 w
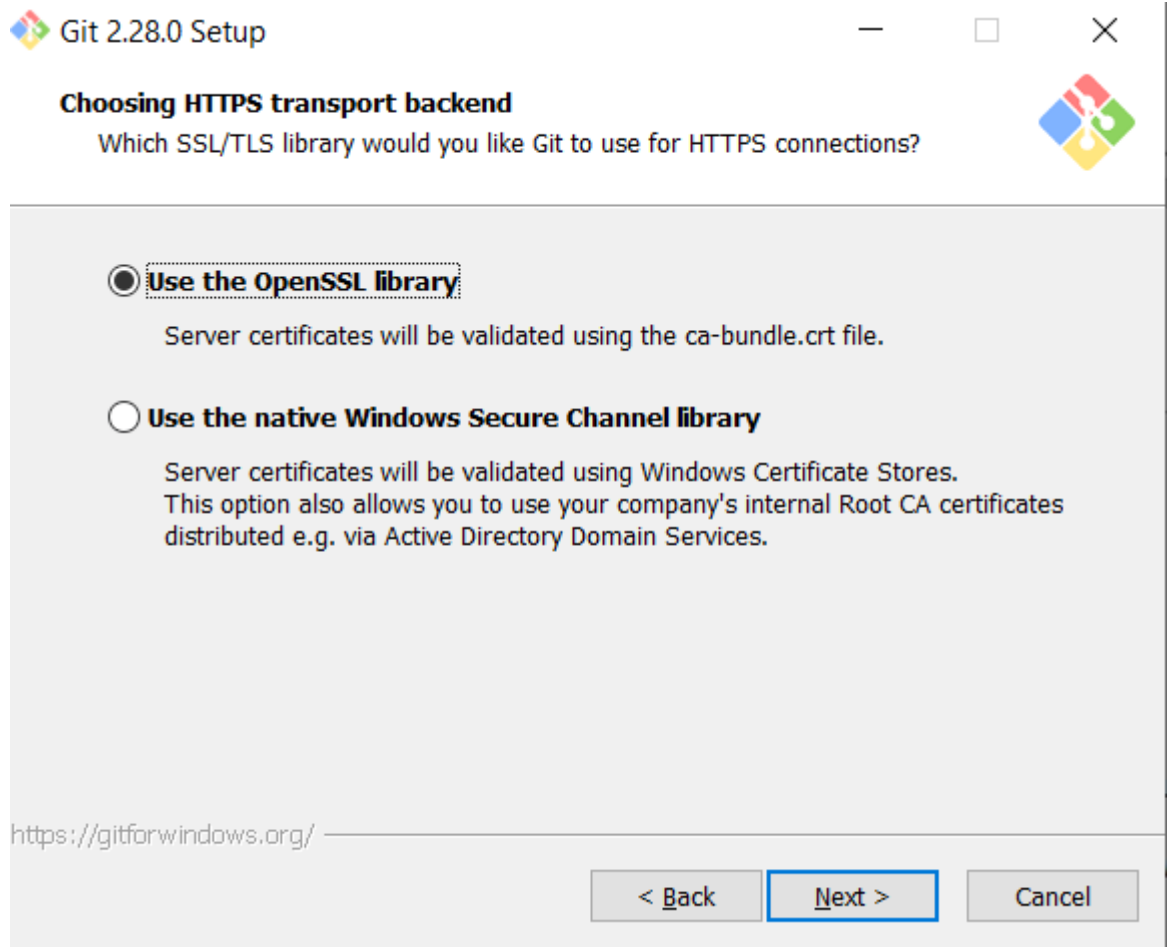only be able to use the Git command line tools from Git Bash.

