

**How to Install Git on Windows**

There are two ways to install Git on Windows:

* **Using the Git installer**. The [GUI](https://phoenixnap.com/glossary/what-is-gui) installer wizard is available on the official Git website and allows you to configure the Git installation step-by-step.
* **Using Winget**. A Linux-style CLI [package manager](https://phoenixnap.com/glossary/what-is-a-package-manager) for Windows that simplifies app installations but offers fewer configuration options as it assumes all default options.

Choose your preferred installation method based on how much control you want over the installation process and your preference for using the [GUI or CLI](https://phoenixnap.com/kb/cli-vs-gui).

**Install Git on Windows via GUI**

The GUI method involves downloading the Git installer wizard from the official website and configuring the installation in a traditional Windows step-by-step manner. This approach is useful if you want full control of the installation process and do not like using the CLI.

**Note:** During the installation, Git prompts you to select a text editor. If you don't have one, we strongly advise you to install it prior to installing Git. Our roundup of the [best text editors for coding](https://phoenixnap.com/kb/best-linux-text-editors-for-coding) may help you decide.

Follow the steps below:

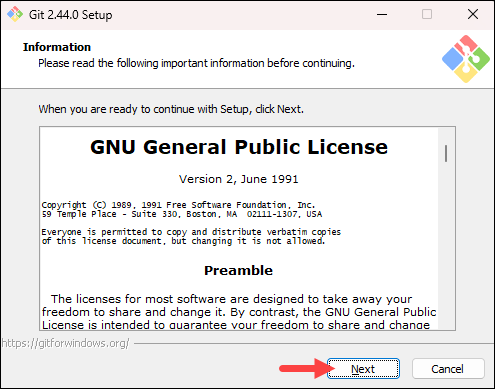
1. Navigate to the [official Git downloads page](https://git-scm.com/download/win) and click the download link for the latest Git version for Windows:



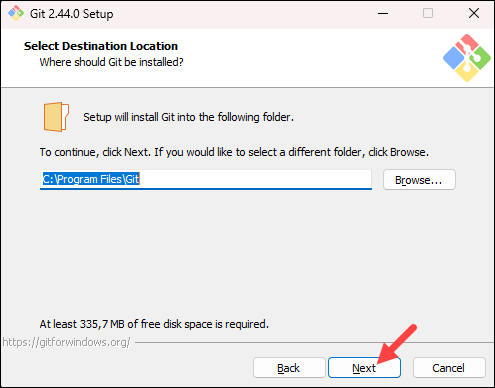
The link contains the latest [64-bit](https://phoenixnap.com/glossary/what-is-x64) Git version for Windows. Alternatively, if you use a [32-bit](https://phoenixnap.com/glossary/what-is-x86) system, download the 32-bit Git installer.

2. Double-click the downloaded [file](https://phoenixnap.com/glossary/what-is-a-file) to extract and launch the installer.

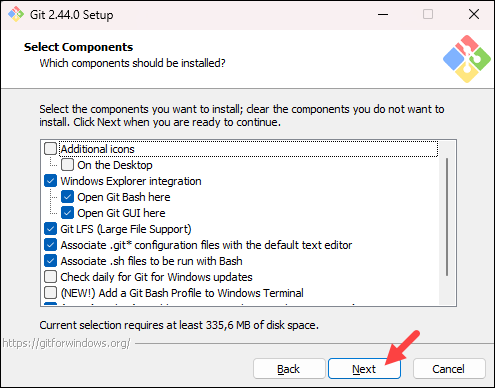
3. Review the [GNU General Public License](https://phoenixnap.com/glossary/gnu-general-public-license), and when you are ready to install, click **Next**.



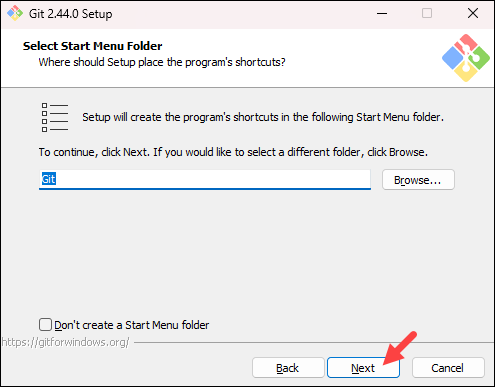
4. The installer prompts you for an installation location. Leave the default one unless you want to change it, and click **Next**.



5. In the component selection screen, leave the defaults unless you need to change them and click **Next**.

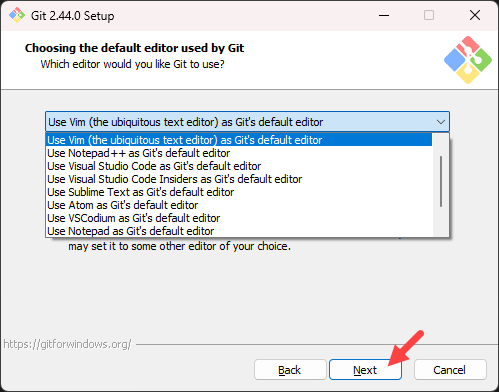


6. The installer offers to create a start menu [folder](https://phoenixnap.com/glossary/what-is-a-folder). Click **Next** to accept and proceed to the next step.

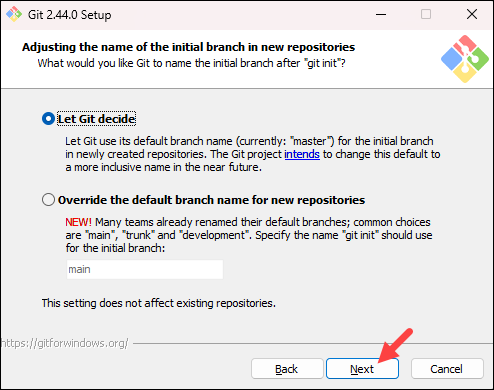


7. Select a text editor you want to use with Git. Use the drop-down menu to select Notepad++ (or whichever text editor you prefer) and click **Next**.

If you prefer to use a CLI text editor in [Git Bash](https://phoenixnap.com/kb/what-is-git-bash" \t "_blank), select [nano](https://phoenixnap.com/kb/use-nano-text-editor-commands-linux" \t "_blank) or [Vim](https://phoenixnap.com/kb/vim-commands-cheat-sheet) from the list.

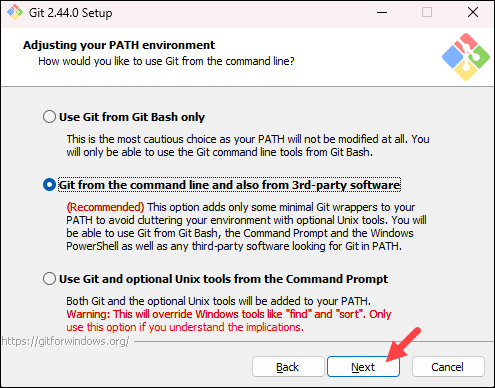


8. The next step allows you to choose a different name for your initial branch. The default is **master**. Unless you are working in a team that requires a different name, leave the default option and click **Next.**

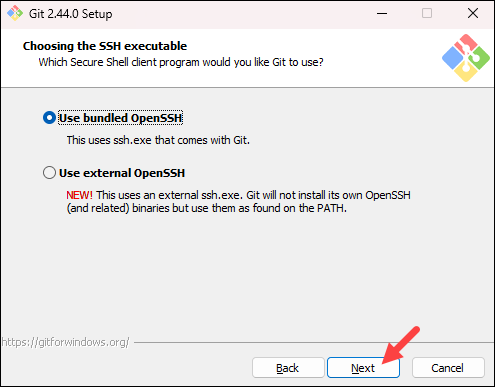


**Note:** If you are new to Git, check out our guides to [Git branch name conventions](https://phoenixnap.com/kb/git-branch-name-convention" \t "_blank) and [Git branching strategies](https://phoenixnap.com/kb/git-branching-strategy" \t "_blank).