○ **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 Windov
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
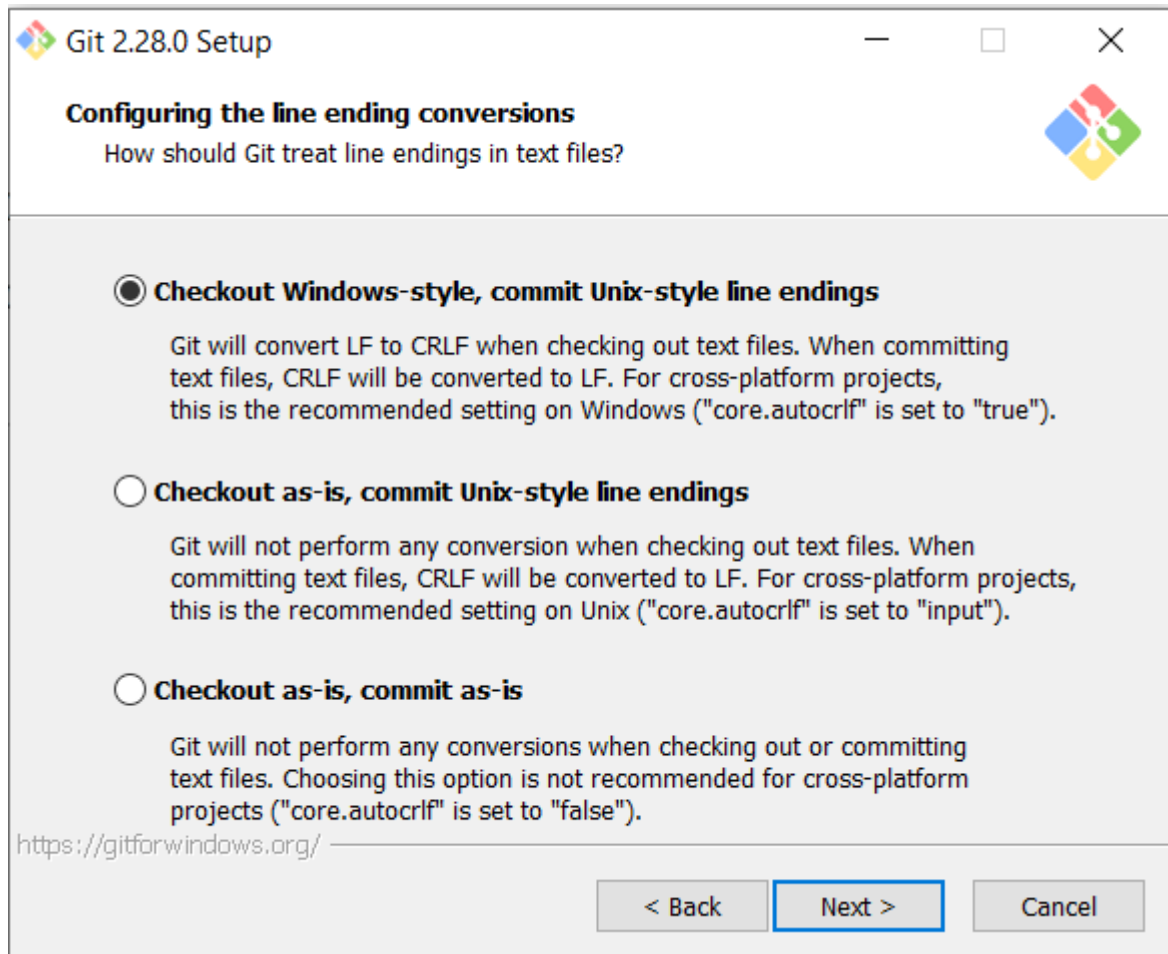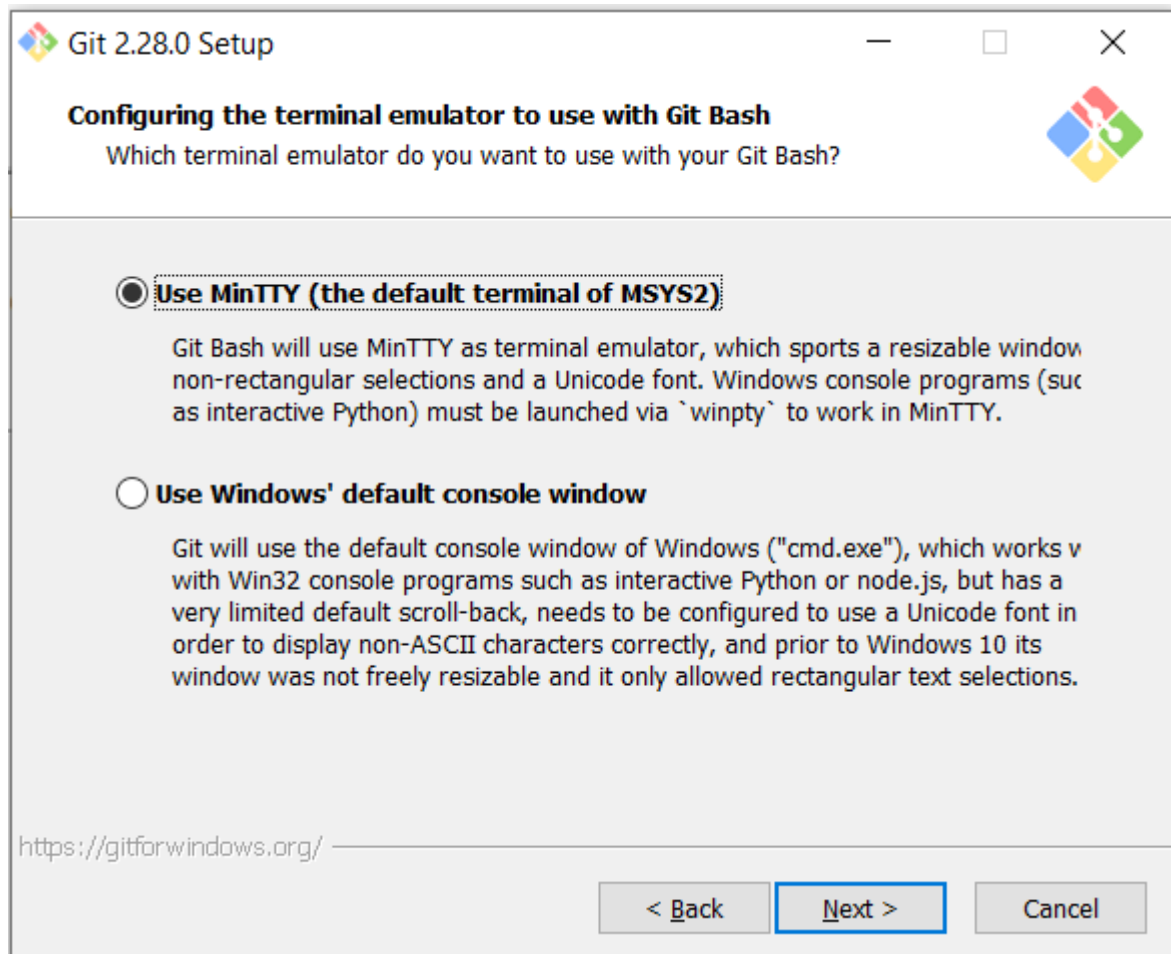