9. The next step allows you to change the **PATH environment**. The **PATH**is the default set of [directories](https://phoenixnap.com/glossary/what-is-a-directory) included when you run a command from the command line. Keep the middle (recommended) selection and click **Next**.

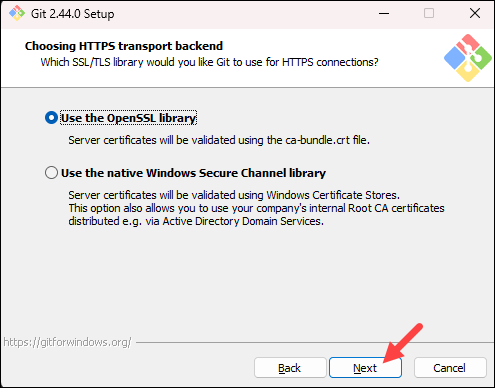


10. The installer prompts you to select the SSH client for Git to use. Git already comes with its own SSH client, so if you don't need a specific one, leave the default option and click **Next.**

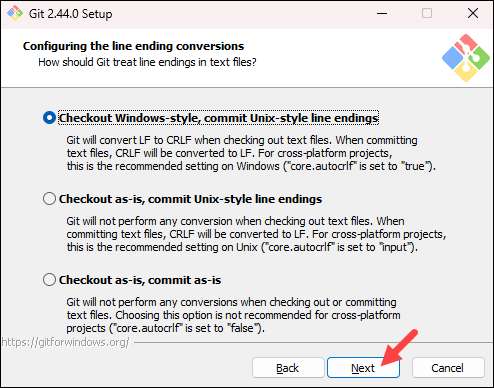


**Note:** Check out our comparison of [SSH and HTTPS for Git](https://phoenixnap.com/kb/git-ssh-vs-https) and which one you should use.

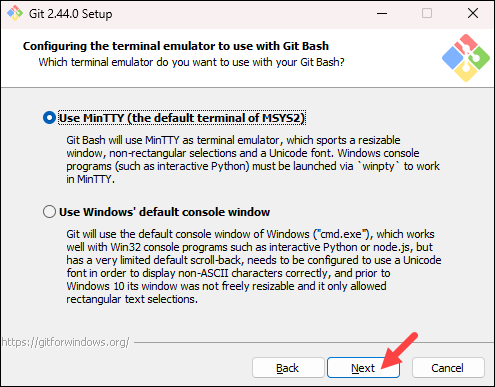
11. The next option relates to server certificates. The default option is recommended for most users. If you work in an Active Directory environment, you may need to switch to Windows Store certificates. Select your preferred option and click **Next**.



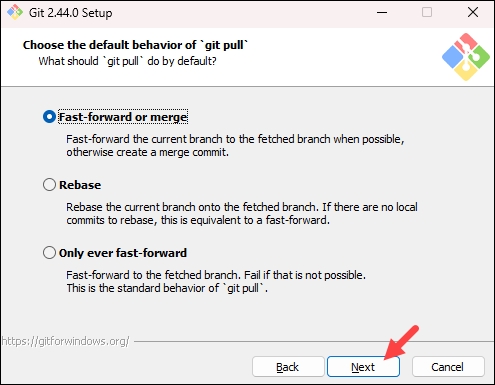
12. The following selection configures line-ending conversion, which relates to the way data is formatted. The default selection is recommended for Windows. Click **Next** to proceed.



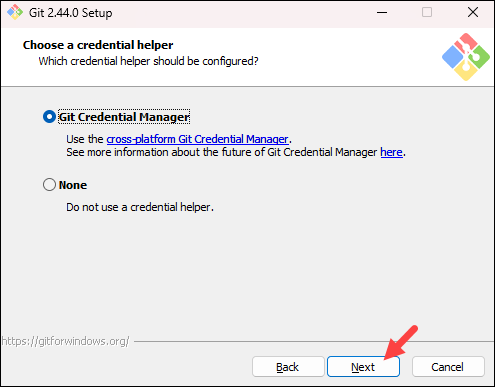
13. Choose the [terminal emulator](https://phoenixnap.com/glossary/terminal-emulation) you want to use. The default MinTTY is recommended for its features. Click **Next** to continue.