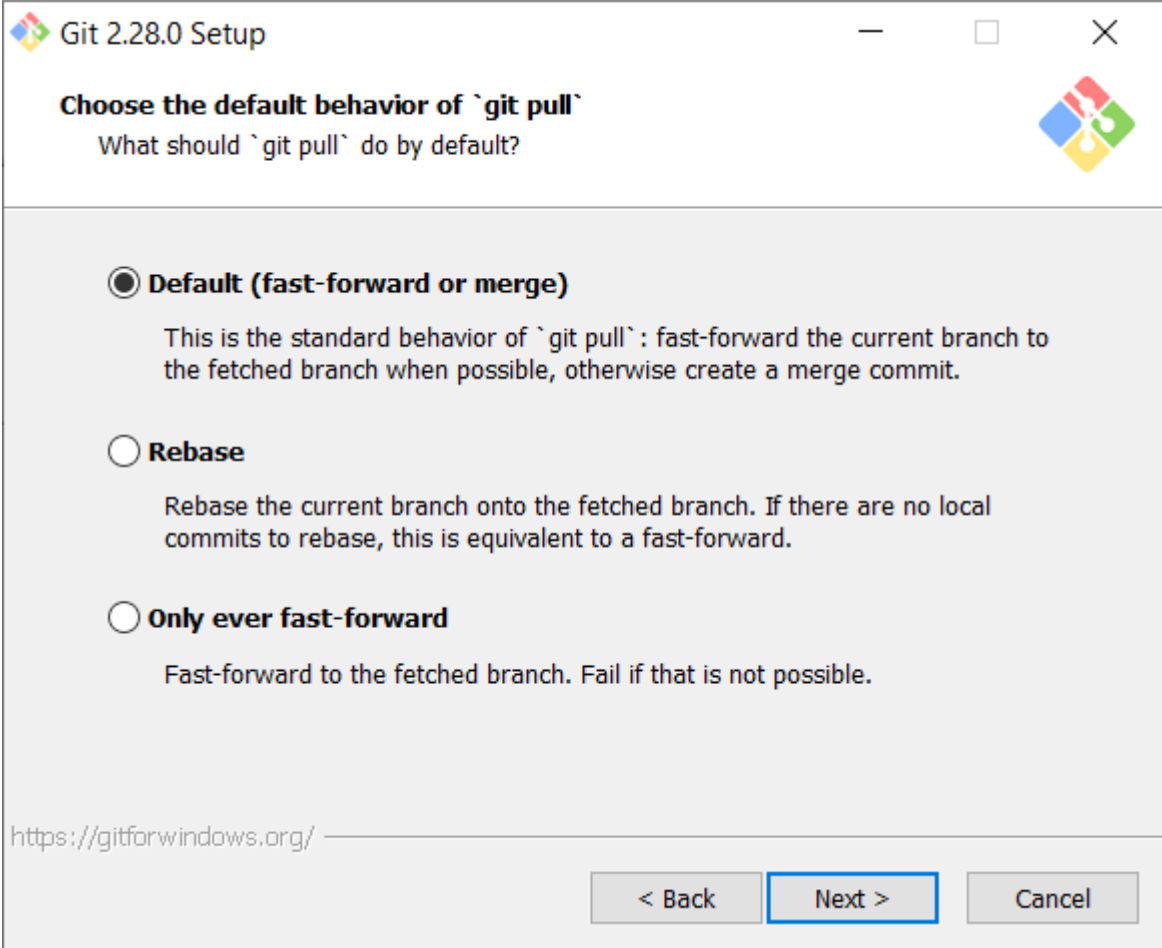use this option if you understand the implications.

https://gitforwindows.org/

< Back   Next >   Cancel

8) Click next

Git 2.28.0 Setup — □ ✕

Choosing HTTPS transport backend
Which SSL/TLS library would you like Git to use for HTTPS connections?

⦿ **Use the OpenSSL library**

Server certificates will be validated using the ca-bundle.crt file.

○ **Use the native Windows Secure Channel library**

Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates
distributed e.g. via Active Directory Domain Services.

https://gitforwindows.org/

< Back   Next >   Cancel

9) Click next

Git 2.28.0 Setup — □ ×

Configuring the line ending conversions
How should Git treat line endings in text files?

◉ **Checkout Windows-style, commit Unix-style line endings**

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

◯ **Checkout as-is, commit Unix-style line endings**

Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

◯ **Checkout as-is, commit as-is**

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

https://gitforwindows.org/

< Back   Next >   Cancel

10) Click next

11) Click next

12) Click next

13) Click next

Git 2.28.0 Setup — □ ✕

Configuring extra options
Which features would you like to enable?

☑ **Enable file system caching**

File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

☐ **Enable symbolic links**

Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.

https://gitforwindows.org/

< Back   Next >   Cancel

14) Click On Install

15) Installation successful
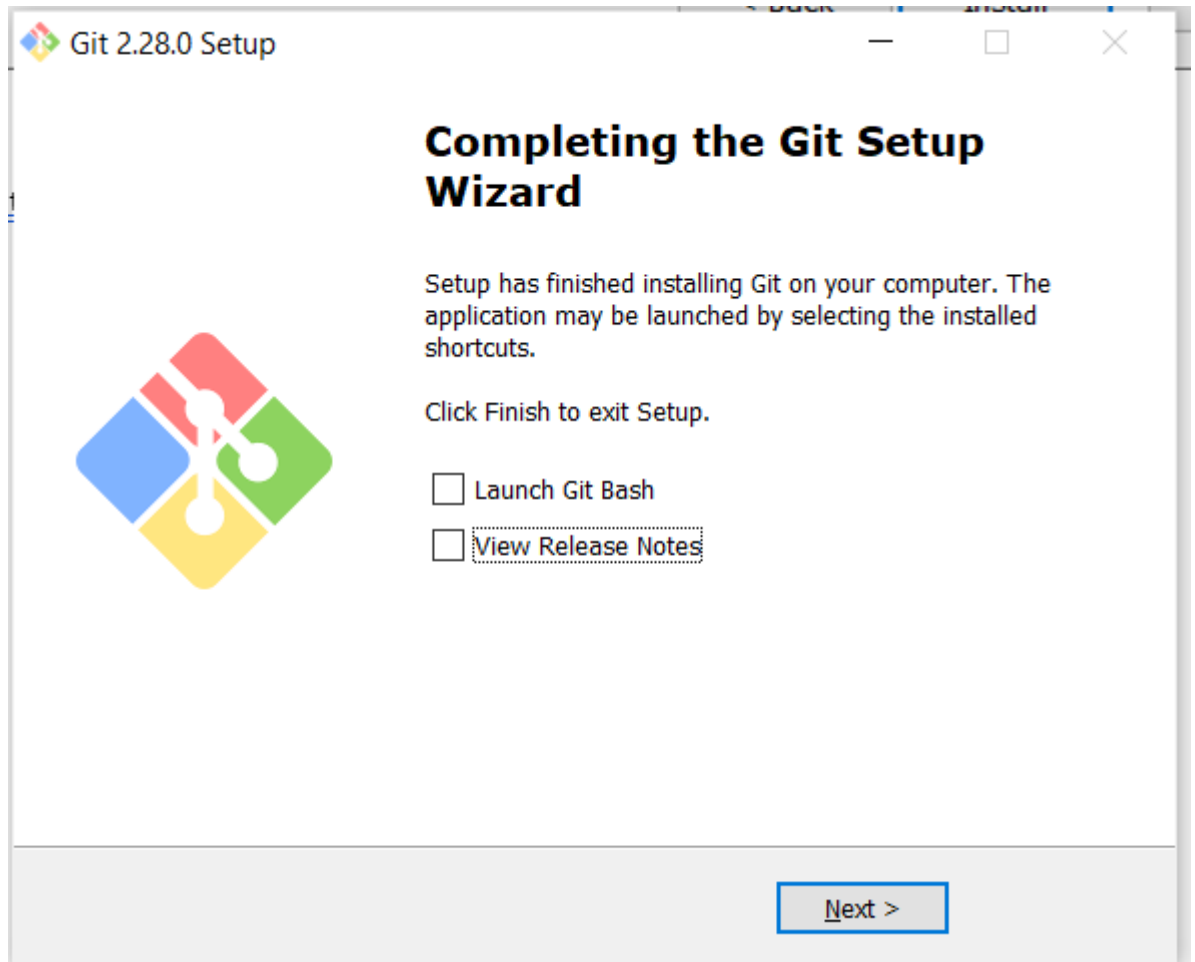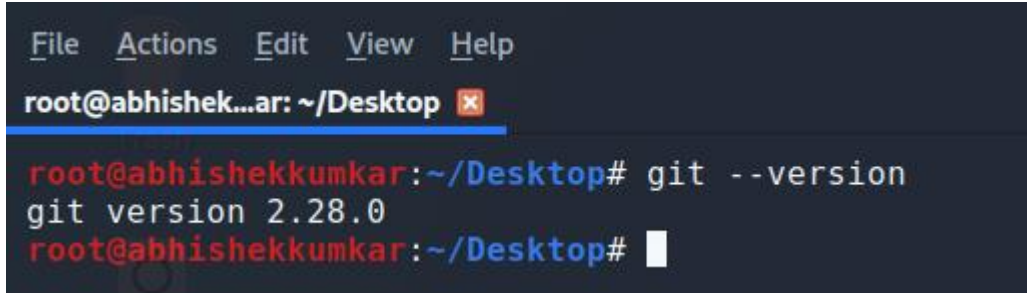
Linux Installation :