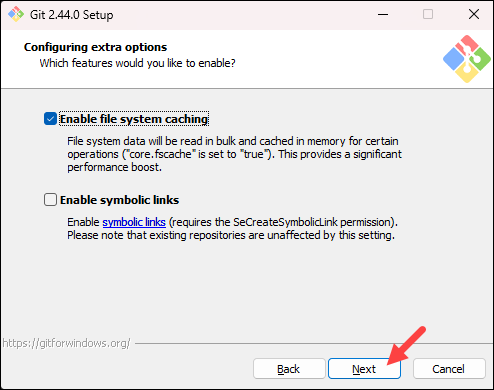
14. The next step allows you to choose what the **git pull** command will do. The default option is recommended unless you specifically need to change its behavior. Click **Next**to continue with the installation.



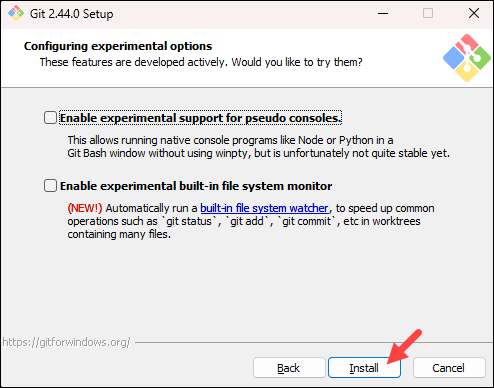
15. The next step is to choose which credential helper to use. Git uses credential helpers to fetch or save credentials. The default option is the most stable one. Select your preferred credential manager and click **Next**.



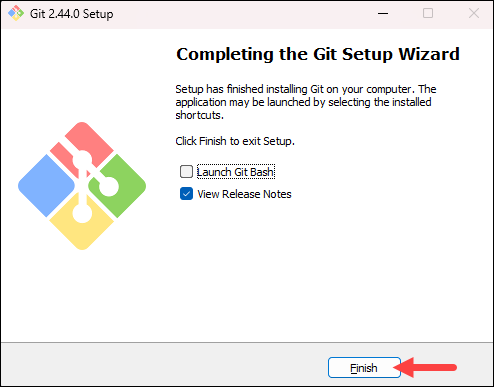
16. The next step lets you decide which extra options to enable. If you use [symbolic links](https://phoenixnap.com/kb/symbolic-link-linux), which represent shortcuts for the command line, tick the box. Keep [file system](https://phoenixnap.com/glossary/filesystem) caching checked and click **Next**.



17. Depending on which Git version you are installing, it may offer to install experimental features. At the time this article was written, the installer offered options to include support for pseudo controls and a built-in file system monitor. For the most stable operation, do not install experimental features and click **Install**.



18. Once the installation is complete, tick the boxes to view the Release Notes or launch Git Bash if you want to start using Git right away, and click **Finish**.

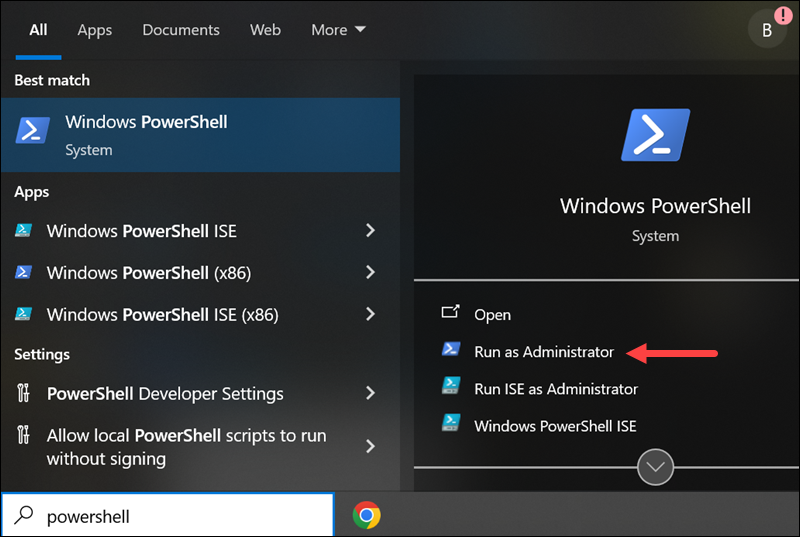


**Install Git on Windows via CMD**

Installing Git on Windows using the command line requires a [working Winget installation](https://phoenixnap.com/kb/install-winget), an Internet connection for the program to download the necessary files, and access to the [Command Prompt or PowerShell](https://phoenixnap.com/kb/powershell-vs-cmd).

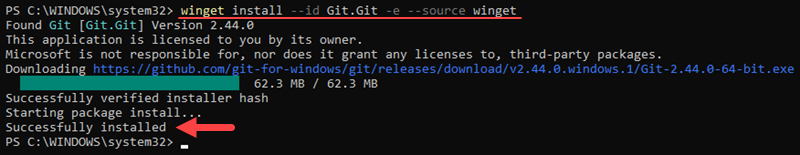
If you prefer using a package manager for installing and managing programs, follow the steps below:

1. Press the **Windows** key and type *powershell*. From the results, select the **Run as administrator** option for **Windows PowerShell**.



2. In PowerShell, run the following command to install the latest Git version:

winget install --id Git.Git -e --source winget



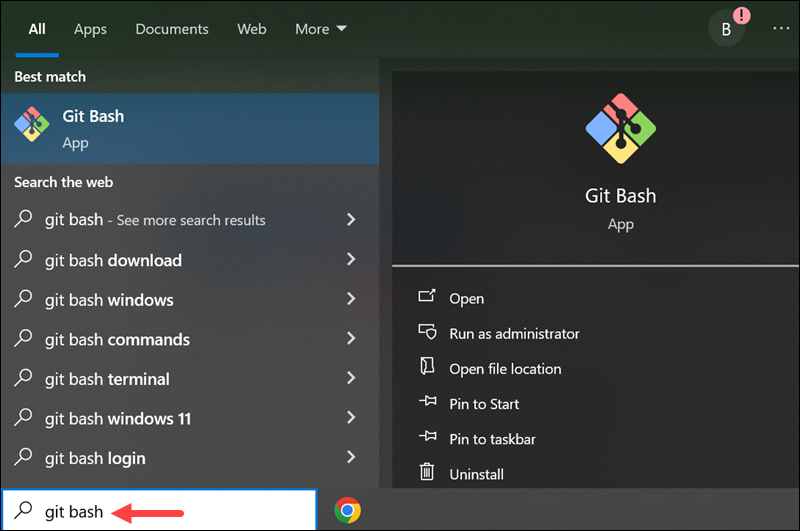
The command downloads and installs Git for Windows. It assumes all the default settings and completes the installation automatically.

**How to Launch Git in Windows**

Git has two modes of use – a **bash scripting shell** (or command line) and a **graphical user interface** (GUI). This section shows how to launch Git after installation.