1) Go to desktop and right click
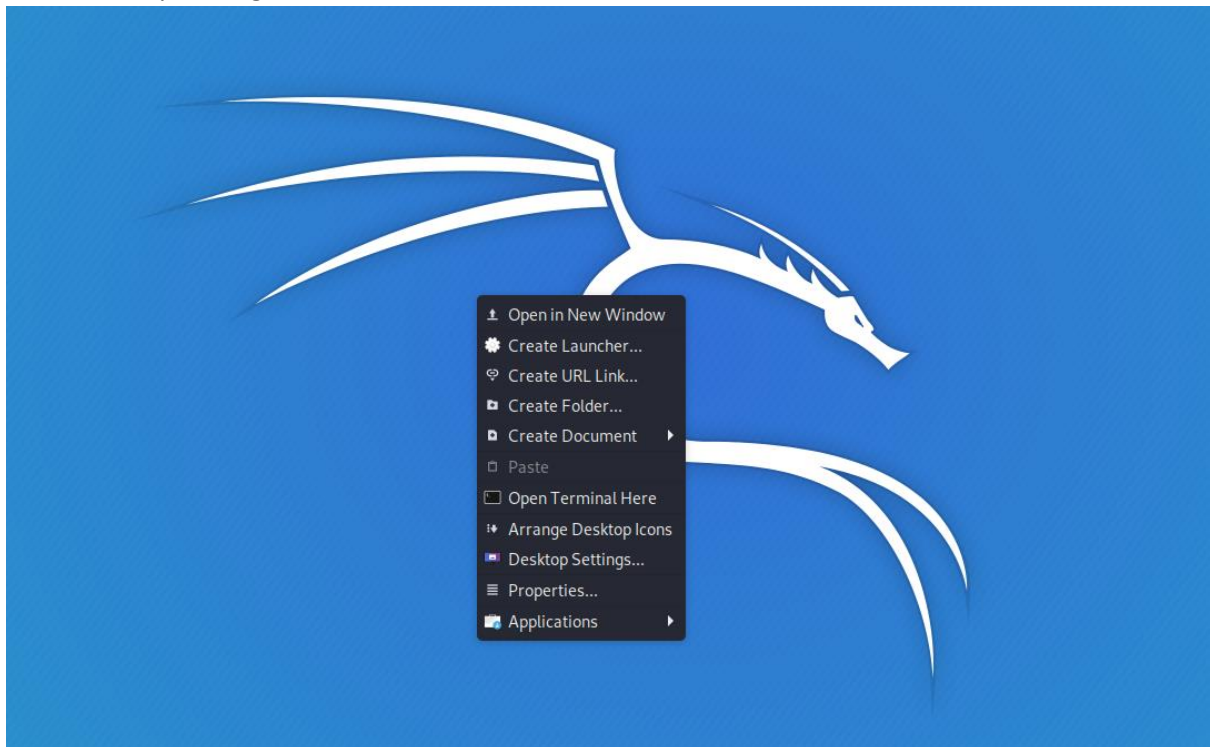
2) Click on Open Terminal Here and type and press enter: apt-get install git
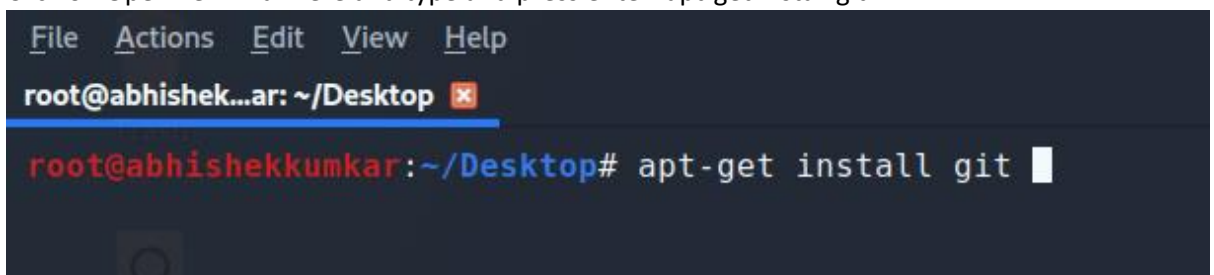
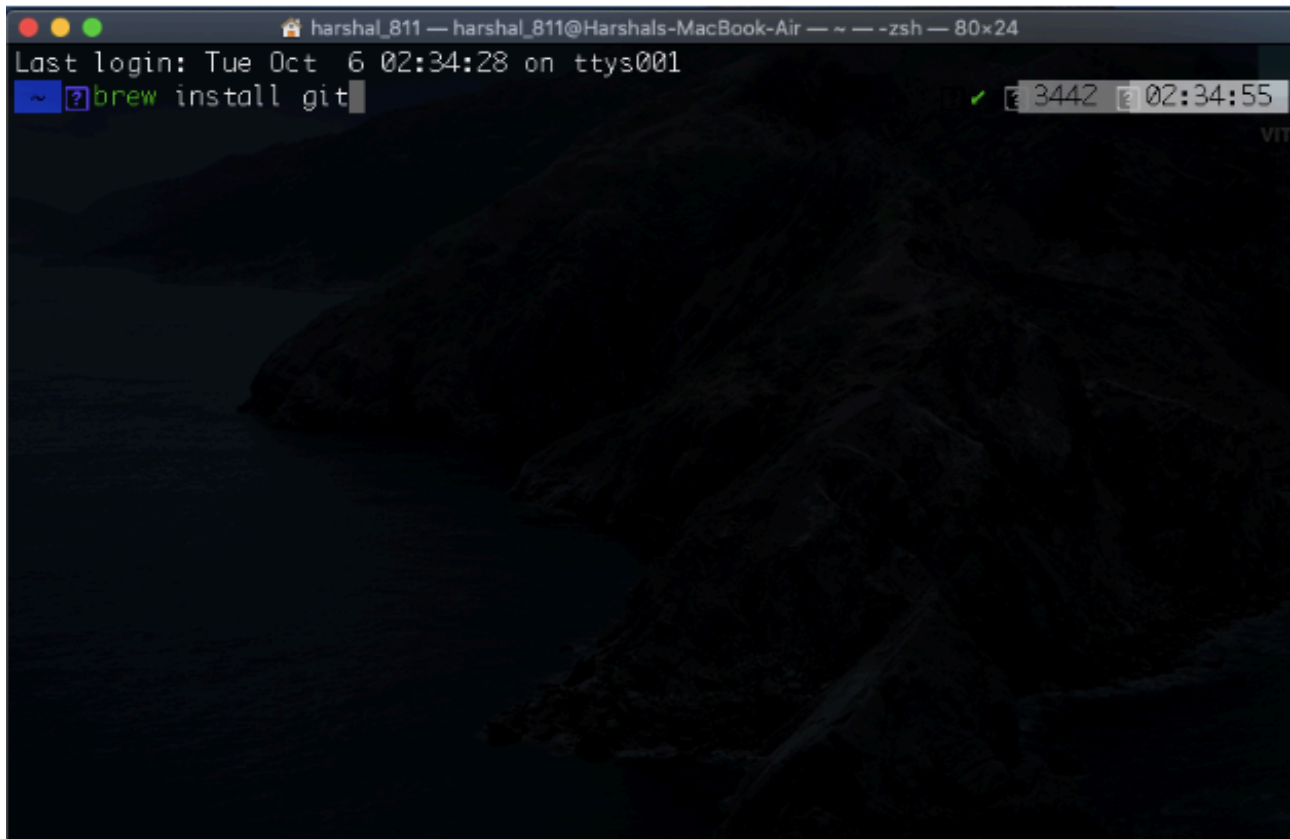3) Enter y and press enter
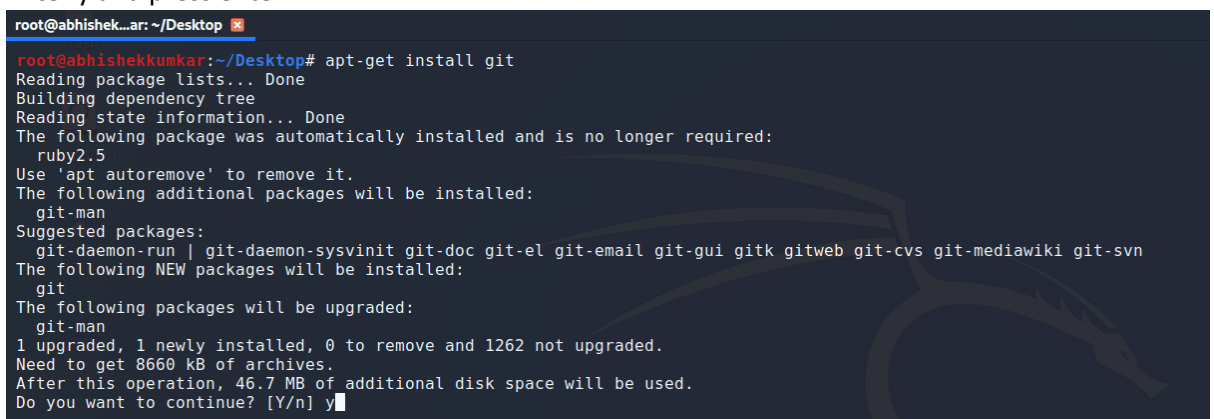
4) Type and press enter : git --version

5) Successfully installed git

<u>Git Installation on MacOS</u>

1. Open your terminal and enter the following command: **brew install git**

NOTE: The command won't work if HomeBrew is not installed on your laptop. So make sure to install HomeBrew as well.
It can be installed in just one command. Refer https://brew.sh/