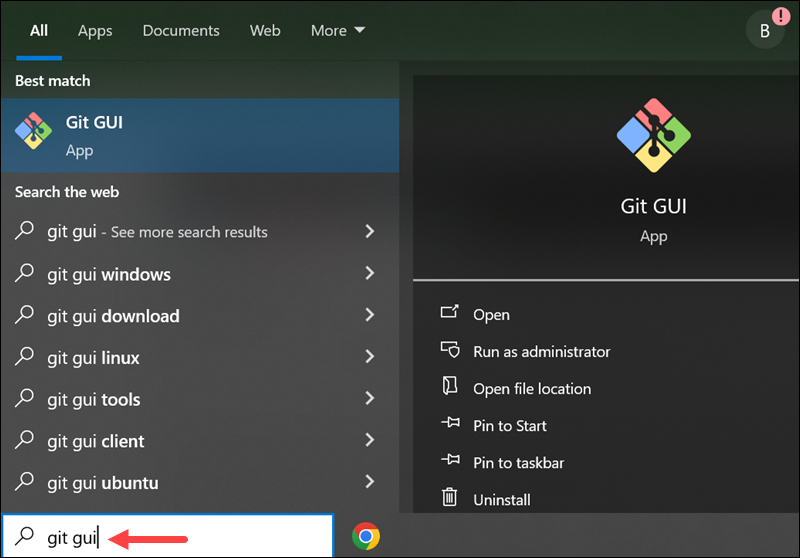
**Launch Git Bash Shell**

To launch **Git Bash**, open the **Windows Start** menu, type *git bash*, and press **Enter** (or click the application icon).



**Launch Git GUI**

To launch **Git GUI**, open the **Windows Start** menu, type *git gui*, and press **Enter** (or click the application icon).



**Configuring Git on Windows**

After installing Git, there are some options that you should configure to customize your Git environment. These configurations stick between upgrades, so they should be done only once after a fresh install. This section shows how to set up Git after installation.

**Configure Identity**

Your identity in Git is your username and email address which Git uses every time you create a commit. To set up your identity, open Git Bash and use the syntax below:

git config --global user.name "[username]"

Replace **[user\_name]** with the actual username you will use. If you have a GitHub account, you can use that username and email.

git config --global user.email [email]

Replace **[email]** with the email you want to use.

**Change Default Text Editor**

The default text editor for Git is the same as your system's default editor unless you specify something different during the installation. You can change the text editor for Git after installation using the following syntax:

git config --global core.editor [path\_to\_editor\_exe\_file]

The path to the editor's executable file can differ based on how your editor is packaged and whether it is a 32-bit or 64-bit system. For example, to set Notepad++ as the default editor for Git, run the following command:

git config --global core.editor "'C:/Program Files/Notepad++/notepad++.exe' -multiInst -notabbar -nosession -noPlugin"

* **-multiInst** - This option tells Notepad++ to open files in a new instance rather than reusing an existing instance. It is useful if you want to edit multiple files simultaneously.
* **-notabbar** - This option hides the tab bar in Notepad++, which is useful when working with a single file at a time.
* **-nosession** - This option prevents Notepad++ from restoring the last session on startup. It opens Notepad++ with a clean slate without previously opened files.
* **-noPlugin** - This option disables all plugins in Notepad++. It is useful if you want to minimize the startup time or if you prefer to work without plugins.

If you want to use a different editor, refer to the official configuration instructions and find the command for the editor you want to use.

**Change Default Branch Name**

The default branch in Git is **master**. However, some developers have started using **main** instead. If your organization wants to change the default branch name, you can do so using the syntax below:

git config --global init.defaultBranch [branch\_name]

For example, to use **main** as the default branch name, run:

git config --global init.defaultBranch main

**Git on Windows: Getting Started With Your First Repo**

This section shows the basic steps for using Git on Windows - from checking your Git version to creating a test directory and making your first commit. Since we will be working with a remote repository, you will need a GitHub username and password.

**How to Check Git Version**

To check which Git version is installed on your system, open the PowerShell and run the following command:

git --version

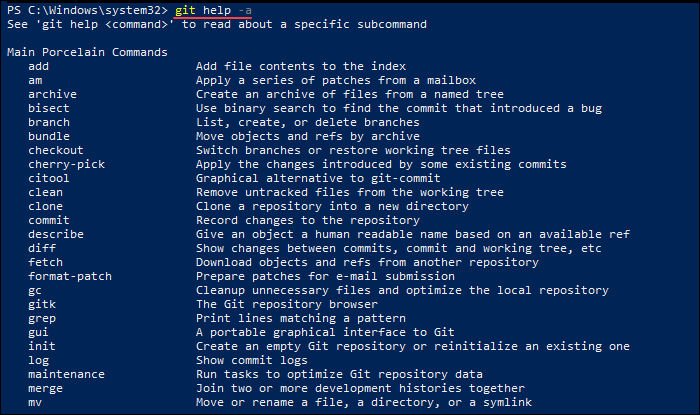
Checking which version of Git is installed on Windows.

The output shows which Git version is installed on the system.

**List All Git Commands**

Git has a help file that lists the most commonly used Git commands. To see the help file, open Git Bash or Windows PowerShell and run the following command:

git help -a



The output contains the most common Git commands.

**Create and Initialize Test Directory**

Follow the steps below to create a new directory and initialize it as a Git repository:

1. Open Git Bash or a Windows PowerShell interface and use the syntax below to create a new test directory (folder):

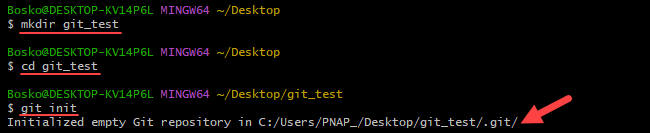
mkdir [directory\_name]

2. Change your location to the newly created directory with the [cd command](https://phoenixnap.com/kb/linux-cd-command" \t "_blank):

cd [directory\_name]

3. Initialize the directory as Git repository by running:

git init



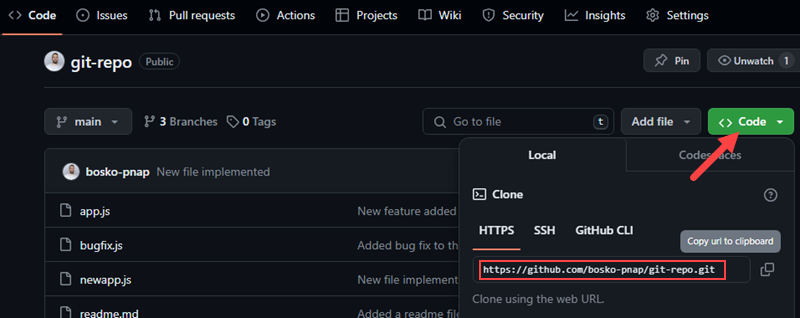
**Clone GitHub Repository**

Follow the steps below to [clone a repository](https://phoenixnap.com/kb/git-clone-submodule) from GitHub to your local Windows machine:

1. In a [web browser](https://phoenixnap.com/glossary/web-browser-definition), navigate to your repository on GitHub.

2. Click the **Code** button and select the [HTTPS or SSH](https://phoenixnap.com/kb/git-ssh-vs-https) option, depending on how you want to secure your connection. For this tutorial, we will use HTTPS. Copy the [URL](https://phoenixnap.com/glossary/url-definition-meaning) for cloning the repository.

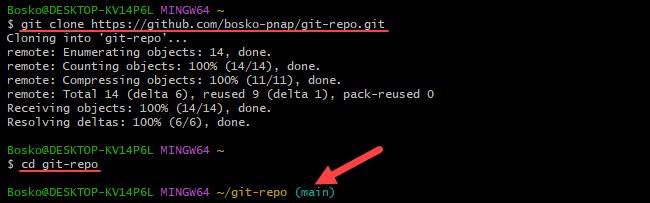
**Note:** To clone a repository using SSH, you must [generate an SSH key pair](https://phoenixnap.com/kb/generate-ssh-key-windows-10) on your Windows workstation and assign the public key to your GitHub account.



3. Open Git Bash or Windows PowerShell and use the syntax below to clone the repository:

git clone [repository\_url]

Replace **[repository\_url]** with the URL you copied on GitHub. For example:



**List Remote Repositories**

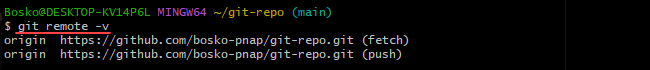
After cloning, your working directory should have a copy of the repository from GitHub. It should contain a directory with the name of the project. Move to the directory with **cd**:

cd [repository\_name]

Replace **[repository\_name]** with the actual name of the repository you downloaded. If it is not working, list the contents of the current directory with the **ls** command. This is helpful if you don't know the exact name or need to check your spelling.

Once you are in the [subdirectory](https://phoenixnap.com/glossary/what-is-a-subdirectory), list the remote repositories with:

git remote -v



The output shows a list of remote repositories associated with your local repository and their corresponding URLs.

**Push Local Files to Remote Repository**

This section shows how to create a new file, add it to the tracking index, create a commit, and ultimately [push the changes to the remote repository](https://phoenixnap.com/kb/git-push-to-remote-branch). Follow the steps below:

1. Open Git Bash and create a new file with the [touch command](https://phoenixnap.com/kb/touch-command-in-linux):

touch text.txt

2. Add your new file to the tracking index:

git add text.txt

If you have created multiple files, you can track all files by running:

git add .

3. Run the following command to make sure the *text.txt* file has been added:

git status

4. Next, commit the changes using the syntax below:

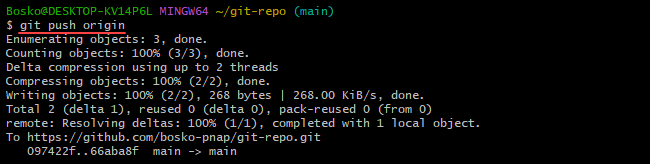
git commit -m "[commit\_message]"

The **-m** option allows you to specify a commit message within the commit command. If you omit the **-m** option, Git opens the default text editor in which you can write the commit message.

5. Finally, push the changes to the remote GitHub repository:

git push [remote\_repository]

Replace **[remote\_repository]** with the name of your remote repo. For example:



Depending on how you have chosen to [authenticate](https://phoenixnap.com/glossary/what-is-authentication) with GitHub, you may need to enter your username and access token.

**Note:** If you are new to Git, read our tutorial on [how Git works](https://phoenixnap.com/kb/how-git-works) to learn more about Git workflow and Git functions.

**Conclusion**

You now have a working installation of Git on your Windows system. If you already have Git installed, you might want to know [how to update Git](https://phoenixnap.com/kb/how-to-update-git). Use Git's features to coordinate work among programmers on a project.

Next, check out our [Git commands cheat sheet](https://phoenixnap.com/kb/git-commands-cheat-sheet" \t "_blank) or learn how to [remove a Git remote](https://phoenixnap.com/kb/git-remove-remote) from a repository.

Was this article helpful?

YesNo

[Share onX (Twitter)](https://twitter.com/intent/tweet?text=How%20to%20Install%20and%20Use%20Git%20on%20Windows&url=https%3A%2F%2Fphoenixnap.com%2Fkb%2Fhow-to-install-git-windows)[Share onFacebook](https://www.facebook.com/sharer/sharer.php?u=https%3A%2F%2Fphoenixnap.com%2Fkb%2Fhow-to-install-git-windows)[Share onLinkedIn](https://www.linkedin.com/shareArticle?mini=1&url=https%3A%2F%2Fphoenixnap.com%2Fkb%2Fhow-to-install-git-windows&title=How%20to%20Install%20and%20Use%20Git%20on%20Windows&source=https%3A%2F%2Fphoenixnap.com%2Fkb&summary=Git%20tracks%20source%20code%20changes%20during%20the%20software%20development%20process.%20It%20can%20help%20coordinate%20work%20among%20programmers.%20This%20tutorial%20uses%20multiple%20images%20to%20show%20you%20how%20to%20install%20Git%20on%20Windows%20in%20a%20few%20simple%20steps%2C%20followed%20by%20a%20few%20basic%20commands%20to%20complete%20the%20initial%20setup.)[Share onEmail](mailto:?body=I%20read%20this%20post%20and%20wanted%20to%20share%20it%20with%20you.%20Here%27s%20the%20link%3A%20https%3A%2F%2Fphoenixnap.com%2Fkb%2Fhow-to-install-git-windows&subject=A%20post%20worth%20sharing%3A%20How%20to%20Install%20and%20Use%20Git%20on%20Windows)



Bosko Marijan

Having worked as an educator and content writer, combined with his lifelong passion for all things high-tech, Bosko strives to simplify intricate concepts and make them user-friendly. That has led him to technical writing at PhoenixNAP, where he continues his mission of spreading knowledge.